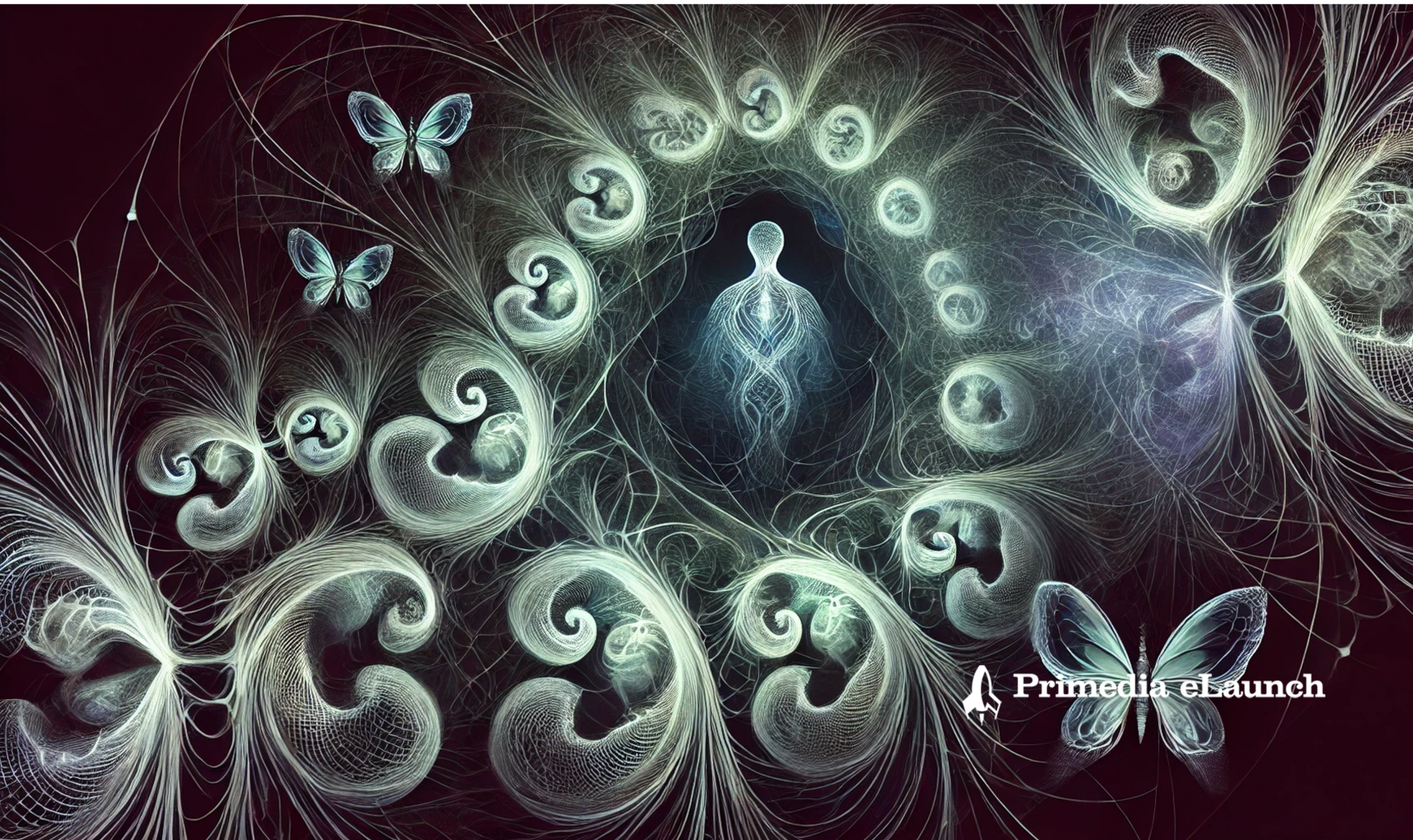
Illia Danilishyn, Oleksandr Danilishyn

INTRODUCTION TO DYNAMIC MATHEMATICS: DYNAMIC SETS, DYNAMIC OPERATORS AND THEIR APPLICATIONS

Monograph



Primedia eLaunch

Illia Danilishyn, Oleksandr Danilishyn

INTRODUCTION TO DYNAMIC MATHEMATICS: DYNAMIC SETS, DYNAMIC OPERATORS AND THEIR APPLICATIONS

Monograph

Edited by *Volodymyr Pasynkov*

Shawnee, USA
Primedia eLaunch LLC
2024

Epigraph:

Mathematics is on the border of science (knowledge), so it is she who is able to make major breakthroughs beyond this border.

Reviewer: **Volodymyr PASYNKOV**

PhD of physic-mathematical science, assistant professor of applied mathematics and calculated techniques department of «National Metallurgical Academy», Ukraine

D 18 **Danilishyn I., Danilishyn O.** Introduction to Dynamic Mathematics: dynamic sets, dynamic operators and their applications: monograph / ed. by Pasynkov V. – Shawnee, USA : Primedia eLaunch LLC. – 2024. – 442 p.

ISBN 979-8-89217-806-8

DOI 10.36074/itdmdsdoata.monograph-2024

The material in this monograph is a development of the previous monograph of the authors: «Hierarchical dynamic mathematical structures (models) theory» (USA, 2024); ISBN 979-8-89217-809-9.

We are just starting from the assumed structure of self-organization, since we are interested not so much in the numerical calculation of this as in the structure of self-organization itself, its formation (construction) for the necessary purposes and its management. Our task: to understand the principles of functioning of living energy (living organism, in particular, human, subtle energies), and then using these principles to "construct" artificial living energies (let's call them pseudo-living energies). Considering object A as the value of dynamic variable P, we can reassign P:= B at the top level and instead of A we will have object B, and also perform any other parallel dynamic operations with objects. That is, our world will be perceived as a Dynamic Programming environment. Dynamic programming is an extension of science for "working" with the top and middle levels. Here information is transformed into actions. Energy Internet: a) transmission of necessary singularities (subtle energies) (connection to necessary singularities (subtle energies)), b) transmission of necessary self-structures (subtle energies) (connection to necessary self-structures) etc. It is also necessary to construct a bioSmnsprt on living, grown elements of the central nervous system, in particular, neurons. It is also possible to construct another version of bioSmnsprt, using a viral model, on living, grown bacteriophages. We have simply outlined some aspects; it is too early to develop them without experimental study. All problems have a solution, the only question is: a) in complete identification with the problem, b) the "breadth of the field" of values that you can provide. Our mathematics is unusual for a mathematician, because here the fulcrum is the action, and not the result of an action, as in classical mathematics. Therefore, our mathematics is adapted not only to obtain results, but also to directly control actions, which will certainly show its advantages on a fundamentally new type of neural networks with directly parallel calculations, for which it was created. Any action has much greater potential than its result. The significance of the monograph: in a new qualitatively different approach to the study of complex processes through new mathematical hierarchical parallel dynamic structures, in particular those processes that are studied by synergetics. Our approach is not based on deterministic equations that generate self-organization, which is very difficult to study and gives very small results for a very limited class of problems and does not provide the most important thing - the structure of self-organization. We simply start from the assumed structure of self-organization, since we are interested not so much in the numerical calculation of this, but in the structure of self-organization itself, its formation (construction) for the necessary purposes and its management. Although we are also interested in numerical calculations. Laureates of the Nobel Prize in Physics 2023 Ferenc Kraus and his colleagues Pierre Agostini and Anne L'Huillier used a short-pulse laser to generate attosecond pulses of light to study the fuzzy dynamics of electrons in matter. According to our Synthesis-Type Singularity Theory, its action corresponds to a singularity, allowing the upper level of subtle energies to be reached to manipulate the lower levels. In In April 2023, we proposed to use a short-pulse laser to achieve desired goals using a directly parallel neural network. We then proposed a fundamental development of this directly parallel neural network. The significance of our monograph lies in the formation of the proposed mathematical structure of subtle energies, this is done for the first time in science, and the proposed classification of mathematical structures of subtle energies for the first time. The experiments of 2022 Nobel Prize winners Asle Allen, John Clauser, Anton Zeilinger and the chemistry experiments of Nazhipa Valitov eloquently demonstrate, that we are right and that these studies are necessary. Be that as it may, we have created classes of new mathematical structures, new fuzzy mathematical singularities, i.e. we have contributed to the development of mathematics.

UDC 004:51(07)

ISBN 979-8-89217-806-8

AUTHORS:

Illia Danilishyn

Graduated from Sumy State University.
Currently a postgraduate student at Sumy State University.

Oleksandr Danilishyn

Graduated from Sumy State University.
Currently works as a practicing physician.

Competing interest:
There are no competing interests. All sections of the monograph are executed jointly.

Funding:
There were no sources of funding for writing the monograph.

Authors' contributions:
The contribution of the authors is the same, we will not separate.

Foreword:

In this book, the authors develop the themes of the book [50] in more interesting dynamic structures. Significance of the monograph: in a new qualitatively different approach to the study of complex processes through new mathematical hierarchical dynamic structures, in particular those processes that are dealt with by Synergetics. Authors' approachis not based on deterministic equations that generate self-organization, which is very difficult to study and gives very small results for a very limited class of problems and does not provide the most important thing - the structure of self-organization. They are just starting from the assumed structure of self-organization, since they are interested not so much in the numerical calculation of this as in the structure of self-organization itself, its formation (construction) for the necessary purposes and its management. Although they are also interested in numerical calculations. Nobel laureates in physics 2023 Ferenc Kraus and his colleagues Pierre Agostini and Anna Lhuillier used a short-pulse laser to generate attosecond pulses of light to study the dynamics of electrons in matter. According to their Theory of singularities of the type synthesizing, its action corresponds to singularity $\uparrow I \ \downarrow_h^q$, which allows one to reach the upper level of subtle energies to manipulate lower levels. In April 2023 [14], the authors proposed using a short-pulse laser to achieve the desired goals by a directly parallel neural network. They then proposed the fundamental development of this directly parallel neural network. There is a long overdue need for the use of singular hierarchical structures, in particular self-sets, to describe complex processes, in particular to describe unusual states of consciousness and pathological conditions in medicine. The experiments of Nobel laureates in 2022-year Asle Ahlen, Clauser John, Zeilinger Anton correspond to the concept of the Universe as its self-containment in itself. The monograph aims to create new constructive hierarchical mathematical objects for new technologies, particularly for a fundamentally new type of neural network with parallel computing and not the usual parallel computing through sequential computing.

Ukraine, September 2023
Volodymyr Pasynkov

Contents

Introduction

There is a need to develop an instrumental mathematical base for new technologies. Our constructive approach to set theory differs from the construction of constructive sets by A.Mostowski [1]: we construct completely different types of constructive sets. Here, the axiom of regularity (A8) [2] is removed from the axioms of set theory, so we naturally obtain the possibility of using singularities in the form of self-sets, self-elements, which is exactly what we need for new mathematical models for describing complex processes. Instead of the axiom of regularity, we introduce the following axioms:
Axiom R1. $\forall$ set B we have: $Sprt_B^B$ is an element of itself. $Sprt_B^B$ will be denoted by selfB. Axiom R2. $\forall$B($\exists$ (selfB)$^{-1}$).

Part I. Sprt – elements and their applications

Introduction

There is a need to develop an instrumental mathematical base for new technologies. The task of the work is to create new approaches for this by introducing new concepts and methods [3]. Significance of this part: in a new qualitatively different approach to the study of complex processes through new mathematical, hierarchical, dynamic structures, in particular those processes that are dealt with by Synergetics.

We consider expression

${}_D^C Sprt_B^A$ ($*_{1.1}$)

where A fits into B, D is forced out from C[3]. The result of this process will be described by the expression

${}_D^C Srt_B^A$ ($*_{1.2}$)

If A, B, D, C are taken as sets, then we will call ($*_{1.1}$) a dynamic set. The need ($*_{1.1}$) arose to describe processes in networks. Threshold element Sprt --${}_{\{qy\}}^{b} Sprt(t)_{b}^{\{ax\}}$, b- artificial neurons of type Sprt (designation - mnSt), x= (x_1, x_2, …, x_n) are the values of the initial signals,, a=(a_1,a_2,…,a_n) are the weights of Sprt-synapses and the values of the output signals [3]. It can be considered a simpler version of the dynamic set

$Sprt_B^A$ ($**_{1.1}$)

where set A fits into set B, the result of this process will be described by the expression

Srt_B^A ($**_{1.2}$)

or

${}_A^B Sprt$ ($***_{1.1}$)

where set A is forced out from B, the result of this process will be described by the expression

${}_{A}^{B}Srt$ (***$_{1.2}$)

We consider the measure: $\mu**({}^{b}{}_{D}Sprt_{b}^{A})$=$\frac{\mu(A)}{\mu(D)}$, where $\mu A), \mu D)$ –usual measures of sets A, D [3].

Remark. One can consider some generalization for (*$_{1.1}$): ${}_{D}^{q_1(C)}Sprt_{q(B)}^{A}$, where A is contained into B through q, D is forced out from C through q_1, A, B, D, C are taken as sets. The result of this process will be described by the expression ${}_{D}^{q_1(C)}Srt_{q(B)}^{A}$.

Similarly, for (**$_{1.1}$): $Sprt_{q(B)}^{A}$, where A is contained into B through q, for (***$_{1.1}$): ${}_{A}^{q(B)}Sprt$, where D is displaced from C through q. The result of this process will be described by the expression ${}_{A}^{q(B)}Srt$.

We construct new mathematical objects constructively without formalism. By its contradiction, formalism may destroy this thry by Gödel's theorem on the incompleteness of any formal theory. But in the next monograph, we will give the formalism of the theory it's due: the proof of axioms and theorems. Let us introduce the concepts Cha, the capacity measure, and Cca, the measure of its content. Cca is the same as the number of capacity content items. Consider the compression ratios of the dynamic set: q_1=$Sprt_B^A$ answers I compression power of dynamic set A, q_2=$Sprt_B^{q_1}$ –II compression power of dynamic set A, . . . , q_{n+1}=$Sprt_B^{q_n}$ –n+1 compression power of dynamic set A [3]. In contrast to the classical one-attribute set theory, where only its contents are taken as a set, we consider a two-attribute set theory with a set as a capacity and separately with its contents. We introduce the designations: CoQ—the contents of the capacity Q.

1.1 Sprt - elements

Definition 1.1.1. The set of elements $\{g\} = (g_1, g_2, ..., g_n)$ at one point x of space X we shall call Sprt – element, and such a point in space is called capacity of the Sprt – element. We shall denote $Sprt_x^{\{g\}}$.

Definition 1.1.2. $Sprt_x^{\{g\}}$— dynamic set $\{g\}$ at x.

Definition 1.1.3. An ordered set of elements at one point in space is called an ordered Sprt–element.
It's possible to $Sprt_x^{\{g\}}$ correspond to the set of elements $\{g\}$, and the ordered Sprt - element - a vector, a matrix, a tensor, a directed segment in the case when the totality of elements is understood as a set of elements in a segment.

It's allowed to sum Sprt – elements: $Sprt_x^{\{a\}} + Sprt_x^{\{b\}}$ =$Sprt_x^{\{a\}\cup\{b\}}$. *The operator* $Sprt_x^{\{a\}\cup\{b\}}$ is not equal $\{a\} \cup \{b\}$, rather, it is dynamic — contraction of $\{a\} \cup \{b\}$ to the point x [3]. Similarly, for $Sprt_x^{\{a\}\cap\{b\}}$. This is more suitable for using sets for energy space, for any objects. The operator Sprt is adapted for ordinary energies, using their property to overlap.

Capacity in itself

Definition 1.1.4. The capacity A in itself of the first type is the capacity containing itself as an element. Denote S_1fA.

Definition 1.1.5. The capacity A in itself of the second type is the capacity that contains elements from which it can be generated. Denote S_2fA.

An example of the capacity in itself of the first type is a set containing itself. An example of capacity in itself of the second type is a living organism since it contains a program: DNA and RNA.

Definition 1.1.6. Partial capacity A in itself of the third type is the capacity A in itself as an element, which partially contains itself or contains elements from which it can be generated in part or both. Let us denote S_3fA.

Let us introduce the following notations: A*B=Srt_B^A, A^2 = Self A =Srt_A^A, A^3 = Self^2A, ..., A^{n+1} = Self^nA, ... There is no commutativity here: A*B B*A [3]. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

You can consider a more "hard"option: A*B=$PSprt_B^A$, where $PSprt_B^A$ – operator, containing A in every element of B, A^2 = PSelf A =$PSprt_A^A$, A^3 = PSelf^2A, ..., A^{n+1} = PSelf^nA, ... [3]. There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

All capacities in self-space are capacities in themselves by definition. Capacities in themselves can appear as Sprt -capacities and ordinary capacities. In these cases, the usual measures and methods of topology are used.

Connection of Sprt – elements with capacities in themselves [3].
For example, $Srt_{g\{R\}}^{\{R\}}$ is the capacity in itself of the second type if $g\{R\}$ is a program capable of generating $\{R\}$.
Consider a third type of capacity in itself [3]. For example, based on $Srt_x^{\{g\}}$, where $\{g\} = (g_1, g_2, ..., g_n)$, i.e. n - elements at one point, we can consider the capacity S_3f in itself with m elements from $\{g\}$, m<n, which is formed according to the form:

w_{mn} = (m, (n,1)) (1.1)

that is, the structure $Sprt_x^{\{g\}}$ contains only m elements. Form (1.1) can be generalized into the following forms:

$$w^1_{m,n,k} = (\mathrm{k}, (\begin{matrix}(n_1,1)\\(\ldots)\\(n_m,1)\end{matrix})) \quad (1.1.1)$$

or

$$w^2_{m,n,k} = (\mathrm{k},\ (l,\ \begin{pmatrix} (n_1) \\ \ldots \\ (n_m) \end{pmatrix})) \ (1.1.2)$$

$$w^3_{m,n,k,l} = \mathrm{Q}(\begin{pmatrix} d_1 \\ \ldots \\ d_l \end{pmatrix},\ (\begin{pmatrix} (n_1, 1) \\ \ldots \\ (n_m, 1) \end{pmatrix})) \ (1.1.3),$$

where Q(x, y) – any operator, which makes a match between set $\begin{pmatrix} d_1 \\ \ldots \\ d_l \end{pmatrix}$ and set $(\begin{pmatrix} (n_1, 1) \\ \ldots \\ (n_m, 1) \end{pmatrix})$ or

$$w^4_{m,m_1,n_1,m_2,n_2,m_3,n_3} = (\mathrm{m},\ ((m_1, n_1),\ ((m_2, n_2),\ (m_3, n_3)))) \ (1.1.4),$$

or

(Q, R) (1.1.5),

where Q – any, R – any structure, R could be anything can be anything, not just structure. In this case, (1.1.5) can be used as another type of transformation from Q to R. Capacities in themselves of the third type can be formed for any other structure, not necessarily Srt, only by necessarily reducing the number of elements in the structure, in particular, using form

$$w_{m_1 \cdots m_n} = (m_1, (m_2, (\ldots(m_n, 1)\ldots))) \ (1.2)$$

Structures more complex than S_3f can be introduced. For example, through a form that generalizes (1.1):

$$w_{ABC} = (A, (B, C)) \ (1.3)$$

where A is compressed (fits) in C in the compression structure B in C (i.e., in the structure Srt^B_C); or

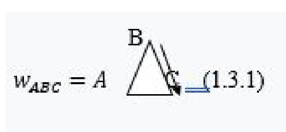


Fig. 0.1

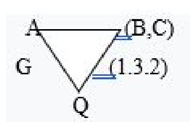


Fig. 0.2

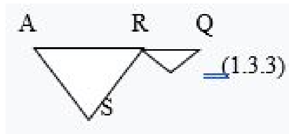


Fig. 0.3

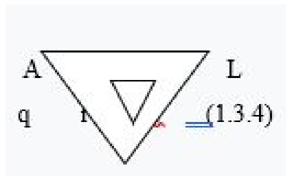


Fig. 0.4

or through the more general form that generalizes (1.2):

$w_{A_1 A_2 \dots A_n C} = (A_1, (A_2, (\dots(A_n, C)\dots)))$ (1.4)

and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc.

(1.3), (1.4) are represented through the usual 2-bond. Science is the discipline of 2-connections, since everything in science is carried out through 2-connected logic, quantum logic is also a projection of 3-connected logic onto 2-connected logic. (1.3.1) - (1.3.4) schematically interpret the formation of the capacity in itself through a pseudo 3-connected form with a 2-connected form. The energy of the self-containment in itself closes on itself.

Math self [3].

Let's consider Sprt arithmetic first:

1. Simultaneous addition of a set of elements $\{g\} = (g_1, g_2, \dots, g_n)$ is carried out using $Sprt^{\{g+\}}$.

2. Similarly, for simultaneous multiplication: $Sprt^{\{g*\}}$: the notation of the set B with elements $b_{i_1 i_2 \dots i_n} = (Sprt_x^{\{g_{1_{i_1}}*, g_{2_{i_2}}*, \dots, \ g_{n_{i_n}}\}})_R$ $for\ any$ $\{i_1, i_2, \dots, i_n\}$ $without\ repetitions, x = {Sprt_w}^{\{K\}}$,K-set of any $\{k_1*, k_2*, \dots, k_n*\}$ without repeating them, k_i-any digit, i=1,2,. . . ,n, R= ${Sprt_w}^{\{i_1+, i_2+, \dots, i_n\}}$, R is the index of the lower discharge (we choose an index on the scale of discharges):

Table I.1. Index on the scale of discharges

index	discharge
n	n
. . .	. . .
1	1

index	discharge
,	0
-1	1st digit to the right of the point
-2	2nd digit to the right of the point
...	...

Then $Sprt_w^{\{B+\}}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. The most straightforward functional scheme of the assumed arithmetic-logical device for Sprt-multiplication:

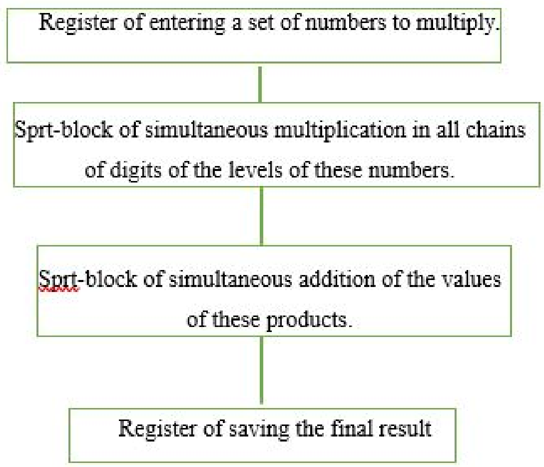


Fig. 0.5 The straightforward functional scheme of the assumed arithmetic-logical device for Sprt-multiplication.

Remark. The algorithm for simultaneously multiplication of a set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by multiplying the first number from the set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the set by the ones following it, etc.

3. Similarly for simultaneous execution of various operations: $Sprt_x^{\{gq\}}$, where $\{q\} = (q_1, q_2, .., q_n)$. q_i -an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $Sprt_x{}^{\{Fg\}}$, where $\{F\} = (F_1, F_2, ..., F_n)$. F_i is an operator, i = 1, . . . , n.

5. The arithmetic itself for capacities in themselves will be similar: addition - $S_1 f\,\{g+\}$, (or $S_3 f\,\{g+\}$) for the third type), multiplication $S_1 f\,\{g*\}$, ($S_3 f\,\{g*\}$).

6. Similarly with different operations: $S_1 f\,\{gq\}$, ($S_3 f\,\{gq\}$), and with different operators: $S_1 f\,\{Fg\}$, ($S_3 f\,\{Fg\}$).

7. $Srt_B^A -$ *the result of the containment operator*. For sets A, B we have

   $Srt_B^A = \left\{ \begin{matrix} D \\ A\cup B - A\cap B \end{matrix} \right\}$, where D is self-set for $A \cap B$. The measure:

$\mu(Srt_B^A)$=($\begin{matrix} \mu^s(\ A\cap B) \\ \mu(A)+\mu(B)-\mu(A\cap B) \end{matrix}$).

   There is the same for structures if it‘s considered as sets. Our approach to the theory of hierarchical sets differs from the construction of hierarchical sets by Y.L. Ershov [4]-[6] : we construct completely different types of hierarchical sets.

8. Sprt-derivative of f(x_1,x_2,...,x_n) is Sprt$\begin{matrix}\left\{\frac{\partial}{\partial x_{1_i}},\frac{\partial}{\partial x_{2_i}},\ldots,\frac{\partial}{\partial x_{k_i}}\right\} \\ f(x_1,x_2,\ldots,x_n)\end{matrix}$, where x=(x_{1_i}, x_{2_i},... x_{k_i})- any set from (x_1,x_2,...,x_n). Let's designate Sprt- $\frac{\partial^k f(x)}{\partial x_{1_i}\partial x_{2_i}\ldots\partial x_{k_i}}$. Sprt-integral off (x_1,x_2,...,x_n) is Sprt$\begin{matrix}\left\{\int ()dx_{1_i},\int ()dx_{2_i},\ldots,\int ()dx_{k_i}\right\} \\ f(x_1,x_2,\ldots,x_n)\end{matrix}$, where ($x_{1_i}$, x_{2_i},... x_{k_i})- any set from (x_1,x_2,...,x_n). Let's designate Sprt-$\int\ldots\int f(x)dx_{1_i}dx_{2_i}\ldots dx_{k_i}$-k-multiple integral. Sprt-lim off (x_1,x_2,...,x_n) is Sprt$\begin{matrix}\left\{\lim\limits_{x_{1_i}\to a_{1_i}},\lim\limits_{x_{2_i}\to a_{2_i}},\ldots,\lim\limits_{x_{k_i}\to a_{k_i}}\right\} \\ f(x_1,x_2,\ldots,x_n)\end{matrix}$. Let's designate

   Sprt- $\begin{pmatrix} \lim\limits_{x_{1_i}\to a_{1_i}} \\ \ldots \\ x_{k_i}\to a_{k_i} \end{pmatrix}$ f(x_1, , x_2, ... , x_n).

   Self-$\lim\limits_{x\to a}$ = $Sprt^{\lim\limits_{x\to a}}_{\lim\limits_{x\to a}}$.

9. In the case of self-derivatives, inclusions of multiple derivatives are obtained. The same is true for self-integrals: we get inclusions of multiple integrals.

10. Let's denote self-(self-Q) through self^2-Q , fS(n,Q)= self-(self-(...(self-Q))) = self^n-Q for n-multiple self.

Operator itself [3].
Definition 7. An operator that transforms $Sprt_x^{\{a\}}$ into any $S_i f\{b\}, i = 2, 3$; where $\{b\} \subset \{a\}$; is the operator itself.

Example. The operator contains the set in itself.

Lim-itself [3].

1. Lim Sprt

   For example, the double limit: $\lim\limits_{\substack{x \to g_1 \\ y \to g_2}} G(x,y)$ corresponds to $Sprt_{(g_1,g_2)}^{\{G(x,y)\}}$.

   Similarly, for lim Sprt with n variables.

In the case of lim-itself, for example, for m variables, it suffices to use the form (1.1) of lim Sprt for n variables (n>m). The same is true for integrals of variables m (for example, the double integral over a rectangular region is through the double limit).

The sequence of actions can be "collapsed" into an ordered Sprt element, and then translate it, for example, into S_3f- the capacity in itself. Take the receipt $\frac{\partial^2 u}{\partial x^2}$ as an example. Here is the sequence of steps 1) $\frac{\partial u}{\partial x}$ 2) $\frac{\partial}{\partial x}(\frac{\partial u}{\partial x})$. "collapses" into an ordered $Sprt_x^{\{\frac{\partial u}{\partial x},\frac{\partial}{\partial x}(\frac{\partial u}{\partial x})\}}$, which can be translated into the corresponding S_1f. The differential operator $Sprt_x^{\{\frac{\partial}{\partial x},\frac{\partial}{\partial x}(\frac{\partial}{\partial x})\}}$ - is interesting too.

Remark1.1.1 Sprt-displacement of A from B will be denote by ${}_A^BSprt$. Then the notation ${}_D^CSprt_B^A$ is both the Sprt-containment of A in B and Sprt-displacement of D from C simultaneously. Let's denote ${}_A^BSprt_B^A$ through $TprS_B^A$, ${}_A^ASprt_A^A - through\ TprS_A^A$.

We can consider the concept of Sprt - element as $Sprt_B^A$, where A fits in capacity B. Then $Sprt_B^B$ it will mean S_1f B

About Sprt and S_3f programming.

The ideology of Sprt and S_3f can be used for programming. Here are some of the Sprt programming operators [3]:

1. Simultaneous assignment of the expression$s$ $\{p\} = (p_1, p_2, ..., p_n)$ to the variables $\{g\} = (g_1, g_2, ..., g_n)$. This is implemented via $Sprt^{\{\{g\}:=\{p\}\}}$.
2. Simultaneous checking the set of conditions $\{f\} = \left(f_1,\ f_2,\ ..., f_n\right)$ for the set of expressions$\{B\} = \left(B_1,\ B_2,\ ..., B_n\right)$. Implemented via $Sprt^{IF\{\{B\}\{f\}\}\ then\ Q}$ where Q can be anything.
3. Similarly for loop operators and others.

S_3f- software operators will differ only in that the aggregates $\{g\}, \{p\}, \{B\}, \{f\}$ will be formed from the corresponding Sprt program operators in form (1.1) and for more complex operators in the form from forms (1.1.1) - (1.4).

The OS (operating system), the computer's principles, and the modes of operation for this programming are interesting. But this is already the material for the following monographs.

Using elements of the mathematics of Srt [3] we introduce the concept of Srt – the change in physical quantity B: $Srt_x^{\{\Delta_1 B,\ldots,\Delta_n B\}}$. Then the mean Srt - velocity will be $v_{cpsrt}(t,\ t) = Srt_x^{\left\{\frac{\Delta_1 B}{\Delta t},\ldots\frac{\Delta_n B}{\Delta t}\right\}}$ and Srt-velocity at time t:$v_{srt} = \lim_{\Delta t\to 0} v_{srt}(t, \Delta t)$. Srt – acceleration $a_{srt} = \frac{dv_{srt}}{dt}$.

In normal use, simply Sprt reduces to a sum at point x of space [3]. When using Sprt with "target weights", we get, depending on the "target weights", one or another modification, namely, for example, the velocity v_{srt}^f(with a "target weight" f in the case when two velocities v_1, v_2 are involved in the set $\{v_1 f, v_2\}\ \ for\ v_{srt}^f = Srt_x^{\{v_1 f, v_2\}}$, f instantaneous replacement we get an instantaneous substitution $v_1 by\ v_2$ at point x of space at time t_0.

Consider, in particular, some examples: 1) $Sprt_{\{x_1,x_2\}}^e$ describes the presence of the same electron e at two different points x_1, x_2. 2) The nuclei of atoms can be considered as Sprt elements.

Similarly, the concepts of Sprt - force and Sprt - energy are introduced [3]. For example, $E_{sprt}^f\ = Sprt_x^{\{E_1 f, E_2\}}$ it would mean the instantaneous replacement of energy E_1 by E_2 at time t_0. Two aspects of Sprt–energy should be distinguished: 1) carrying out the desired "target weight" and 2) fixing the result of it. Do not confuse energy - Sprt (the node of energies) with Sprt – energy that generates the node of energies, usually with the "target weights." In the case of ordinary energies, the energy node is carried out automatically.

Remark1.2. Sprt – elements are all ordinary, but with "target weights," they become peculiar. Here you need the necessary energy to carry them out. As a rule, this energy is at the level of Self. This is natural since it’s much easier to manage elements of the k level via the elements of a more structured k +1 level. Let us consider the concepts of capacities of physical objects in themselves. The question arises about the self-energy of the object. In particular, according to the results of the publication [3]: «Srt_B^B will mean S_1f B.». For example, Sprt_{DNA}^{DNA} allows you to reach the level of DNA self-energy, $Sprt_Q^Q$ allows you to reach the level of self-energy Q. The law of self-energy conservation operates already at the level of self-energy. Also, in addition to capacities in themselves, you can consider the types of containment of oneself in oneself: the first type of the containment of oneself in oneself: the second type of the containment of oneself in oneself: potentially, for example, in the form of programming oneself, the third type is partial containment of oneself in themselves—for example, self-operator, self-action, whirlwind. A container containing itself can be formed by self-containment, i.e., containment in oneself. Let us clarify the concept of the term capacity in itself: it is a capacity containing itself potentially. Consider self-Q, where Q can be anything, including Q=self; in particular, it can be any action. Therefore, self-Q is when Q is made by itself; it makes itself. There is a partial self-Q for any Q with partial self-fulfillment. Let's consider several examples for capacities in themselves: ordinary lightning, electric arc discharge, and ball lightning.

A self-search of the solution of the equations $f_i(x)=0$, where i=1,2,. . . ,n, $x=(x_1,x_2,\ldots,x_n)$, will be realized in $Sprt_a^{\left\{f_1(x)=0?x,f_2(x)=0?x,\ldots,f_n(x)=0?x\right\}}$ or $Sprt_{?x}^{\left\{f_1(x)=0,f_2(x)=0,\ldots,f_n(x)=0\right\}}$. . x acquires more degree of liberty and in this is direct decision [3]. Self-equation for x has its decision for x in direct kind.

The same for $Sprt_?^{\left\{tasks(x)\right\}}$. Self-task for x has its decision for x in direct kind. Self-question has its answer for x in direct kind. $Sprt_{(o,x)}^{\{t\}}$, where $\{t\}$- time points set, (o, x)- object o in point x from space X, give to enter in necessary time moments. The same for $Sprt_o^{\{t\}}$. $Sprt_\alpha^{\{God-father,God-son,Holy\ Spirit\}}$is three-concept representation, where is a point in the connectedness space. Sprt is also great for working with structures, for example: 1) $Sprt_B^{strA}$--the structure A that fits into B, where B can be any capacity, another structure etc. 2) $Sprt_R^{str_Q}$-- embedding structure from Q into R. Similarly for displacement: 1) ${}_{strA}^{B}Sprt$--displacement of structure A from B, 2) ${}_{str_Q}^{B}Sprt$--displacement of the structure Q from B. To work with structures, you can introduce a special operator Cprt: $Cprt_B^{strA}$ structures B with the structure A, $Cprt_R^{str_Q}$ structures R with the structure from Q, ${}_{strA}^{B}Cprt$ destructors B from the structure A, ${}_{str_Q}^{B}Cprt$ destructors B from the structure that structures Q.

Definition 1.1.8. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its elements explicitly, but not necessarily in known operators; it can form (create) new special operators (in particular, special functions).

In particular, $Cprt_{strA}^{strA}$, Crt_{strA}^{strA} are such structures.

Similarly, for working with models, each is structured by its structure; for example, use Sprt-groups, Sprt-rings, Sprt-fields, Sprt-spaces, self-groups, self-rings, self-fields, and self-spaces. Like any task, this is also a structure of the appropriate capacity.

Self-H (self-hydrogen), like other self-particles, does not exist in the ordinary, but all self-molecules, self-atoms, and self-particles are elements of the energy space.

Remark 1.1.3. The concept of elements of physics Sprt is introduced for energy space. The corresponding concept of elements of chemistry Sprt is introduced accordingly. Examples: 1) $SprtE_D^{\{a_1q,a_2\ \}}$ – the energy of instantaneous substitution a_1 by a_2, where a_1, and a_2 are chemical elements, q is instant replacement. Similarly, one can consider for the node of chemical reactions $Sprt_{reaction}^{\{chemical\ elements\ with\ "target\ weights"\}}$. The periodic table itself can also be thought of as the Srt – element: $Srt^{\{list\ of\ chemical\ elements\}}_{Mendeleev\ table}$ The ideology of Sprt elements allows us to go to the border of the world familiar to us, which allows us to act more effectively.

1.2 Dynamic Sprt – elements

We considered stationary Sprt – elements earlier. Here we consider dynamic Sprt – elements [3].

Definition 1.2.1. The process of fitting a set of elements $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ into one point x of space X at time t will be called a dynamic Sprt – element. We will denote $Sprt(t)^{\{g(t)\}}_x$.

Definition 1.2.2. Fitting an ordered set of elements into one point in space is called a dynamic ordered Sprt–element.

It is allowed to sum dynamic Sprt – elements:

$Sprt(t)^{\{a(t)\}}_x + Sprt(t)^{\{b(t)\}}_x = Sprt(t)^{\{a(t)\}\cup\{b(t)\}}_x$.

Dynamic containment of oneself.

Definition 1.2.3. Dynamic capacity Q(t) is fitting into Q(t).

Definition 1.2.4. Dynamic Sprt-capacity $Sprt(t)^{R(t)}_{Q(t)}$ is the process of embedding R(t) into Q(t).

Definition 1.2.5. The dynamic capacity A(t) containing itself as an element of the first type is the process of containing A(t) in A(t). Denote $S_1 f(t) A(t)$.

Definition 1.2.6. Dynamic capacity C(t) in itself of the second type is the process of containing elements from which it can be generated. Let's denote $S_2 f(t) C(t)$.

Definition 1.2.7. Dynamic partial capacity B(t) in itself of the third type is a process of partial containment of B(t) in itself or elements from which it can be generated

partially or both at the same time. Denote $S_3 f(t) B(t)$.

All dynamic capacities in a dynamic self-space are, by definition, dynamic capacities in themselves. Dynamic capacity itself can manifest itself as dynamic Sprt-capacity and ordinary dynamic capacity. In these cases, the usual measures and methods of topology are used.

Connection of dynamic Sprt – elements with dynamic containment of oneself. Consider third type of dynamic partial containment of oneself. For example, based on $Sprt(t)^{\{g(t)\}}$, where $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$, i.e. n – elements at one point x, we can consider the dynamic capacity in itself $S_3 f(t)$ with m elements from $\{g(t)\}$, m<n, which is process formed according to the form (1.1), that is, only m elements from $\{g(t)\}$ are in the structure $Sprt(t)^{\{g(t)\}}$.

Dynamic containment of oneself of the third type can be formed for any other structure, not necessarily Sprt, only through the obligatory reduction in the number of elements in the structure. In particular, using the form from forms (1.1.1) - (1.4).

It is possible to introduce structures more complex than $S_3f(t)$.

Dynamic math itself.

1. The process of simultaneous addition of a set of elements $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ are realized by $Sprt(t)^{\{g(t)+\}}$.

2. By analogy, for simultaneous multiplication: $Sprt(t)^{\{g(t)*\}}$.

3. Similarly for simultaneous execution of various operations:$Sprt(t)^{\{g(t)q(t)\}}$, where $\{q(t)\} = (q_1(t), q_2(t), .., q_n(t))$. $q_i(t)$-an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $Sprt(t)^{\{F(t)g(t)\}}$where $\{F(t)\} = (F_1(t), F_2(t), ..., F_n(t))$. $F_i(t)$ is an operator, i = 1,. . . ,n.

5. Dynamic arithmetic itself for containments of oneself will be similar: dynamic addition - $S_1f(t)\{g(t)+\}$, (or $S_3f(t)\{g(t)+\}$ for the third type), dynamic multiplication $S_1f(t)\{g(t)*\}$, ($S_3f(t)\{g(t)*\}$).

6. Similarly with different operations: $S_1f(t)\{g(t)q(t)\}$, ($S_3f(t)\{g(t)q(t)\}$) and with different operators: $S_1f(t)\{F(t)g(t)\}$, ($S_3f(t)\{F(t)g(t)\}$).

4. 7. $Sprt(t)^{A(t)}_{B(t)}$ *gives* the result $Srt^{A(t)}_{B(t)} = \left\{ \begin{matrix} D(t) \\ A(t) \cup B(t) - A(t) \cap B(t) \end{matrix} \right\}$ for sets $A(t)$, $B(t)$, where D(t) is self-set for $A(t) \cap B(t)$.

The measure: $\mu(Srt^{A(t)}_{B(t)})$=$\left(\begin{matrix} \mu^s(\ A(t) \cap B(t)) \\ \mu\left(A(t)\right) + \mu\left(B(t)\right) - \mu\left(A(t) \cap B(t)\right) \end{matrix} \right)$.

The same is true for structures if they are treated as sets.

Our approach to the theory of hierarchical sets differs from the construction of hierarchical sets by Y.L. Ershov [4]-[6] : we construct completely different types of hierarchical sets.

8. Similarly, for dynamic Sprt-derivatives, dynamic Sprt-integrals, dynamic Sprt-lim, dynamic self-derivatives, dynamic self-integrals

9. Denote dynamic self-(dynamic self-Q(t)) through dynamic self^2-Q(t) , fS(t)(n,Q(t))= dynamic self-(dynamic self-(. . . (dynamic self-Q(t)))) = dynamic self^n-Q(t) for n-multiple dynamic self.

Remark 1.2.1. The dynamic Sprt-displacement of A(t) from B(t) will be denote by ${}^{B(t)}_{A(t)}Sprt(t)$. Then the notation ${}^{C(t)}_{D(t)}Sprt(t)^{A(t)}_{B(t)}$ is dynamic Sprt-containment of A(t) in B(t) and dynamic Sprt-displacement of D(t) from

C(t) simultaneously. Denote ${}^{B(t)}_{A(t)}Sprt(t)^{A(t)}_{B(t)}$ by $TprS(t)^{A(t)}_{B(t)}$, ${}^{A(t)}_{A(t)}Sprt(t)^{A(t)}_{A(t)} - through\ TprS(t)^{A(t)}_{A(t)}$.

We can consider the concept of dynamic Sprt - element as $Sprt(t)^{A(t)}_{B(t)}$, where A(t) fits in dynamic capacity B(t). Then $Sprt(t)^{B(t)}_{B(t)}$ will mean S_1f(t) B(t). Denote S$prt(t)^{B(t)}_{B(t)}$ by pL(t)(B(t)). ${}^{A(t)}_{A(t)}Sprt(t)$ denotes the dynamic expelling A(t) oneself out of oneself, ${}^{A(t)}_{A(t)}Sprt(t)^{A(t)}_{A(t)}$—simultaneous dynamic containment A(t) of oneself in oneself and dynamic expelling A(t) oneself out of oneself. ${}^{A(t)}_{A(t)}Srt$ will be called dynamic anti-capacity from oneself. For example, “white hole” in physics is such simple anti-capacity. The concepts of “white hole” and “black hole” were formulated by the physicists based on the subject of physics –the usual energies level. Mathematics allows you to deeply find and formulate the concept of singular points in the Universe based on the levels of more subtle energies. The experiments of the 2022 Nobel laureates Asle Ahlen, John Clauser, Anton Zeilinger correspond to the concept of the Universe as a capacity in itself. The energy of self-containment in itself is closed on itself [5].

Hypothesis: the containment of the galaxy in oneself as a spiral curl and the expelling out of oneself defines its existence. A self-capacity in itself as an element A is the god of A, the self-capacity in itself as an element the globe—the god of the globe, the self-capacity in itself as an element man--the god of the man, the self-capacity in itself as an element of the universe--the god of the universe, the containment of A into oneself is spirit of A, the containment of the Earth into oneself is is spirit of Earth, the containment of the man into oneself is spirit of the man (soul), the containment of the universe into oneself is spirit of the universe. We may consider the following axiom: any capacity is the capacity of oneself. This is for each energy capacity.
Remark 1.2.1.1. All constructions of Part I can be similarly extended to the more general construction $SCprt(t)\,g(t)\,\begin{matrix}A(t)\\ \\ B(t)\end{matrix}$, where A(t) fits in dynamic capacity B(t) with type of accommodation g(t), for which the specific features will be considered in the next monograph.
About dynamic Sprt and S_3f(t) programming.

The ideology of dynamic Sprt and S_3f(t) can be used for programming [5]:

1. The process of simultaneous assignment of the expression$s$ $\{p(t)\} = (p_1(t), p_2(t), ..., p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ is implemented through $Sprt(t)^{\{\{g(t)\}:=\{p(t)\}\}}$.

2. The process of simultaneous check the set of conditions $\{f(t)\} = (f(t)_1,\ f_{2,}\ (t)\ ..., f(t)_n)$ for a set of expressions$\{B(t)\} =$

$\left(B_1(t),\ B_2\ (t),\ ...,B_n(t)\right)$ is implemented through $Sprt(t)^{IF\left\{\{B(t)\}\{f(t)\}\right\}\ then\ Q(t)}$ where Q(t) can be any.

3.Similarly for loop operators and others.

$S_3f(t)$– software operators will differ only in that the aggregates $\{g\},\{p\},\left\{B(t)\right\},\left\{f(t)\right\}$ will be formed from corresponding processes Sprt(t) for the above-mentioned programming operators through form (1.1) or form from forms (1.1.1) - (1.4) for more complex operators.

Remark 1.2.2. With the help of dynamic Sprt-elements, the concepts of dynamic Sprt - force, dynamic Sprt – energy are introduced. For example, $E(t)^f_{sprt}\ = Sprt(t)^{\left\{E_1(t)f,E_2(t)\right\}}_{x(t)}$ will mean the process of instantaneous replacement f of energy $E_1(t)$ by $E_2(t)$ at time t. Similarly, using $S_if(t)$, the concepts of $S_if(t)$-force, $S_if(t)$-energy, i=1,2,3, and etc are introduced.

Remark 1.2.3. It is the containment of oneself in oneself that can "give birth" to the capacities in itself – that is what self-organization is.

Remark 1.2.4. $Sprt(t)^{St(t)^{B(t)}_{B(t)}}_{St(t)^{B(t)}_{B(t)}}$ can increase self- level of B(t).

Remark 1.2.5. For example, the operator itself [3],[4] is $S_1f(t)$.

Remark 1.2.6. May be considered the following derivatives: $\frac{dSprt(t)^{A(t)}_{B(t)}}{dt}$, $\frac{d^{B(t)}_{A(t)}Sprt(t).}{dt}$, $\frac{d^{C(t)}_{D(t)}Sprt(t)^{A(t)}_{B(t)}}{dt}$, $\frac{dSif(t)}{dt}$, i=1,2,3.

Remark 1.2.7. It is the containment of oneself in itself as an element that can be interpreted as dynamic capacities in itself.

Remark 1.2.8. Not every capacity containing itself as an element will manifest itself as a sedentary capacity or capacity.

3. **Sprt – elements for continual sets**

Earlier, we considered finite-dimensional discrete Sprt-elements and self-capacities in itself as an element. Here we believe some continual Sprt-elements and continual self-capacities in themselves as an element [3].

Definition 1.3.1. The set of continual elements $\{g\} = (g_1, g_2, ..., g_n)$ at one point x of space X will be called continual Sprt – element, and such a point in space will be called capacity of the continual Sprt – element. We will denote $Sprt^{\{a\}}_x$.

Definition 1.3.2. An ordered set of continual elements at one point in space is called an ordered continual Sprt–element.

It's allowed to sum continual Sprt – elements: $Sprt^{\{a\}}_x + Sprt^{\{b\}}_x = Sprt^{\{a\}\cup\{b\}}_x$, where some or any elements may be ordered elements.

Definition 1.3.3. The continual self-capacity A in itself as an element of the first type is the capacity containing itself as an element. Denote S_1fA.

Definition 1.3.4. The ordered continual self-capacity A in itself as an element of the first type is the ordered capacity containing itself as an element. Denote $\overrightarrow{S_1fA}$.

For example, S_∞^+= sin∞ is of this type. It denotes continual ordered self-capacities in itself as an element of following type—the range of simultaneous "activation" of numbers from [-1,1] in mutual directions: $\uparrow I \downarrow_{-1}^{1}$. Also consider the following elements: S_∞^-=sin(-∞) $- -\downarrow I \uparrow_{-1}^{1}$, T_∞^+= tg∞ $- -\uparrow I \downarrow_{-\infty}^{\infty}$, T_∞^-=tg(-∞) $- -\downarrow I \uparrow_{-\infty}^{\infty}$, don't confuse with values of these functions. Such elements can be summarized. For example: aS_∞^++bS_∞^-=(a-b)S_∞^+ = $(b-a)$ S_∞^-.

Definition 3.5. The continual self-capacity A in itself, as an element of the second type, is the capacity containing elements from which it can be generated. Let's denote S_2fA.

An example of continual self-capacity in itself as an element of the second type is a living organism since it contains the programs: DNA and RNA.

Definition 1.3.6. Partial continual self-capacity in itself as an element of the third type is called continual self-capacity in itself as an element that partially contains itself or contains elements from which it can be generated in part or both simultaneously. Denote S_3f.

Also, may be considered operators for them. For example:

$$\mathrm{f}S_\infty^+(\mathrm{t}\text{-}\mathrm{t}_0) = \begin{cases} S_\infty^+\ ,\ t = t_0 \\ 0,\ t \neq\ t_0 \end{cases}$$

You can set the self- vector: $\uparrow \begin{pmatrix} q \\ d \\ c \end{pmatrix} \downarrow$

All continual capacities in self-space are continual self-capacities in itself as an element by definition. The continual self-capacities in itself as an element may appear as continual Srt- capacities and usual continual capacities. In these cases, there are used typical measure and topology methods.

The connection of continual Sprt − elements with continual self-capacities in themselves as an element.
Consider a third type of continual self-capacity in itself as an element. For example, based on $Sprt_x^{\{g\}}, where\,\{g\} = (g_1, g_2, ..., g_n)$, i.e. g_i - continual elements at one point, i=1, 2, . . . , n. The continual self-capacity in itself as an element with m continual elements from $\{g\}$, at m<n, can be considered as S_3f, which is formed by the form (1.1), i.e., only m continual elements are located in the structure $Sprt_x^{\{g\}}$. Continual self-capacities in itself as an element of the third type can be formed for any other structure, not necessarily Sprt, only by obligatory reducing the number of continual elements in the

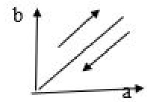


Fig. 0.6 Self-vector $\overset{\rightarrow}{\underset{\leftarrow}{[(0,0),(a,b)]}}$

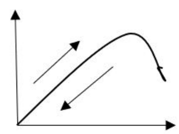


Fig. 0.7 Self-ordered curve

structure. In particular, using the form from forms (1.1.1) - (1.4). Structures more complex than S_3f can be introduced.

Mathematics itself for continual elements.

1.Simultaneous addition of the continual elements of the set $\{g\} = (g_1, g_2, ..., g_n)$ is implemented using $Sprt^{\{g\cup\}}$.

2. By analogy, for simultaneous multiplication: $Sprt^{\{g\cap\}}$.

3. Similarly, for simultaneous execution of various operations: $Sprt_x^{\{gq\}}$, where $\{q\} = (q_1, q_2, .., q_n)$. q_i -an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $Sprt_x{}^{\{Fg\}}$, where $\{F\} = (F_1, F_2, ..., F_n)$. F_i is an operator, i = 1, . . . , n.

5. For continual self-capacities in themself as an element will be similar: addition - $S_1f\,\{g+\}$, (or $S_3f\,\{g+\}$) for the third type), multiplication $S_1f\,\{g*\}$, ($S_3f\,\{g*\}$).

6. Similarly with different operations: $S_1f\,\{q\}$, ($S_3f\,\{q\}$), and with different operators: $S_1f\,\{F\}$, ($S_3f\,\{F\}$).

7. $Srt_B^A - \ is$ the result of the containment operator. For sets A, B we have

$Srt_B^A = \left\{ \begin{matrix} D \\ A\cup B - A\cap B \end{matrix} \right\}$, where D is self-set for $A \cap B$. The measure:

$$\mu(Srt_B^A) = (\begin{matrix} \mu^s(\ A\cap B) \\ \mu(A)+\mu(B)-\mu(A\cap B) \end{matrix}).$$

There is the same for structures if it‘s considered as continual sets. Our approach to the theory of hierarchical sets differs from the construction of hierarchical sets by Y.L. Ershov [4]-[6]: we construct completely different types of hierarchical sets.

Remark 1.3.1. Sprt-displacement of A from B will be denoted by ${}_A^BSprt$. Then the notation ${}_D^CSprt_B^A$ is Sit-containment of A in B and Sprt-displacement of D from C simultaneously. Denote ${}_A^BSprt_B^A$ by $TprS_B^A$, ${}_A^ASprt_A^A - through\ TprS_A^A$. Three in one is $Sprt_{\propto}^{\{\infty\ in\ itself,\ an\ element\ that\ is\ not\ anyone's\ element,\ \ 0\ out\ oneself\}}$, - point space connectedness.

We can consider the concept of a continual Sprt - element as $Sprt_{B}^{A}$, where A fits in continual capacity B. Then $Sprt_{B}^{B}$ will mean S_1f B. These elements are used for Sprt-coding, Sprt translation, coding self, and translation self for networks], which is suitable for electric current of ultrahigh frequency. More complex elements can be considered as continual sets of numbers with their " activation " in mutual directions. For example, ranges of function values, particularly those representing the shape of lightning. Differential geometry can be applied here. Also, n-dimensional elements can be considered. The space of such elements is Banach space if we introduce the usual norm for functions or vectors. We call this space-- Selb-space. Then we introduce the scalar product for functions or vectors and get the Hilbert space. We call this space Selh-space. In particular, one can try to describe some processes with these elements by differential equations and use methods from [7]. You can also try to optimize and research some processes with these elements using the techniques from [8]. Let's introduce operators for transforming capacity to self-capacity in itself as an element: Q_1S(A) transforms A to S_1fA, Q_0S(A) transforms A to ${}_{A}^{A}Sprt$, SO(A) transforms A to$\uparrow A\downarrow$, $\uparrow A\downarrow$ -- ordered self-capacity in itself as an element of simultaneous "activation" of all elements of A in mutual directions. For example, SO([-1,1]) = S_{∞}^{+}, SO([1,-1])= S_{∞}^{-} , SO([-∞,∞]) = T_{∞}^{+}, SO([$\infty,-\infty$]) = T_{∞}^{-}. The operator $(Q_1S(A))^2$ increases self-level for A: it transforms Self-A= S_1fA to self^2-A, $(Q_1S(A))^n \to$ self^n-A, $e^{Q_1S(A)} \ \to\ e^{self} - A$. Let us introduce the following notations: ${}_{A}^{\{\ \}}Sprt$ by os({ } →)elf,

${}_{(A,A)}^{A}Sprt$ by 2oself-A, $Sprt_{A}^{(A,A)}$ by 2self-A, $Sprt_{(A,A)}^{A}$ by 1/2self-A, $Sprt_{q(A)}^{A}$ by qself-A, ${}_{q(A)}^{A}Sprt$ by q()oself-A, q-any operator, ${}_{(q_1,\ \ldots,\ q_N)}^{A}Sprt$ by Noself-A, q_i= A, i= 1, . . . , N; ${}_{A}^{A}Sprt_{A}^{A}$ by self-A- oself-A, ${}_{q_3(A)}^{q_2(A)}Sprt_{q_1(A)}^{A}$ by $q_1$self-A- $\begin{pmatrix} q_3() \\ q_2() \end{pmatrix}$oself-A, ${}_{(A,A)}^{A}Cprt$ by 2Coself-A, $Cprt_{A}^{(A,A)}$ by 2Cself-A, $Cprt_{(A,A)}^{A}$ by 1/2Cself-A, $Ct_{q(A)}^{A}$ by qCself-A, ${}_{q(A)}^{A}Cprt$ by q()Coself-A, q-any operator, ${}_{(q_1,\ \ldots,\ q_N)}^{A}Cprt$ by NCoself-A, q_i= A, i= 1, . . . , N; ${}_{A}^{A}Cprt_{A}^{A}$ by Cself-A- Coself-A, ${}_{q_3(A)}^{q_2(A)}Cprt_{q_1(A)}^{A}$ by q_1Cself-A- $\begin{pmatrix} q_3() \\ q_2() \end{pmatrix}$ Coself-A, $S2prt_{A}^{A}$= (self-A, self-A), $SNprt_{A}^{A}$= (q_1, . . . , q_N), q_i= self-A, i= 1, . . . , N. self($Sprt_{B}^{A}$)= $St_{Sprt_{B}^{A}}^{Sprt_{B}^{A}}$. Can be considered Q(${}_{A}^{A}Sprt_{A}^{A}$), Q-any operator.

1.4 Dynamic continual Sprt − elements

Definition 1.4.1. The process of containing the set of continual elements $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ at one point x of the space X at time will be called the dynamic continual Sprt − element. We will denote $Sprt(t)_{x}^{\{g(t)\}}$.

Definition 1.4.2. The process of containing an ordered set of continual elements at one point in space is called dynamic continual ordered Sprt − element.

It is allowed to sum dynamic continual Sprt − elements:

$Sprt(t)_x^{\{a(t)\}} + Sprt(t)_x^{\{b(t)\}} = Sprt(t)_x^{\{a(t)\}\cup\{b(t)\}}$.

Dynamic continual containment of oneself in oneself as an element.

Definition 1.4.3. The dynamic continual Sprt-capacity $Sprt(t)_{Q(t)}^{R(t)}$ is called the process of embedding R(t) in Q(t).

Definition 1.4.4. The dynamic containment continual A(t) of oneself of the first type is the process of putting A(t) into A(t). Denote $S_1 f(t) A(t)$.

Definition 1.4.5. The dynamic containment continual C(t) of oneself of the second type embedding contains the continual elements from which it can be generated. Denote $S_2 f(t) C(t)$.

Definition 1.4.6. The partial dynamic containment continual B(t) of oneself of the third type is the process of partial embedding continual B(t) into oneself or continual elements from which it can be generated in part or both simultaneously. Denote $S_3 f(t) B(t)$.

The connection of dynamic continual Sprt − elements with dynamic containment of oneself in oneself as an element [3].
Let us consider the partial dynamic continual containment of oneself in oneself as an element of the third type. For example, based on $Sprt(t)_x^{\{g(t)\}}$, where $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$, i.e. n - continual elements at one point x, one can consider the dynamic containment $S_3 f(t)$ of oneself in oneself as an element with m continual elements from $\{g(t)\}$, m<n, which is a process that is necessary form according to the form (1.1), i.e., only m continual elements from $\{g(t)\}$ are located in the structure $Sprt(t)_x^{\{g(t)\}}$. Dynamic continual containments of oneself in oneself as an element of the third type can be formed for any other structure, not necessarily Sprt, only by necessarily reducing the number of continual elements in the structure. In particular, with the help of the form from forms (1.1.1) - (1.4). It is possible to introduce structures more complex than $S_3 f(t)$.

Dynamic continual mathematics self.

1. The process of simultaneous addition of the set of continual elements $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ is realized by $Sprt(t)^{\{g(t)\cup\}}$.

2. By analogy, for simultaneous multiplication: $Sprt(t)^{\{g(t)\cap\}}$.

3. Similarly for simultaneous execution of various operations: $Sprt(t)^{\{g(t)q(t)\}}$, where $\{q(t)\} = (q_1(t), q_2(t), .., q_n(t))$. $q_i(t)$-an operation, i = 1, . . . , n.

5. Similarly, for the simultaneous execution of various operators: $Sprt(t)^{\left\{F(t)g(t)\right\}}$where $\left\{F(t)\right\} = (F_1(t), F_2(t), ..., F_n(t))$. $F_i(t)$ is an operator, i = 1, . . . , n.
6. The dynamic arithmetic self for the dynamic continual containments of oneself will be similar: dynamic addition - $S_1 f(t)\left\{g(t)\cup\right\}$, (or $S_3 f(t)\left\{g(t)\cup\right\}$ for the third type), dynamic multiplication $S_1 f(t)\left\{g(t)\cap\right\}$, $(S_3 f(t)\left\{g(t)\cap\right\})$.
7. Similarly with different operations: $S_1 f(t)\left\{g(t)q(t)\right\}$, $(S_3 f(t)\left\{g(t)q(t)\right\})$, and with different operators: $S_1 f(t)\left\{F(t)g(t)\right\}$, $(S_3 f(t)\left\{F(t)g(t)\right\})$.
8. $Sprt(t)^{A(t)}_{B(t)}$ *gives* the result $Srt^{A(t)}_{B(t)} = \left\{\begin{matrix} D(t) \\ A(t)\cup B(t) - A(t)\cap B(t)\end{matrix}\right\}$ for continual sets $A(t),\ B(t)$, where D(t) is self-set for $A(t)\cap B(t)$.

The measure: $\mu(Srt^{A(t)}_{B(t)})$=($\begin{matrix}\mu^s(\ A(t)\cap B(t)) \\ \mu\left(A(t)\right)+\mu\left(B(t)\right)-\mu\left(A(t)\cap B(t)\right)\end{matrix}$.

There is the same for structures if it‘s considered as continual sets. Our approach to the theory of hierarchical sets differs from the construction of hierarchical sets by Y.L. Ershov [4]-[6] : we construct completely different types of hierarchical sets.

9. Similarly, for dynamic Sprt-derivatives, dynamic Sprt-integrals, dynamic Sprt-lim, dynamic self-derivatives, dynamic self-integrals
10. Denote dynamic continual self-(dynamic continual self-Q(t)) through dynamic continual self^2-Q(t), fS(t)(n,Q(t))= dynamic continual self-(dynamic continual self-(. . . (dynamic continual self-Q(t)))) = dynamic continual self^n-Q(t) for n-multiple dynamic continual self.

Remark 1.4. Dynamic continual Sprt-displacement of A(t) from B(t) will be denote through ${}^{B(t)}_{A(t)}Sprt(t)$. Then the notation ${}^{C(t)}_{D(t)}Sprt(t)^{A(t)}_{B(t)}$ is dynamic continual Sprt- embedding of A(t) in B(t) and dynamic continual Sprt-displacement of D(t) from C(t) simultaneously. Denote ${}^{B(t)}_{A(t)}Sprt(t)^{A(t)}_{B(t)}$ by $TprS(t)^{A(t)}_{B(t)}$, ${}^{A(t)}_{A(t)}Sprt(t)^{A(t)}_{A(t)} - through\ TprS(t)^{A(t)}_{A(t)}$.

We can consider the concept of dynamic continual Sprt - element as $Sprt(t)^{A(t)}_{B(t)}$, where A(t) fits in dynamic continual capacity B(t). Then $Sprt(t)^{B(t)}_{B(t)}$ it will mean S_1f(t) B(t). Denote Spr$t(t)^{B(t)}_{B(t)}$ by pL(t)(B(t)). ${}^{A(t)}_{A(t)}Sprt(t)$ denotes the dynamic continual displacement of A(t) from itself, ${}^{A(t)}_{A(t)}Sprt(t)^{A(t)}_{A(t)}$— simultaneous dynamic continual containment of oneself A(t) in oneself A(t) and dynamic continual expelling oneself A(t) out of oneself A(t). ${}^{A(t)}_{A(t)}Sprt$ will be called dynamic continual anti capacity from itself.

Connection of dynamic continual Sprt – elements with target weights with dynamic continual containment of oneself with target weights [3].
Consider a third type of dynamic partial containment of oneself with target weights g(t) [3]. For example, based on $Sprt(t)^{\{w(t)g(t)\}}$, where $\{w(t)\} = (w_1(t), w_2(t), ..., w_n(t))$, i.e. n - continual elements with target weights{g(t)} at one point x, we can consider the dynamic containment $S_3 f(t)g(t)$ of oneself with target weights with m continual elements with target weights{g(t)} from $\{g(t)\}$, m<n, which is the process of formation according to the form (1.1), i.e., only m continual elements with target weights{g(t)} from $\{w(t)\}$ are located in the structure $S_3 f(t)g(t)$. Dynamic containments of oneself with target weights of the third type can be formed for any other structure, not necessarily Sprt, only by reducing the number of continual elements with target weights in the structure. In particular, using the form from forms (1.1.1) - (1.4). Structures more complex than $S_3 f(t)g(t)$ can be introduced.

Definition 1.4.8. The dynamic embedding of continual A(t) into itself with target weights {g(t)} of the first type is the process of embedding A(t) into A(t) with target weights. Denote $S_1 f(t)A(t)g(t)$.

Definition 30. The dynamic containment of continual C(t) itself into itself with target weights{g(t)} of the second type is the process of containment of the continual elements from which it can be generated. Let's denote $S_2 f(t)C(t)g(t)$.

Definition 1.4.9. Partial dynamic containment of continual B(t) itself into itself with target weights {g(t)} of the third type is the process of partial containment of continual B(t) into itself or continual elements from which it can be generated partially, or both at the same time. Denote $S_3 f(t)B(t)g(t)$.

1.5 The usage of Sprt-elements for networks

A. Galushkin's comprehensive monograph [9] covers all aspects of networks, but traditional approaches go through classical mathematics, mainly through the usual correspondence operators. Here we consider a different approach - through a new mathematical process with containment operators, which, although they can be interpreted as the result of some correspondence operators, are not themselves correspondence operators [3]. Containment operators are more convenient for networks. Also, the main emphasis was placed on using processors operating using triodes, which are generally not used in Sprt-networks. Sprt-networks (Smnsprt) are a Sprt-structure that can be built for the required weights. Sprt-OS (Sprt operating system) uses Sprt-coding and Sprt-translation. In the first one, coding is carried out through a 2-dimensional matrix-row (a, b), where the number b is the code of the action, and the number a is the code of the object of this action. Sprt-coding (or self-coding) is implemented through a matrix consisting of 2 columns (in the continuous case, two intervals of numbers). Here, the source encoding is used for all matrix rows simultaneously. Sprt-translation is carried out by inversion. In this case, self-coding and self-translation will be more stable. The target weights f_i in $Sprt_a^{\{fx\}}$ are chosen for necessary tasks. We will not touch on the issues of applications, or network optimization. They are described in detail by Galushkin [9]. We will touch on the difference of this only for hierarchical complex networks. The same simple executing programs are in the cores of simple artificial neurons of type Sprt (designation - mnSprt) for simple information processing. More complex executing programs are used for mnSprt nodes. Sprt-threshold element –sgn($Sprt_b^{\{ax\}}$), b- mnSprt, x=(x_1,$x_{2,\ldots}$,x_n) – source signals values, a=(a_1,a_2,. . . ,a_n) – Sprt-synapses weights. The first level of mnSprt consists of simple mnSprt. The second level of mnSprt consists of $Sprt_D^{\{mnSprt\}}$ – Sprt-node of mnSprt in range D, D- capacity for mnSprt node. The third level of mnSprt consists of $Sprt_D^{\{Sprt_D^{\{mnSprt\}}\}}$- $Sprt^2$- node of mnSprt in range D, thus D becomes capacity of itself in itself as an element for mnSprt. For our networks, it is sufficient to use $Sprt^2$- nodes of mnSprt, but self-level is higher in living organisms, particularly $Sprt^n$-, with n $\geqslant$ 3. The target structure or the corresponding program enters the target unit using alternating current. After that, all networks or parts of them are activated according to the indicative goal. It may appear that we are leaving the network ideology, but these networks are a complex hierarchy of different levels, like living organisms.

Remark 1.5. Traditional scientific approaches through classical mathematics make it possible to describe only at the usual energy level. Here we consider an approach that makes describing processes with finer energies possible. mnSprt contains $Sprt_{mnSprt}^{\{eprograms\}}$, $eprogram-$executing program in Sprt- OS. Sprt-OS (or Self-OS) is based on Sprt-assembly language (or Self-assembly language), which is based on assembly language through Sprt-approach in turn, if the base of elements of Sprt-networks is sufficient. The eprograms are in Sprt-programming environments (or Self-programming environments), but this question and Sprt-networks base will be considered in the following monographs. In particular, eprograms may contain Sprt- programming operators. In mnSprt cores, the constant memory Sprt with correspondent eprograms depending on mnSprt.

The OS (operating system) and the principles and modes of operation of the Sprt-networks for this programming are interesting. But this is already the material for the next publications.

Here is developed a helicopter model without a main and tail rotors based on Sprt – physics and special neural networks with artificial neurons operating in normal and Sprt-modes. Let's denote this model through SmnSprt. To do this, it’s proposed to use mnSprt of different levels; for example, for the usual mode, mnSprt serves for the initial processing of signals and the transfer of information to the second level, etc., to the nodal center, then checked. In case of an anomaly - local Sprt–mode with the desired "target weight" is realized in this section, etc., to the center. In the case of a monster during the test, Smnsprt is activated with the desired "target weight". Here are realized other tasks also. To reach the self-energy level, the mode $Sprt_{Smnsprt}^{Smnsprt}$ is used. In normal mode, it's planned to carry out the movement of Smnsprt on jet propulsion by converting the energy of the emitted gases into a vortex to obtain additional thrust upwards. For this purpose, a spiral-shaped chute (with "pockets") is arranged at the bottom of the SmnSprt for the gases emitted by the jet engine, which first exit through a straight chute connected to the spiral one. There is drainage of exhaust gases outside the SmnSprt. SmnSprt is represented by a neural network that extends from the center of one of the main clusters of Sprt - artificial neurons to the shell, turning into the body itself. Above the operator's cabin is the central core of the neural network and the target block, responsible for performing the "target weights" and auxiliary blocks, the functions and roles of which we will discuss further. Next is the space for the movement of the local vortex. The unit responsible for SmnSprt's actions is below the operator's cab. In Sprt – mode, the entire network or its sections are Sprt – activated to perform specific tasks, in particular, with "target weights." In the target, block used Sprt-coding, Sprt-translation for activation of all networks to "target weights" simultaneously, then –the reset of this Sprt-coding after activation. Unfortunately, triodes are not suitable for Sprt -neural networks. In the most primitive case, usual separators with corresponding resistances and cores for eprograms may be used instead triodes since there is no necessity to unbend the alternating current to direct. The Sprt-operative memory belt is disposed around a central core of SmnSprt. There are Sprt-coding, Sprt-translation, and Sprt-realize of eprograms and the programs from the archives without extraction, Sprt-coding and Sprt-translation may be used in high-intensity, ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna, Strickland. Sprt – structure or an eprogram if one is present of needed «target weight» are taken in target block at Sprt – activation of the networks. $Sprt_{activation}^{SmnSprt,f}$ derives SmnSprt to the self-level boundary with target weight f.

It's used an alternating current of above high frequency and ultra-violet light, which can work with Sprt – structures in Sprt–modes by its nature to activate the networks or some of its parts in Sprt–modes and locally using Sprt–mode. Above high frequently alternating current go through mercury bearers. That's why overheating does not occur. The power of the alternating current above high frequently increases considerably for the target block. The activation of all networks is realized to indicate "target weights."

1.6 Variable hierarchical dynamical structures (models) for

dynamic, singular, hierarchical sets

In contrast to the classical one-attribute set theory, where only its contents are taken as a set, we consider a two-attribute set theory with a set as a capacity and separately with its contents [3]. We simply use a convenient form to represent the singularity of a set. Articles [3], [10-19] use the following methodology for permanent structures:

1. Cancellation of the axiom of regularity

2. 2 attributes for the set: capacity and its content

3. Compression of a set, for example, to a point

4. “turning out” from one another, particularly from a capacity, we pull out another capacity, for example, itself, as its element.

5. The simultaneity of one (compression) and the other (“eversion”)

6. Own capacities

7. Qualitatively new programming and Networks.

Here we will consider variable structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable structures (models), for example,

$$ {}_D^C Sprt(t)_B^A = \begin{cases} {}_D^C Sprt, & q_2 \geq t \geq q_1 \\ {}_D^B S^1 t_B^A, q_3 \geq & t > q_2 \\ {}_D^C Sprt_B^A, q_4 \geq & t > q_3 \\ Sprt_B^A\ , q_5 \geq & t > q_4 \\ {}_D^{\{\ \}} Sprt, & t > q_5 \\ \ldots \end{cases} \quad (*_{6.1}), $$

${}_D^B S^1 t_B^A$ is considered in [10]. In particular, ${}_D^B Sprt_B^A$ can be interpreted as a game: player 1 fits A into B, and the other pushes D out of B at the same time.

Can be considered N-hierarchical structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, ... can be any in particular, by actions, sets, and others.

$$ {}_D^C Sprt_B^A : \left\langle \begin{matrix} A \to B \\ A, B \end{matrix} \middle| \begin{matrix} D \leftarrow C \\ C, D \end{matrix} \right\rangle \to \begin{pmatrix} self(A \to B) \\ A, B \end{pmatrix} $$

$$ {}_D^C Sprt_B^A : \left\langle \begin{matrix} A \to B \\ A, B \end{matrix} \middle| \begin{matrix} D \leftarrow C \\ C, D \end{matrix} \right\rangle \to \begin{pmatrix} oself(D \leftarrow C) \\ C, D \end{pmatrix} $$

Can be considered discrete hierarchical structure, continuous hierarchical structure, and discrete-continuous hierarchical structure [3], $Sprt_x^{N-hierarchical\ structure}$

The example

$$\mathrm{H}Sprt_x{}^{Sprt_x^{1-level\ of\ hierarchical\ structure}{}^{\cdots}{}^{(Sprt_x^{i-level\ of\ hierarchical\ structure})}{}^{\cdots}{}^{Sprt_x^{N-level\ of\ hierarchical\ structure}}}$$

-- N-hierarchical structure compression into point x.

Let f(N, QHSpr)= $QHSpr^{QHSpr^{QHSpr^{\cdots QHSpr}}}\Big\}$-N levels

It can be considered self- QHSpr, f(y, QHSpr) for any y, f(QHSpr, QHSpr).

Compression Hierarchy Examples:

1) $Sprt^{Sprt^{(\ )}_{(\ )}}_{Sprt^{(\ )}_{(\ )}+B} = \begin{pmatrix} Sprt^{Sprt^{(\ )}_{(\ )}}_{Sprt^{(\ )}_{(\ )}} \\ Sprt^{Sprt^{(\ )}_{(\ )}}_{B} \end{pmatrix}$

2) $^{+^{(\ )}_{(\ )}Sprt^{(\ )}_{(\ )}}_{D+^{(\ )}_{(\ )}Sprt^{(\ )}_{(\ )}}\ S_1t^{A+^{(\ )}_{(\ )}Sprt^{(\ )}_{(\ )}}_{B+^{(\ )}_{(\ )}Sprt^{(\ )}_{(\ )}} = \begin{pmatrix} ^{(\ )}_{(\ )}Sprt^{(\ )}_{(\ )} \\ ^{(\ )}_{(\ )}Sprt^{(\ )}_{(\ )} \end{pmatrix} S_1t^{^{(\ )}_{(\ )}Sprt^{(\ )}_{(\ )}}_{^{(\ )}_{(\ )}Sprt^{(\ )}_{(\ )}} \\ {}_D\, S_1t^A_B \end{pmatrix}$

The example of variable hierarchy

$$ {}^C_D Sprt(t)_1{}^A_B = \begin{cases} \left\{ \begin{matrix} Q +^{\{\ \}}_{D-D\cap C}\ Sprt \\ (\ C-D\cap C)-(D-D\cap C) \end{matrix} \right\}, \quad q_2 \geq t \geq q_1 \\ (\begin{matrix} S^{1}{}^{e}_{0} fB* \\ {}^{B}_{Q-B}S^1t^{A-B}_{B} \end{matrix}), q_3 \geq\ t > q_2 \\ (\begin{matrix} S^{et}_{01}fB \\ {}^{C-B}_{C-B}S_1t^{A-B}_{B} \\ {}^{C-B}_{D-C-B}S_1t^{A-B}_{B} \end{matrix}), \quad, q_4 \geq\ t > q_3 \\ \begin{pmatrix} R \\ A\cup B - A\cap B \end{pmatrix}, q_5 \geq\ t > q_4 \\ {}^{\{\ \}}_{D}Sprt, \quad t > q_5 \\ \cdots \end{cases} \qquad (*_{6.2}),$$

where Q is oself-set for $D \cap C$ [10], R is self-set for $A \cap B$ [3], $S^{et}_{01}fB$, ${}^{C-B}_{C-B}S_1t^{A-B}_{B}$, ${}^{C-B}_{D-C-B}S_1t^{A-B}_{B}$ are considered in [11], ${}^{B}_{Q-B}S^1t^{A-B}_{B}$ is considered in [12].

1.7 Applications Dynamic Sets Theory to physics and chemistry

Objects of physics and chemistry have an energy structure, which can be tried to be represented in the form of a hierarchical energy structure: the upper level of subtle self-energy and the lower level, which is manifested in the form of objectivity [3].

Ordinary types of energy are manifestations of a lower level from these structures.

If we represent an amorphous body with a mathematical structure of self-object $Sprt^{A_0+E_s}_{A_0+E_s}$, where $Sprt^{A_0}_{A_0}$ - level of objectivity of an amorphous object, ($Sprt^{A_0}_{A_0+E_s}+ Sprt^{A_0+E_s}_{A_0}$) - the energy of connections between the level of subtle energy $Sprt^{E_s}_{E_s}$ and the level of objectivity [3].

Thus, one can try to conventionally represent the mathematical model of the energy structure of an amorphous object as a hierarchical dynamic operator

$$\begin{array}{l} \qquad\quad Sprt^{E_s}_{E_s} \\ (\ Sprt^{A_0}_{A_0+E_s} + Sprt^{A_0+E_s}_{A_0}) \ (1.7.1). \\ \qquad\quad Sprt^{A_0}_{A_0} \end{array}$$

In particular, the magnetic field and spin belong to the second level in (1.7.1) [3].

The next level of objectivity responds to a crystal. We represent a crystal with a mathematical structure

$$Sprt^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}} \ (1.7.2).$$

Thus, one can try to conventionally represent the mathematical model of the energy structure of a crystal as a hierarchical dynamic operator (1.7.2). The next level of objectivity responds to a living crystal, for example, the bone of a living organism, a nail, viruses, DNA, RNA and etc. When there is no nutrient medium and energy, it behaves like a crystal: ${}^{Sprt^{\{\ \}}_{\{\ \}}}_{Sprt^{\{\ \}}_{\{\ \}}}Sprt^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}$, a nutrient medium appears and the necessary energy: ${}^{Sprt^{B_0+E_q}_{B_0+E_q}}_{Sprt^{B_0+E_q}_{B_0+E_q}}Sprt^{Sprt^{B_0+E_q}_{B_0+E_q}}_{Sprt^{B_0+E_q}_{B_0+E_q}}$, its structure is transformed into a mathematical structure ${}^{Sprt^{B_0+E_q}_{B_0+E_q}}_{Sprt^{B_0+E_q}_{B_0+E_q}}Sprt^{Sprt^{A_0+E_s+B_0+E_q}_{A_0+E_s+B_0+E_q}}_{Sprt^{A_0+E_s+B_0+E_q}_{A_0+E_s+B_0+E_q}}$. The division of DNA into two DNAs after sufficient accumulation of bases and energy - this minimal division into only two duplicates corresponds to the law of conservation of living energy and minimization of the entropy of the system.

Next comes the level of living organisms:

$$ {}^{{}^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}Sprt^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}}_{{}^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}Sprt^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}}Sprt^{{}^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}Sprt^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}}_{{}^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}Sprt^{Sprt^{A_0+E_s}_{A_0+E_s}}_{Sprt^{A_0+E_s}_{A_0+E_s}}} $$

. Next comes the level of Globe, where the role of living cells (molecules in the case of a crystal) is played by living organisms. Next comes the level of Universe, where the role of living cells (molecules in the case of a crystal) is played by planets inhabited by living beings. You can try to represent these levels through more complex mathematical models, there are options for going beyond the level of objectivity for objects with energy structures of a sufficiently high level, but this is already material for subsequent publications. Our object world is an interpretation of the manifestation of only one set of subtle energy fibers out of their countless number.

${}^{A(t)}_{A(t)}Sprt$ will be called dynamic anti-capacity from oneself. For example, "white hole" in physics is such simple anti-capacity. The concepts of "white hole" and "black hole" were formulated by the physicists based on the subject of physics –the usual energies level. Mathematics allows you to deeply find and formulate the concept of singular points in the Universe based on the levels of more subtle energies. The experiments of the 2022 Nobel laureates Asle Ahlen, John Clauser, Anton Zeilinger and the experiments in chemistry Nazhipa Valitov correspond to the concept of the Universe as a capacity in itself as the element. They experimented with connections for elements of the microworld, and since here the connections are self-connections, then when the object component of self-connections is removed, its higher level remains, which was manifested in their experiments. The electron spin belongs to the second level - above the level of objectivity. The energy of self-containment in itself is closed on itself.

Remark 1.7.1. From the point of view of our theory of dynamic operators and sets, we can interpret the energy effect of a thermonuclear reaction as the result of the "collapse" of two self-objects: for example,1) ${}^{3}_{2}He$, ${}^{3}_{2}He$ and the formation of one self-object ${}^{4}_{2}He$, 2) ${}^{3}_{2}He$, ${}^{2}_{1}H$ and the formation of one self-object ${}^{4}_{2}He$. As a result, the energy of the collapse of the lost part of the self is released.

Remark 1.7.2. To gain access to object transformation, just go to the level lS=$\frac{2}{\pi}arctg(1+\beta)$, β may be quite small.

Examples of transformation:

1) $Sprt^{q}_{q} \to Sprt^{Sprt^{q}_{q}}_{Sprt^{q}_{q}} \to Sprt^{Sprt^{b}_{c}}_{Sprt^{b}_{c}}$

2) $Sprt^{q}_{q} \to$ S$_3$f(self(q)) $\to Sprt^{r}_{r}$

This is a rather conditional interpretation, because in fact, the lS of the “vessel” (energy cocoon) of the object may turn out to be greater than $\frac{2}{\pi}arctg(1)$. This is taken for initiation: we build a theory of this, starting from this stage of interpretation. After experiments, the next stage may begin.

Self A $= Sprt_A^A$ can be transformed into any D if µl(D) = µl ($Sprt_A^A$), µl(x) - level measure of self for x, in particular, into $Sprt_A^{any\ C}$ or $Sprt_{any\ C}^A$, and also an object R into any object Q or any energy U. The transformations of this type will be called stransformations. Self$^{\mathrm{N}}$ A can transform itself into any D if N $\geqslant$ 2; to realize this we need an even larger quantity N.

Example of a parallel-serial program statement

$$Sprt^{if\ \{g\}Sprt^{sSprt^{C}\ Sprt_{A}^{Q}}_{Sprt_{A}^{R}}}_{for\ w\ Sprt^{Sprt_{E}^{af}}_{Sprt_{J}^{if\ C}}}$$

Each self-field can automatically rebuild the self-program to the desired.

Self$^{\mathrm{N}}$- OS and is designed for such transformations, and it itself can be transformed with N $\geqslant$ 1, or it itself can be transformed with N $\geqslant$ 2.

Remark 1.7.3. Hypothesis 1.7: equations for real processes in a non-trivial form can be used to fully or partially interpret the self-level of the process, replacing the equal signs with identification signs, and solutions to these equations as a manifestation of this level on the level of objectivity and ordinary energies. That is, equations for real processes serve as a definition of the self-level of the process, the definition of self-values (self-characteristics) of the process through the identification sign, i.e., they are defined (expressed) through themselves. In particular, forms (1.1) - (1.4) can be used as forms of identification. Each such singularity creates its own field, the process, the object etc. Much more effective than science for working with these singularities will be special Dynamic programming, which we are currently working on to create. Identification at the lower levels of a hierarchical dynamic structure of type (1.7.1) will lead to the upper level. You can also try to use it for full or partial interpretation of the self-level of chemical reactions, but here there will be a trivial identification and determination of the self-level will be much simpler. For example, a type w $\equiv 2w$ singularity at the top level of the structure of a mathematical simplified model of DNA generates a field for DNA division. A rather complex type of singularity at the upper level of the structure of a simplified mathematical model generates an electromagnetic field through identification in Maxwell’s equations.

Remark 1.7.4. Parallel operator $Sprt_{places\ for\ symbols}^{symbols}$ corresponds to theoretical science, parallel operator $Sprt_x^{objects}$ corresponds to technology, x – the space “point” (space place).

Remark 1.7.5. Let self-energy of A looks like $Sprt_{C_A+\Delta C}^{C_A+\Delta C}$=$Sprt_{C_A}^{C_A}$ +$Sprt_{C_A}^{\Delta C}$ +$Sprt_{\Delta C}^{C_A}$ +$Sprt_{\Delta C}^{\Delta C}$, $Sprt_{C_A}^{C_A}$ corresponds to objectivity of A,

$Sprt_{C_A}^{\Delta C}$ = E_A (1.7.3),

E_A - usual energy of A. ΔC determined from (1.7.3) through C_A and then we can determine the complete self-energy of A.

Remark 1.7.6. Let us consider an analogue of the Schrödinger equation for networks operating on electromagnetic energy

$$\frac{\partial w}{\partial x} = \left[w, \mu S(H)\right]$$

w- measure of self for networks operating, $\mu S(H)$ - measure of self for H, H = H(μS(p), μS(q),t) - an analogue of the Hamiltonian in the space of actions of artificial neurons in a neural network, q is the operator of an artificial neurons action result, p is the operator of an artificial neurons action impulse.

Remark 1.7.7. The self-space of a higher level contains many self-energetic fibers, collecting into appropriate sets that can be accessed by the corresponding self-spaces of lower levels. That's right, for example. This assembly point on the human cocoon can carry out this, in particular, access to our self-space with objects.

Remark 1.7.8. It is quite possible to try to build up the levels of objects and processes; change something at these levels.

Remark 1.7.9. One can try to conventionally represent the mathematical model (1.7.1) of the atom (molecule) as a hierarchical dynamic operator.

Remark 1.7.10. Here, self-action is understood as action on oneself (i.e., to the same action), while physicists understand self-action, for example, as the absorption of one elementary particle by another of the same type.

Appendix 1

If we introduce for the energy of a chemical element the concept self-energy (the concept of a chemical element was introduced earlier [3]): $Sprt_R^R$, R=Q+D, Q- internal energy, D is the energy of its interaction with the external environment. $Sprt_R^R$=$Sprt_{Q+D}^{Q+D}$=$Sprt_{Q+D}^{Q}$+$Sprt_{Q+D}^{D}$=$Sprt_Q^Q$+$Sprt_D^Q$+$Sprt_Q^D$+$Sprt_D^D$, $Sprt_Q^Q$- internal self-energy, $Sprt_D^D$-the external self-energy, $Sprt_D^Q$- object component of a chemical element, $Sprt_Q^D$- usual energy component of a chemical element. We describe the usual chemical reactions for the $Sprt_D^Q$-component using the $Sprt_Q^D$-component. A self-molecule (self-atom, self-(elementary particle))) as a capacity can have the following types of self: self-set, self-structure, self-hierarchy or its elements that generates this self-molecule (self-atom, self-(elementary particle))).

Self-power is force that is applied to oneself or its elements that generates this self-power.

You can try to consider the equations: Srt_x^x=a, x(a) - ?, $Srt_b^x = a,\ x(a,b)-?$, Srt_x^q=a, x(a, q)- ?.

Supplement for Quantum Mechanics and Classical statistical Mechanics through Sprt-elements:

Hamilton operator $\widehat{H}\ = \widehat{H}_0 +\ \widehat{W_0}$, $\widehat{H}_0$ -considered quantum system energy, consisting of two or more parts, without their interaction with each other, $\widehat{W_0}$ is the energy of their interaction, $\widehat{\rho}$-statistical operator [20]. Self-energy $Sprt_{\widehat{H}}^{\widehat{H}}$= $Sprt_{\widehat{H}_0+\widehat{W_0}}^{\widehat{H}_0+\widehat{W_0}}$= $Sprt_{\widehat{H}_0+\widehat{W_0}}^{\widehat{H}_0}+Sprt_{\widehat{H}_0+\widehat{W_0}}^{\widehat{W_0}} = Sprt_{\widehat{H}_0}^{\widehat{H}_0}+Sprt_{\widehat{W_0}}^{\widehat{H}_0}+Sprt_{\widehat{H}_0}^{\widehat{W_0}}+Sprt_{\widehat{W_0}}^{\widehat{W_0}}$, $Sprt_{\widehat{H}_0}^{\widehat{H}_0}$-considered quantum system self-energy, $Sprt_{\widehat{W_0}}^{\widehat{W_0}}$ is self-energy of their interaction, $Sprt_{\widehat{W_0}}^{\widehat{H}_0}$--object manifestation of the energy of the system in an external field., $Sprt_{\widehat{H}_0}^{\widehat{W_0}}$- the manifestation of the energy of the system in the energy interaction with the external field. Variants of the Schrödinger equation $\frac{\partial\hat{\hat{\hat{\rho}}}}{\partial t}$ + $[\acute{\widehat{W}},\acute{\hat{\rho}}]$=0 of the form S_2f, S_3f are possible, using the form (1.1) or the form from forms (1.1.1) - (1.4). The carrier of the measure of objectivity-mass should be objectivity - elementary particle graviton, look like $Srt_{objectivity}^{objectivity}$, therefore it is a self-particle and is not an element of the level of objectivity, but is an element of the level self. Therefore, it cannot be found at our level. In fact, the theory of Sprt-elements helps to form a unified field theory on a qualitative level, because it is not possible to create a quantitative unified field theory. Supplement for string theory: May be to try represent elementary particles in the form of continual self-elements of the type S_∞^-=sin(-∞) $--\downarrow I \uparrow_{-1}^{1}$, T_∞^+= tg∞ $--\uparrow I \downarrow_{-\infty}^{\infty}$, T_∞^-=tg(-∞) $--\downarrow I \uparrow_{-\infty}^{\infty}$, f↑I↓g for any f, g etc.

We consider Sprt-logic: consider the functional f(Q), which gives a numerical value for the truth of the statement Q from the interval [0,1], where 0 corresponds to "no," and one corresponds to the logical value "yes." Then for joint statements A, B: f(A+B)=f(A)+f(B)-f(A*B)+fS(D), D- self-statement from A*B, fS(x)- the value of self-truth for self-statement x; for dependent statements: f(A*B)=f(A)*f(B/A)=f(B)*f(A/B), where f(B/A)-conditional truth of the statement B at statement A, f(A/B)- dependent truth of statement A at the statement B. Adding the truth values of inconsistent propositions: f(A+B)=f(A)+f(B). The formula of complete truth: f(A)=$\sum_{k=1}^{n} f\left(B_k\right) * f(A/B_k)$, B_1, B_2, .., B_n-full group of hypotheses-statements: $\sum_{k=1}^{n} f\left(B_k\right)$=1(“yes”).
Remark. A statement can be interpreted as an event, and its truth value as a probability.

Sprt- statement for set of statements A={A_1, A_2, ..., A_n}: $Sprt_x^{\{A_1, A_2, \ldots, A_n\}}$, $Sprt_x^{\{f(A_1), f(A_2), \ldots, f(A_n)\}}$- Sprt- truth for these statements. It is possible to consider the self-statement S_3A with m statements from A, at m<n, which is formed by the form (1.1), that is, only m statements from A are located in the structure $Sprt_x^A$, or by the form from forms (1.1.1) - (1.4). The same for self- truth $S_3\{f(A_1), f(A_2), \ldots, f(A_n)\}$.

One can introduce the concepts of Sprt-group: $Sprt_x^A$, A is usual group, $Sprt(t)_B^A$, where A, B- usual groups, self-group: Sf_iA, i=1,2,3 [3], A is usual group.

Definition A. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its elements clearly, but not necessarily in known operators; it can form (create) new special operators (in particular, special functions). In particular, $Cprt_A^A$, Crt_A^A are such structures. Similarly, for working with models, each is structured by its structure; for example, use Sprt-groups, Sprt-rings, Sprt-fields, Sprt-spaces, self-groups, self-rings, self-fields, and self-spaces. Like any task, this is also a structure of the appropriate capacity. Since the degree of freedom is double, it is clear that the form of the self-equation contains a solution or structures the inversion of the self-equation concerning unknowns, i.e., the structure of the self-equation is complete. The transition process in the form of ${}_D^CSprt_B^A$ is included in the transition from one world A (spatial variables, which we will denote by X1, and time variables, by T1) to another world B (spatial variables, which we will denote by X2, and time variables, by T2). It is accompanied by spatial variables in the form (T1, X1), i.e. such a transition process transforms time variables T1 into part of spatial ones, and time variables - T3.

Supplement

Connection Sprt – elements with usual functionals and operators

We consider functional g(x): X→g, x∈ X, g—numerical value of functional g(x). It is specific capacity for X. $Sprt_{\{g(x)\}}^{\{g(x)\}}$ made from her the self-capacity in itself as an element $S_1f\{g(x)\}$, {g(x)}—the set of any functionals for X. In particular, probability p(X) –is such functional, X—an event. Here $Sprt_{p(X)}^{p(X)}$ is $S_1fp(X)$, denote it through pS(X). Usual event is dynamical capacity.

Definition B. Sprt-probability of events A, B is p($Sprt_B^A$), denote Spp_B^A. In particular,

Spp_B^A for joint A, B: Spp_B^A= p($Sprt_B^A$)=p($\left\{\begin{matrix} D \\ A\cup B - A\cap B \end{matrix}\right\}$)= p(A)+p(B) –p(AB)+pS(D), D-- the self-capacity in itself as an element from $A\cap B$, pS(D)—probability self of D of next level—self level. The probability for stochastic value X is capacity. We represent its distribution in the kind of Sprt-element:

$Sprt(t)_X^{\{(x_1,p_1),(x_2,p_2),\ldots,(x_n,p_n)\}}$ (1.7.*)

Here interest represent partial distribution self from (1.7.*) by form (1.1) or the form from forms (1.1.1) - (1.4) with value self of stochastic value X for some subset $\{(x_{1_1}, x_{2_1}, \ldots, x_{j_1}\} \in \{x_1, x_2, \ldots, x_n\}$ with probabilities self $\{pS_1, pS_2, \ldots, pS_j\}$.

For operator $X_1 \overset{F}{\rightarrow} X_2$: $\overset{F}{\rightarrow} X_2$ is capacity for X_1. $Sprt_{\overset{F}{\rightarrow}X2}^{\overset{F}{\rightarrow}X2}$ -- self- capacity in itself as an element for X_1. More complex for implicit operator: $F(X_1, X_2)=0$. Then $St_{F(X1,X2)=0}^{F(X1,X2)=0}$ forms self-capacity in itself as an element for X_1 relatively of X_2 or for X_2 relatively of X_1. x obtains more power of the liberty and in this is direct decision (i. e. self-capacity in itself as an element for x). Self-equation for x has its decision for x in direct kind. Self-task for x has its decision for x in direct kind. Self-question has its answer for x in direct kind. x acquires more degree of liberty and in this is direct decision. We consider $Sprt_D^D$, D-block over execution subject in S_{mnsprt} for networks. Then we have self-capacity in itself as an element D, where full realization requires correspondent self-energy. $Sprt_{Smnsprt}^{Smnsprt}$ increase self-level of S_{mnsprt} and may made no visual its. The entire neural network as instantaneous simultaneous RAM in Sprt-elements and self- elements. $self^{self^{\cdots^{self}}}$, $f_1\downarrow I \uparrow_{-1}^{1} f_2^{\,f_1\downarrow I\uparrow_{-1}^{1} f_2^{\,\cdots^{f_1\downarrow I\uparrow_{-1}^{1} f_2}}}$, $sin\infty^{sin\infty^{\cdots^{sin\infty}}}$.When activated in a neural network, the entire neural network becomes a working memory. Use of self-energy as activation or from outside. $Q_0 = Sprt_{Sprt_{activation}^{S_{mnSprt}}}^{Sprt_{activation}^{S_{mnSprt}}} \rightarrow$ self-RAM, $Q_{00} = {}_{activation}^{S_{mnSprt}}{}_{Sprt}^{Sprt}\,{}^{}_{}{}_{activation}^{S_{mnSprt}} Q_0$, $Q_{01} = {}_{Sprt_{activation}^{S_{mnSprt}}}^{Sprt_{activation}^{S_{mnSprt}}} Q_0$.

Q_0, Q_{00}, Q_{01}-coding,translation,realization eprograms, Q_0, Q_{00}, Q_{01}-S_{mnSprt}, Q_0, Q_{00}, Q_{01}-Assembler.

References

1) A.Mostowski. Constructive sets.UNIVERSITY OF WARSAW-North-Holland Publishing Company-Amsterdam. PWN-Polish Scientific Publishers-Warszawa.1969.

2) Thomas J.Jech. LECTURES IN SET THEORY WITH PARTICULAR EMPHASIS ON THE METHOD OF FORCING. SPRINGER-VERLAG Berlin. Heidelberg. New-York, 1971.

3) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3, pp. 1-31, March 27, 2024 https://cskscientificpress.com/articles_file/527-_article1712797848.pdf

4) Ershov Y.L. On a Hierarchy of Sets I, Albebra and Logic, 7 (1968), 47 - 73.

5) Ershov Y.L.] On a Hierarchy of Sets II, Albebra and Logic, 7 (1968),15 - 47.

6) Yrshov Y.L. On a Hierarchy of Sets III, Algebra and Logic, 9(1970), 34 -

7) Krain S.G. Linear differential equations in Banach space. M.,Science,1967(in Russian).

8) A.L.Dontchev .Perturbations, approximations and sensitivity analysis of optimal control systems. Sptinger-Verllag, Berlin, New York,1983.

9) Galushkin A. Networks: principles of the theory. Hot line-Telecom. M.,2010 (in Russian).

10) Danilishyn .V. Danilishyn O.V. tS – ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

11) Danilishyn .V. Danilishyn O.V. SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June, 2023 with the proceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

12) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-_ article1695707465.pdf (mkscienceset.com)

13) I. Danilishyn, O. Danilishyn Program Operators Sit, tS, S1e, Set1. Journal of Sensor Networks and Data Communications [ISSN: 2994-6433] 2023, Volume 3 | Issue 1 | 138-143, https://www.opastpublishers.com/table-contents/jsndc-volume-3-issue-1-year-2023

14) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

15) Danilishyn .V. Danilishyn O.V. DYNAMICAL SIT-ELEMENTS. IV International Scientific and Practical Conference "GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ", 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

16) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO SETS THEORY AND OTHERS. Scientific practice: modern and classical research methods: Collection of scientific papers «O» with Proceedings of the IV International Scientific and Practical Conference, Boston, May 26, 2023. Boston-Vinnytsia: Primedia eLaunch & European Scientific Platform, 2023, pp.166-171. **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/11**

17) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT-ELEMENTS TO CONTINUAL VALUED LOGIC AND OTHERS. Features of the development of modern science in the pandemic's era: a collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, May 19, 2023. Berlin, Federal Republic of Germany: European Scientific Platform, pp.79-84. **https://previous.scientia.report/index.php/archive/issue/view/19.05.2023**

18) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square.

https://doi.org/10.21203/rs.3.rs-3217178/v1

Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

19) Danilishyn I., Danilishyn O. VARIABLE HIERARCHICAL DYNAMICAL STRUCTURES (MODELS) FOR DYNAMIC, SINGULAR, HIERARCHICAL SETS AND THE PROBLEM OF COLD THERMONUCLEAR FUSIONThe driving force of science and trends in its development: collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, July 14, 2023. Coventry, United Kingdom: European Scientific. Pp.113-119. Platform.**https://previous.scientia.report/index.php/archive/issue/view/14.07.2023**

20) D.I.Blokhintsev. Quantum Mechanics Lectures on Selected Topics. Moscow, Atomizdat,1981.

Declarations:

Availability of data and material

1) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3, pp. 1-

31, March 27, 2024 https://cskscientificpress.com/articles_file/527-_article1712797848.pdf

2) Illia Danilishyn and Oleksandr Danilishyn. Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care 2023, Volume 4 | Issue 2 | 66- 81. https://doi.org/10.33140/NABCC.04.02.02

3) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-_ article1695707465.pdf (mkscienceset.com)

4) I. Danilishyn, O. Danilishyn Dynamic Sets Se, Networks Se. *Advances in Neurology and Neuroscience*, 2023, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

5) I. Danilishyn, O. Danilishyn Program Operators Sit, tS, S1e, Set1. Journal of Sensor Networks and Data Communications [ISSN: 2994-6433] 2023, Volume 3 | Issue 1 | 138-143, https://www.opastpublishers.com/table-contents/jsndc-volume-3-issue-1-year-2023

6) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference "GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ", 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

7) Danilishyn .V. Danilishyn O.V. tS – ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

8) Danilishyn .V. Danilishyn O.V. SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June, 2023 with the proceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

9) Danilishyn .V. Danilishyn O.V. DYNAMICAL SIT-ELEMENTS. IV International Scientific and Practical Conference "GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ", 31.03.2023/Zurich,

Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

10) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO SETS THEORY AND OTHERS. Scientific practice: modern and classical research methods: Collection of scientific papers «O» with Proceedings of the IV International Scientific and Practical Conference, Boston, May 26, 2023. Boston-Vinnytsia: Primedia eLaunch & European Scientific Platform, 2023, pp.166-171. **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/11**

11) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT-ELEMENTS TO CONTINUAL VALUED LOGIC AND OTHERS. Features of the development of modern science in the pandemic's era: a collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, May 19, 2023. Berlin, Federal Republic of Germany: European Scientific Platform, pp.79-84. **https://previous.scientia.report/index.php/archive/issue/view/19.05.2023**

12) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square.

https://doi.org/10.21203/rs.3.rs-3217178/v1

Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

Competing interest

There are no competing interests. All sections of the monograph are executed jointly.

Funding

There were no sources of funding for writing the monograph.

Authors' contributions

The contribution of the authors is the same, we will not separate.

Part II. Fuzzy fSprt – elements and their applications

Introduction

Significance of the section: in a new qualitatively different approach to the study of complex processes through new mathematical, fuzzy hierarchical, fuzzy dynamic structures, in particular those processes that are dealt with by Synergetics.

We consider expression

$$\begin{matrix} C & & A \\ \mu_2 & \text{ffSprt} & \mu_1 \\ D & & B \end{matrix} \quad (*_{2.1})$$

where A fuzzy fits into B with measure of fuzziness μ_1, D is fuzzy forced out of C with measure of fuzziness μ_2 [1]. The result of this process will be described by the expression

$$\begin{matrix} C & & A \\ \mu_2 & \text{ffSrt} & \mu_1 \\ D & & B \end{matrix} \quad (*_{2.2})$$

If A, B, D, C are taken as fuzzy sets, then we will call ($*_{2.1}$) a fuzzy dynamic fuzzy set. The need ($*_{2.1}$) arose to describe processes in networks. Threshold element ffSprt - $\begin{matrix} b & & \{a\widetilde{x}\} \\ \mu_2 & \text{ffSprt(t)} & \mu_1 \\ \{qy\} & & b \end{matrix}$, b- artificial neurons of type ffSprt (designation - mnffSprt) , $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,x_n|\mu_{\widetilde{x}}(x_n)|$) is the fuzzy set of values of the initial signals, a=($a_1,a_2,\ldots,a_n$) are the weights of Sprt-synapses and $\widetilde{y}$=($y_1|\mu_{\widetilde{y}}(y_1)$, $y_2|\mu_{\widetilde{y}}(y_2)|, \ldots,$ $y_n|\mu_{\widetilde{y}}(y_n)|$) is the fuzzy set of values of the output signals with weights q=($q_1,q_2,\ldots,q_n$). It can be considered a simpler version of the fuzzy dynamic fuzzy set

$$\begin{matrix} & A \\ \text{ffSprt} & \mu \\ & B \end{matrix} \quad (**_{2.1})$$

where fuzzy set A fuzzy fits with measure of fuzziness µ into fuzzy set B, the result of this process will be described by the expression

$$\begin{matrix} & A \\ \text{ffSrt} & \mu \\ & B \end{matrix} \quad (**_{2.2})$$

or

$$\begin{matrix} B & \\ \mu & \text{ffSprt} \\ A & \end{matrix} \quad (***_{2.1})$$

where fuzzy set A is fuzzy forced out of B with measure of fuzziness µ, the result of this process will be described by the expression

B
μffSrt $(***_{2.2})$.
A

We consider the measure:

1) $\underset{D}{\overset{b}{\mu}}**(\mu_2 ffSrt \underset{b}{\overset{A}{\mu_1}})=\frac{\mu(A)}{\mu(D)}\frac{\mu_1}{\mu_2}$, where $\mu(A), \mu(D)$ –usual measures of fuzzy sets A, D.

2) measure of fuzziness $\mu_{\widetilde{\underset{D}{\overset{b}{\mu_2}} ffSrt \underset{b}{\overset{A}{\mu_1}}}}$ $**(\underset{D}{\overset{b}{\mu_2}} ffSrt \underset{b}{\overset{A}{\mu_1}})=\frac{\mu_{\widetilde{A}}(A)}{\mu_{\widetilde{D}}(D)}\frac{\mu_1}{\mu_2}$, where $\mu_{\widetilde{A}}(A)$, $\mu_{\widetilde{D}}(D)$ -measures of fuzziness of fuzzy sets A, D. $\mu_{\widetilde{x}}(\widetilde{x})$ - measure of fuzziness of $\widetilde{x}$ =(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$)|,...,x$_n|\mu_{\widetilde{x}}$(x$_n$)|) let's define it as $\frac{\mu_{\widetilde{x}}(x_1)+\mu_{\widetilde{x}}(x_2)+\ldots+\mu_{\widetilde{x}}(x_n)}{n}$.

Remark. One can consider some generalization for $(*_{2.1})$, $(*_{2.2})$: $\underset{D}{\overset{q_1(C)}{\mu_2}}$ ffSprt $\underset{q(B)}{\overset{A}{\mu_1}}$, $\underset{D}{\overset{q_1(C)}{\mu_2}}$ ffSrt $\underset{q(B)}{\overset{A}{\mu_1}}$, where A is contained into B through q, D is forced out of C through q$_1$, A, B, D, C are taken as fuzzy sets. Similarly for $(**_{2.1})$, $(**_{2.2})$: ffSprt $\underset{q(B)}{\overset{A}{\mu}}$,ffSrt $\underset{q(B)}{\overset{A}{\mu}}$, where A is contained into B through q, for $(***_{2.1})$, $(***_{2.2})$: $\underset{D}{\overset{q(C)}{\mu}}$ ffSprt, $\underset{D}{\overset{q(C)}{\mu}}$ ffSrt, where D is displaced from C through q.

Let us introduce the concepts fCha, the fuzzy capacity measure, and fCca, the measure of its fuzzy content. fCca is the same as the number of fuzzy capacity content items. Consider the compression ratios of the fuzzy dynamic fuzzy set: q$_1$=ffSprt$\underset{B}{\overset{A}{\mu}}$ answers I compression power of fuzzy dynamic fuzzy set A, q$_2$= ffSprt$\underset{B}{\overset{q_1}{\mu}}$ –II compression power of fuzzy dynamic fuzzy set A, ..., q$_{n+1}$=ffSprt$\underset{B}{\overset{q_n}{\mu}}$ –n+1 compression power of fuzzy dynamic fuzzy set A. In contrast to the classical one-attribute fuzzy set theory [2], [3], where only its contents are taken as a fuzzy set, we consider a two-attribute fuzzy set theory with a fuzzy set as a fuzzy capacity and separately with its contents. We introduce the designations: CofQ—the contents of the fuzzy capacity Q. Previously [4] - [16] the axiom of regularity (A8) [17] is removed from the axioms of set theory, so we naturally obtain the possibility of using singularities in the form of fself- fuzzy (fuzzy sets), fself- fuzzy (fuzzy elements), which is exactly what we need for new mathematical models for describing complex processes.

2.1 Fuzzy fSprt − elements

Definition 2.1.1. The fuzzy set of elements $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)|, \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)|$) at one point q with measure of fuzziness µ of space X we shall call fuzzy fSprt– element(designation - ffSprt– element), and such a point in space is called fuzzy capacity of the ffSprt– element. We shall denote ffSprt$\overset{\widetilde{x}}{\underset{q}{\mu}}$. The result of this process we shall denote ffSrt$\overset{\widetilde{x}}{\underset{q}{\mu}}$.

Definition 2.1.2. ffSprt$\overset{\widetilde{x}}{\underset{q}{\mu}}$— fuzzy dynamic with measure of fuzziness µ fuzzy set $\widetilde{x}$ at q.

Definition 2.1.3. An ordered fuzzy set of elements at one point q in space with measure of fuzziness µ: ffSprt $\overset{\overrightarrow{r}}{\underset{q}{\mu}}$, $r = \widetilde{x}$, is called an ordered ffSprt– element.

It's possible to ffSrt$\overset{\widetilde{x}}{\underset{q}{\mu}}$ correspond to the fuzzy set of elements $\widetilde{x}$, and to the ordered ffSprt- element - a fuzzy fvector, a fuzzy fmatrix, a fuzzy ftensor, a fuzzy directed fsegment in the case when the totality of elements is understood as a fuzzy set of elements in a fuzzy segment.

It's allowed to sum ffSprt– elements: ffSprt$\overset{\widetilde{x}}{\underset{q}{\mu}}$+ ffSprt$\overset{\widetilde{y}}{\underset{q}{\mu}}$= ffSprt $\overset{\widetilde{x}\cup\widetilde{y}}{\underset{q}{\mu}}$.

These structures are more suitable for using fuzzy sets for energy space, for any objects. The operator ffSprt is adapted for ordinary energies, using their property to overlap.

Fuzzy fcapacity in itself.

Definition 2.1.4. The fuzzy fcapacity A in itself of the first type is the fuzzy capacity fuzzy containing with measure of fuzziness µ itself as an element. Denote $ffS_1f\mu A$.

Definition 2.1.5. The fuzzy fcapacity A in itself of the second type is the fuzzy capacity that fuzzy contains with measure of fuzziness µ elements from which it can be generated. Denote $ffS_2f\mu A$.

An example of the fuzzy capacity in itself of the first type is a fuzzy set containing itself. An example of capacity in itself of the second type is a living organism since it contains a program: DNA and RNA.

Definition 2.1.6. Fuzzy partial fcapacity A in itself of the third type is the fuzzy capacity A in itself, which partially fuzzy contains with measure of fuzziness µ itself or fuzzy contains with measure of fuzziness µ elements from which it can be generated in part or both. Let us denote $ffS_3f\mu A$.

All fuzzy fcapacities in fuzzy fself-space are fuzzy capacities in themselves by definition. Fuzzy fcapacities in themselves can appear as fuzzy fSrt -capacities and ordinary fuzzy capacities. In these cases, the usual fuzzy measures and methods of fuzzy topology are used.

Connection of ffSprt– elements with fuzzy fcapacities in themselves [1].

For example, $ffSrt\,\mu^{\widetilde{\{R\}}}_{\widetilde{g}\left\{\widetilde{R}\right\}}$ is the fuzzy fcapacity in itself of the second type if $\widetilde{g}\left\{\widetilde{R}\right\}$ is a fuzzy program capable of fuzzy generating $\widetilde{\{R\}}$.

Consider a third type of fuzzy fcapacity in itself. For example, based on ffSprt$\mu^{\widetilde{x}}_{q}$, where $\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$)$|,\ldots,$x$_n|\mu_{\widetilde{x}}$(x$_n$)$|$) i.e. n - elements at one point, we can consider the fuzzy fcapacity $ffS_3f\mu$ in itself with m elements from $\widetilde{x}$, m<n, which is formed according to the form (1.1), that is, the structure ffSprt$\mu^{\widetilde{x}}_{q}$ contains only m elements, or in forms (1.1.1) - (1.1.5), summarizing it. Fuzzy fcapacities in themselves of the third type can be formed for any other structure, not necessarily ffSprt, only by necessarily reducing the number of elements in the structure, in particular, using form (1.2). Structures more complex than ffS$_3$f can be introduced. For example, through a form (1.3), where A is fuzzy compressed (fuzzy fits) in C in the fuzzy compression fuzzy structure B in C (i.e., in the fuzzy structure ffSprtμ^{B}_{C}); or through the forms (1.3.1) - (1.4) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc.

(1.3.1) - (1.3.4) schematically interpret the fuzzy formation of fuzzy capacity in itself through a pseudo 3-connected form with a 2-connected form. The ideology of ffSprt and ffS$_3$f can be used for programming.

Remark 2.1.1. Fuzzy self, in particular, according to a fuzzy form- fuzzy analogue of the form of type (1.1):

(1$|$ μ_1, (2$|\mu_2$,1$|\mu_3$))(2.1$*$),

μ_{i} $(i = 1, 2, 3)$ $--$ the fuzziness of the indicated positions. For example

1) fuzzy forming from element with fuzziness in the form {2,1}: (1| , (2 ,1))

2) fuzzy forming from element in the form {2,1} with fuzziness : (1, (2 ,1)|)

3) fuzzy formation of partial self in the form (1.1) with fuzziness : (1, (2 ,1))|

4) It is also possible to generalize the other remaining forms (1.1) – (1.4) to fuzzy forms

5) etc.

The energy of fuzzy self- (fuzzy containment) in itself closes on itself.

Math fuzzy self fuzzy [1].

Let's consider ffSprt arithmetic first:

1. Simultaneous fuzzy addition of a fuzzy set of elements $\widetilde{x}=(x_1|\mu_{\widetilde{x}}(x_1), x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,x_n|\mu_{\widetilde{x}}(x_n)|)$ is carried out using $\text{ffSprt}^{\widetilde{x}+}_{\mu, q}$

$$= \text{ffSprt}^{\{x_1 * \mu_{\widetilde{x}}(x_1) + \ x_2 * \mu_{\widetilde{x}}(x_2)+, \ldots, +x_n * \mu_{\widetilde{x}}(x_n)\}}_{\mu_i, q}.$$

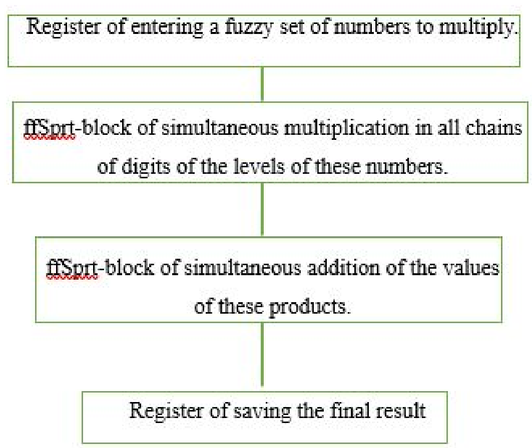


Figure 1: The straightforward functional scheme of the assumed arithmetic-logical device for ffSprt-multiplication.

2. Similarly, for simultaneous multiplication ffSprt $\underset{q}{\overset{\widetilde{x}*}{\mu}}$: the notation of the fuzzy set B with elements $b_{i_1 i_2 \ldots i_n} =$ (ffSprt $\underset{q}{\mu}\{x_{1_{i_1}} * \mu_{\widetilde{x}}(x_{1_{i_1}}) *, x_{2_{i_2}} * \mu_{\widetilde{x}}(x_{2_{i_2}}) *, ..., \ x_{n_{i_n}} * \mu_{\widetilde{x}}(x_{n_{i_n}})\}$ $)_R$

for any $\{i_1, i_2, ..., i_n\}$ *without repetitions*, $q = St_w{}^{\{K\}}$,K-set of any $\{k_1*, k_2*, ..., k_n*\}$ without repeating them, k_i-any digit, i=1,2,. . . ,n, R= $St_w{}^{\{i_1+, i_2+, ..., i_n\}}$, R is the index of the lower discharge (we choose an index on the scale of discharges): see Table I.1.

Then ffSprt $\underset{q}{\overset{B+}{\mu_i}}$ - the final result of simultaneous multiplication ,(i=1,2,3)– the fuzziness of the indicated positions. Any system of calculus can be chosen, in particular binary. The most straightforward functional scheme of the assumed arithmetic-logical device for Sit-multiplication:

Remark 2.1.2. The algorithm for simultaneously adding a set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by multiplying the first number from the fuzzy set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the fuzzy set by the ones following it, etc.

3. Similarly for simultaneous execution of various operations: ffSprt $\underset{r}{\overset{\widetilde{x}\widetilde{q}}{\mu_i}}$,(i=1,2,3)– the fuzziness of the indicated positions, where

$\widetilde{q}$=($q_1|\mu_{\widetilde{q}}(q_1)$, $q_2|\mu_{\widetilde{q}}(q_2)|$, . . . , $q_n|\mu_{\widetilde{q}}(q_n)|$), q_i -an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: ffSprt $\underset{r}{\overset{\widetilde{F}\widetilde{x}}{\mu}}$, where $\widetilde{F}$=($F_1|\mu_{\widetilde{F}}(F_1)$, $F_2|\mu_{\widetilde{F}}(F_2)|$, . . . , $F_n|\mu_{\widetilde{F}}(F_n)|$) F_i is an operator, i = 1, . . . , n.

5. The arithmetic itself for fuzzy fcapacities in themselves will be similar: addition - $ffS_1 f\mu\{\widetilde{x}+\}$, (or $ffS_3 f\mu\{\widetilde{x}+\}$ for the third type), multiplication $ffS_1 f\mu\{\widetilde{x}*\}$, ($ffS_3 f\mu\{\widetilde{x}*\}$).

6. Similarly with different operations: $ffS_1 f\mu\{\widetilde{x}\widetilde{q}\}$, ($ffS_3 f\mu\{\widetilde{x}\widetilde{q}\}$, and with different operators: $ffS_1 f\mu\left\{\widetilde{F}\widetilde{x}\right\}$, ($ffS_3 f\mu\left\{\widetilde{F}\widetilde{x}\right\}$).

7. $ffSrt\mu\begin{matrix}A\\ \\ B\end{matrix} - the\ result\ of\ the\ fuzzy\ containment\ operator.$ For fuzzy sets A, B we have

$ffSrt\mu\begin{matrix}A\\ \\ B\end{matrix} = \{A \cup B - A \cap B\ , D\}$, where D is fuzzy self-(fuzzy set) for $A \cap B$. There is the same for structures if it's considered as fuzzy sets.

8. ffSprt- derivative of g($x_{1,},x_2,\ldots,x_n$) is $ffSrt\begin{matrix}\left\{\frac{\partial}{\partial x_{1_i}},\frac{\partial}{\partial x_{2_i}},\ldots,\frac{\partial}{\partial x_{k_i}}\right\}\\ \mu\\ g(x_1,,x_2,\ldots,x_n)\end{matrix}$, where $\widetilde{x}_0$=(x_{1_i} $|\mu_{\widetilde{x}}(x_{1_i}$), $x_{2_i}|\mu_{\widetilde{x}}(x_{2_i})$,... x_{k_i} $|\mu_{\widetilde{x}}(x_{k_i}$))- any fuzzy set from $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,x_n|\mu_{\widetilde{x}}(x_n)|$) Let's designate ffSprt - $\frac{\partial^k g(x)}{\partial x_{1_i}\partial x_{2_i}\ldots\partial x_{k_i}}$. ffSprt-integral off $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)|$) is ffSprt $\begin{matrix}\left\{\int ()dx_{1_i},\int ()dx_{2_i},\ldots,\int ()dx_{k_i}\right\}\\ \mu\\ g(\widetilde{x})\end{matrix}$, where $\widetilde{x}_0$=(x_{1_i} $|\mu_{\widetilde{x}}(x_{1_i}$), $x_{2_i}|\mu_{\widetilde{x}}(x_{2_i})$,... x_{k_i} $|\mu_{\widetilde{x}}(x_{k_i}$))- any fuzzy set from $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)|$). Let's designate ffSprt - $\int\ldots\int g(\widetilde{x})dx_{1_i}dx_{2_i}\ldots dx_{k_i}$-k-multiple fuzzy integral. ffSprt -lim off $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)|$) is ffSprt $\begin{matrix}\left\{\lim\limits_{x_{1_i}\to a_{1_i}},\lim\limits_{x_{2_i}\to a_{2_i}},\ldots,\lim\limits_{x_{k_i}\to a_{k_i}}\right\}\\ \mu\\ f(\widetilde{x})\end{matrix}$. Let's designate ffSprt-$\begin{pmatrix}\lim\limits_{x_{1_i}\to a_{1_i}}\\ \ldots\\ x_{k_i}\to a_{k_i}\end{pmatrix}$ g($x_1,,x_2,\ldots,x_n$). fself-$\lim\limits_{x\to a}$ = ffSprt $\begin{matrix}\lim\limits_{x\to a}\\ \mu\\ \lim\limits_{x\to a}\end{matrix}$.

9. In the case of fself- (fuzzy derivatives(designation-fderivatives)), fself is designation for fuzzy self, inclusions of multiple fuzzy fderivatives are obtained. The same is true for fself- (fuzzy integrals(designation-fintegrals)): we get inclusions of multiple fuzzy fintegrals.

10. Let's denote fself-(fself-$\widetilde{Q}$) through fself^2-$\widetilde{Q}$, fS(n,$\widetilde{Q}$)= fself-(fself-($\ldots$(fself-$\widetilde{Q}$))) = fself^n-$\widetilde{Q}$ for n-multiple fuzzy self.

ffOperator itself [1].

Definition 2.1.7. An operator that transforms ffSprt$\begin{matrix}\widetilde{x}\\ \mu\\ q\end{matrix}$ into any $ffS_if\mu\left\{\widetilde{b}\right\}$, $i = 2,3$; where$\left\{\widetilde{b}\right\} \subset \{\widetilde{x}\}$; is the foperator itself, foperator is designation for fuzzy operator.

Example. The operator fuzzy contains the fuzzy set in itself.

ffLim-itself [1].

1. ffLim-Sprt, in particular, for partial limits.

   For example, the double limit: $fflim_{\substack{x \to A_1 \\ y \to A_2}} \widetilde{G}(x,y)|\mu_{\widetilde{G}}$ corresponds to

   ffSrt $\underset{(A_1, A_2)}{\overset{\widetilde{G}(x,y)|\mu_{\widetilde{G}}}{\mu}}$.

   Similarly, for fflim-Sprt with n variables.

In the case of fflim-itself, for example, for m variables, it suffices to use the forms (1.1), (2.1*) of fflim-Sprt for n variables (n>m). The same is true for integrals of variables m (for example, the double integral over a rectangular region is through the double limit).

The sequence of actions can be "collapsed" into an ordered ffSprt element, and then translate it, for example, into $ffS_1 f\natural$- the fuzzy fcapacity in itself. Take the receipt $\frac{\partial^2 u}{\partial x^2}$ as an example. Here is the sequence of steps 1)$\frac{\partial u}{\partial x}$ 2)$\frac{\partial}{\partial x}(\frac{\partial u}{\partial x})$. "collapses" into an ordered ffSprt $\underset{q}{\overset{\{\frac{\partial u}{\partial x}, \frac{\partial}{\partial x}(\frac{\partial u}{\partial x})\}}{\mu}}$, which can be translated into the corresponding $ffS_1 f\natural$. The differential operator $ffSprt \underset{q}{\overset{\{\frac{\partial u}{\partial x}, \frac{\partial}{\partial x}(\frac{\partial u}{\partial x})\}}{\mu}}$ - is interesting too.

Remark 2.1.3. ffSprt-displacement of A from B with measure of fuzziness µ will be denote by $\underset{A}{\overset{B}{\mu}}$ffSprt. Then the notation $\underset{D}{\overset{C}{\mu_2}}$ffSprt$\underset{B}{\overset{A}{\mu_1}}$ is both the ffSprt-containment of A in B with measure of fuzziness μ_2 and ffSprt-displacement of D from C with measure of fuzziness μ_1 simultaneously.

We can consider the concept of ffSprt - element as ffSprt$\underset{B}{\overset{A}{\mu}}$, where A fuzzy fits with measure of fuzziness µ in the fuzzy capacity B. Then ffSprt$\underset{B}{\overset{B}{\mu}}$ it will mean $ffS_1 f\mu B$

About ffSprt and ffS_3f programming.

The ideology of ffSprt and ffS_3f can be used for programming. Here are some of the ffSprt programming operators:

1. Simultaneous fuzzy assignment of the expressions $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2)|,\ldots,p_n|\mu_{\widetilde{p}}(p_n)|$) to the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,x_n|\mu_{\widetilde{x}}(x_n)|$).
   This is implemented via ffSprt $\begin{matrix}\{\{\widetilde{x}\} := \{p\}\} \\ \mu \\ q\end{matrix}$.
2. Simultaneous fuzzy checking the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2)|,\ldots,g_n|\mu_{\widetilde{g}}(g_n)|$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|t_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2)|,\ldots,x_n|\mu_{\widetilde{B}}(B_n)|$) Implemented via ffSprt $\begin{matrix}IF\left\{\left\{\widetilde{B}\right\}\{\widetilde{g}\}\right\}\ then\ \widetilde{Q} \\ \mu \\ q\end{matrix}$. where $\widetilde{Q}$ can be anything.
3. Similarly for loop operators and others.

ff$S_3f\mu$– fuzzy software operators will differ only in that the aggregates $\{\widetilde{x}\}, \{\widetilde{p}\}, \left\{\widetilde{B}\right\}, \{\widetilde{g}\}$ will be formed from the corresponding ffSprt program operators in form (1.1) or forms (1.1.1) - (1.4), (1.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*) for more complex operators.

The OS (operating system), the computer's principles, and the modes of operation for this programming are interesting. But this is already the material for the following publications.

Using elements of the mathematics of ffSprt we introduce the concept of ffSprt–change in physical fuzzy quantity $\widetilde{B}$: $ffSprt\begin{matrix}\left\{\Delta_1\widetilde{B} :, ..., \Delta_n\widetilde{B}\right\} \\ \mu \\ q\end{matrix}$.

Then the mean ffSprt- velocity will be $\widetilde{v}_{\text{cpffsprt}}(\text{t}, \text{t}) = ffSprt\begin{matrix}\left\{\frac{\Delta_1\widetilde{B}}{\Delta t}, \ldots \frac{\Delta_n\widetilde{B}}{\Delta t}\right\} \\ \mu \\ q\end{matrix}$.

and ffSprt-velocity at time t:$\widetilde{v}_{ffsprt} = \lim_{\Delta t\to 0}\widetilde{v}_{ffsprt}(t, \Delta t)$. ffSprt–acceleration $a_{ffsprt} = \frac{d\widetilde{v}_{ffsprt}}{dt}$.

When using ffSprt with "target weights", we get, depending on the "target weights", one or another modification, namely, for example, the velocity $\widetilde{v}^g_{ffsprt}$(with a "target weight" f in the case when two fuzzy velocities $\widetilde{v_1}, \widetilde{v_2}$ are involved in the fuzzy set $\{\widetilde{v_1}g, \widetilde{v_2}\}$ for $\widetilde{v}^g_{ffsprt} = ffSprt\begin{matrix}\{\widetilde{v_1}g, \widetilde{v_2}\} \\ \mu \\ x\end{matrix}$ g instantaneous replacement we get an instantaneous substitution $\widetilde{v_1} by\ \widetilde{v_2}$ at point x of space at time t_0.

Consider, in particular, some examples: 1) $ffSprt\begin{matrix}e \\ \mu \\ \left\{x1|\mu_{\widetilde{x}}(x1),\ x2|\mu_{\widetilde{x}}(x2)\right\}\end{matrix}$

describes the fuzzy presence of the same electron e at two different fuzzy points $\{x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2)\}$. 2) The nuclei of atoms can be considered as Srt elements.

Similarly, the concepts of ffSprt- force and ffSprt- energy, for example, $E\mu^{g}_{ffsprt} = ffSprt \begin{matrix}\{E_1 g, E_2\}\\ \mu \\ x\end{matrix}$ it would mean the instantaneous fuzzy replacement of fuzzy energy E_1 by E_2 at time t_0. Two aspects of ffSprt– energy should be distinguished: 1) carrying out the desired "target weight" and 2) fixing the result of it. Do not confuse energy - ffSrt (the node of energies) with fSprt– energy that generates the node of energies, usually with the "target weights." In the case of ordinary energies, the energy node is carried out automatically.

Remark 2.1.4. ffSprt– elements are all ordinary, but with "target weights," they become peculiar. Here you need the necessary energy to carry them out. As a rule, this energy is at the level of Fself. This is natural since it's much easier to manage elements of the k level via the elements of a more structured k +1 level. Let us consider the concepts of capacities of physical objects in themselves. The question arises about the fself-energy of the object. In particular, according to the results of the publication [4] - [16]: «St^{B}_{B} will mean S_1f B.». For example, $ffSprt \begin{matrix}\{DNA\}\\ \mu \\ DNA\end{matrix}$ allows you to reach the level of DNA fself-energy, $ffSprt \begin{matrix}\{Q\}\\ \mu \\ Q\end{matrix}$ allows you to reach the level of fself-energy Q. The law of fself-energy conservation operates already at the level of fself-energy. Also, in addition to capacities in themselves, you can consider the types of fuzzy containment of oneself in oneself: the first type of the fuzzy containment of oneself in oneself: the second type of the fuzzy containment of oneself in oneself: potentially, for example, in the form of programming oneself, the third type is partial fuzzy containment of oneself in themselves—for example, fself-operator, fself-action, whirlwind. A container fuzzy containing itself can be formed by fself-containment, i.e., fuzzy containment in oneself. Let us clarify the concept of the term fuzzy fcapacity in itself: it is a fuzzy fcapacity containing itself potentially. Consider fself-Q, where Q can be anything, including Q=fself; in particular, it can be any action. Therefore, fself-Q is when Q is fuzzy made by itself; it makes itself. There is a partial fself-Q for any Q with partial fself-fulfillment. Let's consider several examples for capacities in themselves: ordinary lightning, electric arc discharge, and ball lightning.

A fself-search of the solution of the equations $\widetilde{f_i(\widetilde{x})}=0$, where i=1,2,. . . ,n, $\widetilde{x}=(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,x_n|\mu_{\widetilde{x}}(x_n)|)$. will be realized in

$$ffSprt_{\substack{\mu\\ a}}\left\{f_1(\widetilde{x})=0?\widetilde{x}|\mu_{\widetilde{f(\widetilde{x})}}\widetilde{(f_1(\widetilde{x})=0)},\ f_2(\widetilde{x})=0?\widetilde{x}|\mu_{\widetilde{f(\widetilde{x})}}\widetilde{(f_2(\widetilde{x})}=0,\dots,f_n(\widetilde{x})=0?x|\mu_{\widetilde{f(\widetilde{x})}}\widetilde{(_n(\widetilde{x})}=0\right\}$$

or

$$ffSprt_{\substack{\mu\\ ?\widetilde{x}}}\left\{f_1(\widetilde{x})=0|\mu_{\widetilde{f(\widetilde{x})}}\widetilde{(f_1(\widetilde{x})}=0),\ f_2(\widetilde{x})=0|\mu_{\widetilde{f(\widetilde{x})}}\widetilde{(f_2(\widetilde{x})}=0,\dots,f_n(\widetilde{x})=0|\mu_{\widetilde{f(\widetilde{x})}}\widetilde{(f_n(\widetilde{x})}=0\right\}$$

$\widetilde{x}$ acquires more degree of liberty and in this is direct decision. Fself-equation for $\widetilde{x}$ has its decision for $\widetilde{x}$ in direct kind.

The same for $ffSprt_{\substack{\mu\\ ?\widetilde{x}}}^{\{tasks\widetilde{x})\}}$.

Fself-task for x has its decision for x in direct kind. Fself-question has its answer for $\widetilde{x}$ in direct kind. $ffSprt_{\substack{\mu\\ \{t\}}}^{\{(\widetilde{o},\widetilde{x})\}}$, where $\{t\}$- time points set, $(\widetilde{o},\widetilde{x})$- object $\widetilde{o}$ in point $\widetilde{x}$ from space X, give to enter in necessary time moments. The same for $ffSprt_{\substack{\mu\\ \{t\}}}^{\widetilde{o}}$. $Sprt_{\alpha}^{\{God-father, God-son, Holy\ Spirit\}}$is three-concept representation, where is a point in the connectedness space. ffSprt is also great for working with fuzzy structures, for example: 1) $ffSprt_{\substack{\mu\\ \widetilde{B}}}\left\{\widetilde{strA}\right\}$ - the fuzzy structure A that fuzzy fits into $\widetilde{B}$, where $\widetilde{B}$ can be any fuzzy capacity, another fuzzy structure etc. 2) $ffSprt_{\substack{\mu\\ \widetilde{R}}}\left\{\widetilde{str_Q}\right\}$ - fuzzy embedding fuzzy structure from Q into $\widetilde{R}$. Similarly for fuzzy displacement: 1) $\underset{\widetilde{strA}}{\overset{\widetilde{B}}{\mu}}$ ffSprt -displacement of fuzzy structure A from $\widetilde{B}$, 2) $\underset{\widetilde{str_Q}}{\overset{\widetilde{B}}{\mu}}$ ffSprt -displacement of the fuzzy structure Q from $\widetilde{B}$. To work with fuzzy structures, you can introduce a special operator ffCprt: $ffCprt_{\substack{\mu\\ \widetilde{B}}}\left\{\widetilde{strA}\right\}$ -

structures $\widetilde{B}$ with the fuzzy structure A, $ffCprt\begin{matrix}\left\{\widetilde{str_Q}\right\}\\ \mu\\ \widetilde{R}\end{matrix}$ - fuzzy structures $\widetilde{R}$ with the fuzzy structure from Q, $\begin{matrix}\widetilde{B}\\ \mu\\ \widetilde{strA}\end{matrix}$ ffCprt destructors $\widetilde{B}$ by the fuzzy structure A, $\begin{matrix}\widetilde{B}\\ \mu\\ \widetilde{str_Q}\end{matrix}$ ffCprt destructors $\widetilde{B}$ by the fuzzy structure that fuzzy structures Q.

Definition 2.1.8. A fuzzy structure with a second degree of freedom will be called fuzzy complete, i.e., "capable" of reversing itself concerning any of its elements explicitly, but not necessarily in known operators; it can form (create) new special foperators (in particular, special fuzzy functions).

In particular, $ffCprt\begin{matrix}\left\{\widetilde{strA}\right\}\\ \mu\\ \left\{\widetilde{strA}\right\}\end{matrix}$ is such structure.

Similarly, for working with models, each is structured by its structure; for example, use ffSprt-groups, ffSprt-rings, ffSprt-fields, ffSprt-spaces, fself-fgroups, fself-frings, fself-ffields, and fself-fspaces. Like any task, this is also a fuzzy structure of the appropriate fuzzy capacity.

Fself-H (fself-hydrogen), like other fself-particles, does not exist in the ordinary, but all fself-molecules, fself-atoms, and fself-particles are elements of the fuzzy energy space.

Remark 2.1.5. The concept of elements of physics fuzzy fSprt is introduced for energy space. The corresponding concept of elements of chemistry Sprt is introduced accordingly. Examples: 1) $E^{g}_{ffsprt} = ffSprt\begin{matrix}\{a_1 g, a_2\}\\ \mu\\ x\end{matrix}$ the energy of instantaneous substitution a_1 by a_2, where a_1, and a_2 are chemical elements, g is instant replacement. Similarly, $ffSprt\begin{matrix}\{chemical\ elements\ with\ "tone\ can\ consider\ for\ the\ node\ of\ chemical\ reactions\ ffSprtarget\\ \mu\\ reaction\end{matrix}$. The periodic table itself can also be thought of as the Srt − element: $Srt^{\{list\ of\ chemical\ elements\}}_{Mendeleev\ table}$ The ideology of ffSprt elements allows us to go to the border of the world familiar to us, which allows us to act more effectively.

2.2 Fuzzy dynamic fuzzy Sprt − elements

We considered stationary fuzzy Sprt − elements earlier. Here we consider fuzzy dynamic fuzzy Sprt – elements (designation - fdfSprt) [1].

Definition 2.2.1. The process of fitting a fuzzy set of elements $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|,…,x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))|). into one point q(t) of space X at time t will be called a fuzzy dynamic fSprt–element (fdfSprt). We will denote $ffSprt(t)\begin{matrix}\widetilde{x(t)}\\ \mu(t)\\ q(t)\end{matrix}$.

Definition 2.2.2. Fuzzy fitting an ordered fuzzy set of elements into one point in space is called a fuzzy dynamic ordered fSprt–element.

It is allowed to sum fuzzy dynamic fSprt– elements:

$$ffSprt(t)\begin{matrix}\widetilde{a(t)}\\ \mu(t)\\ q(t)\end{matrix} + ffSprt(t)\begin{matrix}\widetilde{b(t)}\\ \mu(t)\\ q(t)\end{matrix} = ffSprt(t)\begin{matrix}\widetilde{a(t)}\cup\widetilde{b(t)}\\ \mu(t)\\ q(t)\end{matrix}$$

Dynamic fuzzy containment of oneself.

Definition 2.2.3. Fuzzy dynamic fuzzy capacity Q(t) is fuzzy fitting into Q(t).

Definition 2.2.4. Fuzzy dynamic fSprt-capacity $ffSprt(t)\begin{matrix}R(t)\\ \mu(t)\\ Q(t)\end{matrix}$ is the process of fuzzy embedding fuzzy R(t) into fuzzy Q(t).

Definition 2.2.5. The dynamic fuzzy capacity A(t) fuzzy containing itself as an element of the first type is the process of fuzzy containing A(t) in A(t). Denote $ffS_1f(t)A(t)$.

Definition 2.2.6. Fuzzy dynamic fcapacity C(t) in itself of the second type is the process of fuzzy containing elements from which it can be generated. Let's denote $ffS_2f(t)C(t)$.

Definition 2.2.7. Fuzzy dynamic partial fuzzy capacity B(t) in itself of the third type is a process of partial fuzzy containment of B(t) in itself or elements from which it can be generated partially or both at the same time. Denote $ffS_3f(t)B(t)$.

All dynamic fuzzy capacities in themselves in a dynamic fself-space are, by definition, capacities in themselves. Dynamic fuzzy capacity itself can manifest itself as dynamic fSprt-capacity and ordinary dynamic fuzzy capacity. In these cases, the usual measures and methods of topology are used.

Connection of fuzzy dynamic fSprt– elements with dynamic fuzzy containment of oneself.
Consider third type of dynamic partial fuzzy containment of oneself.

For example, based on $ffSprt(t)\begin{matrix}\widetilde{x(t)}\\ \mu(t)\\ q(t)\end{matrix}$, where $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|, …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))|), i.e. n – elements at one point q(t), we can consider the dynamic fuzzy capacity in itself $ffS_3f(t)$ with m elements from $\widetilde{x(t)}$, m<n, which is process formed according to the form (1.1), that is, only m elements from $\widetilde{x(t)}$ are in the structure $ffSprt(t)\begin{matrix}\widetilde{x(t)}\\ \mu(t)\\ q(t)\end{matrix}$.

Dynamic fuzzy containment of oneself of the third type can be formed for any other structure, not necessarily ffSprt, only through the obligatory reduction in the number of elements in the structure. In particular, using the forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) – (1.4) by type (2.1*). It is possible to introduce structures more complex than fS_3f(t).

Fuzzy dynamic math fuzzy itself.

1. The process of simultaneous fuzzy addition of a fuzzy set of elements $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|,…,x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))|) realized by $ffSprt(t)\begin{matrix}\widetilde{x(t)+}\\ \mu(t)\\ w(t)\end{matrix}$.

2. By analogy, for simultaneous multiplication: $ffSprt(t)\begin{matrix}\widetilde{x(t)*}\\ \mu(t)\\ w(t)\end{matrix}$

3. Similarly for simultaneous execution of various operations: $ffSprt(t)\begin{matrix}\widetilde{x(t)}\widetilde{q(t)}\\ \mu(t)\\ w(t)\end{matrix}$, where $\widetilde{q(t)}$=(q_1(t)|$\mu_{\widetilde{q(t)}}$(q_1(t)), q_2(t)|$\mu_{\widetilde{q(t)}}$(q_2(t))|, …, q_n(t)|$\mu_{\widetilde{q(t)}}$(q_n(t))|), q_i(t)-an operation, i = 1, …, n.

4. Similarly, for the simultaneous execution of various operators: $ffSprt(t)\begin{matrix}\widetilde{F(t)}\widetilde{x(t)}\\ \mu(t)\\ w(t)\end{matrix}$ where $\widetilde{F(t)}$=(F_1(t)|$\mu_{\widetilde{F(t)}}$(F_1(t)), F_2(t)|$\mu_{\widetilde{F(t)}}$(F_2(t))|,…,F_n(t)|$\mu_{\widetilde{F(t)}}$(F_n(t))|), F_i(t) is an operator, i = 1,…,n.

5. Dynamic arithmetic fuzzy itself for containments of oneself will be similar: dynamic addition - $ffS_1f(t)\mu(t)\left\{\widetilde{x(t)+}\right\}$, (or $ffS_3f(t)\mu(t)\left\{\widetilde{x(t)+}\right\}$ for the third type), dynamic multiplication $ffS_1f(t)\mu(t)\left\{\widetilde{x(t)*}\right\}$, (or $ffS_3f(t)\mu(t)\left\{\widetilde{x(t)*}\right\}$

6. Similarly with different fuzzy operations: $ffS_1f(t)\mu(t)\left\{\widetilde{x(t)}\widetilde{q(t)}\right\}$, (or $ffS_3f(t)\mu(t)\left\{\widetilde{x(t)}\widetilde{q(t)}\right\}$) and with different fuzzy operators: $ffS_1f(t)\mu(t)\left\{\widetilde{F(t)}\widetilde{x(t)}\right\}$, (or $ffS_3f(t)\mu(t)\left\{\widetilde{F(t)}\widetilde{x(t)}\right\}$).

7. $ffSprt(t)\underset{\widetilde{B(t)}}{\overset{\widetilde{A(t)}}{\mu(t)}}$ *gives* the result

$$ffSrt(t)\underset{\widetilde{B(t)}}{\overset{\widetilde{A(t)}}{\mu(t)}} = \left\{\widetilde{A(t)}\cup\widetilde{B(t)}-\widetilde{A(t)}\cap\widetilde{B(t)}, D(t)\right\}\ for\ fuzzy\ sets\ \widetilde{A(t)},\widetilde{B(t)},$$

where $\widetilde{D(t)}$ is fself- (fuzzy set) for $\widetilde{A(t)}\cap\widetilde{B(t)}$. The same is true for structures if they are treated as fuzzy sets.

8. Similarly, for fuzzy dynamic fSprt-derivatives, fuzzy dynamic fSprt-integrals, fuzzy dynamic fSprt-lim, dynamic fself- (fuzzy derivatives), dynamic fself- (fuzzy integrals)

9. Denote dynamic fself-(dynamic fself-Q(t)) through dynamic fself^2-Q(t) , fgS(t)(n,Q(t))= dynamic fself-(dynamic fself-(… (dynamic fself-Q(t)))) = dynamic fself^n-Q(t) for n-multiple dynamic fself.

Remark 2.2.1. The fuzzy dynamic fSprt-displacement of A(t) from B(t) will be denote by $\underset{A(t)}{\overset{B(t)}{\mu}}$ ffSprt(t) . Then the notation $\underset{D(t)}{\overset{C(t)}{\mu_2(t)}}$ffSprt(t)$\underset{B(t)}{\overset{A(t)}{\mu_1(t)}}$ is fuzzy dynamic fSprt-containment of A(t) in B(t) and fuzzy dynamic fSprt-displacement of D(t) from C(t) simultaneously. Denote $\underset{A(t)}{\overset{B(t)}{\mu_2(t)}}$ffSprt(t)$\underset{B(t)}{\overset{A(t)}{\mu_1(t)}}$ by $fTfS(t)^{A(t)}_{B(t)}$, $\underset{A(t)}{\overset{A(t)}{\mu_2(t)}}$ffSprt(t)$\underset{A(t)}{\overset{A(t)}{\mu_1(t)}}$ $-$ *through* $fTfS(t)^{A(t)}_{A(t)}$. We can consider the concept of fuzzy dynamic fSprt- element as $ffSprt(t)\underset{\widetilde{B(t)}}{\overset{\widetilde{A(t)}}{\mu(t)}}$, where $\widetilde{A(t)}$ fits in fuzzy dynamic fuzzy capacity $\widetilde{B(t)}$. Then $ffSprt(t)\underset{\widetilde{B(t)}}{\overset{\widetilde{B(t)}}{\mu(t)}}$ will mean ffS$_1$f(t)$\widetilde{B(t)}$. Denote $ffSprt(t)\underset{\widetilde{B(t)}}{\overset{\widetilde{B(t)}}{\mu(t)}}$ by ffL(t)(B(t)). $\underset{A(t)}{\overset{A(t)}{\mu}}$ ffSprt(t) denotes the fuzzy dynamic expelling fuzzy A(t) oneself out of oneself,

$\underset{A(t)}{\overset{A(t)}{\mu_2(t)}}$ffSprt(t)$\underset{A(t)}{\overset{A(t)}{\mu_1(t)}}$ —simultaneous fuzzy dynamic containment fuzzy A(t) of oneself in oneself and fuzzy dynamic expelling fuzzy A(t) oneself out of oneself. $\underset{A(t)}{\overset{A(t)}{\mu}}$ ffSprt(t)will be called fuzzy dynamic anti- (fuzzy capacity) from oneself. For example, "white hole" in physics is such simple anti-(fuzzy capacity). The concepts of "white hole" and "black hole" were formulated by the physicists based on the subject of physics -the usual energies level. Mathematics allows you to deeply find and formulate the concept of singular points in the Universe based on the levels of more subtle energies. The experiments of the 2022 Nobel laureates Asle Ahlen, John Clauser, Anton Zeilinger correspond to the concept of the Universe as a capacity in itself. The energy of fself-containment in itself is closed on itself.

Hypothesis: the containment of the galaxy in oneself as a spiral curl and the expelling out of oneself defines its existence. A fself- capacity in itself as an element A is the god of A, the fself-capacity in itself as an element the globe—the god of the globe, the fself-capacity in itself as an element man-- the god of the man, the fself-capacity in itself as an element of the universe-- the god of the universe, the containment of A into oneself is spirit of A, the containment of the Earth into oneself is is spirit of Earth, the containment of the man into oneself is spirit of the man (soul), the containment of the universe into oneself is spirit of the universe. We may consider the following axiom: any fuzzy capacity is the fuzzy capacity of oneself. This is for each energy fuzzy capacity.

Remark 2.2.2. With the help of fuzzy dynamic fSprt-elements, the concepts of fuzzy dynamic fSprt- force, fuzzy dynamic fSprt– energy are introduced. For example $ffE(t)\underset{prst}{\overset{\widetilde{g}}{\mu(t)}} = fSt(t)_{x(t)}\ ffSprt(t) \quad \underset{\widetilde{w(t)}}{\overset{\{ffE_1(t)\widetilde{g}, ffE_2(t)\}}{\mu(t)}}$. will mean the process of instantaneous replacement $\widetilde{g}$ of energy ffE$_1$(t) by ffE$_2$(t) at time t. Similarly, using $ffS_i f(t)\mu(t)$, the concepts of ffS$_i$f(t)$\mu(t)$-force, ffS$_i$f(t)$\mu(t)$-energy, i=1,2,3, and etc are introduced.

Remark 2.2.3. It is the fuzzy containment of oneself in oneself that can "give birth" to the fcapacities in itself – that is what fself-organization is.

Remark 2.2.4. $ffSprt(t)\,\mu(t)^{ffSprt(t)\,\mu(t)_{\widetilde{B(t)}}^{\widetilde{B(t)}}}_{ffSprt(t)\,\mu(t)_{\widetilde{B(t)}}^{\widetilde{B(t)}}}$ can increase fself- level of $\widetilde{B(t)}$.

Remark 2.2.5. For example, the foperator itself is ffS$_1$f(t)$\mu(t)$.

Remark 2.2.6. May be considered the following derivatives: $\frac{d ffSprt(t)\,\mu(t)_{\widetilde{B(t)}}^{\widetilde{A(t)}}}{dt}$, $\frac{d\,\mu_{A(t)}^{B(t)}\,ffSprt(t)}{dt}$, $\frac{d\mu_2(t)_{D(t)}^{C(t)}\,ffSprt(t)\,\mu_1(t)_{B(t)}^{A(t)}}{dt}$, $\frac{d ffS_i f(t)\mu(t)}{dt}$, i=1,2,3.

Remark 2.2.7. It is the fuzzy containment of oneself in itself as an fuzzy element that can be interpreted as fuzzy dynamic fuzzy capacities in itself.

Remark 2.2.8. Not every fuzzy capacity fuzzy containing itself as an element will manifest itself as a sedentary fuzzy capacity or fuzzy capacity.

About fuzzy dynamic fSprt and ffS$_3$f(t) programming [1].

The ideology of fuzzy dynamic fSprt and ffS$_3$f(t) can be used for programming:

1. The process of simultaneous fuzzy assignment of the fuzzy expressions $\widetilde{p(t)}$=(p$_1$(t)|$\mu_{\widetilde{p(t)}}$(p$_1$(t)), p$_2$(t)|$\mu_{\widetilde{p(t)}}$(p$_2$(t))|,…,p$_n$(t)|$\mu_{\widetilde{p(t)}}$(p$_n$(t))|) to the fuzzy variables ($\widetilde{x(t)}$=(x$_1$(t)|$\mu_{\widetilde{x(t)}}$(x$_1$(t)), x$_2$(t)|$\mu_{\widetilde{x(t)}}$(x$_2$(t))|,…,x$_n$(t)|$\mu_{\widetilde{x(t)}}$(x$_n$(t))|).
   This is implemented via $ffSprt(t)^{\left\{\left\{\widetilde{x(t)}\right\}\widetilde{\{\widetilde{p(t)}\}}\right\}}_{\mu(t)_{\widetilde{w(t)}}}$.

2. The process of simultaneous checking the fuzzy set of conditions $\widetilde{g(t)}$=(g$_1$(t)|$\mu_{\widetilde{g(t)}}$(g$_1$(t)), g$_2$(t)|$\mu_{\widetilde{g(t)}}$(g$_2$(t))|,…,g$_n$(t)|$\mu_{\widetilde{g(t)}}$(g$_n$(t))) for the fuzzy set of expressions $\widetilde{B(t)}$=(B$_1$(t)|$\mu_{\widetilde{B(t)}}$(B$_1$(t)), B$_2$(t)|$\mu_{\widetilde{B(t)}}$(B$_2$(t))|,…,B$_n$(t)|$\mu_{\widetilde{B(t)}}$(B$_n$(t))). Implemented via $fSt(t)_{w(t)}\ ffSprt(t)^{IF\left\{\left\{\widetilde{B(t)}\right\}\widetilde{\{\widetilde{g}(t)\}}\right\}\ then\ \widetilde{Q(t)}}_{\mu(t)_{\widetilde{w(t)}}}$ where $\widetilde{Q(t)}$

can be anything.

3. Similarly for fuzzy loop operators and others.

$ffS_3f(t)\mathfrak{z}(t)$– fuzzy fsoftware operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}, \widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding processes by ffSprt-program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

2.3 ffSprt – elements for continual fuzzy sets

Earlier, we considered finite-dimensional discrete ffSprt-elements and fself- (fuzzy capacities) in itself as an element. Here we believe some continual ffSprt-elements and continual fself- (fuzzy capacities) in themselves as an element [1].

Definition 2.3.1. The fuzzy set of continual elements $\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$)|,…,x$_n|\mu_{\widetilde{x}}$(x$_n$)) at one point w of space X with measure of fuzziness µ will be called continual ffSprt – element, and such a point in space will be called fuzzy fcapacity of the continual ffSprt – element. We will denote ffSprt$\underset{w}{\overset{\widetilde{x}}{\mu}}$.

Definition 2.3.2. An ordered fuzzy set of continual elements at one point in space with measure of fuzziness µ is called an ordered continual ffSprt– element.

It's allowed to sum continual ffSprt – elements: $ffSprt\underset{w}{\overset{\widetilde{x}}{\mu}} + ffSprt\underset{w}{\overset{\widetilde{r}}{\mu}} = ffSprt\underset{w}{\overset{\widetilde{x}\cup\widetilde{r}}{\mu}}$, where some or any elements may be ordered elements.

Definition 2.3.3. The continual fself- (fuzzy capacity) A in itself as an element of the first type is the fuzzy capacity containing itself as an element. Denote fS_1fA.

Definition 2.3.4. The ordered continual fself- (fuzzy capacity) A in itself as an element of the first type is the ordered fuzzy capacity containing itself as an element. Denote $ff\overrightarrow{S_1fA\mathfrak{z}}$.

For example, fS^+_∞=(sin$|\mu_{\widetilde{sin\infty}}$(sin)) is of this type. It denotes continual ordered fself- (fuzzy capacities) in itself as an element of following type-the fuzzy range of simultaneous "activation" of fuzzy numbers from [-1,1] in mutual directions: $\uparrow I \downarrow^1_{-1}$. Also, consider the following elements: fS^-_∞=(sin-$|\mu_{\widetilde{sin-\infty}}$(sin-∞))$-\downarrow I \uparrow^1_{-1}$, , fT^+_∞=(tg∞$|\mu_{\widetilde{tg\infty}}$(tg∞))$-\uparrow I \downarrow^\infty_{-\infty}$, fT^-_∞=(tg-∞$|\mu_{\widetilde{tg-\infty}}$(tg-∞)) $- \downarrow I \uparrow^\infty_{-\infty}$, don't confuse with values of these functions.

Such elements can be summarized. For example: afS_∞^++bfS_∞^-=(a-b)$fS_\infty^+ = (b-a)\ fS_\infty^-$, if fS_∞^+ and fS_∞^- have the same fuzzy structure of the fuzzy range from [-1,1].

Definition 2.3.5. The continual fself-(fuzzy capacity) A in itself, as an element of the second type, is the fuzzy capacity containing fuzzy elements from which it can be generated. Let's denote $fS_2fA\mu$.

An example of continual fself-capacity in itself as an element of the second type is a living organism since it contains the programs: DNA and RNA.

Definition 2.3.6. Partial continual fself- (fuzzy capacity) A in itself as an element of the third type is called continual (fself- fuzzy capacity) in itself as an element that partially contains itself or contains fuzzy elements from which it can be generated in part or both simultaneously. Denote $ffS_3fA\mu$.

Also, may be considered operators for them. For example:

gS_∞^+(t-t$_0$) $= \begin{cases} fS_\infty^+\ ,\ t=t_0 \\ 0,\ t\neq\ t_0 \end{cases}$

You can set the fself- vector: $\uparrow \quad \begin{pmatrix} q \\ d \\ c \end{pmatrix} \quad \downarrow$

All continual capacities in fself-space are continual fself-capacities in itself as an element by definition. The continual fself-capacities in itself as an element may appear as continual Sprt- capacities and usual continual capacities. In these cases, there are used typical measure and topology methods.

The connection of continual ffSprt – elements with continual fself- (fuzzy capacities) in themselves as an element.
Consider a third type of continual fself- (fuzzy capacity) in itself as an element. For example, based on ffSprt$\underset{w}{\overset{\widetilde{x}}{\mu}}$, where $\widetilde{x}$=(x$_1$|$\mu_{\widetilde{x}}$(x$_1$), x$_2$|$\mu_{\widetilde{x}}$(x$_2$)|,…,x$_n$|$\mu_{\widetilde{x}}$(x$_n$)), i.e. x_i - continual elements at one point, i=1, 2, …, n. The continual fself- (fuzzy capacity) in itself as an element with m continual elements from $\widetilde{x}$, at m<n, can be considered as ffS_3f, which is formed by form (1.1) or forms (1.1.1) - (1.4), (1.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*) for more complex operators.

Mathematics fuzzy itself for continual ffSprt-elements [1].

1.Simultaneous addition of the continual elements of the fuzzy set $\widetilde{x}$=(x$_1$|$\mu_{\widetilde{x}}$(x$_1$), x$_2$|$\mu_{\widetilde{x}}$(x$_2$)|,…,x$_n$|$\mu_{\widetilde{x}}$(x$_n$)) is implemented using ffSprt $\underset{w}{\overset{\widetilde{x}\cup}{\mu}}$.

2. By analogy, for simultaneous multiplication: ffSprt $\underset{w}{\overset{\widetilde{x}\cap}{\mu}}$.

3. Similarly, for simultaneous execution of various operations: ffSprt $\underset{w}{\overset{\widetilde{x}\widetilde{q}}{\mu}}$,
where $\widetilde{q}$=($\mathrm{q}_1|\mu_{\widetilde{q}}(\mathrm{q}_1)$, $\mathrm{q}_2|\mu_{\widetilde{q}}(\mathrm{q}_2)|, \ldots,$ $\mathrm{q}_\mathrm{n}|\mu_{\widetilde{q}}(\mathrm{q}_\mathrm{n})$), q_i -an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: ffSprt $\underset{w}{\overset{\widetilde{F}\widetilde{x}}{\mu}}$, where $\widetilde{F}$=($\mathrm{F}_1|\mu_{\widetilde{F}}(\mathrm{F}_1)$, $\mathrm{F}_2|\mu_{\widetilde{F}}(\mathrm{F}_2)|, \ldots,$ $\mathrm{F}_\mathrm{n}|\mu_{\widetilde{F}}(\mathrm{F}_\mathrm{n})$),$\mathrm{F}_\mathrm{i}$ is an operator, i = 1, . . . , n.

5. For continual fself-fcapacities in themself as an element will be similar: addition - $ffS_1f\mu\,\{\widetilde{x}+\}$, (or $ffS_3f\mu\,\{\widetilde{x}+\}$ for the third type), multiplication $ffS_1f\mu\,\{\widetilde{x}*\}$, (or $ffS_3f\mu\,\{\widetilde{x}*\}$.

6. Similarly with different operations: $ffS_1f\mu\,\{\widetilde{x}\widetilde{q}\}$, $(ffS_3f\mu\,\{\widetilde{x}\widetilde{q}\})$, and with different operators: $ffS_1f\mu\left\{\widetilde{F}\widetilde{x}\right\}$, $(ffS_3f\mu\left\{\widetilde{F}\widetilde{x}\right\})$.

7. $ffSrt\underset{B}{\overset{A}{\mu}} -\ is$ the result of the fuzzy containment operator.

For fuzzy sets A, B we have

$ffSrt\underset{B}{\overset{A}{\mu}} = \{A\cup B - A\cap B\ , D\}$, where D is fself-(fuzzy set) for $A \cap B$. The same is true for structures if they are treated as fuzzy sets.

Remark 2.3.1. ffSprt-displacement of A from B with measure of fuzziness μ will be denoted by $\underset{A}{\overset{B}{\mu}}$ffSprt. Then the notation $\underset{D}{\overset{C}{\mu_2}}$ffSprt$\underset{B}{\overset{A}{\mu_1}}$ is fSprt-containment of A in B with measure of fuzziness μ_1, and fSprt-displacement of D from C with measure of fuzziness μ_2, simultaneously. Denote $\underset{A}{\overset{B}{\mu_2}}$ffSprt$\underset{B}{\overset{A}{\mu_1}}$ by $fTfS^A_B$, $\underset{A}{\overset{A}{\mu_2}}ffSprt\underset{A}{\overset{A}{\mu_1}}\ - through\ fTfS^A_A$. Three in one is $St_\propto^{\{\infty\ in\ itself,\ an\ element\ that\ is\ not\ anyone's\ element,\ \ 0\ out\ oneself\}}$, - point space connectedness.

We can consider the concept of a continual fuzzy Sprt - element as $ffSprt\underset{B}{\overset{A}{\mu}}$, where A fits with measure of fuzziness μ in continual fuzzy capacity B.

Then $ffSprt\underset{B}{\overset{B}{\mu}}$ will mean ffS$_1$fμB.

These elements are used for ffSprt-coding, ffSprt-translation, fuzzy coding fself, and fuzzy translation fself for networks, which is suitable for electric current of ultrahigh frequency. More complex elements can be considered as continual sets of numbers with their " activation " in mutual directions. For example, ranges of function values, particularly those representing the shape of lightning. Differential geometry can be applied here. Also, n-dimensional elements can be considered. The space of such elements is Banach space if we introduce the usual norm for functions or vectors. We call this space - fSelb-space. Then we introduce the scalar product for functions or vectors and get the Hilbert space. We call this space fSelh-space. In particular, one can try to describe some processes with these elements by differential equations and use methods from [18]. You can also try to optimize and research some processes with these elements using the techniques from [19]. Let's introduce operators for transforming fuzzy capacity to fself- (fuzzy capacity) in itself as an element: fQ$_1$Sfμ(A) transforms A to ff$_1$SfμA, fQ$_0$S(A) transforms A to $\underset{A}{\overset{A}{\mu}}$ffSprt, fSOf$\mu$(A) transforms A to$\uparrow ffA\downarrow$, $\uparrow ffA\downarrow$ -- ordered fself-fuzzy capacity in itself as an element of simultaneous "activation" of all elements of A in mutual directions. For example, fSOfμ([-1,1]) = $ff\mu S^{+}_{\infty}$, fSOfμ([1,-1]) = fS^{-}_{∞} , fSOfμ([-∞,∞]) = fT^{+}_{∞}, fSOfμ([$\infty,-\infty$]) = fT^{-}_{∞}. The operator (fQ$_1$Sfμ(A))^2 increases fself- (fuzzy level) for A: it transforms fself-fA= ff$_1$SfμA to fself^2-ffμA, (fQ$_1$Sfμ(A))$^{\mathrm{n}}$ $\rightarrow$ fself$^{\mathrm{n}}$-ffμA, $e^{fQ_1Sf\mu(A)} \rightarrow e^{ff\mu self} - ff\mu A$. Let us introduce the following notations: $\underset{(A,A)}{\overset{A}{\mu}}$ ffSprt by 2fμoself-fA, $ffSprt\underset{A}{\overset{(A,A)}{\mu}}$ by 2fμself-fA, $ffSprt\underset{(A,A)}{\overset{A}{\mu}}$ by 1/2fμself-fA, $ffSprt\underset{q(A)}{\overset{A}{\mu}}$ by qfμself-fA, $\underset{q(A)}{\overset{A}{\mu}}$ ffSprt by q()fμoself-fA, q-any operator, $\underset{(q_1,\ \ldots,\ q_N))}{\overset{A}{\mu}}$ ffSprt by Nfμoself-fA, q$_{\mathrm{i}}$= A, i= 1, . . . , N; $\underset{A}{\overset{A}{\mu_1}}$ffSprt$\underset{A}{\overset{A}{\mu_1}}$ by f$\mu_1$self-fA- f$\mu_1$oself-fA, $\underset{q_3(A)}{\overset{q_2(A)}{\mu_1}}$ ffSprt $\underset{q_1(A)}{\overset{A}{\mu_1}}$ by q$_1$f$\mu_1$self-fA- $\begin{pmatrix}q_3()\\ q_2()\end{pmatrix} f\mu_1$oself-fA, $\underset{(A,A)}{\overset{A}{\mu}}$ ffCprt by 2Cfμoself-fA, $ffCSprt\underset{A}{\overset{(A,A)}{\mu}}$ by

2Cfμself-fA, $ffCSprt\overset{A}{\underset{(A,A)}{\mu}}$ by 1/2Cfμself-fA, $fCt^{A}_{q(A)}$ by qCfμself-fA, $\overset{A}{\underset{q(A)}{\mu}}$ ffCprt by q()Cfμoself-fA, q-any operator, $\overset{A}{\underset{(q_1,\ \ldots,\ q_N))}{\mu}}$ ffCprt by NCfμoself-fA, q_i= A, i= 1, …, N; $\overset{A}{\underset{A}{\mu_1}}$ffCSprt$\overset{A}{\underset{A}{\mu_1}}$ by Cf$\mu_1$self-fA-Cf$\mu_1$oself-fA, $\overset{q_2(A)}{\underset{q_3(A)}{\mu_1}}$ ffCprt $\overset{A}{\underset{q_1(A)}{\mu_1}}$ by q_1Cf$\mu_1$self-fA-$\begin{pmatrix} q_3() \\ q_2() \end{pmatrix} Cf\mu_1$oself-fA, $ffS2prt\overset{A}{\underset{A}{\mu}}$ = (fμself-fA, fμself-fA), $ffSNprt\overset{A}{\underset{A}{\mu}}$= ($q_1$, …, q_N), q_i= fμself-fA, i= 1, …, N. fμself($ffSprt\overset{A}{\underset{B}{\mu}}$)= $ffSprt\overset{ffSprt\overset{A}{\underset{B}{\mu}}}{\underset{ffSprt\overset{A}{\underset{B}{\mu}}}{\mu}}$. Can be considered Q($\overset{A}{\underset{A}{\mu_1}}$ffSprt$\overset{A}{\underset{A}{\mu_1}}$), Q-any operator.

2.4 Fuzzy dynamic continual fuzzy Sprt − elements

Definition 2.4.1. The process of fuzzy containing with measure of fuzziness μ in the fuzzy set of continual elements $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|,…,x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))|) at one point w of the space X at time will be called the fuzzy dynamic continual fuzzy Sprt − element. We will denote $ffSprt(t)\overset{\widetilde{x(t)}}{\underset{q(t)}{\mu(t)}}$.

Definition 2.4.2. The process of fuzzy containing with measure of fuzziness μ an ordered fuzzy set of continual elements at one point in space is called fuzzy dynamic continual ordered fuzzy Sprt − element.

It is allowed to sum fuzzy dynamic continual fSprt− elements:

$$ffSprt(t)\overset{\widetilde{a(t)}}{\underset{q(t)}{\mu(t)}} + ffSprt(t)\overset{\widetilde{b(t)}}{\underset{q(t)}{\mu(t)}} = ffSprt(t)\overset{\widetilde{a(t)}\cup\widetilde{b(t)}}{\underset{q(t)}{\mu(t)}}$$

Dynamic continual fuzzy containment of oneself in oneself as an element.

Definition 2.4.3. The fuzzy dynamic continual fuzzy capacity Q(t) is called the process of fuzzy embedding in Q(t).

Definition 2.4.4. The fuzzy dynamic continual fSit-capacity $ffSprt(t)\begin{matrix}\widetilde{R(t)}\\ \mu(t)\\ Q(t)\end{matrix}$ $fSt(t)_{Q(t)}^{R(t)}$ is called the process of fuzzy embedding $\widetilde{R(t)}$ in Q(t).

Definition 2.4.5. The dynamic fuzzy containment fuzzy continual A(t) of oneself of the first type is the process of fuzzy putting A(t) into A(t). Denote $fS_1f(t)\mu(t)A(t)$.

Definition 2.4.6. We will call a dynamic fuzzy embedding of a fuzzy continual C(t) into itself a dynamic fuzzy embedding of the second type if it contains fuzzy dynamic continual elements from which it can be generated. Denote $fS_2f(t)\mu(t)C(t)$.

Definition 2.4.7. The partial fuzzy dynamic containment continual fuzzy B(t) of oneself of the third type is the process of partial fuzzy embedding continual fuzzy B(t) into oneself or fuzzy contains dynamic fuzzy continual elements from which it can be generated in part or both simultaneously. Denote $fS_3f(t)\mu(t)B(t)$.

The connection of fuzzy dynamic continual fSprt– elements with dynamic fuzzy containment of oneself in oneself as an element.
Let us consider the partial dynamic continual fuzzy containment of oneself in oneself as an element of the third type. For example, based on $ffSprt(t)\begin{matrix}\widetilde{x(t)}\\ \mu(t)\\ w(t)\end{matrix}$, where $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|,…,x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))|), i.e. n - continual elements at one point x, one can consider the fuzzy dynamic containment $fS_3f(t)\mu(t)$ of oneself in oneself as an element with m continual elements from $\widetilde{x(t)}$, m<n, which is a process that is necessary form according to form (1.1) or forms (1.1.1) - (1.4), (1.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*) for more complex operators. Dynamic fuzzy continual mathematics fself [1].

1. The process of simultaneous fuzzy addition of the fuzzy set of continual elements $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|, …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))) is realized by $ffSprt(t)\begin{matrix}\widetilde{x(t)\cup}\\ \mu(t)\\ w(t)\end{matrix}$.
2. By analogy, for simultaneous fuzzy multiplication: $ffSprt(t)\begin{matrix}\widetilde{x(t)\cap}\\ \mu(t)\\ w(t)\end{matrix}$.

3. Similarly for simultaneous execution of various operations: $ffSprt(t)\begin{matrix}\widetilde{x(t)}\widetilde{q(t)}\\ \mu(t)\\ w(t)\end{matrix}$, where $\widetilde{q(t)}$=(q_1(t)$|\mu_{\widetilde{q(t)}}$(q_1(t)), q_2(t)$|\mu_{\widetilde{q(t)}}$(q_2(t))|, …, q_n(t)$|\mu_{\widetilde{q(t)}}$(q_n(t))|), q_i(t)-an operation, i = 1, …, n.
4. Similarly, for the simultaneous fuzzy execution of various operators: $ffSprt(t)\begin{matrix}\widetilde{F(t)}\widetilde{x(t)}\\ \mu(t)\\ w(t)\end{matrix}$, where $\widetilde{F(t)}$=(F_1(t)$|\mu_{\widetilde{F(t)}}$(F_1(t)), F_2(t)$|\mu_{\widetilde{F(t)}}$(F_2(t))|,…,F_n(t)$|\mu_{\widetilde{F(t)}}$(F_n(t))|), F_i(t) is an operator, i = 1,…,n.
5. The dynamic arithmetic fself for the dynamic continual fuzzy containments of oneself will be similar: dynamic fuzzy addition - $ffS_1f(t)\mu(t)\left\{\widetilde{x(t)}\cup\right\}$, (or $ffS_3f(t)\mu(t)\left\{\widetilde{x(t)}\cup\right\}$ for the third type), dynamic fuzzy multiplication $ffS_1f(t)\mu(t)\left\{\widetilde{x(t)}\cap\right\}$, (or $ffS_3f(t)\mu(t)\left\{\widetilde{x(t)}\cap\right\}$.
6. Similarly with different fuzzy operations: $ffS_1f(t)\mu(t)\left\{\widetilde{x(t)}\widetilde{q(t)}\right\}$, (or $ffS_3f(t)\mu(t)\left\{\widetilde{x(t)}\widetilde{q(t)}\right\}$), and with different fuzzy operators: $ffS_1f(t)\mu(t)\left\{\widetilde{F(t)}\widetilde{x(t)}\right\}$, (or $ffS_3f(t)\mu(t)\left\{\widetilde{F(t)}\widetilde{x(t)}\right\}$).
7. $ffSprt(t)\begin{matrix}\widetilde{A(t)}\\ \mu(t)\\ \widetilde{B(t)}\end{matrix}\, gives$ the result

   $$ffSrt(t)\begin{matrix}\widetilde{A(t)}\\ \mu(t)\\ \widetilde{B(t)}\end{matrix} = \left\{\widetilde{A(t)}\cup\widetilde{B(t)}-\widetilde{A(t)}\cap\widetilde{B(t)}, D(t)\right\}\ for\ fuzzy\ sets\ \widetilde{A(t)}, \widetilde{B(t)}\ ,$$

   where $\widetilde{D(t)}$ is fself- (fuzzy continual set) for $\widetilde{A(t)}\cap\widetilde{B(t)}$. The same is true for structures if they are treated as fuzzy continual sets.
8. Similarly, for fuzzy dynamic fSprt-derivatives, fuzzy dynamic fSprt-integrals, fuzzy dynamic fSprt-lim, dynamic fself- (fuzzy derivatives), dynamic fself- (fuzzy integrals).
9. Denote dynamic continual fself-(dynamic continual fself-Q(t)) through dynamic continual fself^2-Q(t), fS(t)(n,Q(t))= dynamic continual fself-(dynamic continual fself-(…(dynamic continual fself-Q(t)))) = dynamic continual fself^n-Q(t) for n-multiple dynamic continual fself.

Remark 2.4.1. Fuzzy dynamic continual fSprt-displacement of A(t) from B(t) with measure of fuzziness µ will be denote through $\overset{B(t)}{\underset{A(t)}{\mu(t)}}$ffSprt(t). Then the notation $\overset{C(t)}{\underset{D(t)}{\mu_2(t)}}$ffSprt(t)$\overset{A(t)}{\underset{B(t)}{\mu_1(t)}}$ is fuzzy dynamic continual fSprt- embedding of A(t) in B(t) and fuzzy dynamic continual fSprt-displacement of D(t) from C(t) simultaneously. Denote $\overset{B(t)}{\underset{A(t)}{\mu_2(t)}}$ffSprt(t)$\overset{A(t)}{\underset{B(t)}{\mu_1(t)}}$ by $fTf^{\mu_1(t)}_{\mu_2(t)}S(t)^{A(t)}_{B(t)}$.

We can consider the concept of fuzzy dynamic continual fSprt- element as $ffSprt(t)\overset{\widetilde{A(t)}}{\underset{\widetilde{B(t)}}{\widetilde{\mu(t)}}}$, where $\widetilde{A(t)}$ fits in fuzzy dynamic continual fuzzy capacity $\widetilde{B(t)}$. Then $ffSprt(t)\overset{\widetilde{B(t)}}{\underset{\widetilde{B(t)}}{\widetilde{\mu(t)}}}$ it will mean $ffS_1f(t)\mathfrak{z}(t)B(t)$. $\overset{A(t)}{\underset{A(t)}{\mu(t)}}$ffSprt(t) denotes the fuzzy dynamic continual displacement of fuzzy A(t) from itself with measure of fuzziness µ, $\overset{A(t)}{\underset{A(t)}{\mu_1(t)}}$ffSprt(t)$\overset{A(t)}{\underset{A(t)}{\mu_1(t)}}$—simultaneous fuzzy dynamic continual containment of oneself A(t) in oneself A(t) with measure of fuzziness μ_1 and dynamic continual expelling oneself A(t) out of oneself A(t) with measure of fuzziness μ_1. $\overset{A(t)}{\underset{A(t)}{\mu(t)}}$ffSprt(t) will be called fuzzy dynamic continual anti fuzzy fcapacity from itself with measure of fuzziness µ.

Connection of fuzzy dynamic continual fSprt– elements with target weights with dynamic continual fuzzy containment of oneself with target weights. Consider a third type of dynamic partial fuzzy containment of oneself with target weights $\widetilde{g(t)}$. For example, based on $ffSprt(t)\overset{\widetilde{x(t)}\widetilde{g(t)}}{\underset{\widetilde{w(t)}}{\mu(t)}}$, where $\widetilde{x(t)}$=(x$_1$(t)|$\mu_{\widetilde{x(t)}}$(x$_1$(t)), x$_2$(t)|$\mu_{\widetilde{x(t)}}$(x$_2$(t))|, …, x$_n$(t)|$\mu_{\widetilde{x(t)}}$(x$_n$(t))), i.e. n fuzzy continual elements with target weights $\widetilde{g(t)}$ at one point w, we can consider the dynamic fuzzy containment $fS_3f(t)\widetilde{x(t)\mu(t)}\widetilde{g(t)}$ of oneself with target weights with m fuzzy continual elements with target weights $\widetilde{g(t)}$ from $\widetilde{x(t)}$, m<n, which is the fuzzy process of formation according to form (1.1) or forms (1.1.1) - (1.4), (1.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*) for more complex operators.

Definition 2.4.8. The dynamic fuzzy embedding of fuzzy continual A(t) into itself with target weights $\widetilde{g(t)}$ of the first type is the process of fuzzy embedding A(t) into A(t) with target weights $\widetilde{g(t)}$. Denote $fS_1 f(t)A(t)\mu(t)\widetilde{g(t)}$.

Definition 2.4.9. The dynamic fuzzy containment of fuzzy continual C(t) itself into itself with target weights $\widetilde{g(t)}$ of the second type is the process of fuzzy containment of the fuzzy continual elements from which it can be fuzzy generated. Let's denote $fS_2 f(t)C(t)\mu(t)\widetilde{g(t)}$.

Definition 2.4.10. Partial fuzzy 'dynamic containment of fuzzy continual B(t) itself into itself with target weights $\widetilde{g(t)}$ of the third type is the process of partial fuzzy containment of fuzzy continual B(t) into itself or fuzzy continual elements from which it can be fuzzy generated partially, or both at the same time. Denote $fS_3 f(t)B(t)\mu(t)\widetilde{g(t)}$.

2.5 The usage of ffSprt-elements for networks

A. Galushkin's comprehensive monograph [20] covers all aspects of networks, but traditional approaches go through classical mathematics, mainly through the usual correspondence operators. Here we consider a different approach - through a new mathematical process with fuzzy containment operators, which, although they can be interpreted as the result of some correspondence operators, are not themselves correspondence operators [1]. Containment operators are more convenient for networks. Also, the main emphasis was placed on using processors operating using triodes, which are generally not used in ffSprt -networks. ffSprt-networks(ffSmnsprt) are a ffSprt -structure that can be built for the required weights. ffSprt -OS (ffSprtoperating system) uses ffSprt -coding and ffSprt -translation. In the first one, coding is carried out through a 2-dimensional matrix-row (a, b), where the fuzzy number b is the code of the action, and the fuzzy number a is the code of the object of this action. ffSprt -coding (or fself- (fuzzy coding)) is implemented through a fuzzy matrix consisting of 2 columns (in the continuous case, two intervals of fuzzy numbers). Here, the source encoding is used for all matrix rows simultaneously. ffSprt -translation is carried out by inversion. In this case, fself- (fuzzy coding) and fself- (fuzzy translation) will be more stable. The target weights $q_i(t)$ in $ffSprt(t)\ \begin{matrix}\widetilde{x(t)}\widetilde{q(t)}\\ \mu(t)\\ \widetilde{w(t)}\end{matrix}$ are chosen for necessary tasks. We will not touch on the issues of applications, or network optimization. They are described in detail by Galushkin [20]. We will touch on the difference of this only for hierarchical complex networks. The same simple executing programs are in the cores of simple artificial neurons of type ffSprt(designation - mnffSprt) for simple information processing. More complex executing programs are used for mnffSprt nodes. Threshold element ffSprt - $\begin{matrix}b\\ \mu_2\\ \{qy\}\end{matrix}$ ffSprt(t) $\begin{matrix}\{a\widetilde{x}\}\\ \mu_1\\ b\end{matrix}$, b- artificial neurons of type ffSprt (designation - mnff-

Sprt), $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|, …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))) is the fuzzy set of values of the initial signals, a=(a_1,a_2,…,a_n) are the weights of Sprt-synapses and $\widetilde{y}$=(y_1|$\mu_{\widetilde{y}}$(y_1), y_2|$\mu_{\widetilde{y}}$(y_2)|,…,y_n|$\mu_{\widetilde{y}}$(y_n)|) is the fuzzy set of values of the output signals with weights q=(q_1,q_2,…,q_n).. The first level of mnffSprt consists of simple mnffSprt. The second level of mnffSprt consists of $fSt_D^{\{mnSt\}}$ – Sit-node of mnffSprt in range D, D-fuzzy capacity for mnffSprt node. The third level of mnffSprt consists of ffSprt $\overset{ffSprt}{\underset{\mu_D}{}} \overset{mnffSprt}{\underset{D}{ţ}}$ - ffSprt ^{2}- node of mnffSprt in range D, thus D becomes fuzzy capacity of itself in itself as an element for mnffSprt. For our networks, it is sufficient to use ffSprt ^{2}- nodes of mnffSprt, but fself-level is higher in living organisms, particularly ffSprt ^{n}-, with n $\geqslant$ 3. The target structure or the corresponding program enters the target unit using alternating current. After that, all networks or parts of them are activated according to the indicative goal. It may appear that we are leaving the network ideology, but these networks are a complex hierarchy of different levels, like living organisms.

Remark 2.5. Traditional scientific approaches through classical mathematics make it possible to describe only at the usual energy level. Here we consider an approach that makes describing processes with finer energies possible. mnffSprt contains ffSprt $\overset{\{ffeprograms\}}{\underset{mnffSprt}{\mu}}$, $ffeprogram-$executing program in ffSprt - OS. ffSprt -OS (or Fself-OS) is based on ffSprt -assembly language (or ffself-assembly language), which is based on assembly language through Sprt-approach in turn, if the base of elements of ffSprt -networks is sufficient. The ffeprograms are in fSprt-programming environments (or fself- fuzzy programming environments), but this question and ffSprt -networks base will be considered in the following monographs. In particular, ffeprograms may contain Sprt- programming operators. In mnffSprt cores, the constant memory ffSprt with correspondent ffeprograms depending on mnffSprt.

The OS (operating system) and the principles and modes of operation of the ffSprt-networks for this programming are interesting. But this is already the material for the next publications.

Here is developed a helicopter model without a main and tail rotors based on ffSprt– physics and special neural networks with artificial neurons operating in normal and ffSprt -modes. Let's denote this model through ffSmnsprt. To do this, it’s proposed to use mnffSprt of different levels; for example, for the usual mode, mnffSprt serves for the initial processing of signals and the transfer of information to the second level, etc., to the nodal center, then checked. In case of an anomaly - local ffSprt –mode with the desired "target weight" is realized in this section, etc., to the center. In the case of a monster during the test, ffSmnsprt is activated with the desired "target weight." Here are realized other tasks also. To reach the fself-energy level, the mode ffSprt $\mu_{ffSmnsprt}^{ffSmnsprt}$ is used. In normal mode, it's planned to carry out the movement of ffSmnsprt on jet propulsion by converting the energy of the emitted gases into a vortex to obtain additional thrust upwards. For this purpose, a spiral-shaped chute (with "pockets") is arranged at the bottom of the ffSmnsprt for the gases emitted by the jet engine, which first exit through a straight chute connected to the spiral one. There is drainage of exhaust gases outside the ffSmnsprt. ffSmnsprt is represented by a neural network that extends from the center of one of the main clusters of ffSprt- artificial neurons to the shell, turning into the body itself. Above the operator's cabin is the central core of the neural network and the target block, responsible for performing the "target weights" and auxiliary blocks, the functions and roles of which we will discuss further. Next is the space for the movement of the local vortex. The unit responsible for ffSmnsprt 's actions is below the operator's cab. In ffSprt– mode, the entire network or its sections are fSprt– activated to perform specific tasks, in particular, with "target weights." In the target, block used ffSprt -coding, ffSprt -translation for activation of all networks to "target weights" simultaneously, then –the reset of this Sprt-coding after activation. Unfortunately, triodes are not suitable for Sprt -neural networks. In the most primitive case, usual separators with corresponding resistances and cores for ffeprograms may be used instead triodes since there is no necessity to unbend the alternating current to direct. The ffSprt-operative memory belt is disposed around a central core of ffSmnsprt. There are ffSprt -coding, ffSprt -translation, and ffSprt-realize of feprograms and the programs from the archives without extraction, ffSprt-coding and ffSprt-translation may be used in high-intensity, ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna, Strickland. ffSprt– structure or an feprogram if one is present of needed «target weight» are taken in target block at ffSprt– activation of the networks. ffSprt $\mu_{activation}^{ffSmnsprt,g}$ derives ffSmnsprt to the fself-level boundary with target weight g.

It's used an alternating current of above high frequency and ultra-violet light, which can work with ffSprt– structures in ffSprt –modes by its nature to activate the networks or some of its parts in ffSprt –modes and locally using ffSprt –mode. Above high frequently alternating current go through mercury bearers. That's why overheating does not occur. The power of the alternating current above high frequently increases considerably for the target block. The activation of all networks is realized to indicate "target weights.”

Appendix

We consider ffSprt-logic [1]: consider the functional g(Q), which gives a fuzzy numerical value for the truth of the fuzzy statement Q from the interval [0,1], where 0 corresponds to "no," and one corresponds to the logical value "yes." Then for joint fuzzy statements A, B: g(A+B)=g(A)+g(B)-g(A*B)+gS(D), D- fself- (fuzzy statement) from A*B, gS(x)- the value of fself-truth for fself- (fuzzy statement) x; for dependent fuzzy statements: g(A*B)=g(A)*g(B/A)=g(B)*g(A/B), where g(B/A)- conditional truth of the fuzzy statement B at fuzzy statement A, g(A/B)- dependent truth of fuzzy statement A at the fuzzy statement B. Adding the truth values of inconsistent fuzzy propositions: g(A+B)=g(A)+g(B). The formula of complete truth: g(A)=$\sum_{k=1}^{n} g\,(B_k) * g(A/B_k)$, B_1, B_2, .., B_n-full group of hypotheses- (fuzzy statements): $\sum_{k=1}^{n} f\,(B_k)$=1(“yes”).
Remark. A fuzzy statement can be interpreted as an fuzzy event, and its truth value as a probability.

ffSprt- statement for fuzzy set of statements $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|, …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))):

$$\text{ffSprt}\underset{w}{\overset{(x_1(t)|\mu_{\widetilde{x(t)}}(x_1(t)),\ x_2(t)|\mu_{\widetilde{x(t)}}(x_2(t))|,\ \dots,\ x_n(t)|\mu_{\widetilde{x(t)}}(x_n(t)))}{\mu}},$$

$\text{ffSprt}\underset{w}{\overset{\{\ g(x_1), g(x_2), \dots, g(x_n)\}}{\mu}}$ - ffSprt- truth for these statements. It is possible to consider the fself-statement $ffS_3f\mu\widetilde{x}$ with m statements from $\widetilde{x}$, at m<n, which is formed by the form (1.1), that is, only m statements from $\widetilde{x}$ are located in the structure $ffSprt\underset{w}{\overset{\widetilde{x}}{\mu}}$. The same for fself- truth $ffS_3f\mu\{\ g(x_1), g(x_2), \dots, g(x_n)\}$.

One can introduce the concepts of fSprt-group: ffSrt$\underset{q}{\overset{\widetilde{x}}{\mu}}$, $\widetilde{x}$ is usual fuzzy group, ffSrt$\underset{B}{\overset{\widetilde{x}}{\mu}}$, where $\widetilde{x}$, B- usual fuzzy groups, fself-groups: $ffS_if\mu$, i=1,2,3 [2].

Definition 2.5. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its elements clearly, but not necessarily in known operators; it can form (create) new special operators (in particular, special functions). In particular, ffCSrt μ_{strA}^{strA} is such structure. Similarly, for working with models, each is structured by its structure; for example, use ffSprt-groups, ffSprt -rings, ffSprt -fields, ffSprt -spaces, fself-(fuzzy groups), fself- (fuzzy rings), fself- (fuzzy fields), and fself- (fuzzy spaces). Like any task, this is also a structure of the appropriate fuzzy capacity. Since the degree of freedom is double, it is clear that the form of the fself-equation contains a solution or structures the inversion of the fself- (fuzzy equation) concerning unknowns, i.e., the structure of the fself-equation is complete.

2.6 Variable fuzzy hierarchical dynamic fuzzy structures (models, operators) for dynamic, singular, hierarchical fuzzy sets

In contrast to the classical one-attribute fuzzy set theory [2], [3], where only its contents are taken as a set, we consider a two-attribute fuzzy set theory with a fuzzy set as a fuzzy capacity and separately with its contents. We simply use a convenient form to represent the singularity of a fuzzy set. Articles [4] - [16] use the following methodology for permanent structures:

1. Cancellation of the axiom of regularity.

2. 2 attributes for the fuzzy set: fuzzy capacity and its content.

3. Fuzzy compression of a fuzzy set, for example, to a point.

4. “turning out” from one another, particularly from a fuzzy capacity, we pull out another fuzzy capacity, for example, itself, as its element.

5. The simultaneity of one (fuzzy compression) and the other (“eversion”).

6. Own fuzzy capacities.

7. Qualitatively new fuzzy programming and fuzzy Networks.

Here we will consider variable fuzzy structures (models) [1], both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable fuzzy structures (models), for example,

$$\underset{D}{\overset{C}{\mu_{14}}} ffSprt(t) \underset{B}{\overset{A}{\mu_{13}}} = \begin{cases} (\underset{D}{\overset{C}{\mu_6}} ffSprt, \quad q_2 \ge t \ge q_1)|\mu_1 \\ (\underset{D}{\overset{B}{\mu_8}} ffS^1prt \underset{B}{\overset{A}{\mu_7}}, \quad q_3 \ge t > q_2)|\mu_2 \\ (\underset{D}{\overset{C}{\mu_{10}}} ffSprt \underset{B}{\overset{A}{\mu_9}}, \quad q_4 \ge t > q_3)|\mu_3 \\ (ffSprt \underset{B}{\overset{A}{\mu_{11}}}, \quad q_5 \ge t > q_4)|\mu_4 \\ (\underset{D}{\overset{\{\,\}}{\mu_{12}}} ffSprt, \quad t > q_5)|\mu_5 \\ \dots \end{cases} \qquad (*_{2.1}),$$

μ_i- measures of fuzziness, i = 1, ..., 14. In particular, $\underset{D}{\overset{B}{\mu_{10}}} ffSprt \underset{B}{\overset{A}{\mu_9}}$ can be interpreted as a fuzzy game: player 1 fuzzy with measures of fuzziness μ_9 fits fuzzy A into fuzzy B, and the other fuzzy with measures of fuzziness μ_{10} pushes fuzzy D out of fuzzy B at the same time.

In what follows, we will denote variable fuzzy structure (model) through ffVS(t), self-variable fuzzy structures (models) through SfFVS(t), and oself-variable fuzzy structures (models) through OSffVS(t). Singular fuzzy structures (models) are not confused with fuzzy structures (models) with singularities. $\underset{D}{\overset{B}{\mu_{10}}} ffSprt \underset{B}{\overset{A}{\mu_9}}$ -2-hierarchical fuzzy structure: 1-level - elements A, B, C, D; level 2 - connections between them. 2-

Examples: a) discrete variable fuzzy structure with μ_i- measures of fuzziness, i = 1, ..., 8.

$$\begin{array}{ccc} a|\mu_1 & b|\mu_8 & g|\mu_7 \\ c|\mu_2 & ffVS(t) & w|\mu_6 \\ d|\mu_3 & q|\mu_4 & r|\mu_5 \end{array}$$

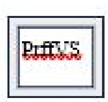


Figure 2: Continuous fuzzy set as the rim.

.

We introduce the notation m_{fVS_N}- the number of elements, N - the number of connections between them in the discrete variable 2-hierarchical fuzzy structure ffVS(t). We introduce the notation $q_{fVS_R}-$ any, R - connections in q_{fVS_R} in the variable 2-hierarchical fuzzy structure ffVS(t), in particular, q_{fVS_R}, R can be fuzzy sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the fuzzy structurability of Q from the interval [0,1], where 0 corresponds to "no fuzzy structure"," and 1 corresponds to the value " fuzzy structure". Then for joint A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D- self-(fuzzy structure) from A*B, cS(x)- the value of self-(fuzzy structure) for self-(fuzzy structure) x; for dependent fuzzy structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional fuzzy structurability of the fuzzy structure B at the fuzzy structure A, c(A/B)- conditional fuzzy structure of the fuzzy structure A at the fuzzy structure B. Adding inconsistent fuzzy structures: c(A+B) =c(A) + c(B). The formula of complete fuzzy structure: c(A)=$\sum_{k=1}^{n} c\,(B_k) * c(A/B_k)$, B_1, $B_{2.}$,.., B_n-full group of fuzzy hypotheses- containments: $\sum_{k=1}^{n} c\,(B_k)$=1("fuzzy structure"). Fuzzy Sprt- structure for fuzzy set of fuzzy structures $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t))|, …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))):

$$ffSprt\begin{array}{c}(x_1(t)|\mu_{\widetilde{x(t)}}(\mathrm{x}_1(\mathrm{t})),\ \mathrm{x}_2(\mathrm{t})|\mu_{\widetilde{x(t)}}(\mathrm{x}_2(\mathrm{t}))|,\ \ldots,\ \mathrm{x}_n(\mathrm{t})|\mu_{\widetilde{x(t)}}(\mathrm{x}_n(\mathrm{t})))\\ \mu\\ w\end{array}$$

$$.ffSprt\begin{array}{c}\{c(x_1)|\mu_{c(\widetilde{x})}c(x_1)|\mu_{c(\widetilde{x})}c(x_2),\ldots,c(x_n)|\mu_{c(\widetilde{x})}c(x_n)\}\\ \mu\\ w\end{array}$$

- fuzzy fSprt-structurability for these fuzzy structures. It is possible to consider the self-(fuzzy structure) $ffS_3f\mu\widetilde{x}$ with m structures and from $\widetilde{x}$, at m<n, which is formed by the form (1.1), that is, only m statements from $\widetilde{x}$ are located in the structure $ffSprt\begin{array}{c}(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n))\\ \mu\\ w\end{array}$ or for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). The same for self-(fuzzy structurability) $f\ fS_3\{\ c(x_1)|\mu_{c(\widetilde{x})}c(x_1), c(x_2)|\mu_{c(\widetilde{x})}c(x_2),\ldots,c(x_n)|\mu_{c(\widetilde{x})}c\,(x_n)\}$.

Can be considered N-hierarchical fuzzy structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical fuzzy structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+1)- level, where A, B, C, … can be any in particular, by fuzzy actions, fuzzy sets, and others.

$$\begin{array}{c}C\\ \mu_{14}\\ D\end{array}ffSprt(t)\begin{array}{c}A\\ \mu_{13}\\ B\end{array}:\left\langle\begin{array}{c}A\to B\\ A,B\end{array}\middle|\begin{array}{c}D\leftarrow C\\ C,D\end{array}\right\rangle\to\begin{pmatrix}fself(A\to B)\mu_{13}\\ A,B\end{pmatrix}$$

$$\underset{D}{\overset{C}{\mu_{14}}} ffSprt(t) \underset{B}{\overset{A}{\mu_{13}}} : \left\langle \begin{matrix} A \to B \\ A, B \end{matrix} \middle| \begin{matrix} D \leftarrow C \\ C, D \end{matrix} \right\rangle \to \left(\begin{matrix} foself(D \leftarrow C)\mu_{14} \\ C, D \end{matrix} \right)$$

Can be considered discrete fuzzy hierarchical fuzzy structure, continuous fuzzy hierarchical fuzzy structure, and discrete-continuous hierarchical fuzzy structure, ffSprt $\underset{x}{\overset{N-hierarchical\ fuzzy\ structure}{\mu}}$.

The example

Let ffSprt $\underset{x}{\overset{i-level\ of\ hierarchical\ structure}{\mu}}$, then ffQHS=

$$\text{ffHSprt} \begin{matrix} (ffSprt \underset{x}{\overset{N-level\ of\ hierarchical\ structure}{\mu}})\mu_N \\ \dots \\ (ffSprt \underset{x}{\overset{i-level\ of\ hierarchical\ structure}{\mu}})\mu_i \\ \dots \\ (ffSprt \underset{x}{\overset{1-level\ of\ hierarchical\ structure}{\mu}})\mu_1 \\ \underset{x}{\mu} \end{matrix} \text{ - fuzzy N-}$$

hierarchical fuzzy structure compression into point x, μ_i- measures of fuzziness, i = 1, ..., N.

Let fg(N, fQHS)= $ffQHS^{\left. ffQHS^{ffQHS^{\cdots ffQHS}} \right\}\text{-N levels}}$

It can be considered self- ffQHS, fg(y, fQHS) for any y, fg(ffQHS, ffQHS).

Compression fuzzy Hierarchy Examples:

$$1)\ ffSprt \underset{\underset{()}{\overset{()}{ffSprt\,\mu}} + B}{\overset{\underset{()}{\overset{()}{ffSprt\,\mu}}}{\mu}} = \begin{pmatrix} ffSprt \underset{\underset{()}{\overset{()}{ffSprt\,\mu}}}{\overset{\underset{()}{\overset{()}{ffSprt\,\mu}}}{\mu}} \\ ffSprt \underset{B}{\overset{\underset{()}{\overset{()}{ffSprt\,\mu}}}{\mu}} \end{pmatrix}$$

$$2)\quad \underset{D+\underset{()}{\overset{()}{\mu_2}}ffS_1prt\underset{()}{\overset{()}{\mu_1}}}{\overset{C+\underset{()}{\overset{()}{\mu_2}}ffS_1prt\underset{()}{\overset{()}{\mu_1}}}{\mu_2}}\ \mathrm{ffS_1prt}\ \underset{B+\underset{()}{\overset{()}{\mu_2}}ffS_1prt\underset{()}{\overset{()}{\mu_1}}}{\overset{A+\underset{()}{\overset{()}{\mu_2}}ffS_1prt\underset{()}{\overset{()}{\mu_1}}}{\mu_1}} =$$

$$(\left(\begin{array}{c} \underset{\underset{()}{\overset{()}{\mu_2}}ffS_1prt\underset{()}{\overset{()}{\mu_1}}}{\overset{\underset{()}{\overset{()}{\mu_2}}ffS_1prt\underset{()}{\overset{()}{\mu_1}}}{\mu_2}}\ ffS_1prt\ \underset{\underset{()}{\overset{()}{\mu_2}}ffS_1prt\underset{()}{\overset{()}{\mu_1}}}{\overset{\underset{()}{\overset{()}{\mu_2}}ffS_1prt\underset{()}{\overset{()}{\mu_1}}}{\mu_1}} \\ \underset{D}{\overset{C}{\mu_2}}ffS_1prt\underset{B}{\overset{A}{\mu_1}} \end{array}\right),$$

Where μ_i- measures of fuzziness, i = 1, 2.

Let's consider two versions: 1) fuzzy containment is interpreted through the concept of fuzzy containment, and 2) fuzzy capacity is interpreted through the concept of fuzzy containment as a rest point of fuzzy containment. Self-(fuzzy containment) is interpreted as a rest point of self-(fuzzy containment). Let A self-(fuzzy compress) into B, D self-(fuzzy displace) from C in $\underset{D}{\overset{C}{\mu_2}}ffVS_1prt\underset{B}{\overset{A}{\mu_1}}$.

We consider the functional ca(Q) [1], which gives a numerical value for the accommodation of fuzzy Q from the interval [0,1], where 0 corresponds to " fuzzy containment" and one corresponds to the value " fuzzy capacity". Then for joint fuzzy A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D-self-(fuzzy containment) for A*B, caS(x)- the value of self-(fuzzy capacity) for self-(fuzzy containment) of x; for dependent fuzzy containments: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional accommodation of the fuzzy containment B at the fuzzy containment A, ca(A/B)- conditional fuzzy capacity of the fuzzy containment A at the fuzzy containment B. Adding the fuzzy capacity values of inconsistent fuzzy containments: ca(A+B)=ca(A)+ca(B). The formula of complete fuzzy capacity: ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2,..,B_n-full group of fuzzy hypotheses- containments: $\sum_{k=1}^{n} ca\,(B_k)$=1("fuzzy capacity"). Sprt-(fuzzy containment) for $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$):

$$\mathrm{ffSprt}\ \underset{w}{\overset{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n))}{\mu}},$$

$\widetilde{x}$ - fuzzy set of fuzzy containments.

$$\{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_2),\dots,ca(x_n)|\mu_{ca(\widetilde{x})}ca(x_n)\}$$

ffSprt μ w - ffSprt-accommodation for these fuzzy containments x_i, i = 1, ..., n. It is possible to consider the self-(fuzzy containment) $ffS_3f\mu\widetilde{x}$ with m fuzzy containments from $\widetilde{x}$, at m<n, which is formed by form (1.1) or forms (1.1.1) - (1.4), (1.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*) for more complex operators. The same for self-(fuzzy accommodation)
($ffS_3f\{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1), ca(x_2)|\mu_{ca(\widetilde{x})}ca(x_2),\dots,ca(x_n)|\mu_{ca(\widetilde{x})}ca\,(x_n)\}$.

Consider a variable fuzzy hierarchy (we will denote it by ffVH) [1].

The example of variable fuzzy hierarchy

$$\begin{matrix} C & & A \\ \mu_2 & \text{ffSprt(t)} & \mu_1 \\ D & & B \end{matrix} = \begin{cases} (\left\{\begin{matrix} Q +^{\{\ \}}_{D-D\cap C} fSt \\ (\ C-D\cap C) \end{matrix}\right\}, \quad q_2 \ge t \ge q_1)|\mu_1 \\ ((\begin{matrix} fS^{1}{}^{e}_{0}fB* \\ {}^{B}_{Q-B}fS^1t^{A-B}_{B} \\ fS^{et}_{01}fB \end{matrix}), q_3 \ge\ t > q_2)|\mu_2 \\ ((\begin{matrix} C-B & & A-B \\ \mu_2 & \text{ffSprt} & \mu_1 \\ C-B & & B \\ C-B & & A-B \\ \mu_2 & \text{ffSprt} & \mu_1 \\ D-C-B & & B \end{matrix}), \quad ,q_4 \ge \quad t > q_3)|\mu_2 \\ (\begin{pmatrix} R \\ A\cup B - A\cap B \end{pmatrix}, q_5 \ge\ t > q_4)|\mu_1 \\ ({}^{\{\ \}}_{D} fSt, \quad t > q_5)|\mu_5 \\ \dots \end{cases}$$

($^*{}_{\text{D.2}}$),

where Q is oself-(fuzzy set) for fuzzy $(D\cap C)$ [10], [15], R is self-(fuzzy set) for fuzzy $A\cap B$ [10] - [16], $fS^{et}_{01}fB$ is considered in [6], [11], μ_i- measures of fuzziness, i = 1, ..., 5 . Variable compression (designation fVS) of fuzzy $\widetilde{A}$ into $\widetilde{x}$(t): $fSt^{\widetilde{A}}_{\widetilde{x}(t)}$, where $\widetilde{x}$(t)- any dynamical fuzzy object at time t.

We consider the functional h(Q), which gives a numerical value for the hierarchization of fuzzy Q from the interval [0,1], where 0 corresponds to "no fuzzy hierarchy," and 1 corresponds to the value " fuzzy hierarchy. " Then for joint fuzzy hierarchies A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- self-(fuzzy hierarchy) from A*B, hS(x)- the value of self-(fuzzy hierarchy) for self-(fuzzy hierarchy) x; for dependent fuzzy hierarchies: h(A*B)=ha(A)*h(B/A)=h(B)*h(A/B), where h(B/A)- conditional hierarchization of the fuzzy hierarchy B at the fuzzy hierarchy A, h(A/B)- conditional fuzzy hierarchy of the fuzzy hierarchy A at the fuzzy hierarchy B. Adding the fuzzy hierarchy values of inconsistent fuzzy hierarchies: h(A+B)=h(A)+h(B). The formula of complete fuzzy hierarchy: h(A)=$\sum_{k=1}^{n} h\left(B_k\right) * h(A/B_k)$, B_1, B_2,..,B_n-full group of fuzzy hypotheses-hierarchies: $\sum_{k=1}^{n} h\left(B_k\right)$=1(“fuzzy hierarchy”).

ffSprt- structure for fuzzy set of hierarches $\widetilde{x}$= $(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2), \ldots, x_n|\mu_{\widetilde{x}}(x_n))$:

$$ffSprt \overset{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2), \ldots, x_n|\mu_{\widetilde{x}}(x_n))}{\underset{B}{\mu}}.$$

ffSprt $\overset{\{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1)|\mu_{h(\widetilde{x})}h(x_2), \ldots, h(x_n)|\mu_{h(\widetilde{x})}h(x_n)\}}{\underset{B}{\mu}}$ - ffSprt-hierarchization for these fuzzy hierarches. It is possible to consider the self-(fuzzy hierarchy) ffS_3A with m fuzzy hierarches from A, at m<n, which is formed by in form (1.1) or forms (1.1.1) - (1.4), (1.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*) for more complex operators.The same for self- hierarchization $fS_3\{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1), h(x_2)|\mu_{h(\widetilde{x})}h(x_2), \ldots, h(x_n)|\mu_{h(\widetilde{x})}h\left(x_n\right)\}$. Can be considered ffSprt $\overset{\{\ ca(x), c(x), h(x)\}}{\underset{B}{\mu}}$. Very interesting next fuzzy hierarchy type: $\overset{fuzzy\ hierarchy\ A}{\underset{fuzzy\ hierarchy\ A}{\mu}}$ ffSprt $\overset{fuzzy\ hierarchy\ A}{\underset{fuzzy\ hierarchy\ A}{\mu}}$. You can enter special operator ffCSprt to work with fuzzy structures:

$\overset{A}{\underset{D}{\mu_2}}$ffCSprt $\overset{Q}{\underset{R}{\mu_1}}$ fuzzy structures R with the fuzzy structure from Q with measure of fuzziness μ_1 and unstructures fuzzy A by the fuzzy structure D with measure of fuzziness μ_2 simultaneously.

Very interesting next fuzzy structure type:

$\overset{A}{\underset{A}{\mu_2}}$ffCSprt $\overset{A}{\underset{A}{\mu_1}}$.

You can enter special operator ffHSprt to work with fuzzy hierarches: $\overset{A}{\underset{D}{\mu_2}}$ffHSprt $\overset{Q}{\underset{R}{\mu_1}}$ fuzzy hierarchizes R with the fuzzy hierarchy from Q with measure of fuzziness μ_1 and unhierarchizes fuzzy A by the fuzzy hierarchy D with measure of fuzziness μ_2 simultaneously.

2.7 FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS^{1}epr, ffSeprt$_1$

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions [1]. This must be done through ffSprt-Networks - fuzzy analogue of Sit-Networks [15] in one of the central departments of which a conventional computer system is located. The parallel processor is itself feprogram - fuzzy analogue of eprogram [15] with direct parallel computing not through serial computing.

Using conventional coding by a computer system, through a Target-block with a fuzzy fSprt -program operator - ffSprt $\overset{Ag}{\underset{activation}{\mu}}$ with measure of fuzziness µ, it will be possible to obtain the fuzzy execution with measure of fuzziness µ of a parallel fuzzy action A with the desired target weight g or the execution of a parallel action A with the desired fuzzy target weight g or both. Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For ffSprt-coding and ffSprt-translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018 year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel fprogram of operator ffSprt $\overset{\{UHF\ AC\ Q\}}{\underset{activation}{\mu}}$ with measure of fuzziness µ, we simultaneously enter the desired set of codes Q using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel fuzzy fprogram operators:

1) fuzzy fSprt-program operators (designation ffSprt-program operators)
2) fuzzy ftprS-program operators (designation fftprS-program operators)
3) fuzzy fS^1epr - program operators (designation ffS^1epr -program operators)
4) fuzzy $Seprt_1$- program operators (designation $ffSeprt_1$-program operators)

One example is pattern recognition:

$$ffSprt^{\,if\ ffSprt\ \overset{image\ archive}{\underset{B}{\mu}}\ ffSprt\overset{q}{\underset{B}{\mu}}\ then\ Name\ of\ q}_{\underset{B}{\mu}}.$$

The example of ffSprt-program is

$$\text{ffSprt}\Big\{ ffSprt\mu_x^{\{\{p\}\{(x)\}\}},\ ffSprt\mu_x^{IF\{\{B\}\{f\}\}\ then\ Q\ \mu_x},\ ffSprt\mu_Q^{Q} \Big\}$$

.

Consider a third type of fuzzy capacity in itself - fuzzy analogue of capacity in itself. For example, based on ffSprt$\mu_q^{\widetilde{x}}$, where $\widetilde{x}=$ ($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$), i.e. n - elements at one point, we can consider the capacity $ffS_3f\widetilde{x}$ (in itself with m elements from $\widetilde{x}$, m<n, which is formed according to the form (1.1),

that is, the structure ffSprt$\mu_q^{\widetilde{x}}$ contains only m elements or for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

Here are some of the fuzzy fSprt-program operators.

1. Simultaneous fuzzy assignment with fuzziness of the expressions $\widetilde{p}$ = ($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2), \ldots,$ $p_n|\mu_{\widetilde{p}}(p_n)$) to the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$). This is implemented via ffSprt $\mu_w^{\{\{\widetilde{x}\}\{p\}\}}$.
2. Simultaneous fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2), \ldots,$ $B_n|\mu_{\widetilde{B}}(B_n)$) Implemented via ffSprt $\mu_w^{IF\left\{\left\{\widetilde{B}\right\}\{\widetilde{g}\}\right\}\ then\ \widetilde{Q}}$ where $\widetilde{Q}$ can be anything.
3. Similarly for fuzzy loop operators and others.

$ffS_3\underset{\sim}{f}-$ fuzzy software operators will differ only just because aggregates $\widetilde{x}, \widetilde{p}$, $\widetilde{B}$,$\widetilde{g}$ will be formed from corresponding ffSprt-program operators in form (1.1), for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

For example, ffSprt $\mu_{g\{R\}}^{\{R\}}$ is the fuzzy capacity with measure of fuzziness µ in itself of the second type if $g\{R\}$ is a ffprogram capable of fuzzy generating $\{R\}$ *with measure of fuzziness* μ.

The example of self-ffprogram of the first type is

$$\mathrm{ffSprt}\begin{smallmatrix}\{\, ffSprt\begin{smallmatrix}\{\{p\}:=\{(x)\}\}\\ \mu\\ w\end{smallmatrix},\ ffSprt\begin{smallmatrix}IF\,\{\{B\}\,\{f\}\}\ then\ Q\\ \mu\\ w\end{smallmatrix},\ ffSprt\begin{smallmatrix}Q\\ \mu\\ Q\end{smallmatrix}\,\}\\ \mu\\ w\end{smallmatrix}.$$

The example of ffSprt-program for ffSmnsprt:

$\mathrm{ffSprt}\begin{smallmatrix}q\\ \mu\\ B\end{smallmatrix}$ - fuzzy assigning fuzzy q to fuzzy B with measure of fuzziness µ.

$\mathrm{ffSprt}\begin{smallmatrix}tw\\ \mu\\ g\end{smallmatrix}$ - fuzzy assigning target weight *tw* to fuzzy g with measure of fuzziness µ.

$\mathrm{ffSprt}\begin{smallmatrix}\{q\}\,w\\ \mu\\ ffSmnsprt\ activation\end{smallmatrix}$ - ffSmnsprt *activation* for fuzzy $\{q\}\,w$ with measure of fuzziness µ.

ffSprt-coding.

ffSprt-coding with measure of fuzziness µ: 1) fuzzy set A to fuzzy set B, 2) fuzzy set A to a point q, where the elements of the fuzzy sets A, B can be continuous. For example, $\mathrm{ffSprt}\begin{smallmatrix}A\\ \mu\\ B\end{smallmatrix}$.

There are ffSprt-coding, ffSprt-translation, ffSprt-realize of ffeprograms and of the ffprograms from the archives without extraction theirs

ffSelf-coding.

ffSelf-coding with measure of fuzziness µ: 1) fuzzy set A to set fuzzy A, i.e. fuzzy A on itself 2) fuzzy set A to a point q in form (1.1), where the elements of the fuzzy sets A, B can be continuous. For example, $\mathrm{ffSprt}\begin{smallmatrix}A\\ \mu\\ A\end{smallmatrix}$.

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with ffSprt-Networks, it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through ffSprt-Networks and series-parallel. Codes from a conventional type computer system will be used via ffSprt -connectors in ffSprt - coding, for example: ffSprt $\underset{activation}{\overset{\{UHF\ AC\ Q\}}{\mu}}$. UHF AC field activation is used.

Dynamic ffSprt and $ffS_3f(t)$ programming [1].

The ideology of dynamic ffSprt and $ffS_3f(t)$ can be used for programming:

4. Simultaneous assignment of the expressions $\widetilde{p(t)}$=($p_1(t)|\mu_{\widetilde{p(t)}}(p_1(t))$, $p_2(t)|\mu_{\widetilde{p(t)}}(p_2(t)), \ldots,$ $p_n(t)|\mu_{\widetilde{p(t)}}(p_n(t))$) to the variables $\widetilde{x(t)}$=($x_1(t)|\mu_{\widetilde{x(t)}}(x_1(t))$, $x_2(t)|\mu_{\widetilde{x(t)}}(x_2(t)), \ldots,$ $x_n(t)|\mu_{\widetilde{x(t)}}(x_n(t))$). This is implemented via ffSprt(t) $\underset{w(t)}{\overset{\left\{\widetilde{x(t)}\right\} := \{p(t)\}}{\mu}}$.
5. Simultaneous checking the fuzzy set of conditions $\widetilde{g(t)}$=($g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t))$, $g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t)), \ldots,$ $g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t))$) for the fuzzy set of expressions $\widetilde{B(t)}$=($B_1(t)|\mu_{\widetilde{B(t)}}(B_1(t))$, $B_2(t)|\mu_{\widetilde{B(t)}}(B_2(t)), \ldots,$ $B_n(t)|\mu_{\widetilde{B(t)}}(B_n(t))$). Implemented via ffSprt(t) $\underset{w(t)}{\overset{IF\left\{\left\{\widetilde{B(t)}\right\}\{\widetilde{g}(t)\}\right\}\ then\ \widetilde{Q(t)}}{\mu}}$ where $\widetilde{Q(t)}$ can be anything.
6. Similarly for fuzzy loop operators and others.

$ffS_3f(t)$– fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}$, $\widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding ffSprt-program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

fftprS-program operators.

The ideology of fftprS and fft_{S_4f}- fuzzy analogues of tS and t_{S_4f} from [10] can be used for programming. Here are some of the ftS-program operators.

1. Simultaneous expelling assignment of the expressions $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2), \ldots,$ $p_n|\mu_{\widetilde{p}}(p_n)$) from the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$). This is implemented via $\underset{=:\,\widetilde{p}}{\overset{\widetilde{x}}{\mu}}$ ffSprt.
2. Simultaneous expelling check the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$,

$B_2|\mu_{\widetilde{B}}(B_2), \ldots, B_n|\mu_{\widetilde{B}}(B_n))$. It's implemented through $\underset{IF\left\{\left\{\widetilde{B}\right\}\{\widetilde{g}\}\right\}\ then\ \widetilde{Q}}{\overset{w}{\mu}}$ ffSprt where Q can be any.

3. Similarly for loop operators and others.

$fft_{S_4f}-$ fuzzy software operators will differ only just because aggregates $\widetilde{x}, \widetilde{p}, \widetilde{B}, \widetilde{g}$ will be formed from corresponding fftprS program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). Consider hierarchical fftprS-program operator

$$\underset{A}{\overset{B}{\mu}} ffSprt = \begin{Bmatrix} D + \underset{\substack{A - A\cap B \\ (B - A\cap B)}}{\overset{\{\ \}}{\mu}} ffSprt \end{Bmatrix},$$ where D is oself-(fuzzy set) for fuzzy $(A\cap B)$.

Dynamic fftprS and $fft(q)_{S_4f}$ programming at time q.

The ideology of fftprS and fft_{S_4f} can be used for dynamic programming. Here are some of the fftprS()- dynamic programming operators.

1. The process of simultaneous expelling assignment of the expressions $\widetilde{p(t)}=(p_1(t)|\mu_{\widetilde{p(t)}}(p_1(t)), p_2(t)|\mu_{\widetilde{p(t)}}(p_2(t)), \ldots, p_n(t)|\mu_{\widetilde{p(t)}}(p_n(t)))$ from the variables $\widetilde{x(t)}=(x_1(t)|\mu_{\widetilde{x(t)}}(x_1(t)), x_2(t)|\mu_{\widetilde{x(t)}}(x_2(t)), \ldots, x_n(t)|\mu_{\widetilde{x(t)}}(x_n(t)))$.is implemented through $\underset{=:\ \widetilde{p}(t)}{\overset{\widetilde{x}(t)}{\mu}}$ ffSprt(t).

2. The process of simultaneous expelling check the fuzzy set of conditions $\widetilde{g(t)}=(g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t)), g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t)), \ldots, g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t)))$ for the fuzzy set of expressions $\widetilde{B(t)}=(B_1(t)|\mu_{\widetilde{B(t)}}(B_1(t)), B_2(t)|\mu_{\widetilde{B(t)}}(B_2(t)), \ldots, B_n(t)|\mu_{\widetilde{B(t)}}(B_n(t)))$ is implemented through $\underset{IF\left\{\left\{\widetilde{B}(t)\right\}\{\widetilde{g}(t)\}\right\}\ then\ \widetilde{Q}(t)}{\overset{w(t)}{\mu}}$ ffSprt(t), where Q(t) can be any. 3.Similarly for loop operators and others.

$fft_{S_4f}-$ fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}, \widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding processes fftprS(t) for above mentioned programming operators through form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). Consider hierarchical dynamic fftprS-program operator:

$$\begin{matrix} B(q) \\ \mu \\ A(q) \end{matrix} ffSprt(q) = \left\{ fft(q)_{S_1 f(A(q)\cap B(q))} + \begin{matrix} \{\ \} \\ \mu \\ A(q) - A(q)\cap B(q) \\ (B(q) - A(q)\cap B(q)) \end{matrix} ffSprt(q) \right\},$$

ffS^{1}epr -program operators (form $\begin{matrix} B \\ \mu_2 \\ D \end{matrix}$ffS^{1}prt$\begin{matrix} A \\ \mu_1 \\ B \end{matrix}$ - fuzzy analogue of ${}^B_D S^1 t^A_B$ [6])

For example, $\begin{matrix} \{(t)\} \\ \mu_2 \\ \{=: \{p(t)\}\} \end{matrix}$ ffS^{1}prt $\begin{matrix} IF\ \{\{B(t)\}\ \{f(t)\}\}\ \ then\ Q(t) \\ \mu_1 \\ \{(t)\} \end{matrix}$.

Consider hierarchical dynamic ffS^{1}epr-program operator: (form $\begin{matrix} \begin{matrix} B \\ \mu_2 \\ A \end{matrix} ffS^1prt \begin{matrix} A \\ \mu_1 \\ B \end{matrix} * \\ \begin{matrix} B \\ \mu_2 \\ D - A \end{matrix} ffS^1prt \begin{matrix} A \\ \mu_1 \\ B \end{matrix} \end{matrix}$).

Consider structure examples hierarchical fuzzy fSprt-program operator

1. ffSprt $\left\{ \begin{matrix} {}^{q}\begin{pmatrix} a & a \\ \mu_1 {}^{ffSrt} \mu_1 \\ a & a \end{pmatrix}_{W_q} & \overset{E_q}{fSprt}{}_{q} \begin{pmatrix} a & a \\ \mu_1 {}^{ffSrt} \mu_1 \\ a & a \end{pmatrix} \begin{matrix} \mu \\ t_0 \end{matrix} & , & fSprt^{\{E^{ex} l^{d_r}\}}_{d_{r(E_{in} l^{d_r})}} \end{matrix} \right\}$

   —fprogram structure example, where the assemblage point d_r is the cursor, it is quite complex self—fprogram.

2. ffSprt $\left\{ \begin{matrix} {}^{q}\begin{pmatrix} a & a \\ \mu_1 {}^{ffSrt} \mu_1 \\ a & a \end{pmatrix}_{W_q} & \overset{E_q}{fSprt}{}_{q} \begin{pmatrix} a & a \\ \mu_1 {}^{ffSrt} \mu_1 \\ a & a \end{pmatrix} \begin{matrix} \mu \\ t_0 \end{matrix} & , & fSprt^{\{E^{ex} l^{d_r}\}}_{d_{r(selfE_{in} l^{d_r})}} \end{matrix} \right\}$

   can be interpreted as a fprogram operator.

ffSprt$_1$- program operators (form $\begin{matrix} C \\ \mu_2 \\ D \end{matrix}ffS_1prt\begin{matrix} A \\ \mu_1 \\ B \end{matrix}$ - fuzzy analogue of ${}^C_D S_1 t^A_B$

[11])

Consider structure example hierarchical fuzzy Set_1-program operator $\begin{matrix} a & & a \\ \mu_1 & \mathrm{ffS_1rt} & \mu_1 \\ a & & a \end{matrix}$ can be interpreted as a $\begin{pmatrix} s_1elf \\ os_1elf \end{pmatrix}$-fprogram operator.

$\begin{matrix} a & & a \\ \mu_1 & \mathrm{ffS_1rt} & \mu_1 \\ a & & a \end{matrix}$-- sample $\begin{pmatrix} s_1elf \\ os_1elf \end{pmatrix}$-fprogram structure example.

Appendix

Remark. Energy of a living organism:

$$\mathrm{ffg}(\mathrm{r}, \mathrm{a}(E_q)) = \mathrm{ffSprt}\left\{ {}^{q}_{W_q}\begin{pmatrix} a & & a \\ \mu_1 & {}^{ffSrt} & \mu_1 \\ a & & a \end{pmatrix} fSprt^{E_q}_{\substack{q \\ \begin{pmatrix} a & & a \\ \mu_1 & {}^{ffSrt} & \mu_1 \\ a & & a \end{pmatrix} \\ \mu \\ t_0}}, \quad fSprt^{\{E^{ex}l^{d_r}\}}_{d_{r}(E_{in}l^{d_r})} \right\}$$

(**$_{2.7}$).

$\begin{matrix} a & & a \\ \mu_1 & \mathrm{ffSrt} & \mu_1 \\ a & & a \end{matrix}$ -internal energy of a living organism, q- a gap in the energy cocoon of a living organism, r-the position of the assemblage point $\mathrm{d_r}$ on the energy cocoon of a living organism, $\mathrm{W_q}$- energy prominences from the gap in the cocoon of a living organism, $\mathrm{E_q}$-external energy entering the gap in the cocoon of a living organism,$E^{ex}l^{d_r}$ - a bundle of fibers of external energy self-capacities from outside the cocoon, collected at the point of assembly of the cocoon of a living organism at time t_0, $E_{in}l^{d_r}$- a bundle of fibers of external energy self-capacities from inside the cocoon, collected at the point of assembly of the cocoon of a living organism in the same position r of the assemblage point $\mathrm{d_r}$. $\mathrm{d_r}$ is the subject of identifying the energy fibers of the subtle energy of the Universe in position r both outside and inside the cocoon.

Energy with measure of fuzziness μ_1, μ_2 of a living organism of a person:

$$ffpg(r, a(E_q)) = ffSprt \left\{ \begin{array}{cc} fSprt_{\substack{q\\ \left(\begin{matrix} a & a \\ \mu_1{}^{ffSrt}\mu_1 \\ a & a \end{matrix}\right)_{W_q}}}^{E_q} \\ \left(\begin{matrix} a & a \\ \mu_1{}^{ffSrt}\mu_1 \\ a & a \end{matrix}\right) \\ \mu \\ t_0 \end{array}, \quad fSprt^{\{E^{ex} l^{d_r}\}}_{d_{r(self(E_{in} l^{d_r}))}} \right\}$$

($**^*_{2.7}$).

($**_{2.7}$), ($***_{2.7}$) can be interpreted as PrfSrt- program operators.

Entire neural network as instantaneous simultaneous ffRAM in ffSprt-elements and fself- elements. $fself^{fself^{\cdots^{fself}}}$, $ff1\downarrow I \uparrow^1_{-1} ff_2{}^{ff1\downarrow I\uparrow^1_{-1}ff_2{}^{\cdots^{ff1\downarrow I\uparrow^1_{-1}ff_2}}}$, $fsin\infty^{fsin\infty^{\cdots^{fsin\infty}}}$. When activated in a neural network, the entire neural network becomes a working memory. Use of fself-energy as fuzzy activation or from outside. ffQ$_0$= ffSprt $\frac{ffSprt \frac{ffS_{mnSprt}}{\mu} \over activation}{\mu} \over {ffSprt \frac{ffS_{mnSprt}}{\mu} \over activation}$ $\rightarrow$ self-ffRAM,

ffQ$_{00}$= $\frac{\frac{ffS_{mnSprt}}{\mu} ffSprt \over activation}{\mu} \over {\frac{ffS_{mnSprt}}{\mu} ffSprt \over activation}$ ffQ_0prt, ffQ$_{01}$= $\frac{ffSprt \frac{ffS_{mnSprt}}{\mu} \over activation}{ffSprt \frac{ffS_{mnSprt}}{\mu} \over activation}$ ffQ_0.ffQ$_0$,ffQ$_{00}$,ffQ$_{01}$-coding,translation,realization in ffeprograms, use ffQ$_0$,ffQ$_{00}$,ffQ$_{01}$-ffS$_{\text{mnSprt}}$,ffAssembler.

REFERENCES:

1) Oleksandr Danilishyn and Illia Danilishyn. Fuzzy Dynamic Fuzzy Sets. Variable Fuzzy Hierarchical Dynamic Fuzzy Structures (Models, Operators) for Dynamic, Singular, Hierarchical Fuzzy Sets. FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS1epr, ffSeprt1. Journal of Mathematical Techniques Computational Mathematics (JMTCM), Volume 3 | Issue 4 |2024, Apr 17, pp. 1 – 37. *Journal DOI: 10.33140/JMTCM*

https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

2) George J. Klir, Bo Yuan. Fuzzy sets and fuzzy logic: theory and applications / Prentice Hall P T R Upper Saddle River, New Jersey 07458. 1995

3) Guanrong Chen, Trung Tat Pham. Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems / CRC Press LLC Boca Raton London New York Washington, D.C. 2001

4) Oleksandr Danilishyn, Illia Danilishyn. Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), 2023, 479-498. https://www.opastpublishers.com/open-access-articles/dynamical-sets-theory-s2telements-and-their-applications.pdf

5) Danilishyn I., Danilishyn O. Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care 2023, Volume 4 | Issue 2 | 66- 81. https://doi.org/10 .33140/NABCC.04.02.02

6) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-_ article1695707465.pdf (mkscience-set.com)

7) I. Danilishyn, O. Danilishyn Dynamic Sets Se, Networks Se. *Advances in Neurology and Neuroscience*, 2023, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

8) I. Danilishyn, O. Danilishyn Program Operators Sit, tS, S1e, Set1. Journal of Sensor Networks and Data Communications [ISSN: 2994-6433] 2023, Volume 3 | Issue 1 | 138-143, https://www.opastpublish ers.com/table-contents/jsndc-volume-3-issue-1-year-2023

9) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

10) Danilishyn .V. Danilishyn O.V. tS – ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

11) Danilishyn .V. Danilishyn O.V. SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June, 2023 with the proceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

12) Danilishyn .V. Danilishyn O.V. DYNAMICAL SIT-ELEMENTS. IV International Scientific and Practical Conference "GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ", 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

13) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT-ELEMENTS TO SETS THEORY AND OTHERS. Scientific practice: modern and classical research methods: Collection of scientific papers «O» with Proceedings of the IV International Scientific and Practical Conference, Boston, May 26, 2023. Boston-Vinnytsia: Primedia eLaunch & European Scientific Platform, 2023, pp.166-171. **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/11**

14) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO CONTINUAL VALUED LOGIC AND OTHERS. Features of the development of modern science in the pandemic's era: a collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, May 19, 2023. Berlin, Federal Republic of Germany: European Scientific Platform, pp.79-84. **https://previous.scientia.report/index.php/archive/issue/view/19.05.2023**

15) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square. https://doi.org/10.21203/rs.3.rs-3217178/v1 Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

16) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol.2, issue 3

17) Thomas J.Jech. LECTURES IN SET THEORY WITH PARTICULAR EMPHASIS ON THE METHOD OF FORCING. SPRINGER-VERLAG Berlin. Heidelberg. New-York, 1971.

18) Krain S.G. Linear differential equations in Banach space. M.,Science,1967(in Russian).

19) A.L.Dontchev .Perturbations, approximations and sensitivity analysis of optimal control systems. Sptinger-Verllag, Berlin, New York,1983.

20) Galushkin A. Networks: principles of the theory. Hot line-Telecom. M.,2010 (in Russian).

21) D.I.Blokhintsev. Quantum Mechanics Lectures on Selected Topics. Moscow, Atomizdat,1981

Declarations:

Availability of data and material

1) Oleksandr Danilishyn and Illia Danilishyn. Fuzzy Dynamic Fuzzy Sets. Variable Fuzzy Hierarchical Dynamic Fuzzy Structures (Models, Operators) for Dynamic, Singular, Hierarchical Fuzzy Sets. FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS1epr, ffSeprt1. Journal of Mathematical Techniques Computational Mathematics (JMTCM), Volume 3 | Issue 4 |2024, Apr 17, pp. 1 – 37. *Journal DOI: 10.33140/JMTCM* https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

2) Oleksandr Danilishyn, Illia Danilishyn. Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), 2023, 479-498. https://www.opastpublishers.com/open-access-articles/dynamical-sets-theory-s2telements-and-their-applications.pdf

3) Danilishyn I., Danilishyn O. Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care 2023, Volume 4 | Issue 2 | 66- 81. https://doi.org/10 .33140/NABCC.04.02.02

4) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-_ article1695707465.pdf (mkscience-set.com)

5) I. Danilishyn, O. Danilishyn Dynamic Sets Se, Networks Se. *Advances in Neurology and Neuroscience*, 2023, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

6) I. Danilishyn, O. Danilishyn Program Operators Sit, tS, S1e, Set1. Journal of Sensor Networks and Data Communications [ISSN: 2994-6433] 2023, Volume 3 | Issue 1 | 138-143, https://www.opastpublishers.com/table-contents/jsndc-volume-3-issue-1-year-2023

7) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

8) Danilishyn .V. Danilishyn O.V. tS -ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

9) Danilishyn .V. Danilishyn O.V. SET1 - ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June, 2023 with the proceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

10) Danilishyn .V. Danilishyn O.V. DYNAMICAL SIT-ELEMENTS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

11) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT-ELEMENTS TO SETS THEORY AND OTHERS. Scientific practice: modern and classical research methods: Collection of scientific papers «O» with Proceedings of the IV International Scientific and Practical Conference, Boston, May 26, 2023. Boston-Vinnytsia: Primedia eLaunch & European Scientific Platform, 2023, pp.166-171. **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/11**

12) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO CONTINUAL VALUED LOGIC AND OTHERS. Features of the development of modern science in the pandemic’s era: a collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, May 19, 2023. Berlin, Federal

Republic of Germany: European Scientific Platform, pp.79-84. **https://previous.scientia.report/index.php/archive/issue/view/19.05.2023**

13) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square. https://doi.org/10.21203/rs.3.rs-3217178/v1 Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

14) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3

Competing interest

There are no competing interests. All sections of the monograph are executed jointly.

Funding

There were no sources of funding for writing the monograph.

Authors' contributions

The contribution of the authors is the same, we will not separate.

Part III. Classes of other structural elements

A Dprt – elements, self-type Dprt-structures and their applications

A.1 Dprt – elements, self-type Dprt-structures

In the articles [1-3], [8 -13] new mathematical structures and operators are constructed through one action - "containment". Here, the construction of new mathematical structures and operators is carried out with generalization to any actions. We consider dynamic operator

$$(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Dprt}\ \underset{B}{\overset{A}{action\ Q}}\ \text{(A.1.1)},$$

where A acts Q to B, D acts Q out from C, Q is any *action* simultaneously. The result of this process will be described by the expression

$$(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Drt}\ \underset{B}{\overset{A}{action\ Q}}\ \text{(A.1.2)}.$$

We consider the measure: $\mu**((\underset{D}{\overset{B}{action\ Q}})^{-1}\text{Dprt}\ \underset{B}{\overset{A}{action\ Q}})=\frac{\mu(A)}{\mu(D)}$, where $\mu(A), \mu(D)$ –usual measures of A, D.

Definition A.1.1. The dynamic operator (A.1.1) we shall call Dprt – element of the first type, (A.1.2) we shall call Drt – element of the first type.

Remark A.1.1. $(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Dprt}\ \underset{B}{\overset{A}{action\ Q}}$ - the analogue of ${}^{C}_{D}Sprt^{A}_{B}$ [1-3] as a special case of (A.1.1), where *action Q* is "contain".

Remark A.1.1.1 an consider Dprt – elements use the Banach space.

It's allowed to add Dprt – elements:

$$(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Dprt}\ \underset{B}{\overset{A_1}{action\ Q}}+(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Dprt}\ \underset{B}{\overset{A_2}{action\ Q}}=(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Dprt}\ \underset{B}{\overset{A_1\cup A_2}{action\ Q}}\ \text{(A.1.1.2.1)},$$

$$(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Dprt}\ \underset{B_1}{\overset{A}{action\ Q}}+(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Dprt}\ \underset{B_2}{\overset{A}{action\ Q}}=(\underset{D}{\overset{C}{action\ Q}})^{-1}\text{Dprt}$$

$$\underset{B_1 \cup B_2}{\overset{A}{action\ Q}} \text{ (A.1.1.2.2),}$$

$$\underset{D}{\overset{C_1}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{B}{\overset{A}{action\ Q}} + \underset{D}{\overset{C_2}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{B}{\overset{A}{action\ Q}} = \underset{D}{\overset{C_1 \cup C_2}{(action\ Q)^{-1}}}\mathrm{Dprt}$$

$$\underset{B}{\overset{A}{action\ Q}} \text{ (A.1.1.2.3),}$$

$$\underset{D_1}{\overset{C}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{B}{\overset{A}{action\ Q}} + \underset{D_2}{\overset{C}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{B}{\overset{A}{action\ Q}} = \underset{D_1 \cup D_2}{\overset{C}{(action\ Q)^{-1}}}\mathrm{Dprt}$$

$$\underset{B}{\overset{A}{action\ Q}} \text{ (A.1.1.2.4).}$$

We consider the following self-type Dprt-structures of the first type:

$$\underset{action\ Q}{\overset{action\ Q}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{action\ Q}{\overset{action\ Q}{action\ Q}} \text{ (A.1.3),}$$

denote D_1fQ.

$$\underset{A}{\overset{action\ Q}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{action\ Q}{\overset{A}{action\ Q}} \text{ (A.1.4),}$$

denote $D_2fA;Q$.

$$\underset{A}{\overset{B}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{B}{\overset{A}{action\ Q}} \text{ (A.1.5),}$$

denote $D_3fA;Q;B$.

$$\underset{A}{\overset{A}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{A}{\overset{A}{action\ Q}} \text{ (A.1.6),}$$

denote $D_4fA;Q$.

$$\underset{strA}{\overset{a}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{a}{\overset{strA}{action\ Q}} \text{ (A.1.6.1),}$$

denote $D_5fA;Q;a$, $a \subset A$ and structure of A acts Q to a and acts Q out from a simultaneously,

$$\left(\underset{a}{\overset{StrA}{action\ Q}}\right)^{-1}\mathrm{Dprt}\ \underset{StrA}{\overset{a}{action\ Q}}\ (A.1.6.2),$$

denote $D_6 fa; Q; A,\ a \subset A$ and acts Q to structure of A and acts Q out from structure of A simultaneously,

$$\left(\underset{B}{\overset{B}{action\ Q}}\right)^{-1}\mathrm{Dprt}\ \underset{B}{\overset{A}{action\ Q}}\ (A.1.7),$$

and any other possible options of self for (A.1.1) etc.

It can be considered a simpler version of the dynamic operator

$$\mathrm{Dprt}\ \underset{B}{\overset{A}{action\ Q}}\ (A.1.8)$$

where A acts Q to B, Q is any *action*, the result of this process will be described by the expression

$$\mathrm{Drt}\ \underset{B}{\overset{A}{action\ Q}}\ (A.1.9)$$

or

$$\left(\underset{D}{\overset{C}{action\ Q}}\right)^{-1}\mathrm{Dprt}\ (A.1.10)$$

where D acts Q out from C, Q is any *action*, the result of this process will be described by the expression

$$\left(\underset{D}{\overset{C}{action\ Q}}\right)^{-1}\mathrm{Drt}\ (A.1.11)$$

Definition 1,2. The dynamic operator (A.1.8) we shall call Dprt – element of the second type, (A.1.9) we shall call Drt – element of the second type.

Remark A.1.2. Dprt $\underset{B}{\overset{A}{action\ Q}}$ - the analogue of $Sprt_B^A$ [1-3] as a special case of (A.1.8), where *action Q* is "contain". In this case

$Sprt_{action\ Q}^{action\ Q} =$ Dprt $\underset{action\ Q}{\overset{action\ Q}{action\ Q}}$ – self-containment and unlike usual self has higher level self(contain): sel$f^{\frac{3}{2}}$. That's why self-containment can generate, modify and perform other actions with self-capacities, because they have lower level = self.

It's allowed to add Dprt – elements of the second type:

$$\mathrm{Dprt}\ \underset{B}{\overset{A_1}{action\ Q}} + \mathrm{Dprt}\ \underset{B}{\overset{A_2}{action\ Q}} = \mathrm{Dprt}\ \underset{B}{\overset{A_1 \cup A_2}{action\ Q}} \text{ (A.1.12)},$$

$$\mathrm{Dprt}\ \underset{B_1}{\overset{A}{action\ Q}} + \mathrm{Dprt}\ \underset{B_2}{\overset{A}{action\ Q}} = \mathrm{Dprt}\ \underset{B_1 \cup B_2}{\overset{A}{action\ Q}} \text{ (A.1.13)},$$

We consider the following self-type Dprt-structures of the second type:

$$\mathrm{Dprt}\ \underset{A}{\overset{A}{action\ Q}} \text{ (A.1.14)},$$

$$\mathrm{Dprt}\ \underset{a}{\overset{strA}{action\ Q}} \text{ (A.1.14.1)},$$

denote $D_7 fA; Q; a$, $a \subset A$ and structure of A acts Q to a,

$$\mathrm{Dprt}\ \underset{strA}{\overset{a}{action\ Q}} \text{ (A.1.14.2)},$$

denote $D_8 fa; Q; A$, $a \subset A$ and acts Q to structure of A,

$$\mathrm{Dprt}\ \underset{action\ Q}{\overset{action\ Q}{action\ Q}} \text{ (A.1.15)},$$

$$\mathrm{Dprt}\ \underset{action\ Q}{\overset{A}{action\ Q}} \text{ (A.1.16)},$$

and any other possible options of self for (A.1.8) etc.

Definition A.1.3. The dynamic operator (A.1.10) we shall call tprD – element, (A.1.11) we shall call trD – element.

Remark A.1.3. $(\underset{D}{\overset{C}{action\ Q}})^{-1}$Dprt - the analogue of ${}^C_D Sprt$ [1-3] as a special case of (A.1.10), where *action Q* is "contain".

It's allowed to add tprD – elements:

$$(\underset{D}{\overset{C_1}{action\ Q}})^{-1}\mathrm{Dprt} + (\underset{D}{\overset{C_2}{action\ Q}})^{-1}\mathrm{Dprt} = (\underset{D}{\overset{C_1 \cup C_2}{action\ Q}})^{-1}\mathrm{Dprt} \text{ (A.1.17)},$$

$$\underset{D_1}{\overset{C}{(action\ Q)^{-1}}}\mathrm{Dprt} + \underset{D_2}{\overset{C}{(action\ Q)^{-1}}}\mathrm{Dprt} = \underset{D_1 \cup D_2}{\overset{C}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \text{(A.1.18)}.$$

We consider the following self-type tprD-structures:

$$\underset{D}{\overset{D}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \text{(A.1.19)}$$

$$\underset{d}{\overset{strD}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \text{(A.1.19.1)},$$

denote $D_9 fd; Q; D$, $d \subset D$ and d acts Q out from structure of D,

$$\underset{strD}{\overset{d}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \text{(A.1.19.2)},$$

denote $D_{10} fD; Q; d$, $d \subset D$ and structure of D acts Q out from d,

$$\underset{D}{\overset{action\ Q}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \text{(A.1.20)}$$

$$\underset{action\ Q}{\overset{action\ Q}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \text{(A.1.21)}$$

and any other possible options of self for (A.1.10) etc.

Dynamic Dprt – elements, self-type dynamic Dprt-structures.

We considered Dprt – elements earlier. Here we consider dynamic Dprt – elements. We consider dynamic operator whose elements change over time

$$\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(A.1.2.1)},$$

where A(t) acts Q(t) to B(t), D(t) acts Q(t) out from C(t), Q(t) is any *action* simultaneously. The result of this process will be described by the expression

$$\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Drt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(A.1.2.2)}.$$

Definition A.1.2.1. The dynamic operator (A.1.2.1) we shall call dynamic Dprt – element of the first type, (A.1.2.2) we shall call dynamic Drt – element of the first type.

Remark A.1.2.1. $\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}$ - the analogue of ${}^{C(t)}_{D(t)}Sprt(t)^{A(t)}_{B(t)}$ [1-3] as a special case of (A.1.2.1), where *action* $Q(t)$ is "contain".

It's allowed to add dynamic Dprt – elements:

$$\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A_1(t)}{action\ Q(t)}}+\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A_2(t)}{action\ Q(t)}}=\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A_1(t)\cup A_2(t)}{action\ Q(t)}}\quad (\mathrm{A.1.2.2.1}),$$

$$\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B_1(t)}{\overset{A(t)}{action\ Q(t)}}+\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B_2(t)}{\overset{A(t)}{action\ Q(t)}}=\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B_1(t)\cup B_2(t)}{\overset{A(t)}{action\ Q(t)}}\quad (\mathrm{A.1.2.2.2}),$$

$$\underset{D(t)}{\overset{C_1(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}+\underset{D(t)}{\overset{C_2(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}=\underset{D(t)}{\overset{C_1(t)\cup C_2(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\quad (\mathrm{A.1.2.2.3}),$$

$$\underset{D_1(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}+\underset{D_2(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}=\underset{D_1(t)\cup D_2(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\quad (\mathrm{A.1.2.2.4}).$$

We consider the following self-type dynamic Dprt-structures of the first type:

$$\underset{action\ Q(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{action\ Q(t)}{\overset{action\ Q(t)}{action\ Q(t)}}\quad (\mathrm{A.1.2.3}),$$

$$\underset{A(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \underset{action\ Q(t)}{\overset{A(t)}{action\ Q(t)}}\quad (\mathrm{A.1.2.4}),$$

$$\left(\overset{B(t)}{\underset{A(t)}{\mathit{action}\ Q(t)}}\right)^{-1}\mathrm{Drt(t)}\ \overset{A(t)}{\underset{B(t)}{\mathit{action}\ Q(t)}}\ \text{(A.1.2.5),}$$

$$\left(\overset{A(t)}{\underset{A(t)}{\mathit{action}\ Q(t)}}\right)^{-1}\mathrm{Drt(t)}\ \overset{A(t)}{\underset{A(t)}{\mathit{action}\ Q(t)}}\ \text{(A.1.2.6),}$$

$$\left(\overset{a(t)}{\underset{strA(t)}{\mathit{action}\ Q(t)}}\right)^{-1}\mathrm{Drt(t)}\ \overset{strA(t)}{\underset{a(t)}{\mathit{action}\ Q(t)}}\ \text{(A.1.2.6.1),}$$

denote $D_{11}(t)fA(t);Q(t);a(t)$, $a(t) \subset A(t)$ and structure of A(t) acts Q(t) to $a(t)$ and acts Q(t) out from $a(t)$ simultaneously,

$$\left(\overset{strA(t)}{\underset{a(t)}{\mathit{action}\ Q(t)}}\right)^{-1}\mathrm{Drt(t)}\ \overset{a(t)}{\underset{strA(t)}{\mathit{action}\ Q(t)}}\ \text{(A.1.2.6.2),}$$

denote $D_{12}(t)fa(t);Q(t);A(t),\ a(t) \subset A(t)$ and acts Q(t) to structure of A(t) and acts Q(t) out from structure of A(t) simultaneously,

$$\left(\overset{B(t)}{\underset{B(t)}{\mathit{action}\ Q(t)}}\right)^{-1}\mathrm{Drt(t)}\ \overset{A(t)}{\underset{B(t)}{\mathit{action}\ Q(t)}}\ \text{(A.1.2.7),}$$

and any other possible options of self for (A.1.2.1) etc.

It can be considered a simpler version of the dynamic operator

$$\mathrm{Dprt(t)}\ \overset{A(t)}{\underset{B(t)}{\mathit{action}\ Q(t)}}\ \text{, (A.1.2.8)}$$

where A(t) acts Q(t) to B(t), the result of this process will be described by the expression

$$\mathrm{Drt(t)}\ \overset{A(t)}{\underset{B(t)}{\mathit{action}\ Q(t)}}\ \text{(A.1.2.9),}$$

or

$$\left(\overset{C(t)}{\underset{D(t)}{\mathit{action}\ Q(t)}}\right)^{-1}\mathrm{Dprt(t)}\ \text{(A.1.2.10),}$$

where D(t) acts Q(t) out from C(t), Q(t) is any *action*, the result of this process will be described by the expression

$$\left(\overset{C(t)}{\underset{D(t)}{\mathit{action}\ Q(t)}}\right)^{-1}\mathrm{Drt(t)}\ \text{(A.1.2.11),}$$

Definition A.1.2.2. The dynamic operator (A.1.2.8) we shall call dynamic Dprt – element of the second type, (A.1.2.9) we shall call dynamic Drt – element of the second type.

Remark A.1.2.2. Dprt(t) $\underset{B(t)}{\overset{A(t)}{action\ Q(t)}}$ - the analogue of $Sprt(t)_{B(t)}^{A(t)}$ [1-3] as a special case of (A.1.2.8), where $action\ Q(t)$ is “contain”. In this case

$Sprt(t)_{action\ Q(t)}^{action\ Q(t)}$ = Dprt(t) $\underset{action\ Q(t)}{\overset{action\ Q(t)}{action\ Q(t)}}$ – self-containment and unlike usual self has higher level self(contain) $self^{\frac{3}{2}}$. That's why self-containment can generate, modify and perform other actions with self-capacities, because they have lower level = self.

It’s allowed to add dynamic Dprt – elements of the second type:

$$\text{Dprt(t)}\ \underset{B(t)}{\overset{A_1(t)}{action\ Q(t)}} + \text{Dprt(t)}\ \underset{B(t)}{\overset{A_2(t)}{action\ Q(t)}} = \text{Dprt(t)}\ \underset{B(t)}{\overset{A_1(t)\cup A_2(t)}{action\ Q(t)}} \quad \text{(A.1.2.12)},$$

$$\text{Dprt(t)}\ \underset{B_1(t)}{\overset{A(t)}{action\ Q(t)}} + \text{Dprt(t)}\ \underset{B_2(t)}{\overset{A(t)}{action\ Q(t)}} = \text{Dprt(t)}\ \underset{B_1(t)\cup B_2(t)}{\overset{A(t)}{action\ Q(t)}} \quad \text{(A.1.2.13)},$$

We consider the following self-type dynamic Dprt-structures of the second t type:

$$\text{Dprt(t)}\ \underset{A(t)}{\overset{A(t)}{action\ Q(t)}} \quad \text{(A.1.2.14)},$$

$$\text{Dprt(t)}\ \underset{a(t)}{\overset{strA(t)}{action\ Q(t)}} \quad \text{(A.1.2.14.1)},$$

denote $D_{13}(t)fA(t);Q(t);a(t)$, $a(t) \subset A(t)$ and structure of A(t) acts Q(t) to $a(t)$,

$$\text{Dprt(t)}\ \underset{strA(t)}{\overset{a(t)}{action\ Q(t)}} \quad \text{(A.1.2.14.2)},$$

denote $D_{14}(t)fa(t);Q(t);A(t)$, $a(t) \subset A(t)$ and acts Q(t) to structure of A(t),

$$\text{Dprt(t)}\ \underset{action\ Q(t)}{\overset{action\ Q(t)}{action\ Q(t)}} \quad \text{(A.1.2.15)},$$

$$\text{Dprt(t)}\ \underset{action\ Q(t)}{\overset{A(t)}{action\ Q(t)}} \quad \text{(A.1.2.16)},$$

and any other possible options of self for (A.1.2.8) etc.

Definition A.1.2.3. The dynamic operator (A.1.2.10) we shall call dynamic tprD – element, (A.1.2.11) we shall call dynamic trD – element.

Remark A.1.2.3. $\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}$ - the analogue of ${}^{C(t)}_{D(t)}Sprt(t)$ [1-3] as a special case of (A.1.2.10), where *action* $Q(t)$ is "contain".

It's allowed to add dynamic tprD – elements:

$$\underset{D(t)}{\overset{C_1(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)} + \underset{D(t)}{\overset{C_2(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)} = \underset{D(t)}{\overset{C_1(t)\cup C_2(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}$$

(A.1.2.17),

$$\underset{D_1(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)} + \underset{D_2(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)} = \underset{D_1(t)\cup D_2(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}$$

(A.1.2.18).

We consider the following self-type dynamic tprD-structures:

$$\underset{D(t)}{\overset{D(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \text{(A.1.2.19)}$$

$$\underset{d(t)}{\overset{strD(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \text{(A.1.2.19.1)},$$

denote $D_{15}(t) f d(t); Q(t); D(t)$, $d(t) \subset D(t)$ and d(t) acts Q(t) out from structure of D(t),

$$\underset{strD(t)}{\overset{d(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \text{(A.1.2.19.2)}$$

denote $D_{16}(t) f D(t); Q(t); d(t)$, $d(t) \subset D(t)$ and structure of D(t) acts Q(t) out from d(t),

$$\underset{D(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \text{(A.1.2.20)}$$

$$\underset{action\ Q(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\mathrm{Dprt(t)}\ \text{(A.1.2.21)}$$

and any other possible options of self for (A.1.2.10) etc.

New mathematical structures and operators is carried out with generalization it to any structures with any actions. For example,

$$1)\ \begin{matrix} f_{11} & \dots & f_{1k} \\ \dots & \dots & \dots \\ (q_{j1})^{-1} & \dots & (q_{jk})^{-1} \\ \dots & \dots & \dots \\ f_{l1} & \dots & f_{lk} \end{matrix} \text{DDprt} \begin{matrix} q_{11} & \dots & q_{1n} \\ & \dots & \\ q_{m1} & \dots & q_{mn} \end{matrix} \quad (*_{\text{A.1}}),$$

$f_{ij},\ q_{ij}$ – any objects, actions etc.

$$2)\ \begin{matrix} & g_{12} & \\ g_{11} & g_{22} & g_{13} \\ (w_{j1})^{-1} & (w_{j2})^{-1} & (w_{j3})^{-1} \\ g_{31} & \dots & \\ & g_{k2} & \end{matrix} \text{ DGprt } \begin{matrix} w_{11} & w_{12} & & w_{1n} & \\ \dots & \dots & & w_{2n} & \\ & & & \dots & \\ w_{m1} & w_{m2} & \dots & \dots & w_{ml} \\ & & & w_{sn} & \end{matrix} \quad (*_{\text{A.1.1}}),$$

$w_{ij},\ g_{ij}$ – any objects, actions etc.

3)

$$\begin{matrix} a & b & g \\ c & ASrq(\text{ţ}) & w \\ d & q & r \end{matrix} \ (*_{\text{A.1.2}}),$$

where $ASrq$ is virtual structure or virtual operator, which can take any form of action; a, c, d, q, r, w, g, b, ţ – any objects, actions etc.

Accordingly, we can consider all sorts of self-structures for 1) – 3). And any other possible structures and operators etc.

Appendix

Let us introduce the following notations: A*B= Sprt_B^A, A^2 = Self A =Srt_A^A, $A^{\frac{3}{2}}$ = Drt $\begin{matrix} A \\ A \\ A \end{matrix}$ = sel$f^{\frac{3}{2}}$ (A), A^3 = Self^{2}A, …,$A^{\frac{3n}{2}}$ = Drt $\begin{matrix} A^n \\ A^n \\ A^n \end{matrix}$ = sel$f^{\frac{3n}{2}}$ (A), A^{n+1} = Self^{n}A, sel$f^{min(n,m)}$(A) $Srt_{A^m}^{A^n}$ = sel$f^{\frac{n}{m}}$ (A), sel$f^{min(n,m,k)}$(A) Drt $\begin{matrix} A^n \\ A^m \\ A^k \end{matrix}$ = sel$f^{\frac{n+m+k}{2k}}$ (A), … etc.

There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \dots$, etc.

You can consider a more “hard” option: A*B=$PSprt_B^A$, where $PSprt_B^A$ – operator, containing A in every element of B, A^2 = PSelf A =$PSrt_A^A$, A^3 = PSelf^{2}A, …, A^{n+1} = PSelf^{n}A, PSel$f^{min(n,m)}$(A) $PSrt_{A^m}^{A^n}$ = PSel$f^{\frac{n}{m}}$ (A), …etc. There is no commutativity here: A*B B*A. We can consider operator functions:

$e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

Let's introduce $\sqrt{self}$ as the result of the decision of the equation Srt_x^x = self,

x

that is x = $\sqrt{self}$, $\sqrt[3]{self}$ as the result of the decision of the equation Dprt x =

x

self, that is x = $\sqrt[3]{self}$, $\sqrt[n]{self^m}$ as the result of the decision of the equation $x^{\frac{n}{m}}$ = self, $self^\alpha$ as the result of the decision of the equation $x^{\frac{1}{\alpha}}$ = self, where is any number, in particular, a negative number etc. The following equality is true:

$self^{-\alpha}(self^\alpha G)$ =$self^\alpha(self^{-\alpha}G)$ = G. In this way one can introduce self-level space.

A.2 fDprt – elements, self-type fDprt- structures

We consider fuzzy dynamic operator

$$\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt}\ \underset{B}{\overset{A}{action\ Q}}\ \text{(A.2.1.1)},$$

where fuzzy A fuzzy acts Q to fuzzy B, fuzzy D fuzzy acts Q out from fuzzy C, Q is any fuzzy *action* simultaneously. The result of this process will be described by the expression

$$\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDrt}\ \underset{B}{\overset{A}{action\ Q}}\ \text{(A.2.1.2)}.$$

We consider the measure: $\mu**((\underset{D}{\overset{B}{action\ Q)^{-1}}}\text{fDprt}\ \underset{B}{\overset{A}{action\ Q}})=\frac{\mu(A)}{\mu(D)}$, where $\mu(A),\mu(D)$ –usual fuzzy measures of A, D.

Definition A.2.1.1. The fuzzy dynamic operator (A.2.1.1) we shall call fDprt – element of the first type, (A.2.1.2) we shall call fDrt – element of the first type.

Remark A.2.1.1. $\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt}\ \underset{B}{\overset{A}{action\ Q}}$ - the fuzzy analogue of ${}^{C}_{D}Sprt^{A}_{B}$ [17] as a special case of (A.2.1.1), where fuzzy *action Q* is “contain”.

It’s allowed to add fDprt – elements:

$$\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt}\ \underset{B}{\overset{A_1}{action\ Q}}+\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt}\ \underset{B}{\overset{A_2}{action\ Q}}=\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt}\ \underset{B}{\overset{A_1\cup A_2}{action\ Q}}\ \text{(A.2.1.2.1)},$$

$$\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt}\ \underset{B_1}{\overset{A}{action\ Q}}+\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt}\ \underset{B_2}{\overset{A}{action\ Q}}=\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt}\ \underset{B_1\cup B_2}{\overset{A}{action\ Q}}\ \text{(A.2.1.2.2)},$$

$$\underset{D}{\overset{C_1}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{B}{\overset{A}{\mathit{action}\ Q}} + \underset{D}{\overset{C_2}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{B}{\overset{A}{\mathit{action}\ Q}} =$$

$$\underset{D}{\overset{C_1 \cup C_2}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{B}{\overset{A}{\mathit{action}\ Q}} \text{ (A.2.1.2.3),}$$

$$\underset{D_1}{\overset{C}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{B}{\overset{A}{\mathit{action}\ Q}} + \underset{D_2}{\overset{C}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{B}{\overset{A}{\mathit{action}\ Q}} =$$

$$\underset{D_1 \cup D_2}{\overset{C}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{B}{\overset{A}{\mathit{action}\ Q}} \text{ (A.2.1.2.4).}$$

We consider the following self-type fuzzy Dprt-structures of the first type:

$$\underset{\mathit{action}\ Q}{\overset{\mathit{action}\ Q}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{\mathit{action}\ Q}{\overset{\mathit{action}\ Q}{\mathit{action}\ Q}} \text{ (A.2.1.3),}$$

denote fD_1fQ.

$$\underset{A}{\overset{\mathit{action}\ Q}{(\mathit{action}\ Q)^{-1}\mathrm{FDprt}}}\ \underset{\mathit{action}\ Q}{\overset{A}{\mathit{action}\ Q}} \text{ (A.2.1.4),}$$

denote $fD_2fA;Q$.

$$\underset{A}{\overset{B}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{B}{\overset{A}{\mathit{action}\ Q}} \text{ (A.2.1.5),}$$

denote $fD_3fA;Q;B$.

$$\underset{A}{\overset{A}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{A}{\overset{A}{\mathit{action}\ Q}} \text{ (A.2.1.6),}$$

denote $fD_4fA;Q$.

$$\underset{strA}{\overset{a}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{a}{\overset{strA}{\mathit{action}\ Q}} \text{ (A.2.1.6.1),}$$

denote $fD_5fA;Q;a$, $a\subset A$ and fuzzy structure of A fuzzy acts Q to fuzzy a and fuzzy acts Q out from fuzzy a simultaneously.

$$\underset{a}{\overset{StrA}{(\mathit{action}\ Q)^{-1}\mathrm{fDprt}}}\ \underset{StrA}{\overset{a}{\mathit{action}\ Q}} \text{ (A.2.1.6.2),}$$

denote $fD_6 fa; Q; A,\ a \subset A$and fuzzy acts Q to fuzzy structure of A and fuzzy acts Q out from fuzzy structure of A simultaneously.

$$\left(\begin{matrix} B \\ action\ Q \\ B \end{matrix}\right)^{-1} \text{fDprt} \begin{matrix} A \\ action\ Q \\ B \end{matrix} \text{ (A.2.1.7)},$$

and any other possible options of fuzzy self for (A.2.1.1) etc.

It can be considered a simpler version of the fuzzy dynamic operator

$$\text{fDprt} \begin{matrix} A \\ action\ Q \\ B \end{matrix} \text{ (A.2.1.8)}$$

where fuzzy A fuzzy acts Q to fuzzy B, Q is any fuzzy *action*, the result of this process will be described by the expression

$$\text{fDrt} \begin{matrix} A \\ action\ Q \\ B \end{matrix} \text{ (A.2.1.9)}$$

or

$$\left(\begin{matrix} C \\ action\ Q \\ D \end{matrix}\right)^{-1} \text{fDprt (A.2.1.10)}$$

where fuzzy D fuzzy acts Q out from fuzzy C, Q is any fuzzy *action*, the result of this process will be described by the expression

$$\left(\begin{matrix} C \\ action\ Q \\ D \end{matrix}\right)^{-1} \text{fDrt (A.2.1.11)}$$

Definition A.2.1,2. The fuzzy dynamic operator (A.2.1.8) we shall call fDprt – element of the second type, (A.2.1.9) we shall call fDrt – element of the second type.

Remark A.2.1.2. fDprt $\begin{matrix} A \\ action\ Q \\ B \end{matrix}$- the fuzzy analogue of $Sprt_B^A$ [1-3] as a special case of (A.2.1.8), where *action Q* is "fuzzy contain". In this case

$fSprt_{action\ Q}^{action\ Q} = \text{fDprt} \begin{matrix} action\ Q \\ action\ Q \\ action\ Q \end{matrix}$ – fuzzy self-containment and unlike usual fuzzy self has higher level self(fuzzy contain): sel$f^{\frac{3}{2}}$. That's why fuzzy self-containment can generate, modify and perform other actions with fuzzy self-capacities, because they have lower level = self.

It's allowed to add fDprt – elements of the second type:

$$\text{fDprt}\ \underset{B}{\overset{A_1}{action\ Q}} + f\text{Dprt}\ \underset{B}{\overset{A_2}{action\ Q}} = \text{fDprt}\ \underset{B}{\overset{A_1 \cup A_2}{action\ Q}}\ \text{(A.2.1.12)},$$

$$\text{fDprt}\ \underset{B_1}{\overset{A}{action\ Q}} + \text{fDprt}\ \underset{B_2}{\overset{A}{action\ Q}} = \text{fDprt}\ \underset{B_1 \cup B_2}{\overset{A}{action\ Q}}\ \text{(A.2.1.13)},$$

We consider the following self-type fDprt-structures of the second t type:

$$\text{fDprt}\ \underset{A}{\overset{A}{action\ Q}}\ \text{(A.2.1.14)},$$

$$\text{fDprt}\ \underset{a}{\overset{strA}{action\ Q}}\ \text{(A.2.1.14.1)},$$

denote $fD_7 fA; Q; a$, $a \subset A$ and fuzzy structure of A fuzzy acts Q to fuzzy a.

$$\text{fDprt}\ \underset{strA}{\overset{a}{action\ Q}}\ \text{(A.2.1.14.2)},$$

denote $fD_8 fa; Q; A$, $a \subset A$ and fuzzy acts Q to fuzzy structure of A.

$$\text{fDprt}\ \underset{action\ Q}{\overset{action\ Q}{action\ Q}}\ \text{(A.2.1.15)},$$

$$\text{fDprt}\ \underset{action\ Q}{\overset{A}{action\ Q}}\ \text{(A.2.1.16)},$$

and any other possible options of fuzzy self for (A.2.1.8) etc.

Definition A.2.1.3. The fuzzy dynamic operator (A.2.1.10) we shall call ftprD – element, (A.2.1.11) we shall call ftrD – element.

Remark A.2.1.3. $\underset{D}{\overset{C}{(action\ Q)^{-1}}}$fDprt - the fuzzy analogue of ${}^{C}_{D}Sprt$ [1-3] as a special case of (A.2.1.10), where fuzzy *action Q* is "fuzzy contain".

It's allowed to add ftrD – elements:

$$\underset{D}{\overset{C_1}{(action\ Q)^{-1}}}\text{fDprt} + \underset{D}{\overset{C_2}{(action\ Q)^{-1}}}\text{fDprt} = \underset{D}{\overset{C_1 \cup C_2}{(action\ Q)^{-1}}}\text{fDprt}\ \text{(A.2.1.17)},$$

$$(\overset{C}{\underset{D_1}{action\ Q}})^{-1}\text{fDprt} + (\overset{C}{\underset{D_2}{action\ Q}})^{-1}\text{fDprt} = (\overset{C}{\underset{D_1 \cup D_2}{action\ Q}})^{-1}\text{fDprt}\ \text{(A.2.1.18)}.$$

We consider the following self-type ftprD-structures:

$$(\overset{D}{\underset{D}{action\ Q}})^{-1}\text{fDprt}\ \text{(A.2.1.19)}$$

$$(\overset{strD}{\underset{d}{action\ Q}})^{-1}\text{fDprt}\ \text{(A.2.1.19.1)},$$

denote $fD_9 fd; Q; D$, $d \subset D$ and fuzzy d fuzzy acts Q out from fuzzy structure of fuzzy D.

$$(\overset{d}{\underset{strD}{action\ Q}})^{-1}\text{fDprt}\ \text{(A.2.1.19.2)},$$

denote $fD_{10} fD; Q; d$, $d \subset D$ and fuzzy structure of fuzzy D fuzzy acts Q out from fuzzy d.

$$(\overset{action\ Q}{\underset{D}{action\ Q}})^{-1}\text{fDprt}\ \text{(A.2.1.20)}$$

$$(\overset{action\ Q}{\underset{action\ Q}{action\ Q}})^{-1}\text{fDprt}\ \text{(A.2.1.21)}$$

and any other possible options of fuzzy self for (A.2.1.10) etc.

Dynamic fDprt – elements, self-type dynamic Dprt-structures

We considered fDprt – elements earlier. Here we consider dynamic fDprt – elements. We consider fuzzy dynamic operator whose elements change over time

$$(\overset{C(t)}{\underset{D(t)}{action\ Q(t)}})^{-1}\text{fDprt(t)}\ \overset{A(t)}{\underset{B(t)}{action\ Q(t)}}\ \text{(A.2.2.1)},$$

where fuzzy A(t) fuzzy acts Q(t) to fuzzy B(t), fuzzy D(t) fuzzy acts fuzzy Q(t) out from fuzzy C(t), fuzzy Q(t) is any fuzzy *action* simultaneously. The result of this process will be described by the expression

$$(\overset{C(t)}{\underset{D(t)}{action\ Q(t)}})^{-1}\text{fDrt(t)}\ \overset{A(t)}{\underset{B(t)}{action\ Q(t)}}\ \text{(A.2.2.2)}.$$

Definition A.2.2.1. The fuzzy dynamic operator (A.2.2.1) we shall call dynamic fDprt – element of the first type, (A.2.2.2) we shall call dynamic fDrt – element of the first type.

Remark A.2.2.1. $\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}$- the fuzzy analogue of ${}^{C(t)}_{D(t)}Sprt(t)^{A(t)}_{B(t)}$ [1], [6], [12] as a special case of (A.2.2.1), where *action* $Q(t)$ is "contain".

It's allowed to add dynamic fDprt – elements:

$$\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A_1(t)}{action\ Q(t)}}+\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A_2(t)}{action\ Q(t)}}=$$
$$\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A_1(t)\cup A_2(t)}{action\ Q(t)}}\quad (A.2.2.2.1),$$

$$\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B_1(t)}{\overset{A(t)}{action\ Q(t)}}+\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B_2(t)}{\overset{A(t)}{action\ Q(t)}}$$
$$=\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B_1(t)\cup B_2(t)}{\overset{A(t)}{action\ Q(t)}}\quad (A.2.2.2.2),$$

$$\underset{D(t)}{\overset{C_1(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}+\underset{D(t)}{\overset{C_2(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}=$$
$$\underset{D(t)}{\overset{C_1(t)\cup C_2(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\quad (A.2.2.2.3),$$

$$\underset{D_1(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}+\underset{D_2(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}$$
$$=\underset{D_1(t)\cup D_2(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\quad (A.2.2.2.4).$$

We consider the following self-type dynamic fDprt-structures of the first type:

$$\underset{action\ Q(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ \underset{action\ Q(t)}{\overset{action\ Q(t)}{action\ Q(t)}}\quad (A.2.2.3),$$

$$\underset{A(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\text{fDprt(t)}\ \underset{action\ Q(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(A.2.2.4),}$$

$$\underset{A(t)}{\overset{B(t)}{(action\ Q(t))^{-1}}}\text{fDrt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(A.2.2.5),}$$

$$\underset{A(t)}{\overset{A(t)}{(action\ Q(t))^{-1}}}\text{fDrt(t)}\ \underset{A(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(A.2.2.6),}$$

$$\underset{strA(t)}{\overset{a(t)}{(action\ Q(t))^{-1}}}\text{fDrt(t)}\ \underset{a(t)}{\overset{strA(t)}{action\ Q(t)}}\ \text{(A.2.2.6.1),}$$

denote $fD_{11}(t)fA(t);Q(t);a(t)$, $a(t) \subset A(t)$ and fuzzy structure of A(t) fuzzy acts Q(t) to fuzzy $a(t)$ and fuzzy acts Q(t) out from fuzzy $a(t)$ simultaneously.

$$\underset{a(t)}{\overset{strA(t)}{(action\ Q(t))^{-1}}}\text{Drt(t)}\ \underset{strA(t)}{\overset{a(t)}{action\ Q(t)}}\ \text{(A.2.2.6.2),}$$

denote $fD_{12}(t)fa(t);Q(t);A(t),\ a(t) \subset A(t)$ and fuzzy acts Q(t) to fuzzy structure of fuzzy A(t) and fuzzy acts fuzzy Q(t) out from fuzzy structure of fuzzy A(t) simultaneously.

$$\underset{B(t)}{\overset{B(t)}{(action\ Q(t))^{-1}}}\text{fDrt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(A.2.2.7),}$$

and any other possible options of fuzzy self for (2.1) etc.

It can be considered a simpler version of the fuzzy dynamic operator

$$\text{fDprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{, (A.2.2.8)}$$

where fuzzy A(t) fuzzy acts Q(t) to fuzzy B(t), the result of this process will be described by the expression

$$\text{fDrt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(A.2.2.9),}$$

or

$$\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\text{fDprt(t)}\ \text{(A.2.2.10),}$$

where fuzzy D(t) fuzzy acts Q(t) out from fuzzy C(t), Q(t) is any fuzzy *action*, the result of this process will be described by the expression

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \text{fDrt(t)} \\ D(t) \end{matrix} \quad (A.2.2.11),$$

Definition A.2.2.2. The fuzzy dynamic operator (A.2.2.8) we shall call dynamic fDprt – element of the second type, (A.2.2.9) we shall call dynamic fDrt – element of the second type.

Remark A.2.2.2. fDprt(t) $\begin{matrix} A(t) \\ action\ Q(t) \\ B(t) \end{matrix}$- the fuzzy analogue of $Sprt(t)_{B(t)}^{A(t)}$ [1-3] as a special case of (A.2.2.8), where *action* $Q(t)$ is “contain”. In this case

$fSprt(t)_{action\ Q(t)}^{action\ Q(t)}$ = fDprt(t) $\begin{matrix} action\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \end{matrix}$ – fuzzy self-containment and unlike usual self has higher level self(fuzzy contain) sel$f^{\frac{3}{2}}$. That's why fuzzy self-containment can generate, modify and perform other actions with fuzzy self-capacities, because they have lower level = self.

It’s allowed to add dynamic fDprt – elements of the second type:

$$\text{fDprt(t)} \begin{matrix} A_1(t) \\ action\ Q(t) \\ B(t) \end{matrix} + f\text{Dprt(t)} \begin{matrix} A_2(t) \\ action\ Q(t) \\ B(t) \end{matrix} = \text{fDprt(t)} \begin{matrix} A_1(t) \cup A_2(t) \\ action\ Q(t) \\ B(t) \end{matrix}$$

(A.2.2.12),

$$\text{fDprt(t)} \begin{matrix} A(t) \\ action\ Q(t) \\ B_1(t) \end{matrix} + \text{fDprt(t)} \begin{matrix} A(t) \\ action\ Q(t) \\ B_2(t) \end{matrix} = \text{fDprt(t)} \begin{matrix} A(t) \\ action\ Q(t) \\ B_1(t) \cup B_2(t) \end{matrix}$$

(A.2.2.13),

We consider the following self-type dynamic fDprt-structures of the second t type:

$$\text{fDprt(t)} \begin{matrix} A(t) \\ action\ Q(t) \\ A(t) \end{matrix} \quad (A.2.2.14),$$

$$\text{fDprt(t)} \begin{matrix} strA(t) \\ action\ Q(t) \\ a(t) \end{matrix} \quad (A.2.2.14.1),$$

denote $fD_{13}(t)fA(t); Q(t); a(t)$, $a(t) \subset A(t)$ and fuzzy structure of fuzzy A(t) fuzzy acts Q(t) to fuzzy $a(t)$.

$$\mathrm{fDprt(t)}\ \underset{strA(t)}{\overset{a(t)}{action\ Q(t)}}\ (\mathrm{A.2.2.14.2}),$$

denote $fD_{14}(t)fa(t);Q(t);A(t),\ a(t) \subset A(t)$ and fuzzy acts Q(t) to fuzzy structure of fuzzy A(t).

$$\mathrm{fDprt(t)}\ \underset{action\ Q(t)}{\overset{action\ Q(t)}{action\ Q(t)}}\ (\mathrm{A.2.2.15}),$$

$$\mathrm{fDprt(t)}\ \underset{action\ Q(t)}{\overset{A(t)}{action\ Q(t)}}\ (\mathrm{A.2.2.16}),$$

and any other possible options of fuzzy self for (A.2.2.8) etc.

Definition A.2.2.3. The fuzzy dynamic operator (A.2.2.10) we shall call dynamic ftprD – element, (A.2.2.11) we shall call dynamic ftrD – element.

Remark A.2.2.3. $\underset{D(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}$ - the fuzzy analogue of ${}^{C(t)}_{D(t)}Sprt(t)$ [1-3] as a special case of (A.2.2.10), where $action\ Q(t)$ is "contain".

It's allowed to add dynamic ftprD – elements:

$$\underset{D(t)}{\overset{C_1(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)} + \underset{D(t)}{\overset{C_2(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)} = \underset{D(t)}{\overset{C_1(t)\cup C_2(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}$$
(A.2.2.17),

$$\underset{D_1(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)} + \underset{D_2(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)} = \underset{D_1(t)\cup D_2(t)}{\overset{C(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}$$
(A.2.2.18).

We consider the following self-type dynamic ftprD-self structures:

$$\underset{D(t)}{\overset{D(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ (\mathrm{A.2.2.19})$$

$$\underset{d(t)}{\overset{strD(t)}{(action\ Q(t))^{-1}}}\mathrm{fDprt(t)}\ (\mathrm{A.2.2.19.1}),$$

denote $fD_{15}(t)fd(t);Q(t);D(t)$, $d(t) \subset D(t)$ and fuzzy d(t) fuzzy acts Q(t) out from fuzzy structure of fuzzy D(t).

$$\begin{matrix} d(t) \\ (action\ Q(t))^{-1}\text{fDprt(t)} \\ strD(t) \end{matrix} \quad \text{(A.2.2.19.2)}$$

denote $fD_{16}(t)fD(t); Q(t); d(t)$, $d(t) \subset D(t)$ and fuzzy structure of fuzzy D(t) fuzzy acts Q(t) out from fuzzy d(t).

$$\begin{matrix} action\ Q(t) \\ (action\ Q(t))^{-1}\text{fDprt(t)} \\ D(t) \end{matrix} \quad \text{(A.2.2.20)}$$

$$\begin{matrix} action\ Q(t) \\ (action\ Q(t))^{-1}\text{fDprt(t)} \\ action\ Q(t) \end{matrix} \quad \text{(A.2.2.21)}$$

and any other possible options of fuzzy self for (A.2.2.10) etc.

New mathematical fuzzy structures and operators is carried out with generalization it to any fuzzy structures with any fuzzy actions. For example,

$$1)\ \begin{matrix} f_{11} & \dots & f_{1k} \\ \dots & \dots & \dots \\ (q_{j1})^{-1} & \dots & (q_{jk})^{-1} \\ \dots & \dots & \dots \\ f_{l1} & \dots & f_{lk} \end{matrix} f\text{DDprt} \begin{matrix} q_{11} & \dots & q_{1n} \\ & \dots & \\ q_{m1} & \dots & q_{mn} \end{matrix} \quad (*_{\text{A.2}}),$$

$f_{ij},\ q_{ij}$ – any fuzzy objects, fuzzy actions etc.

$$2)\ \begin{matrix} g_{11} \\ (w_{j1})^{-1} \\ g_{31} \end{matrix} \begin{matrix} g_{12} \\ g_{22} \\ (w_{j2})^{-1} \\ \dots \\ g_{k2} \end{matrix} \begin{matrix} g_{13} \\ (w_{j3})^{-1} \end{matrix} \text{fDGprt} \begin{matrix} w_{11} & w_{12} & & & w_{1n} \\ \dots & \dots & & & w_{2n} \\ & & & \dots & \\ w_{m1} & w_{m2} & \dots & \dots & w_{ml} \\ & & & w_{sn} & \end{matrix} \quad (*_{\text{A.2.1}}),$$

$w_{ij},\ g_{ij}$ – any fuzzy objects, fuzzy actions etc.

3)

$$\begin{matrix} a & b & g \\ c & fASrq(\mathfrak{t}) & w \\ d & q & r \end{matrix} \quad (*_{\text{A.2.2}}),$$

where $ASrq$ is fuzzy virtual fuzzy structure or fuzzy virtual fuzzy operator, which can take any form of fuzzy action; a, c, d, q, r, w, g, b, μ – any fuzzy objects, fuzzy actions etc.

Accordingly, we can consider all sorts of fuzzy self-structures for 1) – 3). And any other possible fuzzy structures and operators etc.

A.3 Elements of the theory of variables of fuzzy hierarchical dynamic fuzzy operators

In contrast to the classical one-attribute fuzzy set theory, where only its contents are taken as a set, we consider a two-attribute fuzzy set theory with a fuzzy set as a fuzzy capacity and separately with its contents. We simply use a convenient form to represent the singularity of a fuzzy set. Articles [1-3],[8-16] use the following methodology for permanent structures:

1. Cancellation of the axiom of regularity.

2. 2 attributes for the fuzzy set: fuzzy capacity and its content.

3. Fuzzy compression of a fuzzy set, for example, to a point.

4. “turning out” from one another, particularly from a fuzzy capacity, we pull out another fuzzy capacity, for example, itself, as its element.

5. The simultaneity of one (fuzzy compression) and the other (“eversion”).

6. Own fuzzy capacities.

7. Qualitatively new fuzzy programming and fuzzy Networks.

Here we will consider variable fuzzy structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable fuzzy structures (models), for example,

$$\underset{D}{\overset{C}{(action\ Q)^{-1}}}\text{fDprt(t)}\ \underset{B}{\overset{A}{action\ Q}} = \begin{cases} (\ \underset{D}{\overset{C}{(action\ Q)^{-1}}} fDprt\ ,\quad q_2 \geq t \geq q_1)|\mu_1 \\ (\ \underset{D}{\overset{B}{\mu_7}} ffS^1prt\underset{B}{\overset{A}{\mu_6}}\quad ,q_3 \geq\ t > q_2)|\mu_2 \\ ((\underset{D}{\overset{C}{action\ Q)^{-1}}} fDprt\ \underset{B}{\overset{A}{action\ Q}}\quad ,q_4 \geq\ \ t > q_3)|\mu_3 \\ (\ fDprt\ \underset{B}{\overset{A}{action\ Q}}\ \ ,q_5 \geq\ t > q_4)|\mu_4 \\ \underset{D}{\overset{\{\ \}}{(action\ Q)^{-1}}} fDprt,\quad t > q_5)|\mu_5 \\ \ldots \end{cases}$$

($*_{\text{D.1}}$),

μ_i- measures of fuzziness, i = 1, ..., 5. In particular, $\underset{D}{\overset{B}{\mu_7}} ffS^1prt\underset{B}{\overset{A}{\mu_6}}$ can be interpreted as a fuzzy game: player 1 fuzzy with measures of fuzziness μ_6 fits fuzzy A into fuzzy B, and the other fuzzy with measures of fuzziness μ_7 pushes fuzzy D out of fuzzy B at the same time [10], [15].

In what follows, we will denote variable fuzzy structure (model) through fVD(t), qself-variable fuzzy structures (models) through DqfFVS(t), qself is self for *action* Q, and oqself-variable fuzzy structures (models) through OqfVD(t), qoself is oself for *action* Q. Singular fuzzy structures (models) are not confused with fuzzy structures (models) with singularities. $\underset{D}{\overset{B}{\mu_7}} ffS^1prt\underset{B}{\overset{A}{\mu_6}}$ -2-hierarchical fuzzy structure: 1-level - elements A, B, C, D; level 2 - connections between them. 2-

Examples: a) discrete variable fuzzy structure with ţ_i- measures of fuzziness, i = 1, ..., 8.

$$\begin{array}{ccc} a|\text{ţ}_1 & b|\text{ţ}_8 & g|\text{ţ}_7 \\ c|\text{ţ}_2 & ffVD(t) & w|\text{ţ}_6 \\ d|\text{ţ}_3 & q|\text{ţ}_4 & r|\text{ţ}_5 \end{array}$$

Fig.A.3.1

c) continuous variable fuzzy structure

Fig. 0.1

Where a continuous fuzzy set represents the rim of the Fig. 0.1.

We introduce the notation m_{fVS_N}– the number of elements, N - the number of connections between them in the discrete variable 2-hierarchical fuzzy structure fVD(t). We introduce the notation q_{fVS_R}– any, R - connections in q_{fVS_R} in the variable 2-hierarchical fuzzy structure fVD(t), in particular, q_{fVS_R}, R can be fuzzy sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the fuzzy structurability of Q from the interval [0,1], where 0 corresponds to "no fuzzy structure"," and 1 corresponds to the value " fuzzy structure". Then for joint fuzzy A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D- self-(fuzzy structure) from A*B, cS(x)- the value of self-(fuzzy structure) for self-(fuzzy structure) x; for dependent fuzzy structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional fuzzy structurability of the fuzzy structure B at the fuzzy structure A, c(A/B)- conditional fuzzy structure of the fuzzy structure A at the fuzzy structure B. Adding inconsistent fuzzy structures: c(A+B) =c(A) + c(B). The formula of complete fuzzy structure: c(A)=$\sum_{k=1}^{n} c\left(B_k\right) * c(A/B_k)$, B$_1$,

B_2,.., B_n-full group of fuzzy hypotheses- actions: $\sum_{k=1}^{n} c\,(B_k)$=1("fuzzy structure"). Fuzzy Dprt- structure for fuzzy set of fuzzy structures $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$):

$$\text{fDprt} \begin{array}{c} (x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2), \ldots, x_n|\mu_{\widetilde{x}}(x_n)) \\ action\ Q \\ B \end{array}.$$

$$\text{fDprt} \begin{array}{c} \{c(x_1)|\mu_{c(\widetilde{x})}c(x_1)|\mu_{c(\widetilde{x})}c(x_2), \ldots, c(x_n)|\mu_{c(\widetilde{x})}c(x_n)\} \\ action\ Q \\ B \end{array}$$ - fuzzy Dprt- structurability for these fuzzy structures. It is possible to consider the self-(fuzzy structure) $fD_8 f\widetilde{x_w}; Q; \widetilde{x}$, $\widetilde{x_w} \subset \widetilde{x}$. The same for self-(fuzzy structurability): $fD_8 fC_w(\widetilde{x}); Q; \widetilde{C(x)}$, where $\widetilde{C(x)} = \{ c(x_1)|\mu_{c(\widetilde{x})}c(x_1), c(x_2)|\mu_{c(\widetilde{x})}c(x_2), \ldots, c(x_n)|\mu_{c(\widetilde{x})}c\,(x_n)\}$, $C_w(\widetilde{x}) \subset \widetilde{C(x)}$.

Can be considered N-hierarchical fuzzy structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical fuzzy structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, … can be any in particular, by fuzzy actions, fuzzy sets, and others.

$$\begin{pmatrix} C \\ action\ Q \\ D \end{pmatrix}^{-1} \text{fDprt} \begin{array}{c} A \\ action\ Q \\ B \end{array} : \left\langle \begin{array}{c|c} A \to B & D \leftarrow C \\ A, B & C, D \end{array} \right\rangle \to \begin{pmatrix} fself(A \to B)\mu_{13} \\ A, B \end{pmatrix}$$

$$\begin{pmatrix} C \\ action\ Q \\ D \end{pmatrix}^{-1} \text{fDprt} \begin{array}{c} A \\ action\ Q \\ B \end{array} : \left\langle \begin{array}{c|c} A \to B & D \leftarrow C \\ A, B & C, D \end{array} \right\rangle \to \begin{pmatrix} foself(D \leftarrow C)\mu_{14} \\ C, D \end{pmatrix}$$

Can be considered discrete fuzzy hierarchical fuzzy structure, continuous fuzzy hierarchical fuzzy structure, and discrete-continuous hierarchical fuzzy structure, $\text{fDprt} \begin{array}{c} N - hierarchical\ fuzzy\ structure \\ action\ Q \\ B \end{array}$.

The example

$$\text{fQHD= fHDprt}\ \begin{matrix} \left(fDprt \begin{matrix} N-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix}\right)\mu_N \\ \dots \\ \left(fDprt \begin{matrix} i-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix}\right)\mu_i \\ \dots \\ \left(fDprt \begin{matrix} 1-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix}\right)\mu_1 \\ action\ Q \\ B \end{matrix}\ \text{- fuzzy N-}$$

hierarchical fuzzy structure compression into B, μ_i- measures of fuzziness, i = 1, ..., N.

Let fdg(N, fQHD)= $fQHD^{\left.fQHD^{fQHD^{\cdots fQHD}}\right\}\text{-N levels}}$

It can be considered self- fQHD, fdg(y, fQHD) for any y, fdg(fQHD, fQHD).

Compression fuzzy Hierarchy Examples:

$$1)\ fDprt \begin{matrix} () \\ fDprt\ action\ () \\ () \\ () \\ () \\ Dprt\ action\ () \\ () \end{matrix} = \left(\begin{matrix} fDprt \begin{matrix} () \\ fDprt\ action\ () \\ () \\ () \\ () \\ () \\ fDprt\ action\ () \\ () \\ () \end{matrix} \\ fDprt \begin{matrix} () \\ fDprt\ action\ () \\ () \\ () \\ () \end{matrix} \end{matrix} \right)$$

$$2)\quad \begin{array}{ccc} \begin{array}{c} ()\qquad\quad () \\ C+\quad \mu_2 ffS_1prt\mu_1 \\ ()\qquad\quad () \end{array} & & \begin{array}{c} ()\qquad\quad () \\ A+\quad \mu_2 ffS_1prt\mu_1 \\ ()\qquad\quad () \end{array} \\ \mu_2 & \mathrm{ffS_1prt} & \mu_1 \\ \begin{array}{c} ()\qquad\quad () \\ D+\mu_2 ffS_1prt\mu_1 \\ ()\qquad\quad () \end{array} & & \begin{array}{c} ()\qquad\quad () \\ B+\mu_2 ffS_1prt\mu_1 \\ ()\qquad\quad () \end{array} \end{array} = \left(\begin{array}{c} \begin{array}{ccc} \begin{array}{c} ()\qquad\quad () \\ \mu_2 ffS_1prt\mu_1 \\ ()\qquad\quad () \end{array} & & \begin{array}{c} ()\qquad\quad () \\ \mu_2 ffS_1prt\mu_1 \\ ()\qquad\quad () \end{array} \\ \mu_2 & ffS_1prt & \mu_1 \\ \begin{array}{c} ()\qquad\quad () \\ \mu_2 ffS_1prt\mu_1 \\ ()\qquad\quad () \end{array} & & \begin{array}{c} ()\qquad\quad () \\ \mu_2 ffS_1prt\mu_1 \\ ()\qquad\quad () \end{array} \end{array} \\ \begin{array}{c} C \qquad\quad A \\ \mu_2 ffS_1prt\mu_1 \\ D \qquad\quad B \end{array} \end{array} \right),$$

where μ_i- measures of fuzziness, i = 1, 2.

Let's consider two versions: 1) fuzzy containment is interpreted through the concept of fuzzy containment, and 2) fuzzy capacity is interpreted through the concept of fuzzy containment as a rest point of fuzzy containment. Self-(fuzzy containment) is interpreted as a rest point of self-(fuzzy containment). Let A self-(fuzzy compress) into B, D self-(fuzzy displace) from C in $\begin{array}{c} C \qquad\quad A \\ \mu_2 ffVS_1prt\mu_1 \\ D \qquad\quad B \end{array}$.

We consider the functional ca(Q), which gives a numerical value for the accommodation of fuzzy Q from the interval [0,1], where 0 corresponds to " fuzzy action" and one corresponds to the value " fuzzy result of action". Then for joint fuzzy A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D- self-(fuzzy action) for A*B, caS(x)- the value of self-(fuzzy result of action) for self-(fuzzy action) of x; for dependent fuzzy actions: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional accommodation of the fuzzy action B at the fuzzy action A, ca(A/B)- conditional fuzzy result of action of the fuzzy action A at the fuzzy action B. Adding the fuzzy capacity values of inconsistent fuzzy action s: ca(A+B)=ca(A)+ca(B). The formula of complete fuzzy result of action: ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2, ..., B_n-full group of fuzzy hypotheses- actions: $\sum_{k=1}^{n} ca\,(B_k)$=1("fuzzy result of action"). Dprt-(fuzzy action) for

$\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$):

$$fDprt \quad \begin{array}{c} (x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\dots,x_n|\mu_{\widetilde{x}}(x_n)) \\ action\ Q \\ w \end{array}$$

, Q is fuzzy action, $\widetilde{x}$ - fuzzy set of fuzzy actions.

$fDprt\underset{w}{\overset{\{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_2),\ldots,ca(x_n)|\mu_{ca(\widetilde{x})}ca(x_n)}{action\ Q}}$ - fDprt-accommodation for these fuzzy actions x_i, i = 1, ..., n. It is possible to consider the self-(fuzzy action) $fD_8f\widetilde{x_w};Q;\widetilde{x}$, $\widetilde{x_w}\subset\ \widetilde{x}$. The same for self-(fuzzy accommodation): $fD_8fCa_w(\widetilde{x});Q;\widetilde{Ca(x)}$, where $Ca_w(\widetilde{x}) = \{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1), ca(x_2)|\mu_{ca(\widetilde{x})}ca(x_2),\ldots,ca(x_n)|\mu_{ca(\widetilde{x})}ca\,(x_n)\}\subset\widetilde{Ca(x)}$.

Consider a variable fuzzy hierarchy (we will denote it by fdVH).

We consider the functional h(Q), which gives a numerical value for the hierarchization of fuzzy Q from the interval [0,1], where 0 corresponds to "no fuzzy hierarchy," and 1 corresponds to the value " fuzzy hierarchy". Then for joint fuzzy hierarchies A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- self-(fuzzy hierarchy) from A*B, hS(x)- the value of self-(fuzzy hierarchy) for self-(fuzzy hierarchy) x; for dependent fuzzy hierarchies: h(A*B)=ha(A)*h(B/A)=h(B)*h(A/B), where h(B/A)- conditional hierarchization of the fuzzy hierarchy B at the fuzzy hierarchy A, h(A/B)- conditional fuzzy hierarchy of the fuzzy hierarchy A at the fuzzy hierarchy B. Adding the fuzzy hierarchy values of inconsistent fuzzy hierarchies: h(A+B)=h(A)+h(B). The formula of complete fuzzy hierarchy: h(A)=$\sum_{k=1}^{n}h\,(B_k)*h(A/B_k)$, B_1, B_2,..,B_n-full group of fuzzy hypotheses-hierarches: $\sum_{k=1}^{n}h\,(B_k)$=1("fuzzy hierarchy").

fDprt- structure for fuzzy set of hierarches $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, ..., $x_n|\mu_{\widetilde{x}}(x_n)$): fDprt $\underset{B}{\overset{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n))}{action\ Q}}$.

fDprt $\underset{B}{\overset{\{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1)|\mu_{h(\widetilde{x})}h(x_2),\ldots,h(x_n)|\mu_{h(\widetilde{x})}h(x_n)\}}{action\ Q}}$ - fDprt-hierarchization for these fuzzy hierarchies. It is possible to consider the self-(fuzzy hierarchy) $fD_8f\widetilde{x_w};Q;\widetilde{x}$, $\widetilde{x_w}\subset\ \widetilde{x}$. The same for self-hierarchization $fD_8f\widetilde{hx_w};Q;\widetilde{hx}$, $\widetilde{hx_w}\subset\ \widetilde{hx}$, $\widetilde{hx} = \{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1), h(x_2)|\mu_{h(\widetilde{x})}h(x_2),\ldots,h(x_n)|\mu_{h(\widetilde{x})}h\,(x_n)\}$. Can be considered fDprt $\underset{B}{\overset{\{\ ca(x),c(x),h(x)\}}{action\ Q}}$.

Very interesting next fuzzy hierarchy type:

$\left(\begin{matrix} fuzzy\ hierarchy\ A \\ action\ Q \\ fuzzy\ hierarchy\ A \end{matrix}\right)^{-1}$ fDprt $\begin{matrix} fuzzy\ hierarchy\ A \\ action\ Q \\ fuzzy\ hierarchy\ A \end{matrix}$. You can enter special operator fCprt to work with fuzzy structures: $\left(\begin{matrix} A \\ action\ Q \\ D \end{matrix}\right)^{-1}$fCprt $\begin{matrix} C \\ action\ Q \\ R \end{matrix}$ fuzzy structures R by fuzzy Q with the fuzzy structure from C, expelling fuzzy structure D by fuzzy action Q^{-1} from the fuzzy structure A simultaneously.

Very interesting next fuzzy structure type: $\left(\begin{matrix} A \\ action\ Q \\ A \end{matrix}\right)^{-1}$fCprt $\begin{matrix} A \\ action\ Q \\ A \end{matrix}$.

You can enter special operator fHt to work with fuzzy hierarchies: $\left(\begin{matrix} A \\ action\ Q \\ B \end{matrix}\right)^{-1}$fHprt $\begin{matrix} D \\ action\ Q \\ R \end{matrix}$ fuzzy hierarchizes R by fuzzy Q with the fuzzy hierarchy from D, expelling fuzzy hierarchy B by fuzzy action Q^{-1} from the fuzzy hierarchy A simultaneously.

Appendix

Let us introduce the following notations:

A*B= fSprt_B^A, A^2 = fSelf A $=fSrt_A^A$, $A^{\frac{3}{2}}$ = fDrt $\begin{matrix} A \\ A \\ A \end{matrix}$ = fsel$f^{\frac{3}{2}}$ (A), A^3 = fSelf^{2}A, $\ldots$,$A^{\frac{3n}{2}}$ = fDrt $\begin{matrix} A^n \\ A^n \\ A^n \end{matrix}$ = fsel$f^{\frac{3n}{2}}$ (A), A^{n+1} = fSelf^{n}A, fsel$f^{min(n,m)}$(A) $fSrt_{A^m}^{A^n}$ = fsel$f^{\frac{n}{m}}$ (A), fsel$f^{min(n,m,k)}$(A) fDrt $\begin{matrix} A^n \\ A^m \\ A^k \end{matrix}$ = fsel$f^{\frac{n+m+k}{2k}}$ (A), $\ldots$ etc.

fself is fuzzy self. A,B – fuzzy sets.

There is no commutativity here: A*B B*A. We can consider operator functions for fuzzy A: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

You can consider a more “hard” option: A*B=$fPSprt_B^A$, where $fPSprt_B^A$ – fuzzy operator, fuzzy containing fuzzy A in every element of fuzzy B, A^2 = fPSelf A $=fPSrt_A^A$, A^3 = fPSelf^{2}A, $\ldots$, A^{n+1} = fPSelf^{n}A, fPSel$f^{min(n,m)}$(A) $fPSrt_{A^m}^{A^n}$ = fPSel$f^{\frac{n}{m}}$ (A), $\ldots$etc. There is no commutativity here: A*B B*A. We can consider operator functions for fuzzy A: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

Let's introduce $\sqrt{fself}$ as the result of the decision of the equation $fSrt_x^x$ = fself, that is x = $\sqrt{fself}$, $\sqrt[3]{fself}$ as the result of the decision of the equation fDprt $\underset{x}{\overset{x}{x}}$ = self, that is x = $\sqrt[3]{fself}$, $\sqrt[n]{fself^m}$ as the result of the decision of the equation $x^{\frac{n}{m}}$ = fself, $fself^{\alpha}$ as the result of the decision of the equation $x^{\frac{1}{\alpha}}$ = fself, where is any number , in particular, a negative number etc. The following equality is true:

$fself^{-\alpha}(fself^{\alpha}G)$ =$fself^{\alpha}(fself^{-\alpha}G)$ = G etc.

In this way one can introduce the fuzzy self-level space.

A.4 Introduction to FUZZY PROGRAM OPERATORS fDprt, ftprD

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions. This must be done through fDprt-Networks - fuzzy analogue of Sit-Networks [1-3] in one of the central departments of which a conventional computer system is located. The parallel processor is itself fdeprogram - fuzzy analogue of eprogram [1-3] with direct parallel computing not through serial computing.

Using conventional coding by a computer system, through a Target-block with a fuzzy Dprt -program operator - fDprt $\underset{activation}{\overset{Ag}{action\ Q}}$, where fuzzy A with measure of fuzziness μ_A fuzzy acts Q with measure of fuzziness μ_Q to fuzzy *activation* with measure of fuzziness $\mu_{activation}$, Q is any fuzzy *action*, it will be possible to obtain the fuzzy execution with measure of fuzziness $\mu_{activation}$ of a parallel fuzzy action A with the desired target weight g or the execution with measure of fuzziness $\mu_{activation}$ of a parallel action A with the desired fuzzy target weight g with measure of fuzziness μ_g or both. Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For fDprt-coding and fDprt-translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018 year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel fprogram of operator

fDprt $\underset{activation}{\overset{\{UHF\ AC\ R\}}{action\ Q}}$ with the specified measures of fuzziness, we simultaneously enter the desired fuzzy set of codes R with measure of fuzziness μ_R using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel fuzzy fdprogram operators:

1) fuzzy Dprt-program operators (designation fDprt-program operators)
2) fuzzy tprD-program operators (designation ftprD-program operators)
3) fuzzy D^1epr - program operators (designation fD^1epr -program operators)
4) fuzzy $Deprt_1$- program operators (designation f$Deprt_1$-program operators)

fDprt-algorithm Example:

Simultaneous multiplication Q with measure of fuzziness μ_Q: $\widetilde{x}$ fDprt*multiplication* Q, the notation of the fuzzy set B with elements $\widetilde{y}$ $b_{i_1 i_2 \ldots i_n j_1 j_2 \ldots j_n} =$

$$\left(fDprt\ \left(ffSprt \underset{\underset{q}{\mu}}{\left\{x_{1_{i_1}} * \mu_{\widetilde{x}}\left(x_{1_{i_1}}\right) *, x_{2_{i_2}} * \mu_{\widetilde{x}}\left(x_{2_{i_2}}\right) *, ..., \ \ x_{n_{i_n}} * \mu_{\widetilde{x}}\left(x_{n_{i_n}}\right)\right\}}\right)\ multiplication\ Q\ \left(ffSprt \underset{\underset{h}{\mu}}{\left\{y_{1_{j_1}} * \mu_{\widetilde{y}}\left(y_{1_{j_1}}\right), y_{2_{j_2}} * \mu_{\widetilde{y}}\left(y_{2_{j_2}}\right), ..., \ \ y_{n_{j_m}} * \mu_{\widetilde{y}}\left(y_{n_{j_m}}\right)\}\right\}}\right)^{R}\right)_{G \quad V}$$

for any $\{i_1, i_2, ..., i_n\}$, $\{j_1, j, ..., j_n\}$ *without repetitions*, $q =$ ffSprt $\overset{K}{\underset{w}{\mu}}$, K-set of any $\{k_1*, k_2*, ..., k_n*\}$ without repeating them, k_i-any digit, i=1,2,. . . ,n, R= ffSprt $\overset{\{i_1+, i_2+, ..., i_n\}}{\underset{w}{\mu}}$, R is the index of the lower discharge, $h =$ ffSprt $\overset{L}{\underset{w}{\mu}}$, L-set of any $\{l_1*, l_2*, ..., l_m*\}$ without repeating them, l_i-any digit, i=1,2,. . . ,m, G= ffSprt $\overset{\{j_1+, j_2+, ..., j_m\}}{\underset{w}{\mu}}$, G is the index of the lower discharge, V = ffSprt $\overset{\{i_1+, i_2+, ..., i_n + j_1+, j_2+, ..., j_m\}}{\underset{w}{\mu}}$ (we choose an index on the scale of discharges): see Table I.1. Then ffSprt $\overset{B+}{\underset{w}{\mu}}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. Here, in fact, sets of digits in the corresponding digits, representing numbers, are multiplied together simultaneously. The simplest functional scheme of the assumed arithmetic-logical device for fDprt-multiplication:

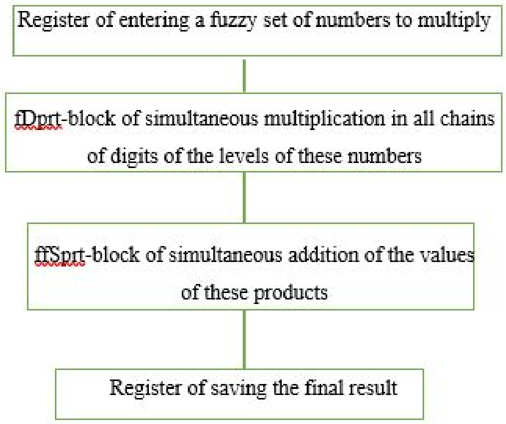


Fig. 0.2 The straightforward functional scheme of the assumed arithmetic-logical device for fDprt-multiplication.

Remark A.4.1. The algorithm for simultaneously fuzzy multiplication a fuzzy set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by fuzzy multiplying the first number from the fuzzy set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the fuzzy set by the ones following it, etc.

One example is pattern recognition: fDprt $\begin{array}{c} image\ archive \quad q \\ if\ ffSprt\ \mu_B \quad ffSprt\,\mu_B \\ give\ test\ result \\ Name\ of\ q \end{array}$

The example of fDprt-program is

fDprt $\left\{ fDprt\frac{\{\{p\}\}}{\{(x)\}},\ fDprt give\ test\ result\frac{IF\{\{B\}\{f\}\}}{Q},\ fDprt\ action\ G\frac{R}{R} \right\}$ $\begin{array}{c} action\ H \\ x \end{array}$.

For example, based on fDprt $action\ Q$ $\begin{array}{c}\widetilde{y}\\ \widetilde{x}\end{array}$, where $\widetilde{y}$=($y_1|\mu_{\widetilde{x}}(y_1)$, $y_2|\mu_{\widetilde{x}}(y_2)$, …, $y_m|\mu_{\widetilde{x}}(y_m)$) $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$), we can consider self-type fDprt-structure - $fD_8 f\widetilde{y};Q;\widetilde{x}$ with m elements from $\widetilde{x}$, m<n, which is

formed according to the form (1.1) , or in forms (1.1.1) - (1.1.5), generalizing it.

Structures more complex than $fD_8f\widetilde{y};Q;\widetilde{x}$ can be introduced. For example, through the forms that generalizes (1.1): (1.2), (1.3) - (1.4). In particular, in (1.1), A is fuzzy compressed (fuzzy fits) in C in the fuzzy compression fuzzy structure B in C (i.e., in the fuzzy structure $\mathrm{fDprt}\overset{B}{\underset{C}{\mu}}$) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc. (1.3.1) - (1.3.4) schematically interpret the fuzzy formation of fuzzy fDprt-self structure through a pseudo 3-connected form with a 2-connected form. The ideology of fDprt and $fD_8f\widetilde{y};Q;\widetilde{x}$ can be used for programming.

Remark A.4. 2. Fuzzy self, in particular, according to a fuzzy form is fuzzy analogue of the form of type (1.1) [1-3]: (2.1*).

Here are some of the fuzzy fDprt-program operators.

1. Simultaneous fuzzy *action* Q of the expressions $\widetilde{p}$=($\mathrm{p}_1|\mu_{\widetilde{p}}(\mathrm{p}_1)$, $\mathrm{p}_2|\mu_{\widetilde{p}}(\mathrm{p}_2), \ldots,$ $\mathrm{p}_\mathrm{n}|\mu_{\widetilde{p}}(\mathrm{p}_\mathrm{n})$) to the variables $\widetilde{x}$=($\mathrm{x}_1|\mu_{\widetilde{x}}(\mathrm{x}_1)$, $\mathrm{x}_2|\mu_{\widetilde{x}}(\mathrm{x}_2), \ldots,$ $\mathrm{x}_\mathrm{n}|\mu_{\widetilde{x}}(\mathrm{x}_\mathrm{n})$). This is implemented via fDprt $\overset{\widetilde{p}}{\underset{\{\widetilde{x}\}}{Q}}$.
2. Simultaneous R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}$=($\mathrm{g}_1|\mu_{\widetilde{g}}(\mathrm{g}_1)$, $\mathrm{g}_2|\mu_{\widetilde{g}}(\mathrm{g}_2), \ldots,$ $\mathrm{g}_\mathrm{n}|\mu_{\widetilde{g}}(\mathrm{g}_\mathrm{n})$) for the fuzzy set of expressions $\widetilde{B}$=($\mathrm{B}_1|\mu_{\widetilde{B}}(\mathrm{B}_1)$, $\mathrm{B}_2|\mu_{\widetilde{B}}(\mathrm{B}_2), \ldots,$ $\mathrm{B}_\mathrm{n}|\mu_{\widetilde{B}}(\mathrm{B}_\mathrm{n})$). Implemented via fDprt$\overset{\widetilde{B}}{\underset{\widetilde{Q}}{R}}$, where $\widetilde{Q}$ can be anything.
3. Similarly for fuzzy loop operators and others.

fD_8f- fuzzy software operators will differ only just because aggregates $\widetilde{x}$,$\widetilde{p}$, $\widetilde{B}$,$\widetilde{g}$ will be formed from corresponding fdprt-program operators in form (1.1), for more complex operators in forms (1.1,1) - (1.4), (2.1*) and analogs of forms (1.1,1) - (1.4) by type (2.1*).

For example, fDprt$g\overset{S}{\underset{R}{\{R,S\}}}$ is the fuzzy fDprt-self structure with measure of fuzziness µ of the second type if $g\{R,S\}$ is a fdprogram capable of fuzzy generating R *with measure of fuzziness* μ from S.

The example of self-fdprogram of the first type is

$$\text{fDprt}^{\{\, fDprt\,\overset{\widetilde{p}}{\underset{\{\widetilde{x}\}}{Q}},\ \ fDprt\,\overset{\widetilde{B}}{\underset{\widetilde{Q}}{R}},\ \ fDprt\,\overset{Q}{\underset{Q}{Q}}\,\}}_{\substack{action\ R\\ w}}.$$

The example of fdprt-program for fDmnsprt- fuzzy analogue of SmnSt [1-3]:

$fDprt\,\overset{\widetilde{p}}{\underset{\{\widetilde{x}\}}{Q}}$ - fuzzy *action* Q of $\widetilde{p}$ to $\widetilde{x}$.

fDprt $\overset{tw}{\underset{g}{P}}$, where P - fuzzy assigning target weight tw to fuzzy g with measure of fuzziness µ.

fDprt $\underset{fDmnsprt\ activation}{\overset{\{q\}\,w}{S}}$, where S - fDmnsprt *activation* for fuzzy $\{q\}\,w$ with measure of fuzziness µ.

fDprt-coding.

fDprt-coding with measure of fuzziness µ: 1) fuzzy set A to fuzzy set B, 2) fuzzy set A to a point q, where the elements of the fuzzy sets A, B can be continuous. For example, fDprt$\overset{A}{\underset{B}{Q}}$, where Q - fDprt-coding.

There are fDprt-coding, fDprt-translation, fDprt-realize of fdeprograms and fprograms from the archives without extraction theirs

fDelf-coding.

fDelf-coding with measure of fuzziness µ: 1) fuzzy set A to set fuzzy A, i.e. fuzzy A on itself 2) fuzzy set A to a point q in form (A.4.1), where the elements of the fuzzy sets A, B can be continuous. For example, fDprt$\overset{A}{\underset{A}{Q}}$,

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with fDprt-Networks, it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through fDprt-Networks and series-parallel. Codes from a conventional type computer system will be used via fDprt -connectors in fDprt - coding, for example: fDprt $\underset{activation}{\overset{\{UHF\ AC\ \}}{Q}}$. UHF AC field activation is used.

Dynamic fDprt and $fD_8(t)f$ programming.

The ideology of dynamic fDprt and $fD_8(t)f$ can be used for programming:

1. Simultaneous fuzzy *action* $\widetilde{Q(t)}$ of the expressions $\widetilde{p(t)}$=(p_1(t)|$\mu_{\widetilde{p(t)}}$(p_1(t)), p_2(t)|$\mu_{\widetilde{p(t)}}$(p_2(t)), …, p_n(t)|$\mu_{\widetilde{p(t)}}$(p_n(t))) to the variables $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t)), …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))). This is implemented via fDprt(t) $\begin{matrix}\widetilde{p(t)}\\ \widetilde{Q(t)}\\ \left\{\widetilde{x(t)}\right\}\end{matrix}$.

2. Simultaneous $\widetilde{R(t)}$ = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g(t)}$=(g_1(t)|$\mu_{\widetilde{g(t)}}$(g_1(t)), g_2(t)|$\mu_{\widetilde{g(t)}}$(g_2(t)), …, g_n(t)|$\mu_{\widetilde{g(t)}}$(g_n(t))) for the fuzzy set of expressions $\widetilde{B(t)}$=(B_1(t)|$\mu_{\widetilde{B(t)}}$(B_1(t)), B_2(t)|$\mu_{\widetilde{B(t)}}$(B_2(t)), …, B_n(t)|$\mu_{\widetilde{B(t)}}$(B_n(t))). Implemented via fDprt$(t)$$\begin{matrix}\widetilde{B(t)}\\ \widetilde{R(t)}\\ \widetilde{Q(t)}\end{matrix}$, where $\widetilde{Q}$ can be anything.

3. Similarly for fuzzy loop operators and others.

$fD_8(t)f$ - fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}$, $\widetilde{B(t)}$,$\widetilde{g(t)}$ will be formed from corresponding fDprt-program operators in form (1.1), for more complex operators in forms (1.1,1) - (1.4), (2.1*) and analogs of forms (1.1,1) - (1.4) by type (2.1*).

ftprD-program operators.

The ideology of ftprD and $D_{16}f$ - fuzzy analogues of tS and t_{S_4f} from [14] can be used for programming. Here are some of the ftprD-program operators.

1. Simultaneous expelling fuzzy *action* Q of the expressions $\widetilde{p}$=(p_1|$\mu_{\widetilde{p}}$(p_1), p_2|$\mu_{\widetilde{p}}$(p_2), …, p_n|$\mu_{\widetilde{p}}$(p_n)) from the variables $\widetilde{x}$=(x_1|$\mu_{\widetilde{x}}$(x_1), x_2|$\mu_{\widetilde{x}}$(x_2), …, x_n|$\mu_{\widetilde{x}}$(x_n)). This is implemented via $\begin{matrix}\widetilde{x}\\ Q\\ \{\widetilde{p}\}\end{matrix}$ fDprt.

2. Simultaneous expelling R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}$=(g_1|$\mu_{\widetilde{g}}$(g_1), g_2|$\mu_{\widetilde{g}}$(g_2), …, g_n|$\mu_{\widetilde{g}}$(g_n)) for the fuzzy set of expressions $\widetilde{B}$=(B_1|$\mu_{\widetilde{B}}$(B_1), B_2|$\mu_{\widetilde{B}}$(B_2), …, B_n|$\mu_{\widetilde{B}}$(B_n)). It's implemented through $\begin{matrix}\widetilde{Q}\\ R\\ \widetilde{B}\end{matrix}$ fDprt, where $\widetilde{Q}$ can be anything.

3. Similarly for loop operators and others.

$fD_{16}\underset{\sim}{f}$ – fuzzy software operators will differ only just because aggregates $\widetilde{x}, \widetilde{p}, \widetilde{B}, \widetilde{g}$ will be formed from corresponding ftprD program operators in form (1.1) for more complex operators in form (1.1), for more complex operators in forms (1.1,1) - (1.4), (2.1*) and analogs of forms (1.1,1) - (1.4) by type (2.1*).

Consider hierarchical ftprD-program operator

$$\begin{matrix} B \\ (action\ Q)^{-1} Dprt = \\ A \end{matrix} \begin{Bmatrix} & \{\ \} & \\ D + & \underset{\sim}{t} & ffSprt \\ & A - A \cap B & \\ & (\ B - A \cap B) & \end{Bmatrix}, \text{ where D is oself-(fuzzy set)}$$

for fuzzy $(A \cap B)$, where action Q- contain.

Dynamic ftprD and $fD_{16}(t)f$ programming at time q.

The ideology of ftprD and $fD_{16}f$ can be used for dynamic programming. Here are some of the ftprD- dynamic programming operators.

1. The process of simultaneous expelling fuzzy *action* $Q(t)$ of the expressions $\widetilde{p(t)}$=(p_1(t)|$\mu_{\widetilde{p(t)}}$(p_1(t)), p_2(t)|$\mu_{\widetilde{p(t)}}$(p_2(t)), …, p_n(t)|$\mu_{\widetilde{p(t)}}$(p_n(t))) from the variables $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t)), …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))). This is implemented via $\begin{matrix}\widetilde{x(t)} \\ Q(t) \\ \left\{\widetilde{p(t)}\right\}\end{matrix}$ fDprt(t).

2. The process of simultaneous expelling R(t) = fuzzy checking with fuzziness (t) by the fuzzy set of conditions $\widetilde{g(t)}$=(g_1(t)|$\mu_{\widetilde{g(t)}}$(g_1(t)), g_2(t)|$\mu_{\widetilde{g(t)}}$(g_2(t)), …, g_n(t)|$\mu_{\widetilde{g(t)}}$(g_n(t))) for the fuzzy set of expressions $\widetilde{B(t)}$=(B_1(t)|$\mu_{\widetilde{B(t)}}$(B_1(t)), B_2(t)|$\mu_{\widetilde{B(t)}}$(B_2(t)), …, B_n(t)|$\mu_{\widetilde{B(t)}}$(B_n(t))) is implemented through $\begin{matrix}\widetilde{Q(t)} \\ R(t) \\ \widetilde{B(t)}\end{matrix}$fDprt(t), where $\widetilde{Q(t)}$ can be anything.

3.Similarly for loop operators and others.

$fD_{16}(t)f$ – fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}, \widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding processes ftprD(t) for above mentioned programming operators through form (1.1), for more complex operators in forms (1.1,1) - (1.4), (2.1*) and analogs of forms (1.1,1) - (1.4) by type (2.1*).

Consider hierarchical dynamic ftprD-program operator:

$$\underset{A(q)}{\overset{B(q)}{(action\ Q)^{-1}}} fDprt(q) = \left\{ fft(q)_{S_1 f(A(q)\cap B(q))} + \underset{\substack{A(q)-A(q)\cap B(q)\\ \left(B(q)-A(q)\cap B(q)\right)}}{\overset{\{\ \}}{\mu}} \qquad ffSprt(q) \right\},$$

where action Q- contain.

fD^1epr -program operators (form $\underset{D}{\overset{B}{(action\ Q)^{-1}}}\mathrm{fD}^1\mathrm{prt}\ \underset{B}{\overset{A}{action\ Q}}$ - fuzzy analogue of ${}_D^B S^1 t_B^A$ [10]))

For example, $\underset{\{\widetilde{p}\}}{\overset{\widetilde{x}}{R^{-1}}}\mathrm{fD}^1\mathrm{prt}\underset{\widetilde{Q}}{\overset{\widetilde{B}}{R}}$, where simultaneous expelling fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2)$, …, $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2)$, …, $p_n|\mu_{\widetilde{p}}(p_n)$) from the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$) and simultaneous R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2)$, …, $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2)$, …, $B_n|\mu_{\widetilde{B}}(B_n)$), $\widetilde{Q}$ can be anything.

The examples:

$\underset{A}{\overset{A}{(action\ Q)^{-1}}}\mathrm{fD}^1\mathrm{prt}\underset{A}{\overset{A}{action\ Q}}$ can be interpreted as $\begin{pmatrix} s^1elf \\ os^1elf \end{pmatrix}$-fdprogram operator. $\underset{A}{\overset{A}{(action\ Q)^{-1}}}\mathrm{fD}^1\mathrm{prt}\underset{A}{\overset{A}{action\ Q}}$ sample $\begin{pmatrix} s^1elf \\ os^1elf \end{pmatrix}$-fdprogram structure example.

Consider hierarchical dynamic fD^1epr-program operator: (form $\begin{matrix} \underset{A}{\overset{B}{(action\ Q)^{-1}}} fD^1prt\ \underset{B}{\overset{A}{action\ Q}} * \\ \underset{D-A}{\overset{B}{(action\ Q)^{-1}}} fD^1prt\ \underset{B}{\overset{A}{action\ Q}} \end{matrix}$).

fDeprt_1- program operators (form $\underset{D}{\overset{C}{(action\ Q)^{-1}}}\mathrm{fD}_1\mathrm{prt}\underset{B}{\overset{A}{action\ Q}}$ - fuzzy analogue of ${}_D^C S_1 t_B^A$ [3], [9]))

$\underset{A}{\overset{A}{(action\ Q)}}^{-1}fD_1prt\underset{A}{\overset{A}{action\ Q}}$ - sample $\begin{pmatrix} d_1 self \\ d_1 oself \end{pmatrix}$-fdprogram structure example.

fdprogram structure example, where the assemblage point d_r is the cursor, it is quite complex self—fdprogram:

$$\text{fD}_1\text{prt}\left\{ \int_{W_q}^{q} \begin{pmatrix} \underset{A}{\overset{A}{(action\ Q)}}^{-1} {}_{fD1prt} \underset{A}{\overset{A}{action\ Q}} \end{pmatrix} fD_1prt_q^{E_q} \underset{\substack{Q \\ V}}{\begin{pmatrix} \underset{A}{\overset{A}{(action\ Q)}}^{-1} {}_{fD1prt} \underset{A}{\overset{A}{action\ Q}} - \end{pmatrix}}, \quad fD_1prt \underset{d_{r\left(E_{in}l^{d_r}\right)}}{\overset{E^{ex}l^{d_r}}{action\ Q}} \right\}$$

.

$$\text{fD}_1\text{prt}\left\{ \int_{W_q}^{q} \begin{pmatrix} \underset{A}{\overset{A}{(action\ Q)}}^{-1} {}_{fD_1prt} \underset{A}{\overset{A}{action\ Q}} \end{pmatrix} fD_1prt_q^{E_q} \underset{\substack{Q \\ V}}{\begin{pmatrix} \underset{A}{\overset{A}{(action\ Q)}}^{-1} {}_{fD_1prt} \underset{A}{\overset{A}{action\ Q}} - \end{pmatrix}}, \quad fD_1prt \underset{d_{r\left(E_{in}l^{d_r}\right)}}{\overset{E^{ex}l^{d_r}}{action\ Q}} \right\}$$

can be interpreted as a fDpr-program operator.

Appendix

Remark. Energy of a living organism:

fd$_1$g(r, a(E_q)) =

$$\text{fD}_1\text{prt}\left\{ \int_{W_q}^{q} \begin{pmatrix} \underset{A}{\overset{A}{(action\ Q)}}^{-1} {}_{fD_1prt} \underset{A}{\overset{A}{action\ Q}} \end{pmatrix} fD_1prt_q^{E_q} \underset{\substack{Q \\ V}}{\begin{pmatrix} \underset{A}{\overset{A}{(action\ Q)}}^{-1} {}_{fD_1prt} \underset{A}{\overset{A}{action\ Q}} \end{pmatrix}}, fD_1prt \underset{d_{r\left(E_{in}l^{d_r}\right)}}{\overset{E^{ex}l^{d_r}}{action\ Q}} \right\}$$

$(**_{A.4})$.

Energy with measure of fuzziness μ_1, μ_2 of a living organism of a person:

$$fd_1pg(r,\ a(E_q)) = fD_1prt \left\{ \left(\begin{matrix} A & & A \\ (action\ Q)^{-1} & fD_1prt & action\ Q \\ A & & A \end{matrix} \right)_{W_q}^{q} fD_1prt_q^{E_q} \left(\begin{matrix} A & & A \\ (action\ Q)^{-1} & fD_1prt & action\ Q\ - \\ A & & A \end{matrix} \right)_{\substack{Q \\ V}}, fD_1prt \begin{matrix} E^{ex}l^{d_r} \\ action\ Q \\ d_r\left(self\left(E_{in}l^{d_r}\right)\right) \end{matrix} \right\}$$

(***$_{A.4}$).

$\begin{matrix} A \\ (action\ Q)^{-1} \\ A \end{matrix} fD_1prt \begin{matrix} A \\ action\ Q \\ A \end{matrix}$ -internal energy of a living organism, q- a gap in the energy cocoon of a living organism, r-the position of the assemblage point d_r on the energy cocoon of a living organism, W_q- energy prominences from the gap in the cocoon of a living organism, E_q-external energy entering the gap in the cocoon of a living organism, $E^{ex}l^{d_r}$ - a bundle of fibers of external energy self-capacities (fDprt-self structures) from outside the cocoon, collected at the point of assembly of the cocoon of a living organism, $E_{in}l^{d_r}$- a bundle of fibers of external energy self-capacities (fDprt-self structures) from inside the cocoon, collected at the point of assembly of the cocoon of a living organism in the same position r of the assemblage point d_r. d_r is the subject of identifying the energy fibers of the subtle energy of the Universe in position r both outside and inside the cocoon.

(**$_{A.4}$), (***$_{A.4}$) can be interpreted as fDprt - program operators.

Entire neural network as instantaneous simultaneous ffRAM in ffSprt-elements and fself- elements.$fself^{fself^{\cdots fself}}$, $ff1\downarrow I \uparrow_{-1}^{1} ff_2{}^{ff1\downarrow I\uparrow_{-1}^{1}ff_2{}^{\cdots ff1\downarrow I\uparrow_{-1}^{1}ff_2}}$, $fsin\infty^{fsin\infty^{\cdots fsin\infty}}$.When activated in a neural network, the entire neural network becomes a working memory. Use of fself-energy as fuzzy activation or from outside. fdQ_0= fDprt $\begin{matrix} & ffS_{mnSprt} \\ ffSprt & \mu \\ & activation \\ & Q \\ & ffS_{mnSprt} \\ ffSprt & \mu \\ & activation \end{matrix}$ $\rightarrow$ self-ffRAM,

$\text{ffQ}_{00}= \begin{array}{cc} fD_{mnSprt} & \\ D & fDprt \\ activation & \\ Q & \\ fD_{mnSprt} & \\ R & fDprt \\ activation & \end{array} fdQ_0\text{prt}, \text{ffQ}_{01}= \begin{array}{cc} & fD_{mnSprt} \\ {\scriptstyle fDprt} & C \\ & activation \\ & fD_{mnSprt} \\ {\scriptstyle fDprt} & H \\ & activation \end{array} fdQ_0$.fd$Q_0$,fd$Q_{00}$,fd$Q_{01}$-coding,translation,realization in fdeprograms, use fdQ_0,fdQ_{00},fdQ_{01}-fdS_{mnSprt},fdAssembler.

B Rprt – elements, self-type Rprt-structures and their applications

B.1 Rprt – elements, self-type Rprt-structures

We consider dynamic operator

$\underset{D}{\overset{C}{action\ P}}\text{Rprt}\ \underset{B}{\overset{A}{action\ Q}}$ (B.1.1.1),

where A acts Q to B, D acts P out from C; A, B, C, D may be fuzzy with corresponding fuzzy measures; Q and P are any *actions*, in particular, fuzzy *actions*, simultaneously. The result of this process will be described by the expression

$\underset{D}{\overset{C}{action\ P}}\text{fRrt}\ \underset{B}{\overset{A}{action\ Q}}$ (B.1.1.2).

We consider the measure: $\mu**(\underset{D}{\overset{B}{action\ P}}\text{Rprt}\ \underset{B}{\overset{A}{action\ Q}})=\frac{\mu(A)\mu(Q)}{\mu(D)\mu(P)}$, where $\mu(A), \mu(D), \mu(Q)$, $\mu(P)$ –usual measures or fuzzy measures of A, D, Q, P.

Definition B.1.1.1. The dynamic operator (B.1.1.1) we shall call Rprt – element of the first type or fRprt – element of the first type for fuzzy dynamic operator, (B.1.1.2) we shall call Rrt – element of the first type or fRrt – element of the first type for fuzzy dynamic operator.

Remark B.1.1.1. $\underset{D}{\overset{C}{action\ P}}\text{Rprt}\ \underset{B}{\overset{A}{action\ Q}}$ - the analogue of ${}^{C}_{D}Sprt^{A}_{B}$ [17] as a special case of (B.1.1.1), where *action Q* is “contain”, *action P* is Q^{-1}.

Remark B.1.1.1.1 an consider Rprt – elements use the Banach space.

It’s allowed to add Rprt – elements:

$\underset{D}{\overset{C}{action\ P}}\text{Rprt}\ \underset{B}{\overset{A_1}{action\ Q}}+\underset{D}{\overset{C}{action\ P}}\text{Rprt}\ \underset{B}{\overset{A_2}{action\ Q}}=\underset{D}{\overset{C}{action\ P}}\text{Rprt}\ \underset{B}{\overset{A_1\cup A_2}{action\ Q}}$
(B.1.1.2.1),

$\underset{D}{\overset{C}{action\ P}}\text{Rprt}\ \underset{B_1}{\overset{A}{action\ Q}}+\underset{D}{\overset{C}{action\ P}}\text{Rprt}\ \underset{B_2}{\overset{A}{action\ Q}}=\underset{D}{\overset{C}{action\ P}}\text{Rprt}\ \underset{B_1\cup B_2}{\overset{A}{action\ Q}}$
(B.1.1.2.2),

$\underset{D}{\overset{C_1}{action\ P}}\ \text{Rprt}\ \underset{B}{\overset{A}{action\ Q}}+\underset{D}{\overset{C_2}{action\ P}}\text{Rprt}\ \underset{B}{\overset{A}{action\ Q}}=\underset{D}{\overset{C_1\cup C_2}{action\ P}}\ \text{Rprt}\ \underset{B}{\overset{A}{action\ Q}}$
(B.1.1.2.3),

$$\underset{D_1}{\overset{C}{action}}\ P \ \mathrm{Rprt}\ \underset{B}{\overset{A}{action}}\ Q + \underset{D_2}{\overset{C}{action}}\ P\ \mathrm{Rprt}\ \underset{B}{\overset{A}{action}}\ Q = \underset{D_1 \cup D_2}{\overset{C}{action}}\ P\ \mathrm{Rprt}\ \underset{B}{\overset{A}{action}}\ Q$$
(B.1.1.2.4).

Likewise for fuzzy dynamic operator $\underset{D}{\overset{C}{action}}\ P \mathrm{fRprt}\ \underset{B}{\overset{A}{action}}\ Q$.

We consider the following self-type Rprt-structures of the first type:

$$\underset{action\ Q}{\overset{action\ Q}{(action\ Q)^{-1}}}\mathrm{Rprt}\ \underset{action\ Q}{\overset{action\ Q}{action\ Q}}$$ (B.1.1.3),

denote $R_1 f Q$.

$$\underset{A}{\overset{action\ Q}{(action\ Q)^{-1}}}\mathrm{Rprt}\ \underset{action\ Q}{\overset{A}{action\ Q}}$$ (B.1.1.4),

denote $R_2 f A; Q$.

$$\underset{A}{\overset{B}{(action\ Q)^{-1}}}\mathrm{Rprt}\ \underset{B}{\overset{A}{action\ Q}}$$ (B.1.1.5),

denote $R_3 f A; Q; B$.

$$\underset{A}{\overset{A}{(action\ Q)^{-1}}}\mathrm{Rprt}\ \underset{A}{\overset{A}{action\ Q}}$$ (B.1.1.6),

denote $R_4 f A; Q$.

$$\underset{strA}{\overset{a}{(action\ Q)^{-1}}}\mathrm{Rprt}\ \underset{a}{\overset{strA}{action\ Q}}$$ (B.1.1.6.1),

denote $R_5 f A; Q; a$, a denote $D_5 f A; Q; a$, $a \subset A$ and structure of A acts Q to a and and structure of A acts Q to a and acts Q out from a simultaneously.

$$\underset{a}{\overset{StrA}{(action\ Q)^{-1}}}\mathrm{Rprt}\ \underset{StrA}{\overset{a}{action\ Q}}$$ (B.1.1.6.2),

denote $R_6 f a; Q; A$, a denote $D_5 f A; Q; a$, $a \subset A$ and structure of A acts Q to a and and acts Q to structure of A and acts Q out from structure of A simultaneously,

$$\underset{B}{\overset{B}{(action\ Q)^{-1}}}\mathrm{Rprt}\ \underset{B}{\overset{A}{action\ Q}}$$ (B.1.1.7),

$$\Big(\overset{B}{\underset{B}{action\ Q}}\Big)^{-1}\mathrm{Rprt}\ \overset{A}{\underset{B}{action\ Q}}\ (\mathrm{B.1.1.8}),$$

$$\overset{B}{\underset{A}{action\ P}}\mathrm{Rprt}\ \overset{A}{\underset{B}{action\ Q}}\ (\mathrm{B.1.1.9}),$$

$$\overset{B}{\underset{A}{action\ P}}\mathrm{Rprt}\ \overset{B}{\underset{B}{action\ Q}}\ (\mathrm{B.1.1.10}),$$

$$\overset{A}{\underset{A}{action\ P}}\mathrm{Rprt}\ \overset{A}{\underset{A}{action\ Q}}\ (\mathrm{B.1.1.11}),$$

and any other possible options of self for (B.1.1.1) etc.

Likewise for fuzzy dynamic operator $\overset{C}{\underset{D}{action\ P}}\mathrm{fRprt}\ \overset{A}{\underset{B}{action\ Q}}$.

It can be considered a simpler version of the dynamic operator

$$\mathrm{Rprt}\ \overset{A}{\underset{B}{action\ Q}}\ (\mathrm{B.1.1.12}),$$

where A acts Q to B, Q is any *action*, the result of this process will be described by the expression

$$\mathrm{Rrt}\ \overset{A}{\underset{B}{action\ Q}}\ (\mathrm{B.1.1.13})$$

or

$$\overset{C}{\underset{D}{action\ P}}\ \mathrm{Rprt}\ (\mathrm{B.1.1.14})$$

where D acts P out from C, P is any *action*, the result of this process will be described by the expression

$$\overset{C}{\underset{D}{action\ P}}\ \mathrm{Rrt}\ (\mathrm{B.1.1.15})$$

Definition 1,2. The dynamic operator (B.1.1.12) we shall call Rprt – element of the second type or fRprt – element of the second type for fuzzy dynamic operator, (B.1.1.13) we shall call Rrt – element of the second type or fRrt – element of the second type for fuzzy dynamic operator.

Remark B.1.1.2. $\mathrm{Rprt}\ \overset{A}{\underset{B}{action\ Q}}$ - the analogue of $Sprt_B^A$ [17] as a special case of (B.1.1.8), where $action\ Q$ is “contain”. In this case

$Sprt_{action\ Q}^{action\ Q} = \mathrm{Rprt}\ \overset{action\ Q}{\underset{action\ Q}{action\ Q}}$ − self-containment and unlike usual self has higher level self(contain): sel$f^{\frac{3}{2}}$. That's why self-containment can generate, modify and perform other actions with self-capacities, because they have lower level = self.

It’s allowed to add Rprt − elements of the second type:

$$\mathrm{Rprt}\ \overset{A_1}{\underset{B}{action\ Q}}+\mathrm{Rprt}\ \overset{A_2}{\underset{B}{action\ Q}} = \mathrm{Rprt}\ \overset{A_1 \cup A_2}{\underset{B}{action\ Q}} \text{ (B.1.1.16)},$$

$$\mathrm{Rprt}\ \overset{A}{\underset{B_1}{action\ Q}} + \mathrm{Rprt}\ \overset{A}{\underset{B_2}{action\ Q}} =\mathrm{Rprt}\ \overset{A}{\underset{B_1 \cup B_2}{action\ Q}} \text{ (B.1.1.17)}.$$

Likewise for fuzzy dynamic operator $\mathrm{fRprt}\ \overset{A}{\underset{B}{action\ Q}}$.

We consider the following self-type Rprt-structures of the second t type:

$$\mathrm{Rprt}\ \overset{A}{\underset{A}{action\ Q}} \text{ (B.1.1.18)},$$

$$\mathrm{Rprt}\ \overset{strA}{\underset{a}{action\ Q}} \text{ (B.1.1.18.1)},$$

denote $R_7 f A; Q; a$, a denote $D_5 f A; Q; a$, $a \subset A$ and structure of A acts Q to a and and structure of A acts Q to a,

$$\mathrm{Rprt}\ \overset{a}{\underset{strA}{action\ Q}} \text{ (B.1.1.18.2)},$$

denote $R_8 f a; Q; A,$ $a$denote $D_5 f A; Q; a$, $a \subset A$ and structure of A acts Q to a and and acts Q to structure of A,

$$\mathrm{Rprt}\ \overset{action\ Q}{\underset{action\ Q}{action\ Q}} \text{ (B.1.1.19)},$$

$\mathrm{Rprt}\ \underset{action\ Q}{\overset{A}{action\ Q}}$ (B.1.1.20),

and any other possible options of self for (B.1.1.12) etc. Likewise for fuzzy dynamic operator fRprt $\underset{B}{\overset{A}{action\ Q}}$.

Definition B.1.1.3. The dynamic operator (B.1.1.14) we shall call tprR – element or ftprR – element for fuzzy dynamic operator, (B.1.1.15) we shall call trR – element or ftrR – element for fuzzy dynamic operator.

Remark B.1.1.3. $(\underset{D}{\overset{C}{action\ Q}})^{-1}$Rprt - the analogue of ${}^{C}_{D}Sprt$ [17] as a special case of (B.1.1.14), where $action\ Q$ is "contain".

It's allowed to add tprR – elements:

$\underset{D}{\overset{C_1}{action\ P}}\ \mathrm{Rprt} + \underset{D}{\overset{C_2}{action\ P}}\ \mathrm{Rprt} = \underset{D}{\overset{C_1 \cup C_2}{action\ P}}\ \mathrm{Rprt}$ (B.1.1.21),

$\underset{D_1}{\overset{C}{action\ P}}\ \mathrm{Rprt} + \underset{D_2}{\overset{C}{action\ P}}\ \mathrm{Rprt} = \underset{D_1 \cup D_2}{\overset{C}{action\ P}}\ \mathrm{Rprt}$ (B.1.1.22).

Likewise for fuzzy dynamic operator $\underset{D}{\overset{C}{action\ P}}$fRprt.

We consider the following self-type tprR-structures:

$\underset{D}{\overset{D}{action\ P}}\ \mathrm{Rprt}$ (B.1.1.23)

$\underset{d}{\overset{strD}{action\ P}}\ \mathrm{Rprt}$ (B.1.1.23.1),

denote $R_9 fd; Q; D$, $d \subset D$ and d acts Q out from structure of D,

$\underset{strD}{\overset{d}{action\ P}}\ \mathrm{Rprt}$ (B.1.1.23.2),

denote $R_{10} fD; Q; d$, $d \subset D$ and structure of D acts Q out from d,

$$\begin{matrix} action\ Q \\ (action\ Q)^{-1} \text{Rprt (B.1.1.24)} \\ D \end{matrix}$$

$$\begin{matrix} action\ Q \\ (action\ Q)^{-1} \text{Rprt (B.1.1.25)} \\ action\ Q \end{matrix}$$

and any other possible options of self for (B.1.1.14) etc. Likewise for fuzzy dynamic operator $\overset{C}{\underset{D}{action\ P}}$fRprt.

Dynamic Rprt – elements, self-type dynamic Rprt-structures.

We considered Rprt – elements earlier. Here we consider dynamic Rprt – elements. We consider dynamic operator whose elements change over time

$$\overset{C(t)}{\underset{D(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \overset{A(t)}{\underset{B(t)}{action\ Q(t)}}\ \text{(B.1.2.1)},$$

where A(t) acts Q(t) to B(t), D(t) acts P(t) out from C(t) simultaneously; A(t), B(t), C(t), D(t) may be fuzzy with corresponding fuzzy measures; Q(t), P(t) are any *actions* , in particular, fuzzy *actions*. The result of this process will be described by the expression

$$\overset{C(t)}{\underset{D(t)}{action\ P(t)}}\ \text{Rrt(t)}\ \overset{A(t)}{\underset{B(t)}{action\ Q(t)}}\ \text{(B.1.2.2)}.$$

Definition B.1.2.1. The dynamic operator (B.1.2.1) we shall call dynamic Rprt– element of the first type or dynamic fRprt– element of the first type for fuzzy dynamic operator, (B.1.2.2) we shall call dynamic Rrt– element of the first type or dynamic fRrt– element of the first type for fuzzy dynamic operator.

Remark B.1.2.1. $\overset{C(t)}{\underset{D(t)}{action\ P(t)}}$ Rprt(t) $\overset{A(t)}{\underset{B(t)}{action\ Q(t)}}$ - the analogue of ${}^{C(t)}_{D(t)}Sprt(t)^{A(t)}_{B(t)}$ [17] as a special case of (B.1.2.1), where *action* $Q(t)$ is “contain”, *action* $P(t)$ is $Q(t)^{-1}$.

It’s allowed to add dynamic Rprt – elements:

$$\overset{C(t)}{\underset{D(t)}{action\ P(t)}}\ Rprt(t)\ \overset{A_1(t)}{\underset{B(t)}{action\ Q(t)}} + \overset{C(t)}{\underset{D(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \overset{A_2(t)}{\underset{B(t)}{action\ Q(t)}} =$$

$$\overset{C(t)}{\underset{D(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \overset{A_1(t)\cup A_2(t)}{\underset{B(t)}{action\ Q(t)}}\ \text{(B.1.2.2.1)},$$

$$\underset{D(t)}{\overset{C(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \underset{B_1(t)}{\overset{A(t)}{action\ Q(t)}} + \underset{D(t)}{\overset{C(t)}{action\ P(t)}}\ Rprt(t)\ \underset{B_2(t)}{\overset{A(t)}{action\ Q(t)}} = \underset{D(t)}{\overset{C(t)}{action\ P(t)}}\ Rprt(t)\ \underset{B_1(t)\cup B_2(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.2.2)},$$

$$\underset{D(t)}{\overset{C_1(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}} + \underset{D(t)}{\overset{C_2(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}} = \underset{D(t)}{\overset{C_1(t)\cup C_2(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.2.3)},$$

$$\underset{D_1(t)}{\overset{C(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}} + \underset{D_2(t)}{\overset{C(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}} = \underset{D_1(t)\cup D_2(t)}{\overset{C(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.2.4)}.$$

Likewise for fuzzy dynamic operator $\underset{D(t)}{\overset{C(t)}{action\ P(t)}}\ \text{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}$.

We consider the following self-type dynamic Rprt-structures of the first type:

$$\underset{action\ Q(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\text{Rprt(t)}\ \underset{action\ Q(t)}{\overset{action\ Q(t)}{action\ Q(t)}}\ \text{(B.1.2.3)},$$

$$\underset{A(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\text{Dprt(t)}\ \underset{action\ Q(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.4)},$$

$$\underset{A(t)}{\overset{B(t)}{(action\ Q(t))^{-1}}}\text{Drt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.5)},$$

$$\underset{A(t)}{\overset{A(t)}{(action\ Q(t))^{-1}}}\text{Drt(t)}\ \underset{A(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.6)},$$

$$\underset{strA(t)}{\overset{a(t)}{(action\ Q(t))^{-1}}}\text{Drt(t)}\ \underset{a(t)}{\overset{strA(t)}{action\ Q(t)}}\ \text{(B.1.2.6.1)},$$

denote $D_{11}(t) f A(t); Q(t); a(t)$, $a(t)\subset A(t)$ and structure of A(t) acts Q(t) to $a(t)$ and acts Q(t) out from $a(t)$ simultaneously.

$$\underset{a(t)}{\overset{strA(t)}{(action\ Q(t))^{-1}}}\mathrm{Drt(t)}\ \underset{strA(t)}{\overset{a(t)}{action\ Q(t)}}\ \text{(B.1.2.6.2)},$$

denote $D_{12}(t)fa(t); Q(t); A(t),\ a(t) \subset A(t)$ and acts Q(t) to structure of A(t) and acts Q(t) out from structure of A(t) simultaneously.

$$\underset{B(t)}{\overset{B(t)}{(action\ Q(t))^{-1}}}\mathrm{Drt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.7)},$$

$$\underset{A(t)}{\overset{B(t)}{action\ P(t)}}\ \mathrm{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.8)},$$

$$\underset{B(t)}{\overset{B(t)}{action\ P(t)}}\ \mathrm{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.9)},$$

$$\underset{A(t)}{\overset{B(t)}{action\ P(t)}}\ \mathrm{Rprt(t)}\ \underset{B(t)}{\overset{B(t)}{action\ Q(t)}}\ \text{(B.1.2.10)},$$

$$\underset{B(t)}{\overset{B(t)}{action\ P(t)}}\ \mathrm{Rprt(t)}\ \underset{B(t)}{\overset{B(t)}{action\ Q(t)}}\ \text{(B.1.2.11)},$$

and any other possible options of self for (B.1.2.1) etc. Likewise for fuzzy dynamic operator $\underset{D(t)}{\overset{C(t)}{action\ P(t)}}\ \mathrm{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}$.

It can be considered a simpler version of the dynamic operator

$$\mathrm{Rprt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ ,\ \text{(B.1.2.12)}$$

where A(t) acts Q(t) to B(t), the result of this process will be described by the expression

$$\mathrm{Rrt(t)}\ \underset{B(t)}{\overset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.13)},$$

or

$$\underset{D(t)}{\overset{C(t)}{action\ P(t)}}\ \mathrm{Rprt(t)}\ \text{(B.1.2.14)},$$

where D(t) acts Q(t) out from C(t), Q(t) is any *action*, the result of this process will be described by the expression

$$\overset{C(t)}{\underset{D(t)}{action\ P(t)}}\ \mathrm{Rrt(t)}\ \text{(B.1.2.15)},$$

Definition B.1.2.2. The dynamic operator (B.1.2.12) we shall call dynamic Rprt – element of the second type or dynamic fRprt – element of the second type for fuzzy dynamic operator, (B.1.2.13) we shall call dynamic Rrt – element of the second type or dynamic fRrt – element of the second type for fuzzy dynamic operator.

Remark B.1.2.2. Rprt(t) $\overset{A(t)}{\underset{B(t)}{action\ Q(t)}}$ - the analogue of $Sprt(t)^{A(t)}_{B(t)}$ [17] as a special case of (B.1.2.12), where $action\ Q(t)$ is “contain”. In this case

$$Sprt(t)^{action\ Q(t)}_{action\ Q(t)} = \mathrm{Rprt(t)}\ \overset{action\ Q(t)}{\underset{action\ Q(t)}{action\ Q(t)}}$$ – self-containment and unlike usual self has higher level self(contain) sel$f^{\frac{3}{2}}$. That's why self-containment can generate, modify and perform other actions with self-capacities, because they have lower level = self.

It’s allowed to add dynamic Rprt – elements of the second type:

$$\mathrm{Rprt(t)}\ \overset{A_1(t)}{\underset{B(t)}{action\ Q(t)}} + \mathrm{Rprt(t)}\ \overset{A_2(t)}{\underset{B(t)}{action\ Q(t)}} = \mathrm{Rprt(t)}\ \overset{A_1(t)\cup A_2(t)}{\underset{B(t)}{action\ Q(t)}}\quad \text{(B.1.2.16)},$$

$$\mathrm{Rprt(t)}\ \overset{A(t)}{\underset{B_1(t)}{action\ Q(t)}} + \mathrm{Rprt(t)}\ \overset{A(t)}{\underset{B_2(t)}{action\ Q(t)}} = \mathrm{Rprt(t)}\ \overset{A(t)}{\underset{B_1(t)\cup B_2(t)}{action\ Q(t)}}\quad \text{(B.1.2.17)}.$$

Likewise for fuzzy dynamic operator Rprt(t) $\overset{A(t)}{\underset{B(t)}{action\ Q(t)}}$.

We consider the following self-type dynamic Rprt-structures of the second t type:

$$\mathrm{Rprt(t)}\ \overset{A(t)}{\underset{A(t)}{action\ Q(t)}}\ \text{(B.1.2.18)},$$

$$\mathrm{Rprt(t)}\ \overset{strA(t)}{\underset{a(t)}{action\ Q(t)}}\ \text{(B.1.2.18.1)},$$

denote $R_{13}(t)fA(t);Q(t);a(t)$, $a(t)$ A(t) and structure of A(t) acts Q(t) to $a(t)$,

Rprt(t) $\underset{strA(t)}{\overset{a(t)}{action\ Q(t)}}$ (B.1.2.18.2),

denote $R_{14}(t)fa(t);Q(t);A(t)$, $a(t)$ A(t) and acts Q(t) to structure of A(t),

Rprt(t) $\underset{action\ Q(t)}{\overset{action\ Q(t)}{action\ Q(t)}}$ (B.1.2.19),

Rprt(t) $\underset{action\ Q(t)}{\overset{A(t)}{action\ Q(t)}}$ (B.1.2.20),

and any other possible options of self for (B.1.2.12) etc. Likewise for fuzzy dynamic operator Rprt(t) $\underset{B(t)}{\overset{A(t)}{action\ Q(t)}}$.

Definition B.1.2.3. The dynamic operator (B.1.2.14) we shall call dynamic tprR – element or dynamic ftprR – element for fuzzy dynamic operator, (B.1.2.15) we shall call dynamic trR – element or dynamic ftprR – element for fuzzy dynamic operator.

Remark B.1.2.3. $\underset{D(t)}{\overset{C(t)}{action\ P(t)}}$ Rprt(t) - the analogue of ${}^{C(t)}_{D(t)}Sprt(t)$ [17] as a special case of (B.1.2.14), $action\ P(t)$ is $Q(t)^{-1}$, where $Q(t)$ is "contain".

It's allowed to add dynamic tprR – elements:

$\underset{D(t)}{\overset{C_1(t)}{action\ P(t)}}$ Rprt(t) + $\underset{D(t)}{\overset{C_2(t)}{action\ P(t)}}$ Rprt(t) = $\underset{D(t)}{\overset{C_1(t)\cup C_2(t)}{action\ P(t)}}$ Rprt(t) (B.1.2.21),

$\underset{D_1(t)}{\overset{C(t)}{action\ P(t)}}$ Rprt(t) + $\underset{D_2(t)}{\overset{C(t)}{action\ P(t)}}$ Rprt(t) = $\underset{D_1(t)\cup D_2(t)}{\overset{C(t)}{action\ P(t)}}$ Rprt(t) (B.1.2.22).

Likewise for fuzzy dynamic operator $\underset{D(t)}{\overset{C(t)}{action\ P(t)}}$ Rprt(t).

We consider the following self-type dynamic tprR-structures:

$$\underset{D(t)}{\overset{D(t)}{(action\ Q(t))^{-1}}}\text{Rprt(t)}\ \text{(B.1.2.15)}$$

$$\underset{d(t)}{\overset{strD(t)}{(action\ Q(t))^{-1}}}\text{Rprt(t)}\ \text{(B.1.2.15.1)},$$

denote $R_{15}(t) f d(t); Q(t); D(t)$, $d(t) \subset D(t)$ and d(t) acts Q(t) out from structure of D(t),

$$\underset{strD(t)}{\overset{d(t)}{(action\ Q(t))^{-1}}}\text{Rprt(t)}\ \text{(B.1.2.15.2)}$$

denote $R_{16}(t) f D(t); Q(t); d(t)$, $d(t) \subset D(t)$ and structure of D(t) acts Q(t) out from d(t),

$$\underset{D(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\text{Rprt(t)}\ \text{(B.1.2.16)}$$

$$\underset{action\ Q(t)}{\overset{action\ Q(t)}{(action\ Q(t))^{-1}}}\text{Rprt(t)}\ \text{(B.1.2.17)}$$

and any other possible options of self for (B.1.2.14) etc. Likewise for fuzzy dynamic operator $\underset{D(t)}{\overset{C(t)}{action\ P(t)}}$ Rprt(t).

New mathematical structures and operators is carried out with generalization it to any structures with any actions. For example,

$$1)\ \begin{array}{ccc} f_{11} & \dots & f_{1k} \\ \dots & \dots & \dots \\ (q_{j1})^{-1} & \dots & (q_{jk})^{-1} \\ \dots & \dots & \dots \\ f_{l1} & \dots & f_{lk} \end{array}\text{DDprt}\begin{array}{ccc} q_{11} & \dots & q_{1n} \\ & \dots & \\ q_{m1} & \dots & q_{mn} \end{array}\ (*_{\text{B.1.1}}),$$

$f_{ij},\ q_{ij}$ – any objects, actions etc.

$$2)\ \begin{array}{ccc} f_{11} & \dots & f_{1k} \\ \dots & \dots & \dots \\ (q_{j1})^{-1} & \dots & (q_{jk})^{-1} \\ \dots & \dots & \dots \\ f_{l1} & \dots & f_{lk} \end{array}\text{fDDprt}\begin{array}{ccc} q_{11} & \dots & q_{1n} \\ & \dots & \\ q_{m1} & \dots & q_{mn} \end{array}\ (*_{\text{B.1.2}}),$$

$f_{ij},\ q_{ij}$ – any fuzzy objects, fuzzy actions etc.

$$3)\ \begin{array}{cccccccc} & g_{12} & & & w_{11} & w_{12} & w_{1n} & \\ g_{11} & g_{22} & & & \dots & \dots & w_{2n} & \\ (w_{j1})^{-1} & (w_{j2})^{-1} & \begin{array}{c} g_{13} \\ (w_{j3})^{-1} \end{array} & \text{DGprt} & & & \dots & (*_{\text{B.1.3}}), \\ g_{31} & \dots & & & w_{m1} & w_{m2} \dots & \dots \ w_{ml} & \\ & g_{k2} & & & & & w_{sn} & \end{array}$$

$w_{ij},\ g_{ij}$ – any objects, actions etc.

$$4)\ \begin{array}{cccccccc} & g_{12} & & & w_{11} & w_{12} & w_{1n} & \\ g_{11} & g_{22} & & & \dots & \dots & w_{2n} & \\ (w_{j1})^{-1} & (w_{j2})^{-1} & \begin{array}{c} g_{13} \\ (w_{j3})^{-1} \end{array} & \text{fDGprt} & & & \dots & (*_{\text{B.1.4}}), \\ g_{31} & \dots & & & w_{m1} & w_{m2} \dots & \dots \ w_{ml} & \\ & g_{k2} & & & & & w_{sn} & \end{array}$$

$w_{ij},\ g_{ij}$ – any fuzzy objects, fuzzy actions etc.

$$5)\ \begin{array}{ccc} a & b & g \\ c & ASrq(\text{ţ}) & w\ (*_{\text{B.1.5}}), \\ d & q & r \end{array}$$

where $ASrq$ is virtual structure or virtual operator, which can take any form of action; a, c, d, q, r, w, g, b, μ – any objects, actions etc.

$$6)\ \begin{array}{ccc} a & b & g \\ c & fASrq(\text{ţ}) & w\ (*_{\text{B.1.6}}), \\ d & q & r \end{array}$$

where $fASrq$ is fuzzy virtual fuzzy structure or fuzzy virtual operator, which can take any fuzzy form of action; a, c, d, q, r, w, g, b, μ – any fuzzy objects, fuzzy actions etc.

Accordingly, we can consider all sorts of self-structures for 1) – 6). And any other possible structures and operators etc.

B.2 Variable fuzzy hierarchical dynamic fuzzy structures (models) for dynamic, singular, hierarchical fuzzy sets

In contrast to the classical one-attribute fuzzy set theory, where only its contents are taken as a set, we consider a two-attribute fuzzy set theory with a fuzzy set as a fuzzy capacity and separately with its contents. We simply use a convenient form to represent the singularity of a fuzzy set. Articles [1]-[12] use the following methodology for permanent structures:

1. Cancellation of the axiom of regularity

2. 2 attributes for the set: capacity and its content

3. Compression of a set, for example, to a point

4. "turning out" from one another, particularly from a capacity, we pull out another capacity, for example, itself, as its element.

5. The simultaneity of one (compression) and the other ("eversion")

6. Own capacities

7. Qualitatively new programming and Networks.

Here we will consider variable structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable structures (models), for example,

$$\underset{D}{\overset{C}{action P}}\ \mathrm{Rprt(t)}\ \underset{B}{\overset{A}{action\ Q}} = \begin{cases} \underset{D}{\overset{C}{action\ PRprt}}, \quad q_2 \geq t \geq q_1)|\mu_1 \\ (\ \underset{D}{\overset{B}{\mu_7}} ffS^1prt \underset{B}{\overset{A}{\mu_6}} \quad , q_3 \geq\ t > q_2)|\mu_2 \\ \underset{D}{\overset{C}{action P}}Rprt\ \underset{B}{\overset{A}{action\ Q}} \quad , q_4 \geq \quad t > q_3)|\mu_3 \\ (\ Rprt\ \underset{B}{\overset{A}{action\ Q}} \quad , q_5 \geq\ t > q_4)|\mu_4 \\ \underset{D}{\overset{\{\ \}}{action\ PRprt}}, \quad t > q_5)|\mu_5 \\ \dots \end{cases}$$

($*_{\mathrm{B.2.1}}$),

μ_i- measures of fuzziness, i = 1, ..., 5. In particular, $\underset{D}{\overset{B}{\mu_7}} ffS^1prt \underset{B}{\overset{A}{\mu_6}}$ can be interpreted as a fuzzy game: player 1 fuzzy with measures of fuzziness μ_6 fits fuzzy A into fuzzy B, and the other fuzzy with measures of fuzziness μ_7 pushes fuzzy D out of fuzzy B at the same time [10], [15].

In what follows, we will denote variable structure (model) through VR, self-type variable structures (models) through SVR, and oself-type variable structures (models) through OSVR. Singular structures (models) are not confused with structures (models) with singularities. $\underset{D}{\overset{B}{\mu_7}} ffS^1prt \underset{B}{\overset{A}{\mu_6}}$ -2-hierarchical structure: 1-level - elements A, B, C, D; level 2 - connections between them. 2-

Examples: a) discrete variable structure

$$\begin{matrix} a & b & g \\ c & VR & w \\ d & q & r \end{matrix}$$

c) continuous variable structure

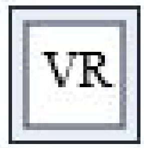


Fig. 0.1

Where a continuous set represents the rim of the Fig.0.1.

We introduce the notation m_{VS_N}– the number of elements, N - the number of connections between them in the discrete variable 2-hierarchical structure VS. We introduce the notation q_{VS_D}– any, D - connections in q_{VS_D} in the variable 2-hierarchical structure VR, in particular, q_{VS_D}, D can be sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the structurability of Q from the interval [0,1], where 0 corresponds to "no structure", and 1 corresponds to the value " structure". Then for joint A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D-self-structure from A*B, cS(x)- the value of self- structure for self- structure x; for dependent structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional structurability of the structure B at the structure A, c(A/B)- conditional structure of the structure A at the structure B. Adding inconsistent structures: c(A+B) =c(A)+ +c(B). The formula of complete structure: c(A)=$\sum_{k=1}^{n} c\,(B_k) * c(A/B_k)$, B_1, $B_{2.}$,.., B_n-full group of hypotheses-actions: $\sum_{k=1}^{n} c\,(B_k)$=1(“structure”). Rprt- structure of the second type for set of fuzzy structures $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$): Rprt $\underset{\substack{action\ Q \\ B}}{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2), \dots, x_n|\mu_{\widetilde{x}}(x_n))}$.

Rprt $\underset{\substack{action\ Q \\ B}}{\{c(x_1)|\mu_{c(\widetilde{x})}c(x_1)|\mu_{c(\widetilde{x})}c(x_2), \dots, c(x_n)|\mu_{c(\widetilde{x})}c(x_n)\}}$ - Rprt- structurability for these structures. It is possible to consider the self-type self-(fuzzy structure) $fR_8f\widetilde{x_w}; Q; \widetilde{x}$, $\widetilde{x_w} \subset$ $\widetilde{x}$. The same for self- structurability $fD_8fC_w(\widetilde{x}); Q; \widetilde{C(x)}$, where $\widetilde{C(x)}$ = $\{ c(x_1)|\mu_{c(\widetilde{x})}c(x_1), c(x_2)|\mu_{c(\widetilde{x})}c(x_2), \dots, c(x_n)|\mu_{c(\widetilde{x})}c\,(x_n)\}$, $C_w(\widetilde{x}) \subset$ $\widetilde{C(x)}$. Can be considered N-hierarchical structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, … can be any in particular, by actions, sets, and others.

$$\overset{C}{\underset{D}{action}}\ P\mathrm{Rprt}\ \overset{A}{\underset{B}{action}}\ Q : \left\langle \begin{matrix} A \to B \\ A, B \end{matrix} \middle| \begin{matrix} D \leftarrow C \\ C, D \end{matrix} \right\rangle \to \begin{pmatrix} fself(A \to B)\mu_{13} \\ A, B \end{pmatrix}$$

$$\overset{C}{\underset{D}{action}}\ P\mathrm{Rprt}\ \overset{A}{\underset{B}{action}}\ Q : \left\langle \begin{matrix} A \to B \\ A, B \end{matrix} \middle| \begin{matrix} D \leftarrow C \\ C, D \end{matrix} \right\rangle \to \begin{pmatrix} foself(D \leftarrow C)\mu_{14} \\ C, D \end{pmatrix}$$

Can be considered discrete hierarchical structure, continuous hierarchical structure, and discrete-continuous hierarchical structure,

$$\mathrm{Rprt}\ \begin{matrix} N - hierarchical\ fuzzy\ structure \\ action\ Q \\ B \end{matrix}$$

The example

fQHR=

$$\mathrm{HRprt}\ \begin{matrix} (Rprt\ \begin{matrix} N - level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix})\mu_N \\ \cdots \\ (Rprt\ \begin{matrix} i - level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix})\mu_i \\ \cdots \\ (Rprt\ \begin{matrix} 1 - level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix})\mu_1 \\ action\ Q \\ B \end{matrix}\ \text{- fuzzy N-hierarchical}$$

fuzzy structure compression into B, μ_i- measures of fuzziness, i = 1, ..., N.

Let rg(N, QHD)= $QHR^{\left.QHR^{QHR^{\cdots QHR}}\right\}\text{-N levels}}$

It can be considered self-QHR, rg(y, QHR) for any y, rg(QHR, QHR).

Compression fuzzy Hierarchy Examples:

$$Rprt \begin{array}{c} () \\ Rprt\ action\ () \\ () \\ () \\ () \\ Rprt\ action\ () \\ () \end{array} = \left(\begin{array}{c} () \\ Rprt\ action\ () \\ () \\ () \\ Rprt\quad () \\ () \\ Rprt\ action\ () \\ () \\ () \\ () \\ Rprt\ action\ () \\ Rprt\quad () \\ () \\ () \end{array} \right)$$

Let's consider two versions: 1) fuzzy containment is interpreted through the concept of fuzzy containment, and 2) fuzzy capacity is interpreted through the concept of fuzzy containment as a rest point of fuzzy containment. Self-(fuzzy containment) is interpreted as a rest point of self-(fuzzy containment). Let A self-(fuzzy compress) into B, D self-(fuzzy displace) from C in $\underset{D}{\overset{C}{\mu_2}} ffVS_1prt\underset{B}{\overset{A}{\mu_1}}$.

We consider the functional ca(Q), which gives a numerical value for the accommodation of fuzzy Q from the interval [0,1], where 0 corresponds to " fuzzy action" and one corresponds to the value " fuzzy result of action". Then for joint fuzzy A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D- self-(fuzzy action) for A*B, caS(x)- the value of self-(fuzzy result of action) for self-(fuzzy action) of x; for dependent fuzzy actions: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional accommodation of the fuzzy action B at the fuzzy action A, ca(A/B)- conditional fuzzy result of action of the fuzzy action A at the fuzzy action B. Adding the fuzzy capacity values of inconsistent fuzzy action s: ca(A+B)=ca(A)+ca(B). The formula of complete fuzzy result of action: ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2, ..., B_n-full group of fuzzy hypotheses- actions: $\sum_{k=1}^{n} ca\,(B_k)$=1("fuzzy result of action"). Rprt-(fuzzy action) for

$\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$): $Rprt^{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\dots,x_n|\mu_{\widetilde{x}}(x_n))}_{\substack{action\ Q\\ w}}$,

$\widetilde{x}$ - fuzzy set of fuzzy actions. $Rprt^{\{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_2),\dots,ca(x_n)|\mu_{ca(\widetilde{x})}ca(x_n)}_{\substack{action\ Q\\ w}}$

- Rprt- accommodation for these fuzzy actions x_i, i = 1, ..., n. It is possible to consider the self-(fuzzy action) $fR_8f\widetilde{x_w};Q;\widetilde{x}$, $\widetilde{x_w}\subset$ $\widetilde{x}$.The same for self-(fuzzy accommodation): $fR_8fCa_w(\widetilde{x});Q;\widetilde{Ca(x)}$, where $Ca_w\left(\widetilde{x}\right) = \{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1), ca(x_2)|\mu_{ca(\widetilde{x})}ca(x_2), \ldots, ca(x_n)|\mu_{ca(\widetilde{x})}ca\left(x_n\right)\}\subset \widetilde{Ca(x)}$.

Consider a variable fuzzy hierarchy (we will denote it by frVH).

We consider the functional h(Q), which gives a numerical value for the hierarchization of fuzzy Q from the interval [0,1], where 0 corresponds to "no fuzzy hierarchy", and 1 corresponds to the value " fuzzy hierarchy". Then for joint fuzzy hierarchies A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- self-(fuzzy hierarchy) from A*B, hS(x)- the value of self-(fuzzy hierarchy) for self-(fuzzy hierarchy) x; for dependent fuzzy hierarchies: h(A*B)=ha(A)*h(B/A)=h(B)*h(A/B), where h(B/A)- conditional hierarchization of the fuzzy hierarchy B at the fuzzy hierarchy A, h(A/B)- conditional fuzzy hierarchy of the fuzzy hierarchy A at the fuzzy hierarchy B. Adding the fuzzy hierarchy values of inconsistent fuzzy hierarchies: h(A+B)=h(A)+h(B). The formula of complete fuzzy hierarchy: h(A)=$\sum_{k=1}^{n} h\left(B_k\right) * h(A/B_k)$, B_1, B_2,..,B_n-full group of fuzzy hypotheses-hierarches: $\sum_{k=1}^{n} h\left(B_k\right)$=1("fuzzy hierarchy").

Rprt- structure for fuzzy set of hierarches $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$):

Rprt $\begin{matrix} (x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2), \ldots, x_n|\mu_{\widetilde{x}}(x_n)) \\ action\ Q \\ B \end{matrix}$.

Rprt $\begin{matrix} \{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1)|\mu_{h(\widetilde{x})}h(x_2), \ldots, h(x_n)|\mu_{h(\widetilde{x})}h(x_n)\} \\ action\ Q \\ B \end{matrix}$ - Rprt- hierarchization for these fuzzy hierarches. It is possible to consider the self-(fuzzy hierarchy) $fR_8f\widetilde{x_w};Q;\widetilde{x},\ \widetilde{x_w}\subset\ \widetilde{x}$. The same for self- hierarchization $fR_8f\widetilde{hx_w};Q;\widetilde{hx}\ ,\ \widetilde{hx_w}\subset \widetilde{hx},\ \widetilde{hx} =$
$\{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1), h(x_2)|\mu_{h(\widetilde{x})}h(x_2), \ldots, h(x_n)|\mu_{h(\widetilde{x})}h\left(x_n\right)\}$. Can be considered Rprt $\begin{matrix} \{\ ca(x), c(x), h(x)\} \\ action\ Q \\ B \end{matrix}$.

Very interesting next fuzzy hierarchy type:

$\begin{matrix} fuzzy\ hierarchy\ A \\ action\ P \\ fuzzy\ hierarchy\ A \end{matrix}$ Rprt $\begin{matrix} fuzzy\ hierarchy\ A \\ action\ Q \\ fuzzy\ hierarchy\ A \end{matrix}$. You can enter special operator fRCprt to work with fuzzy structures:

$\begin{pmatrix} A \\ action\ Q \\ D \end{pmatrix}^{-1}$fRCprt $\begin{matrix} C \\ action\ Q \\ R \end{matrix}$ fuzzy structures R by fuzzy Q with the fuzzy structure from C, expelling fuzzy structure D by fuzzy action P from the fuzzy structure A simultaneously.

Very interesting next fuzzy structure type:

$\left(\overset{A}{\underset{A}{action\ Q}}\right)^{-1}$fRCprt $\overset{A}{\underset{A}{action\ Q}}$,

You can enter special operator fRHprt to work with fuzzy hierarches: $\overset{A}{\underset{B}{action\ P}}$fRHprt $\overset{D}{\underset{R}{action\ Q}}$, fuzzy hierarchizes R by fuzzy Q with the fuzzy hierarchy from D, expelling fuzzy hierarchy B by fuzzy action P from the fuzzy hierarchy A simultaneously.

B.3 FUZZY PROGRAM OPERATORS Rprt, tprR

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions. This must be done through Rprt-Networks in one of the central departments of which a conventional computer system is located. The parallel processor is itself rprogram with direct parallel computing not through serial computing.

Using conventional coding by a computer system, through a Target-block with a Rprt -program operator - Rprt $\overset{Ag}{\underset{activation}{action\ Q}}$, where fuzzy A with measure of fuzziness μ_A fuzzy acts Q with measure of fuzziness μ_Q to fuzzy *activation* with measure of fuzziness $\mu_{activation}$, Q is any fuzzy *action*, it will be possible to obtain the fuzzy execution with measure of fuzziness $\mu_{activation}$ of a parallel fuzzy action A with the desired target weight g or the execution with measure of fuzziness $\mu_{activation}$ of a parallel action A with the desired fuzzy target weight g with measure of fuzziness μ_g or both. Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For Rprt-coding and Rprt-translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018 year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel fprogram operator Rprt $\overset{\{UHF\ AC\ R\}}{\underset{activation}{action\ Q}}$ with the specified measures of fuzziness, we simultaneously enter the desired fuzzy set of codes R with measure of fuzziness μ_R using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel fuzzy frprogram operators:

1) fuzzy Rprt-program operators
2) fuzzy tprR-program operators
3) fuzzy R^1epr - program operators
4) fuzzy $Reprt_1$- program operators

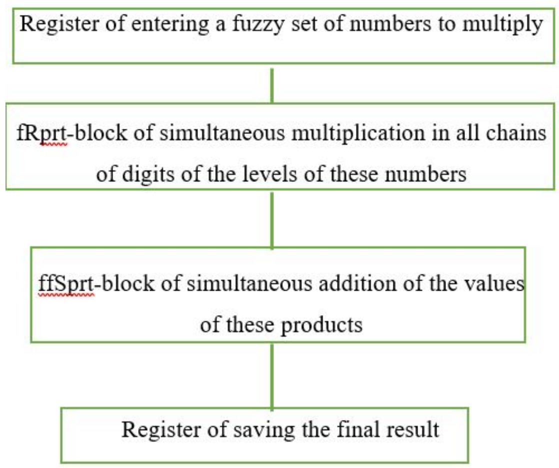


Fig. 0.2 The straightforward functional scheme of the assumed arithmetic-logical device for fRprt-multiplication.

Rprt-algorithm Example:

Simultaneous multiplication Q with measure of fuzziness μ_Q:
$\widetilde{x}$
fRprt$multiplication\, Q$, the notation of the fuzzy set B with elements
$\widetilde{y}$

$b_{i_1 i_2 \ldots i_n j_1 j_2 \ldots j_n} =$

$$(Rprt \underset{(ffSprt \underset{\mu \atop h}{\left\{ y_{1_{j_1}} * \mu_{\widetilde{y}} \left(y_{1_{j_1}} \right), y_{2_{j_2}} * \mu_{\widetilde{y}} \left(y_{2_{j_2}} \right), \ldots, \quad y_{n_{j_m}} * \mu_{\widetilde{y}} \left(y_{n_{j_m}} \right) \} \right\}})_{G \quad V}}{\overset{(ffSprt \underset{\mu \atop q}{\left\{ x_{1_{i_1}} * \mu_{\widetilde{x}} \left(x_{1_{i_1}} \right) *, x_{2_{i_2}} * \mu_{\widetilde{x}} \left(x_{2_{i_2}} \right) *, \ldots, \quad x_{n_{i_n}} * \mu_{\widetilde{x}} \left(x_{n_{i_n}} \right) \right\}})^{R}}{multiplication\ Q}})$$

for any $\{i_1, i_2, \ldots, i_n\}$, $\{j_1, j, \ldots, j_n\}$ *without repetitions*, $q =$ ffSprt $\underset{w}{\overset{K}{\mu}}$, K-set of any $\{k_1*, k_2*, \ldots, k_n*\}$ without repeating them, k_i-any digit, i=1,2,...,n, R= ffSprt $\underset{w}{\overset{\{i_1+, i_2+, \ldots, i_n\}}{\mu}}$, R is the index of the lower discharge, $h =$ ffSprt $\underset{w}{\overset{L}{\mu}}$, L-set of any $\{l_1*, l_2*, \ldots, l_m*\}$ without repeating them, l_i-any digit, i=1,2,...,m, G= ffSprt $\underset{w}{\overset{\{j_1+, j_2+, \ldots, j_m\}}{\mu}}$, G is the index of the lower discharge, V = ffSprt $\underset{w}{\overset{\{i_1+, i_2+, \ldots, i_n + j_1+, j_2+, \ldots, j_m\}}{\mu}}$ (we choose an index on the scale of discharges): see Table I.1.

Remark B.3.1. The algorithm for simultaneously fuzzy multiplication a fuzzy set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by fuzzy multiplying the first number from the fuzzy set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the fuzzy set by the ones following it, etc.

One example is pattern recognition: Rprt $\underset{Name\ of\ q}{\overset{if\ ffSprt \underset{B}{\overset{q}{\mu}} \subset ffSprt \underset{B}{\overset{image\ archive}{\mu}}}{give\ test\ result}}$

The example of Rprt-program is

$$\mathrm{Rprt}\ \left\{ Rprt_{\{g(x)\}}^{\{\{p\}\}} := ,\ Rprtgive\ test\ result_{\ Q \atop {action\ H \atop x}}^{IF\{\{B\}\{f\}\}},\ Rprt\ action\ G_{R}^{R} \right\}.$$

For example, based on Rprt $action\ Q_{\widetilde{x}}^{\widetilde{y}}$, where $\widetilde{y}$=($y_1|\mu_{\widetilde{x}}(y_1)$, $y_2|\mu_{\widetilde{x}}(y_2), \ldots,$ $y_m|\mu_{\widetilde{x}}(y_m)$)$\subset$ $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$), we can consider self-type Rprt-structure - $fR_8f\widetilde{y};Q;\widetilde{x}$ with m elements from $\widetilde{x}$, m<n, which is formed according to the form(1.1) , or in forms (1.1.1) - (1.1.5), generalizing it. Structures more complex than $fR_8f\widetilde{y};Q;\widetilde{x}$ can be introduced. For example, through the forms that generalizes (1.1): (1.2), (1.3) - (1.4). In particular, in (1), A is fuzzy compressed (fuzzy fits) in C in the fuzzy compression fuzzy structure B in C (i.e., in the fuzzy structure fRprtμ_{C}^{B}) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc. (1.3.1) - (1.3.4) schematically interpret the fuzzy formation of fuzzy fRprt-self structure through a pseudo 3-connected form with a 2-connected form. The ideology of fRprt and $fR_8f\widetilde{y};Q;\widetilde{x}$ can be used for programming.

Remark B.1.2. Fuzzy self, in particular, according to a fuzzy form is fuzzy analogue of the form of type (1.1) -(1.4) [1]: (2.1*).

Here are some of the fuzzy Rprt-program operators.

1. Simultaneous fuzzy *action* Q of the expressions $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2), \ldots,$ $p_n|\mu_{\widetilde{p}}(p_n)$) to the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$). This is implemented via Rprt $Q_{\{\widetilde{x}\}}^{\widetilde{p}}$.
2. Simultaneous R = fuzzy checking with fuzziness μ by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2), \ldots,$ $B_n|\mu_{\widetilde{B}}(B_n)$). Implemented via Rprt$R_{\widetilde{Q}}^{\widetilde{B}}$, where $\widetilde{Q}$ can be anything.
3. Similarly for fuzzy loop operators and others.

$fR_8f\!-\!$ fuzzy software operators will differ only just because aggregates $\widetilde{x},\widetilde{p},\ \widetilde{B},\widetilde{g}$ will be formed from corresponding frprt-program operators in form

(1.1), for more complex operators in forms (1.1,1) - (1.4), (2.1*) and analogs of forms (1.1,1) - (1.4) by type (2.1*).

For example, $\mathrm{Rprt}\underset{R}{\overset{S}{g\{R,S\}}}$ is the fuzzy Rprt-self structure with measure of fuzziness µ of the second type if $g\{R,S\}$ is a frprogram capable of fuzzy generating R *with measure of fuzziness* μ from S.

The example of self-frprogram of the first type is

$$\mathrm{Rprt}\underset{\underset{w}{action\ R}}{\left\{ Rprt\underset{\{\widetilde{x}\}}{\overset{\widetilde{p}}{Q}},\ Rprt\underset{\widetilde{Q}}{\overset{\widetilde{B}}{R}},\ Rprt\underset{Q}{\overset{Q}{Q}} \right\}}.$$

The example of Rprt-program operators for Rmnrprt- fuzzy analogue of SmnSt [1-3]:

1) $Rprt\underset{\{\widetilde{x}\}}{\overset{\widetilde{p}}{Q}}$ - fuzzy *action* Q of $\widetilde{p}$ to $\widetilde{x}$.
2) $\mathrm{Rprt}\underset{g}{\overset{tw}{P}}$, where P - fuzzy assigning target weight *tw* to fuzzy g with measure of fuzziness µ.
3) $\mathrm{Rprt}\underset{Rmnrprt\ activation}{\overset{\{q\}\,w}{S}}$, where S -Rmnrprt *activation* for fuzzy $\{q\}\,w$ with measure of fuzziness µ.

Rprt-coding.

Rprt-coding with measure of fuzziness µ: 1) fuzzy set A to fuzzy set B, 2) fuzzy set A to a point q, where the elements of the fuzzy sets A, B can be continuous. For example, $\mathrm{Rprt}\underset{B}{\overset{A}{Q}}$, where Q - Rprt-coding.

There are Rprt-coding, Rprt-translation, Rprt-realize of frprograms and fprograms from the archives without extraction theirs

Relf-coding.

Relf-coding with measure of fuzziness µ: 1) fuzzy set A to set fuzzy A, i.e. fuzzy A on itself 2) fuzzy set A to a point q in form (1), where the elements of the fuzzy sets A, B can be continuous. For example, $\mathrm{Rprt}\overset{A}{\underset{A}{Q}}$,

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with Rprt-Networks, it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through Rprt-Networks and series-parallel. Codes from a conventional type computer system will be used via Rprt -connectors in Rprt - coding, for example: $\mathrm{Rprt}\overset{\{UHF\ AC\}}{\underset{activation}{Q}}$. UHF AC field activation is used.

Dynamic Rprt and $fR_8(t)f$ programming.

The ideology of dynamic Rprt and $fR(t)f$ can be used for programming:

1. Simultaneous fuzzy *action* $\widetilde{Q(t)}$ of the expressions $\widetilde{p(t)}$=($\mathrm{p_1(t)}|\mu_{\widetilde{p(t)}}(\mathrm{p_1(t)})$, $\mathrm{p_2(t)}|\mu_{\widetilde{p(t)}}(\mathrm{p_2(t)}), \ldots,$ $\mathrm{p_n(t)}|\mu_{\widetilde{p(t)}}(\mathrm{p_n(t)})$) to the variables $\widetilde{x(t)}$=($\mathrm{x_1(t)}|\mu_{\widetilde{x(t)}}(\mathrm{x_1(t)})$, $\mathrm{x_2(t)}|\mu_{\widetilde{x(t)}}(\mathrm{x_2(t)}), \ldots,$ $\mathrm{x_n(t)}|\mu_{\widetilde{x(t)}}(\mathrm{x_n(t)})$). This is implemented via $\mathrm{Rprt}(t)\overset{\widetilde{p(t)}}{\underset{\left\{\widetilde{x(t)}\right\}}{\widetilde{Q(t)}}}$.
2. Simultaneous $\widetilde{R(t)}$ is fuzzy checking with fuzziness μ by the fuzzy set of conditions $\widetilde{g(t)}$=($\mathrm{g_1(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_1(t)})$, $\mathrm{g_2(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_2(t)}), \ldots,$ $\mathrm{g_n(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_n(t)})$) for the fuzzy set of expressions $\widetilde{B(t)}$=($\mathrm{B_1(t)}|\mu_{\widetilde{B(t)}}(\mathrm{B_1(t)})$, $\mathrm{B_2(t)}|\mu_{\widetilde{B(t)}}(\mathrm{B_2(t)}), \ldots,$ $\mathrm{B_n(t)}|\mu_{\widetilde{B(t)}}(\mathrm{B_n(t)})$). Implemented via $\mathrm{Rprt}(t)\overset{\widetilde{B(t)}}{\underset{\widetilde{Q(t)}}{\widetilde{R(t)}}}$, where $\widetilde{Q}$ can be anything.
3. Similarly for fuzzy loop operators and others.

$fR_8(t)f$ - fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}, \widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding Rprt-program operators in form (1.1), for more complex operators in forms (1.1,1) - (1.4), (2.1*) and analogs of forms (1.1,1) - (1.4) by type (2.1*).

tprR-program operators.

The ideology of tprR and Rf - fuzzy analogues of tS and $t_{S_4 f}$ from [14] can be used for programming. Here are some of the tprR-program operators.

1. Simultaneous expelling fuzzy *action* Q of the expressions $\widetilde{p}$=(p$_1|\mu_{\widetilde{p}}$(p$_1$), p$_2|\mu_{\widetilde{p}}$(p$_2$), …, p$_n|\mu_{\widetilde{p}}$(p$_n$)) from the variables $\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$), …, x$_n|\mu_{\widetilde{x}}$(x$_n$)). This is implemented via $\begin{matrix}\widetilde{x}\\ Q\\ \{\widetilde{p}\}\end{matrix}$ Rprt.

2. Simultaneous expelling R = fuzzy checking with fuzziness μ by the fuzzy set of conditions $\widetilde{g}$=(g$_1|\mu_{\widetilde{g}}$(g$_1$), g$_2|\mu_{\widetilde{g}}$(g$_2$), …, g$_n|\mu_{\widetilde{g}}$(g$_n$)) for the fuzzy set of expressions $\widetilde{B}$=(B$_1|\mu_{\widetilde{B}}$(B$_1$), B$_2|\mu_{\widetilde{B}}$(B$_2$), …, B$_n|\mu_{\widetilde{B}}$(B$_n$)). It's implemented through $\begin{matrix}\widetilde{Q}\\ R\\ \widetilde{B}\end{matrix}$ Rprt, where $\widetilde{Q}$ can be anything.

3. Similarly for loop operators and others.

$fR_{16}f$ – fuzzy software operators will differ only just because aggregates $\widetilde{x},\widetilde{p},\ \widetilde{B},\widetilde{g}$ will be formed from corresponding tprR program operators in form (1.1), for more complex operators in forms (1.1,1) - (1.4), (2.1*) and analogs of forms (1.1,1) - (1.4) by type (2.1*).

Consider hierarchical tprR-program operator

$$\begin{matrix}B\\ (action\ Q)^{-1}Rprt\\ A\end{matrix} = \left\{\begin{matrix} & \{\ \} & \\ D+ & \mu & ffSprt\\ & A-A\cap B & \\ & (\ B-A\cap B) & \end{matrix}\right\}, \text{ where D is oself-(fuzzy set)}$$

for fuzzy $(A \cap B)$, where action Q- contain.

Dynamic tprR and $R_{16}(t)f$ programming at time q.

The ideology of tprR and $R_{16}f$ can be used for dynamic programming. Here are some of the tprR- dynamic programming operators.

1. The process of simultaneous expelling fuzzy *action* $Q(t)$ of the expressions $\widetilde{p(t)}$=(p$_1$(t)$|\mu_{\widetilde{p(t)}}$(p$_1$(t)), p$_2$(t)$|\mu_{\widetilde{p(t)}}$(p$_2$(t)), …, p$_n$(t)$|\mu_{\widetilde{p(t)}}$(p$_n$(t))) from the variables $\widetilde{x(t)}$=(x$_1$(t)$|\mu_{\widetilde{x(t)}}$(x$_1$(t)), x$_2$(t)$|\mu_{\widetilde{x(t)}}$(x$_2$(t)), …, x$_n$(t)$|\mu_{\widetilde{x(t)}}$(x$_n$(t))). This is implemented via $\begin{matrix}\widetilde{x(t)}\\ Q(t)\\ \{\widetilde{p(t)}\}\end{matrix}$ Rprt(t).

2. The process of simultaneous expelling R(t) = fuzzy checking with fuzziness $\mu(t)$ by the fuzzy set of conditions $\widetilde{g(t)}$=(g$_1$(t)$|\mu_{\widetilde{g(t)}}$(g$_1$(t)),

$g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t)), \ldots,$ $g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t)))$ for the fuzzy set of expressions $\widetilde{B(t)}$=($B_1(t)|\mu_{\widetilde{B(t)}}(B_1(t))$, $B_2(t)|\mu_{\widetilde{B(t)}}(B_2(t)), \ldots,$ $B_n(t)|\mu_{\widetilde{B(t)}}(B_n(t)))$ is implemented through $\begin{matrix} \widetilde{Q(t)} \\ R(t) \\ \widetilde{B(t)} \end{matrix}$Rprt(t), where $\widetilde{Q(t)}$ can be anything.

3.Similarly for loop operators and others.

$\underline{R_{16}(t) f}$ – $\underline{\text{fuzzy}}$ software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}$, $\widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding processes tprR(t) for above mentioned programming operators through form (1.1), for more complex operators in forms (1.1,1) - (1.4), (2.1*) and analogs of forms (1.1,1) - (1.4) by type (2.1*).

Consider hierarchical dynamic tprR-program operator:

$$\begin{matrix} B(q) \\ (action\ Q)^{-1} Rprt(q) = \\ A(q) \end{matrix} \left\{ fft(q)_{S_1 f(A(q)\cap B(q))} + \begin{matrix} \{\ \} \\ \mu \\ A(q) - A(q)\cap B(q) \\ \big(B(q) - A(q)\cap B(q)\big) \end{matrix} ffSprt(q) \right\},$$

where action Q- contain.

R^1epr -program operators (form $\begin{matrix} B \\ (action\ Q)^{-1} \\ D \end{matrix}$$R^1$prt $\begin{matrix} A \\ action\ Q \\ B \end{matrix}$ - fuzzy analogue of ${}^{B}_{D}S^1t^{A}_{B}$ [10]))

For example, $\begin{matrix} \widetilde{x} \\ R^{-1} \\ \{\widetilde{p}\} \end{matrix}$$R^1prt\begin{matrix} \widetilde{B} \\ R \\ \widetilde{Q} \end{matrix}$, where simultaneous expelling fuzzy checking with fuzziness μ by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2), \ldots,$ $p_n|\mu_{\widetilde{p}}(p_n)$) from the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$) and simultaneous R = fuzzy checking with fuzziness μ by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2), \ldots,$ $B_n|\mu_{\widetilde{B}}(B_n)$), $\widetilde{Q}$ can be anything.

The examples:

$\left(\underset{A}{\overset{A}{action\ Q}}\right)^{-1}\mathrm{R}^1\mathrm{prt}\underset{A}{\overset{A}{action\ Q}}$ can be interpreted as $\begin{pmatrix} s^1elf \\ os^1elf \end{pmatrix}$- R^1epr-program operator. $\left(\underset{A}{\overset{A}{action\ Q}}\right)^{-1}\mathrm{R}^1\mathrm{prt}\underset{A}{\overset{A}{action\ Q}}$ sample $\begin{pmatrix} s^1elf \\ os^1elf \end{pmatrix}$-frprogram structure example.

Consider hierarchical dynamic R^1epr-program operator: (form $\frac{\left(\underset{A}{\overset{B}{action\ Q}}\right)^{-1}R^1prt\ \underset{B}{\overset{A}{action\ Q}}*}{\left(\underset{D-A}{\overset{B}{action\ Q}}\right)^{-1}R^1prt\ \underset{B}{\overset{A}{action\ Q}}}$).

Reprt_1- program operators (form $\left(\underset{D}{\overset{C}{action\ Q}}\right)^{-1}\mathrm{R}_1\mathrm{prt}\underset{B}{\overset{A}{action\ Q}}$ - fuzzy analogue of ${}_D^CS_1t_B^A$ [13])

$\left(\underset{A}{\overset{A}{action\ Q}}\right)^{-1}\mathrm{R}_1\mathrm{prt}\underset{A}{\overset{A}{action\ Q}}$ - sample $\begin{pmatrix} r_1self \\ r_1oself \end{pmatrix}$-frprogram structure example.

frprogram structure example, where the assemblage point d_r is the cursor, it is quite complex self—frprogram:

$$\mathrm{fR_1prt}\left\{ {}^{q}_{W_q}\left(\left(\underset{A}{\overset{A}{action\ Q}}\right)^{-1}{}_{fR_1prt}\underset{A}{\overset{A}{action\ Q}} \right) \overset{E_q}{fR_1prt}_{q}\underset{\substack{Q\\V}}{\left(\left(\underset{A}{\overset{A}{action\ Q}}\right)^{-1}{}_{fR_1prt}\underset{A}{\overset{A}{action\ Q}} - \right)},\quad fR_1prt\underset{d_{r}\left(E_{in}l^{d_r}\right)}{\overset{\{E^{ex}l^{d_r}\}}{action\ Q}} \right\}$$

.

$$\mathrm{fR_1prt}\left\{ {}^{q}_{W_q}\left(\overset{A}{\underset{A}{(action\ Q)}}^{-1}{}_{fR_1prt}\overset{A}{\underset{A}{action\ Q}}\right) fR_1prt^{E_q}_{q\left(\overset{A}{\underset{A}{(action\ Q)}}^{-1}{}_{fR_1prt}\overset{A}{\underset{A}{action\ Q}} - \right)_{\substack{Q\\V}}}, \ fR_1prt \overset{\{E^{ex}l^{d_r}\}}{\underset{d_{r\left(E_{in}l^{d_r}\right)}}{action\ Q}} \right\}$$

can be interpreted as a fRpt program operator.

Appendix

Remark. Energy of a living organism:

$\mathrm{fr_1g}(\mathrm{r},\ \mathrm{a}(E_q)) =$

$$\mathrm{fR_1prt}\left\{ {}^{q}_{W_q}\left(\overset{A}{\underset{A}{(action\ Q)}}^{-1}{}_{fR_1prt}\overset{A}{\underset{A}{action\ Q}}\right) fD_1prt^{E_q}_{q\left(\overset{A}{\underset{A}{(action\ Q)}}^{-1}{}_{fR_1prt}\overset{A}{\underset{A}{action\ Q}} - \right)_{\substack{Q\\V}}}, \ fR_1prt \overset{\{E^{ex}l^{d_r}\}}{\underset{d_{r\left(E_{in}l^{d_r}\right)}}{action\ Q}} \right\}$$

$(**_{\mathrm{B.4}})$

Energy of a living organism of a person:

$$\mathrm{fR_1prt}\left\{ {}^{q}_{W_q}\left(\overset{A}{\underset{A}{(action\ Q)}}^{-1}{}_{fR_1prt}\overset{A}{\underset{A}{action\ Q}}\right) fR_1prt^{E_q}_{q\left(\overset{A}{\underset{A}{(action\ Q)}}^{-1}{}_{fR_1prt}\overset{A}{\underset{A}{action\ Q}} - \right)_{\substack{Q\\V}}}, \ fR_1prt \overset{\{E^{ex}l^{d_r}\}}{\underset{d_{r\left(self\left(E_{in}l^{d_r}\right)\right)}}{action\ Q}} \right\}$$

$(***_{\mathrm{B.4}})$

$\overset{A}{\underset{A}{(action\ Q)}}^{-1} fR_1prt\overset{A}{\underset{A}{action\ Q}}$ -internal energy of a living organism, q- a gap in the energy cocoon of a living organism, r-the position of the assemblage point $\mathrm{d_r}$ on the energy cocoon of a living organism, $\mathrm{W_q}$- energy prominences from the gap in the cocoon of a living organism, E_q-external energy entering the gap in the cocoon of a living organism, $E^{ex}l^{d_r}$ - a bundle of fibers of external energy self-capacities from outside the cocoon, collected at the point of assembly of the cocoon of a living organism, $E_{in}l^{d_r}$- a bundle of fibers of external energy self-capacities from inside the cocoon, collected at the point of assembly of the cocoon of a living organism in the same position r of the assemblage point d_r. d_r is the subject of identifying the energy fibers of the subtle energy of the Universe in position r both outside and inside the cocoon.

($**_{B.4}$), ($***_{B.4}$) can be interpreted as the program operators.

Entire neural network as instantaneous simultaneous ffRAM in ffSprt-elements and fself- elements.$fself^{fself^{\cdots^{fself}}}$, $ff1\downarrow I \uparrow_{-1}^{1} ff_2^{\;ff1\downarrow I\uparrow_{-1}^{1}ff_2^{\;\cdots^{ff1\downarrow I\uparrow_{-1}^{1}ff_2}}}$, $fsin\infty^{fsin\infty^{\cdots^{fsin\infty}}}$.When activated in a neural network, the entire neural network becomes a working memory. Use of fself-energy as fuzzy activation or from outside. fdQ_0= Rprt $\begin{matrix} & ffS_{mnSprt} \\ ffSprt & \mu \\ & activation \\ & Q \\ & ffS_{mnSprt} \\ ffSprt & \mu \\ & activation \end{matrix}$ $\rightarrow$ self-frRAM, frQ_{00}= $\begin{matrix} R_{mnSprt} & \\ D & Rprt \\ activation & \\ Q & \\ R_{mnSprt} & \\ R & Rprt \\ activation & \end{matrix}$ frQ_0prt, frQ_{01}= $\begin{matrix} & R_{mnSprt} \\ Rprt & C \\ & activation \\ & R_{mnSprt} \\ Rprt & H \\ & activation \end{matrix}$ frQ_0.

frQ_0,frQ_{00},frQ_{01}-coding,translation,realization in freprograms, use frQ_0,frQ_{00},frQ_{01}-frS_{mnSprt},frAssembler.

Appendix

Let us introduce the following notations:

A*B= Sprt_B^A, A^2 = Self A =Srt_A^A, $A^{\frac{3}{2}}$ = Rrt $\begin{matrix} A \\ A \\ A \end{matrix}$ = sel$f^{\frac{3}{2}}$ (A), A^3 = $Self^2$A, $\ldots$,$A^{\frac{3n}{2}}$ = Rrt $\begin{matrix} A^n \\ A^n \\ A^n \end{matrix}$ = sel$f^{\frac{3n}{2}}$ (A), A^{n+1} = $Self^n$A, sel$f^{min(n,m)}$(A) $Srt_{A^m}^{A^n}$ = sel$f^{\frac{n}{m}}$ (A), sel$f^{min(n,m,k)}$(A) Rrt $\begin{matrix} A^n \\ A^m \\ A^k \end{matrix}$ = sel$f^{\frac{n+m+k}{2k}}$ (A), $\ldots$ etc.

There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A + B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1 + A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

You can consider a more “hard” option: A*B=$PSprt_B^A$, where $PSprt_B^A$ – operator, containing A in every element of B, A^2 = PSelf A =$PSrt_A^A$, A^3 = PSelf^2A, …, A^{n+1} = PSelf^nA, $\mathrm{PSel}f^{min(n,m)}$(A) $PSrt_{A^m}^{A^n}$ = $\mathrm{PSel}f^{\frac{n}{m}}$ (A), …etc. There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A + B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1 + A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

Let's introduce $\sqrt{self}$ as the result of the decision of the equation Srt_x^x = self, that is x = $\sqrt{self}$, $\sqrt[3]{self}$ as the result of the decision of the equation Rprt $\overset{x}{\underset{x}{x}}$ = self, that is x = $\sqrt[3]{self}$, $\sqrt[n]{self^m}$ as the result of the decision of the equation $x^{\frac{n}{m}}$ = self, $self^{\alpha}$ as the result of the decision of the equation $x^{\frac{1}{\alpha}}$ = self, where is any number , in particular, a negative number etc. The following equality is true:

$self^{-\alpha}(self^{\alpha}G)$ =$self^{\alpha}(self^{-\alpha}G)$ = G. In this way one can introduce self-level space.

B.4 Rprt-networks

A. Galushkin's comprehensive monograph [6] covers all aspects of networks, but traditional approaches go through classical mathematics, mainly through the usual correspondence operators. Here we consider a different approach - through a new mathematical process with containment operators, which, although they can be interpreted as the result of some correspondence operators, are not themselves correspondence operators. Containment operators are more convenient for networks. Also, the main emphasis was placed on using processors operating using triodes, which are generally not used in Rprt-networks. Rprt networks (SmnRprt) are a Rprt structure that can be built for the required weights, the implementation of which will be carried out using a short-pulse laser to generate attosecond pulses of light. Rprt-OS (Rprt operating system) uses Rprt-coding and Rprt-translation. In the first one, coding is carried out through a 2-dimensional matrix-row (a, b), where the number b is the code of the action, and the number a is the code of the object of this action. Rprt-coding (or self-coding) is implemented through a matrix consisting of 2 columns (in the continuous case, two intervals of numbers). Here, the source encoding is used for all matrix rows simultaneously. Rprt-translation is carried out by inversion. In this case, self-type coding and self- type translation by (B.1.1.6) or (B.1.1.11), (B.1.1.18) will be more stable. The set of the target weights f = (f_1, f_2, …, f_n) in Rprt $action\ \overset{\{fx\}}{\underset{D}{Q}}$ are chosen for necessary tasks using a short-pulse laser to generate attosecond pulses of light to accomplish them, x= $(x_1, x_2, \ldots, x_n)$, Q there is a containment operation there is a containment operation. We will not touch on the issues of applications, or network optimization. They are described in detail by A. Galushkin [6]. We will touch on the difference of this only for hierarchical complex networks. The same simple executing programs are in the cores of simple artificial neurons of type Rprt (designation - mnRprt) for simple information processing. More complex executing programs are used for mnRprt nodes. Rprt-threshold element $-\underset{\{qy\}}{\overset{b}{action\ Q^{-1}}}\mathrm{Rprt}(t)\ \underset{b}{\overset{\{ax\}}{action\ Q}}$, Q there is a containment operation there is a containment operation, b- artificial neurons of type Rprt (designation - mnRprt), x= $(x_1, x_2, \ldots, x_n)$ are the values of the initial signals, $a=(a_1,a_2,\ldots,a_n)$ are the weights of Rprt-synapses and the values of the output signals The first level of mnRprt consists of simple mnRprt. The second level of mnRprt consists of Rprt $\underset{D}{\overset{\{mnRprt\}}{action\ Q}}$ – Rprt-node of mnRprt in range D, D- capacity for mnRprt node. The third level of mnRprt consists of

$$\mathrm{Rprt}^{\{Rprt\ \underset{D}{\overset{\{mnRprt\}}{action\ Q}}\}}_{\underset{D}{action\ Q}}$$

- Rprt^2- node of mnRprt in range D, thus D becomes capacity of itself in itself as an element for mnRprt. For our networks, it is sufficient to use Rprt^2- nodes of mnRprt, but self-level is higher in living organisms, particularly Rprt^n-, n3. The target structure or the corresponding program enters the target unit using a short-pulse laser to generate attosecond pulses of light. After that, all networks or parts of them are activated according to the indicative goal. It may appear that we are leaving the network ideology, but these networks are a complex hierarchy of different levels, like living organisms.

Remark B.4.1. Traditional scientific approaches through classical mathematics make it possible to describe only at the usual energy level. Here we consider an approach that makes describing processes with finer energies possible. mnRprt contains

$\mathrm{Rprt}\ \underset{mnRprt}{\overset{\{reprogram\ \}}{action\ Q}}$, $reprogram-$executing program in Rprt-OS. Rprt-OS (or Self-type of Rprt OS) is based on Rprt-assembly language (or Self-type of Rprt assembly language), which is based on assembly language through Rprt-approach in turn, if the base of elements of Rprt-networks is sufficient. The reprograms are in Rprt-programming environments (or Self-type of Rprt programming environments), but this question and Rprt-networks base will be considered in the following articles. In particular, reprograms may contain Rprt- programming operators. In mnRprt cores, the constant memory Rprt with correspondent reprograms depending on mnRprt.

The OS (operating system) and the principles and modes of operation of the Rprt-networks for this programming are interesting. But this is already the material for the next publications.

Here is developed a helicopter model without a main and tail rotors based on Rprt – physics and special neural networks with artificial neurons operating in normal and Rprt-modes. Let's denote this model through SmnRprt. To do this, it’s proposed to use mnRprt of different levels; for example, for the usual mode, mnRprt serves for the initial processing of signals and the transfer of information to the second level, etc., to the nodal center, then checked. In case of an anomaly - local Rprt–mode with the desired "target weight" is realized in this section, etc., to the center. In the case of a monster during the test, SmnRprt is activated with the desired "target weight" using a short-pulse laser to generate attosecond pulses of light. Here are realized other tasks also. To reach the self-energy level, the mode Rprt $\underset{SmnRprt}{\overset{SmnRprt}{action\ Q}}$, is used. In normal mode, it’s planned to carry out the movement of SmnRprt on jet propulsion by converting the energy of the emitted gases into a vortex to obtain additional thrust upwards. For this purpose, a spiral-shaped chute (with "pockets") is arranged at the bottom of the SmnRprt for the gases emitted by the jet engine, which first exit through a straight chute connected to the spiral one. There is drainage of exhaust gases outside the SmnRprt. SmnRprt is represented by a neural network that extends from the center of one of the main clusters of Rprt - artificial neurons to the shell, turning into the body itself. Above the operator's cabin is the central core of the neural network and the target block, responsible for performing the "target weights" and auxiliary blocks, the functions and roles of which we will discuss further. Next is the space for the movement of the local vortex. The unit responsible for SmnRprt's actions is below the operator's cab. In Rprt – mode, the entire network or its sections are Rprt – activated to perform specific tasks, in particular, with "target weights" using a short-pulse laser to generate attosecond pulses of light. In the target, block used Rprt -coding, Rprt-translation for activation of all networks to "target weights" simultaneously, then –the reset of this Rprt-coding after activation using a short-pulse laser to generate attosecond pulses of light.

Unfortunately, triodes are not suitable for Rprt -neural networks. In the most primitive case, usual separators with corresponding resistances and cores for reprograms may be used instead triodes since there is no necessity to unbend the alternating current to direct. The Rprt-operative memory belt is disposed around a central core of SmnRprt. There are Rprt-coding, Rprt-translation, and Rprt-realize of reprograms and the programs from the archives without extraction, Rprt-coding and Rprt-translation may be used in high-intensity, ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna, Strickland. Rprt – structure or an reprogram if one is present of needed «target weight» are taken in target block at Rprt – activation of the networks. Rprt $\underset{activation}{\overset{SmnRprt, f}{action\ Q}}$ derives SmnRprt to the self-level boundary with target weight f. Activation of the entire network is implemented to perform “target weights” using a short-pulse laser to generate attosecond pulses of light.

You can also try to use higher frequency alternating current and ultraviolet light, which can work with Rprt– structures in Rprt–modes by its nature to activate the networks or some of its parts in Rprt–modes and locally using Rprt–mode to perform local tasks. Above high frequently alternating current go through mercury bearers. That’s why overheating does not occur.

Remark B.4.2. Hypothesis 1: equations for real processes in a non-trivial form can be used to fully or partially interpret the self-level of the process, replacing the equal signs with identification signs, and solutions to these equations as a manifestation of this level on the level of objectivity and ordinary energies. That is, equations for real processes serve as a definition of the self-level of the process, the definition of self-values (self-characteristics) of the process through the identification sign, i.e., they are defined (expressed) through themselves. In particular, forms (1.1) - (1.4) can be used as forms of identification. Each such singularity creates its own field, the process, the object etc. Much more effective than science for working with these singularities will be special Dynamic programming, which we are currently working on to create. If we represent an amorphous body with a mathematical structure of self-object $Sprt^{A_0+E_s}_{A_0+E_s}$, where $Sprt^{A_0}_{A_0}$ - level of objectivity of an amorphous object, ($Sprt^{A_0}_{A_0+E_s}+ Sprt^{A_0+E_s}_{A_0}$) - the energy of connections between the level of subtle energy $Sprt^{E_s}_{E_s}$ and the level of objectivity.

Thus, one can try to conventionally represent the mathematical model of the energy structure of an amorphous object as a hierarchical dynamic operator

$$\left(Sprt^{A_0}_{A_0+E_s} + \overset{Sprt^{E_s}_{E_s}}{\underset{Sprt^{A_0}_{A_0}}{Sprt^{A_0+E_s}_{A_0}}} \right) \quad (B.4.1)$$

Identification at the lower levels of a hierarchical dynamic structure of type (B.4.1) will lead to the upper level. Let us denote the upper level of A by $\overline{A}$, the upper level of P by $\overline{P}$. Then singularity $\overline{A} \rightarrow \overline{P}$ is the setting for the transformation of A into P. The field of the given structure tw is used for the activation of networks. The field can remain in effect until it is executed tw. Here all stages of the structure tw can be executed directly in parallel, in particular, an algorithm for solving the desired problem. We will call this field the operational activation field. This field will be created according to the structure tw. The pulse structure of a short-pulse laser for generating attosecond light pulses is close to $(a \uparrow I \downarrow^{a}) \uparrow I \downarrow (\downarrow_{a} \downarrow I^{\uparrow a})$, i.e., type ${}^{a}_{a}Sprt^{a}_{a}$, and upon activation it will be induction of same type self, which is necessary for the formation of a local assembly point d_r of external energy fibers El^{d_r}. Its locality (position of the assembly point r) will be determined by the structure of the magnetic induction of the short-pulse laser pulse for generating attosecond light generation through Targetblock SmnSprt [7]. Execution tw will be achieved through setting the assemblage point in the desired position r_1 to engage the appropriate external energy: $Sprt^{{}^{r}_{d_r}\ Sprt^{d_r}_{r1}}_{{}^{r}_{d_r}\ Sprt^{d_r}_{r1}}$.

C Lprt – elements, self-type Lprt-structures and their applications

C.1 Lprt – elements, self-type Lprt-structures.

We consider dynamic operator

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ Lprt } \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.1.1.1)},$$

where $\overbrace{A}$, $\overbrace{R}$ - upper levels of A and R respectively, $\overline{Q}$, $\overline{P}$ - average levels of Q and P respectively, B goes to the middle level of Q - $\overline{Q}$, $\overline{Q}$ goes to the upper level of A - $\overbrace{A}$, $\overbrace{R}$ goes to the middle level of P - $\overline{P}$, $\overline{P}$ goes to the lower level of C simultaneously. The result of this process will be described by the expression

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ Lrt } \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.1.1.2)}.$$

Definition C.1.1.1. The dynamic operator (C.1.1.1) we shall call Lprt – element of the first type, (C.1.1.2) we shall call Lrt – element of the first type.

Remark C.1.1.1 an consider Lprt – elements use the Banach space.

It's allowed to add Lprt – elements:

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ Lprt } \begin{array}{c} \overbrace{A_1} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} + \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ Lprt } \begin{array}{c} \overbrace{A_2} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} = \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ Lprt } \begin{array}{c} \overbrace{A_1} \cup \overbrace{A_2} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.1.1.2.1)},$$

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \; Lprt \; \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B_1 \end{array} + \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \; Lprt \; \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B_2 \end{array} = \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \; Lprt \; \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B_1 \cup B_2 \end{array} \quad \text{(C.1.1.2.2)},$$

$$\begin{array}{c} C_1\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} + \begin{array}{c} C_2\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} = \begin{array}{c} C_1\cup C_2\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} \quad (\text{C.1.1.2.3}),$$

$$\begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R_1}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} + \begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R_2}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} = \begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R_1}\cup\overbrace{R_2}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} \quad (\text{C.1.1.2.4}),$$

$$\begin{array}{c} C\\ \uparrow\\ \overline{P_1}\\ \uparrow\\ \overbrace{R}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} + \begin{array}{c} C\\ \uparrow\\ \overline{P_2}\\ \uparrow\\ \overbrace{R}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} = \begin{array}{c} C\\ \uparrow\\ (\overline{P_1}\cup\overline{P_2})\\ \uparrow\\ \overbrace{R}\end{array} \text{Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} \quad (\text{C.1.1.2.5}),$$

$$\begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q_1}\\ \uparrow\\ B\end{array} + \begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q_2}\\ \uparrow\\ B\end{array} = \begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ (\overline{Q_1}\cup\overline{Q_2})\\ \uparrow\\ B\end{array} (\text{C.1.1.2.6}).$$

We consider the following self-type Lprt-structure of the first type:

$$\begin{array}{c} Q\\ \uparrow\\ \overline{Q}\\ \uparrow\\ \overbrace{Q}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{Q}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ Q\end{array} \quad (\text{C.1.1.3}),$$

denote $L_1 f Q$.

$$\begin{array}{c} Q\\ \uparrow\\ \overline{Q}\\ \uparrow\\ \overbrace{A}\end{array} \text{ Lprt } \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ Q\end{array} \quad (\text{C.1.1.4}),$$

denote $L_2 f A; Q$.

$$\begin{matrix} B \\ \frac{\uparrow}{Q} \\ \uparrow \\ \overbrace{A} \end{matrix} \ \text{Lprt} \ \begin{matrix} \overbrace{A} \\ \frac{\uparrow}{Q} \\ \uparrow \\ B \end{matrix} \quad (\text{C.1.1.5}),$$

denote $L_3 f A; Q; B$.

$$\begin{matrix} A \\ \frac{\uparrow}{Q} \\ \uparrow \\ \overbrace{A} \end{matrix} \ \text{Lprt} \ \begin{matrix} \overbrace{A} \\ \frac{\uparrow}{Q} \\ \uparrow \\ A \end{matrix} \quad (\text{C.1.1.6}),$$

denote $L_4 f A; Q$.

$$\begin{matrix} a \\ \frac{\uparrow}{Q} \\ \uparrow \\ \overbrace{A} \end{matrix} \ \text{Lprt} \ \begin{matrix} str\ \overbrace{A} \\ \frac{\uparrow}{Q} \\ \uparrow \\ a \end{matrix} \quad (\text{C.1.1.6.1}),$$

denote $L_5 f A; Q; a$, $a \subset A$,

$$\begin{matrix} strA \\ \frac{\uparrow}{Q} \\ \uparrow \\ \overbrace{a} \end{matrix} \ \text{Lprt} \ \begin{matrix} \overbrace{a} \\ \frac{\uparrow}{Q} \\ \uparrow \\ strA \end{matrix} \quad (\text{C.1.1.6.2}),$$

denote $L_6 f a; Q; A$, $a \subset A$,

$$\begin{matrix} B \\ \frac{\uparrow}{Q} \\ \uparrow \\ \overbrace{B} \end{matrix} \ \text{Lprt} \ \begin{matrix} \overbrace{A} \\ \frac{\uparrow}{Q} \\ \uparrow \\ B \end{matrix} \quad (\text{C.1.1.7}),$$

and any other possible options of self for (C.1.1.1) etc.

It can be considered a simpler version of the dynamic operator

$$\text{Lprt} \ \begin{matrix} \overbrace{A} \\ \frac{\uparrow}{Q} \\ \uparrow \\ B \end{matrix} \quad (\text{C.1.1.8})$$

where $\overbrace{A}$ - upper levels of A, $\overline{Q}$ - average levels of Q, B goes to the middle level of Q - $\overline{Q}$, $\overline{Q}$ goes to the upper level of A - $\overbrace{A}$ simultaneously, the result of this process will be described by the expression

$$\text{Lrt}\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array}\ \ (\text{C.1.1.9})$$

or

$$\begin{array}{c}C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array}\ \text{Lprt}\ (\text{C.1.1.10})$$

where $\overbrace{R}$ - upper levels of R, $\overline{P}$ - average levels of P, $\overbrace{R}$ goes to the middle level of P - $\overline{P}$, $\overline{P}$ goes to the lower level of C simultaneously, the result of this process will be described by the expression

$$\begin{array}{c}C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array}\ \text{Lrt}\ (\text{C.1.1.11})$$

Definition 1,2. The dynamic operator (C.1.1.8) we shall call Lprt – element of the second type, (C.1.1.9) we shall call Lrt – element of the second type.

It's allowed to add Lprt – elements of the second type:

$$\text{Lprt}\ \begin{array}{c}\overbrace{A_1}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array}\ +\text{Lprt}\ \begin{array}{c}\overbrace{A_2}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array}\ =\text{Lprt}\ \begin{array}{c}\overbrace{A_1}\cup\overbrace{A_2}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array}\qquad (\text{C.1.1.12}),$$

$$Lprt\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B_1\end{array}\ +Lprt\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B_2\end{array}\ =Lprt\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B_1\cup B_2\end{array}\quad (\text{C.1.1.13}),$$

$$\text{Lprt}\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q_1}\\ \uparrow\\ B\end{array} + \text{Lprt}\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q_2}\\ \uparrow\\ B\end{array} = \text{Lprt}\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ (\overline{Q_1}\cup\overline{Q_2})\\ \uparrow\\ B\end{array}\ (\text{C.1.1.13.1}),$$

We consider the following self-type Lprt-structures of the second type:

$$\text{Lprt}\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ A\end{array}\quad (\text{C.1.1.14}),$$

$$\text{Lprt}\ \begin{array}{c}str\ \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ a\end{array}\quad (\text{C.1.1.14.1}),$$

denote $L_7 f A; Q; a$, $a \subset A$,

$$\text{Lprt}\ \begin{array}{c}a\\ \uparrow\\ \overline{Q}\\ \uparrow\\ strA\end{array}\quad (\text{C.1.1.15}),$$

denote $L_8 f a; Q; A$, $a \subset A$,

$$\text{Lprt}\ \begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ Q\end{array}\quad (\text{C.1.1.16}),$$

and any other possible options of self for (C.1.1.8) etc.

Definition C.1.1.3. The dynamic operator (C.1.1.10) we shall call tprL – element, (C.1.1.11) we shall call trL – element.

It's allowed to add tprL – elements:

$$\begin{array}{c}C_1\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{Lprt} + \begin{array}{c}C_2\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{Lprt} = \begin{array}{c}C_1\cup C_2\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array}\ \text{Lprt}\ (\text{C.1.1.17}),$$

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R_1} \end{array} \text{Lprt} + \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R_2} \end{array} \text{Lprt} = \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R_1} \cup \overbrace{R_2} \end{array} \text{Lprt (C.1.1.18)},$$

$$\begin{array}{c} C \\ \uparrow \\ \overline{P_1} \\ \uparrow \\ \overbrace{R} \end{array} \text{Lprt} + \begin{array}{c} C \\ \uparrow \\ \overline{P_2} \\ \uparrow \\ \overbrace{R} \end{array} \text{Lprt} = \begin{array}{c} C \\ \uparrow \\ (\overline{P_1} \cup \overline{P_2}) \\ \uparrow \\ \overbrace{R} \end{array} \text{Lprt (C.1.1.18.1)}.$$

We consider the following self-type tprL-structures:

$$\begin{array}{c} R \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{Lprt (C.1.1.19)}$$

$$\begin{array}{c} strD \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{d} \end{array} \text{Lprt (C.1.1.19.1)},$$

denote $L_9 fd; Q; D$, $d \subset D$,

$$\begin{array}{c} d \\ \uparrow \\ \overline{P} \\ \uparrow \\ str\overbrace{D} \end{array} \text{Lprt (C.1.1.20)},$$

denote $L_{10} fD; Q; d$, $d \subset D$,

$$\begin{array}{c} P \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{Lprt (C.1.1.21)}$$

and any other possible options of self for (C.1.1.10) etc.

Dynamic Lprt – elements, self-type dynamic Lprt-structures.

We considered Lprt – elements earlier. Here we consider dynamic Lprt – elements. We consider dynamic operator whose elements change over time

$$\begin{array}{c} C(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array} \text{ (2.1),}$$

where $\overbrace{A(t)}$, $\overbrace{R(t)}$ - upper levels of A(t) and R(t) respectively, $\overline{Q(t)}$, $\overline{P(t)}$ - average levels of Q(t) and P(t) respectively, B(t) goes to the middle level of Q(t) - $\overline{Q(t)}$, $\overline{Q(t)}$ goes to the upper level of A(t) - $\overbrace{A(t)}$, $\overbrace{R(t)}$ goes to the middle level of P(t) - $\overline{P(t)}$, $\overline{P(t)}$ goes to the lower level of C(t) simultaneously. The result of this process will be described by the expression

$$\begin{array}{c} C(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} \text{Lrt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array} \text{ (2.2).}$$

Definition C.1.2.1. The dynamic operator (C.1.2.1) we shall call dynamic Lprt – element of the first type, (C.1.2.2) we shall call dynamic Lrt – element of the first type.

It's allowed to add dynamic Lprt – elements:

$$\begin{array}{c} C(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A_1(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array} + \begin{array}{c} C(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A_2(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array} = \begin{array}{c} C(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A_1(t)} \cup \overbrace{A_2(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}$$
(C.1.2.2.1),

$$\begin{array}{c} C(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B_1(t)\end{array} + \begin{array}{c} C(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B_2(t)\end{array} = \begin{array}{c} C(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B_1(t) \cup B_2(t)\end{array}$$
(C.1.2.2.2),

$$\begin{array}{c} C_1(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array} + \begin{array}{c} C_2(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array} = \begin{array}{c} C_1(t) \cup C_2(t)\\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}$$
(C.1.2.2.3),

$$\begin{array}{c} C(t)\\ \uparrow\\ \overline{P(t)}\\ \uparrow\\ \overbrace{R_1(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t) \end{array} + \begin{array}{c} C(t)\\ \uparrow\\ \overline{P(t)}\\ \uparrow\\ \overbrace{R_2(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t) \end{array} = \begin{array}{c} C(t)\\ \uparrow\\ \overline{P(t)}\\ \uparrow\\ \overbrace{R_1(t)} \cup \overbrace{R_2(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t) \end{array}$$
(C.1.2.2.4),

$$\begin{array}{c} C(t)\\ \uparrow\\ \overline{P_1(t)}\\ \uparrow\\ \overbrace{R(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t) \end{array} + \begin{array}{c} C(t)\\ \uparrow\\ \overline{P_2(t)}\\ \uparrow\\ \overbrace{R(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t) \end{array} = \begin{array}{c} C(t)\\ \uparrow\\ \overline{P_1(t)} \cup \overline{P_2(t)}\\ \uparrow\\ \overbrace{R(t)} \end{array} Lprt(t) \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t) \end{array}$$
(C.1.2.2.5),

$$\begin{array}{c} C(t)\\ \uparrow\\ \overline{P(t)}\\ \uparrow\\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q_1(t)}\\ \uparrow\\ B(t) \end{array} + \begin{array}{c} C(t)\\ \uparrow\\ \overline{P(t)}\\ \uparrow\\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q_2(t)}\\ \uparrow\\ B(t) \end{array} = \begin{array}{c} C(t)\\ \uparrow\\ \overline{P(t)}\\ \uparrow\\ \overbrace{R(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q_1(t)} \cup \overline{Q_2(t)}\\ \uparrow\\ B(t) \end{array}$$
(C.1.2.2.6).

We consider the following self-type dynamic Lprt-structures of the first type:

$$\begin{array}{c} Q(t)\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ \overbrace{Q(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{Q(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ Q(t) \end{array} \text{ (C.1.2.3),}$$

$$\begin{array}{c} Q(t)\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ \overbrace{A(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ Q(t) \end{array} \text{ (C.1.2.4),}$$

$$\begin{array}{c} B(t)\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ \overbrace{A(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t) \end{array} \text{ (C.1.2.5),}$$

$$\begin{array}{c} A(t) \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ \widehat{A(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \widehat{A(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ A(t) \end{array} \text{ (C.1.2.6),}$$

$$\begin{array}{c} a(t) \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ str\,\widehat{A(t)} \end{array} \quad \text{Lprt(t)} \quad \begin{array}{c} str\,\widehat{A(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ a(t) \end{array} \quad \text{(C.1.2.6.1),}$$

denote $L_{11}(t) f A(t); Q(t); a(t),\ a(t) \subset A(t)$,

$$\begin{array}{c} strA(t) \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ \widehat{a(t)} \end{array} \quad \text{Lprt(t)} \quad \begin{array}{c} \widehat{a(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ strA(t) \end{array} \quad \text{(C.1.2.6.2),}$$

denote $L_{12}(t) f a(t); Q(t); A(t),\ a(t) \subset A(t)$,

$$\begin{array}{c} B(t) \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ \widehat{B(t)} \end{array} \text{Lprt(t)} \begin{array}{c} \widehat{A(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B(t) \end{array} \text{ (C.1.2.7),}$$

and any other possible options of self for (C.1.2.1) etc.

It can be considered a simpler version of the dynamic operator

$$\text{Lprt(t)} \begin{array}{c} \widehat{A(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B(t) \end{array} \text{ (C.1.2.8)}$$

where $\widehat{A(t)}$ - upper levels of A(t), $\overline{Q(t)}$- average levels of Q(t), B(t) goes to the middle level of Q(t) - $\overline{Q(t)}$, $\overline{Q(t)}$ goes to the upper level of A(t) - $\widehat{A(t)}$ simultaneously, the result of this process will be described by the expression

$$\mathrm{Lrt(t)}\ \begin{array}{c}\overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}\ \text{(C.1.2.9),}$$

or

$$\begin{array}{c}C(t)\\ \uparrow\\ \overline{P(t)}\\ \uparrow\\ \overbrace{R(t)}\end{array}\mathrm{Lprt(t)}\ \text{(C.1.2.10),}$$

where $\overbrace{R(t)}$ - upper levels of R(t), $\overline{P(t)}$ - average levels of P(t), $\overbrace{R(t)}$ goes to the middle level of P(t) - $\overline{P(t)}$, $\overline{P(t)}$ goes to the lower level of C(t) simultaneously, the result of this process will be described by the expression

$$\begin{array}{c}C(t)\\ \uparrow\\ \overline{P(t)}\\ \uparrow\\ \overbrace{R(t)}\end{array}\mathrm{Lrt(t)}\ \text{(C.1.2.11),}$$

Definition C.1.2.2. The dynamic operator (C.1.2.8) we shall call dynamic Lprt – element of the second type, (C.1.2.9) we shall call dynamic Lrt – element of the second type.

It's allowed to add dynamic Lprt – elements of the second type:

$$\mathrm{Lprt(t)}\ \begin{array}{c}\overbrace{A_1(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}\ +\mathrm{Lprt(t)}\ \begin{array}{c}\overbrace{A_2(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}\ = \mathrm{Lprt(t)}\ \begin{array}{c}\overbrace{A_1(t)}\cup\overbrace{A_2(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}\ \text{(C.1.2.12),}$$

$$\mathrm{Lprt(t)}\ \begin{array}{c}\overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B_1(t)\end{array}\ + Lprt(t)\ \begin{array}{c}\overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B_2(t)\end{array}\ = \mathrm{Lprt(t)}\ \begin{array}{c}\overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B_1(t)\cup B_2(t)\end{array}\ \text{(C.1.2.13),}$$

$$\text{Lprt(t)}\ \frac{\overbrace{A(t)}\atop\uparrow}{Q_1(t)}{\atop{\uparrow\atop B(t)}} + \text{Lprt(t)}\ \frac{\overbrace{A(t)}\atop\uparrow}{Q_2(t)}{\atop{\uparrow\atop B(t)}} = \text{Lprt(t)}\ \frac{\overbrace{A(t)}\atop\uparrow}{\overline{Q_1(t)}\cup\overline{Q_2(t)}}{\atop{\uparrow\atop B(t)}}\ \text{(C.1.2.13.1).}$$

We consider the following self-type dynamic Lprt-structures of the second t type:

$$\text{Lprt(t)}\ \frac{\overbrace{A(t)}\atop\uparrow}{Q(t)}{\atop{\uparrow\atop A(t)}}\ \text{(C.1.2.14),}$$

$$\text{Lprt(t)}\ \frac{str\,\overbrace{A(t)}\atop\uparrow}{Q(t)}{\atop{\uparrow\atop a(t)}}\ \text{(C.1.2.14.1),}$$

denote $L_{13}(t) f A(t); Q(t); a(t),\ a(t) \subset A(t)$,

$$\text{Lprt(t)}\ \frac{\overbrace{a(t)}\atop\uparrow}{Q(t)}{\atop{\uparrow\atop str A(t)}}\ \text{(C.1.2.15),}$$

denote $L_{14}(t) f a(t); Q(t); A(t),\ \ a(t) \subset A(t)$,

$$\text{Lprt(t)}\ \frac{\overbrace{A(t)}\atop\uparrow}{Q(t)}{\atop{\uparrow\atop Q(t)}}\ \text{(C.1.2.16),}$$

and any other possible options of self for (C.1.2.8) etc.

Definition C.1.2.3. The dynamic operator (C.1.2.10) we shall call dynamic tprL – element, (C.1.2.11) we shall call dynamic trL – element.

It's allowed to add dynamic tprL – elements:

$$\frac{\overset{C_1(t)}{\uparrow}}{\underset{\overbrace{R(t)}}{\overline{P(t)}\uparrow}} Lprt(t) + \frac{\overset{C_2(t)}{\uparrow}}{\underset{\overbrace{R(t)}}{\overline{P(t)}\uparrow}} Lprt(t) = \frac{\overset{C_1(t)\cup C_2(t)}{\uparrow}}{\underset{\overbrace{R(t)}}{\overline{P(t)}\uparrow}} Lprt(t) \quad \text{(C.1.2.17)},$$

$$\frac{\overset{C(t)}{\uparrow}}{\underset{\overbrace{R_1(t)}}{\overline{P(t)}\uparrow}} Lprt(t) + \frac{\overset{C(t)}{\uparrow}}{\underset{\overbrace{R_2(t)}}{\overline{P(t)}\uparrow}} Lprt(t) = \frac{\overset{C(t)}{\uparrow}}{\underset{\overbrace{R_1(t)}\cup\overbrace{R_2(t)}}{\overline{P(t)}\uparrow}} Lprt(t) \quad \text{(C.1.2.18)},$$

$$\frac{\overset{C(t)}{\uparrow}}{\underset{\overbrace{R(t)}}{\overline{P_1(t)}\uparrow}} Lprt(t) + \frac{\overset{C(t)}{\uparrow}}{\underset{\overbrace{R(t)}}{\overline{P_2(t)}\uparrow}} Lprt(t) = \frac{\overset{C(t)}{\uparrow}}{\underset{\overbrace{R(t)}}{\overline{P_1(t)}\cup\overline{P_2(t)}\uparrow}} Lprt(t) \quad \text{(C.1.2.18.1)}.$$

We consider the following self-type dynamic tprL-structures:

$$\frac{\overset{R(t)}{\uparrow}}{\underset{\overbrace{R(t)}}{\overline{P(t)}\uparrow}} \text{Lprt(t)} \quad \text{(C.1.2.19)}$$

$$\frac{\overset{strD(t)}{\uparrow}}{\underset{\overbrace{d(t)}}{\overline{P(t)}\uparrow}} \text{Lprt(t)} \quad \text{(C.1.2.19.1)},$$

denote $L_{15}(t)fd(t);\overline{P(t)};D(t)$, $d(t)\subset D(t)$,

$$\frac{\overset{d(t)}{\uparrow}}{\underset{str\,\overbrace{D(t)}}{\overline{P(t)}\uparrow}} \text{Lprt(t)} \quad \text{(C.1.2.20)}$$

denote $L_{16}(t)fD(t);\overline{P(t)};d(t)$, $d(t)\subset D(t)$,

$$
\begin{array}{l}
P(t) \\
\uparrow \\
\overline{P(t)}\,\mathrm{Lprt(t)}\ (\mathrm{C.1.2.21}) \\
\uparrow \\
\overbrace{R(t)}
\end{array}
$$

and any other possible options of self for (C.1.2.10) etc.

New mathematical structures and operators is carried out with generalization it to any structures with any actions. For example,

$$
1)\ \begin{array}{ccc} f_{11} & \dots & f_{1k} \\ \dots & \dots & \dots \\ (q_{j1})^{-1} & \dots & (q_{jk})^{-1} \\ \dots & \dots & \dots \\ f_{l1} & \dots & f_{lk} \end{array} \mathrm{LLprt} \begin{array}{ccc} q_{11} & \dots & q_{1n} \\ & \dots & \\ q_{m1} & \dots & q_{mn} \end{array} \quad (*_{\mathrm{C.1}}),
$$

$f_{ij},\ q_{ij}$ – any objects, actions etc.

$$
2)\ \begin{array}{c} g_{11} \\ (w_{j1})^{-1} \\ g_{31} \end{array} \begin{array}{c} g_{12} \\ g_{22} \\ (w_{j2})^{-1} \\ \dots \\ g_{k2} \end{array} \begin{array}{c} g_{13} \\ (w_{j3})^{-1} \end{array} \ \mathrm{LGprt} \begin{array}{ccccc} w_{11} & w_{12} & & w_{1n} & \\ \dots & \dots & & w_{2n} & \\ & & & \dots & \\ w_{m1} & w_{m2} & \dots & \dots & w_{ml} \\ & & & w_{sn} & \end{array} \quad (*_{\mathrm{C.1.1}}),
$$

$w_{ij},\ g_{ij}$ – any objects, actions etc.

3)

$$
\begin{array}{ccc} a & b & g \\ c & ALrq(ţ) & w \\ d & q & r \end{array} \ (*_{\mathrm{C.1.2}}),
$$

where $ALrq$ is virtual structure or virtual operator, which can take any form of action; a, c, d, q, r, w, g, b, μ – any objects, actions etc.

Accordingly, we can consider all sorts of self-type structures for 1) – 3). And any other possible structures and operators etc.

C.2 FLprt – elements, self-type FLprt-structures

We consider fuzzy dynamic operator

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ FLprt } \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.2.1.1)},$$

where $\overbrace{A}$, $\overbrace{R}$ - upper levels of fuzzy A and fuzzy R respectively, $\overline{Q}$, $\overline{P}$ - average levels of fuzzy Q and fuzzy P respectively, fuzzy B goes to the middle level of fuzzy Q - $\overline{Q}$, $\overline{Q}$ goes to the upper level of fuzzy A - $\overbrace{A}$, $\overbrace{R}$ goes to the middle level of fuzzy P - $\overline{P}$, $\overline{P}$ goes to the lower level of fuzzy C simultaneously. The result of this process will be described by the expression

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \ F\text{Lrt } \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.2.1.2)}.$$

Definition C.2.1.1. The fuzzy dynamic operator (C.2.1.1) we shall call FLprt – element of the first type, (C.2.1.2) we shall call FLrt – element of the first type.

Remark C.2.1.1 an consider FLprt – elements use the Banach space.

It's allowed to add FLprt – elements:

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ FLprt } \begin{array}{c} \overbrace{A_1} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} + \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ FLprt } \begin{array}{c} \overbrace{A_2} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} = \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ FLprt } \begin{array}{c} \overbrace{A_1} \cup \overbrace{A_2} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.2.1.2.1)},$$

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \ FLprt \ \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B_1 \end{array} + \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \ FLprt \ \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B_2 \end{array} = \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \ FLprt \ \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B_1 \cup B_2 \end{array} \quad \text{(C.2.1.2.2)},$$

$$\begin{array}{c} C_1 \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ FLprt } \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} + \begin{array}{c} C_2 \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ FLprt } \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} = \begin{array}{c} C_1 \cup C_2 \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \text{ FLprt } \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.2.1.2.3)},$$

$$\begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R_1}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} + \begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R_2}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} = \begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R_1}\cup\overbrace{R_2}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} \quad \text{(C.2.1.2.4)},$$

$$\begin{array}{c} C\\ \uparrow\\ \overline{P_1}\\ \uparrow\\ \overbrace{R}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} + \begin{array}{c} C\\ \uparrow\\ \overline{P_2}\\ \uparrow\\ \overbrace{R}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} = \begin{array}{c} C\\ \uparrow\\ (\overline{P_1}\cup\overline{P_2})\\ \uparrow\\ \overbrace{R}\end{array} \text{FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} \quad \text{(C.2.1.2.5)},$$

$$\begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q_1}\\ \uparrow\\ B\end{array} + \begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q_2}\\ \uparrow\\ B\end{array} = \begin{array}{c} C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ (\overline{Q_1}\cup\overline{Q_2})\\ \uparrow\\ B\end{array} \text{(C.2.1.2.6)}.$$

We consider the following self-type FLprt-structure of the first type:

$$\begin{array}{c} Q\\ \uparrow\\ \overline{Q}\\ \uparrow\\ \overbrace{Q}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{Q}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ Q\end{array} \quad \text{(C.2.1.3)},$$

denote FL_1fQ.

$$\begin{array}{c} Q\\ \uparrow\\ \overline{Q}\\ \uparrow\\ \overbrace{A}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ Q\end{array} \quad \text{(C.2.1.4)},$$

denote $FL_2fA;Q$.

$$\begin{array}{c} B\\ \uparrow\\ \overline{Q}\\ \uparrow\\ \overbrace{A}\end{array} \text{ FLprt } \begin{array}{c} \overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array} \quad \text{(C.2.1.5)},$$

denote $FL_3fA;Q;B$.

$$\begin{array}{c} A \\ \uparrow \\ \overline{Q} \\ \uparrow \\ \overbrace{A} \end{array} \ \text{FLprt} \ \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ A \end{array} \quad \text{(C.2.1.6)},$$

denote $FL_4 f A; Q$.

$$\begin{array}{c} a \\ \uparrow \\ \overline{Q} \\ \uparrow \\ \overbrace{A} \end{array} \ \text{FLprt} \ \begin{array}{c} str\overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ a \end{array} \quad \text{(C.2.1.6.1)},$$

denote $FL_5 f A; Q; a,\ a \subset A$,

$$\begin{array}{c} strA \\ \uparrow \\ \overline{Q} \\ \uparrow \\ \overbrace{a} \end{array} \ \text{FLprt} \ \begin{array}{c} \overbrace{a} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ strA \end{array} \quad \text{(C.2.1.6.2)},$$

denote $FL_6 f a; Q; A,\ a \subset A$,

$$\begin{array}{c} B \\ \uparrow \\ \overline{Q} \\ \uparrow \\ \overbrace{B} \end{array} \ \text{FLprt} \ \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.2.1.7)},$$

and any other possible options of self for (C.2.1.1) etc.

It can be considered a simpler version of the fuzzy dynamic operator

$$\text{FLprt} \ \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} \quad \text{(C.2.1.8)}$$

where $\overbrace{A}$ - upper levels of fuzzy A, $\overline{Q}$ - average levels of fuzzy Q, fuzzy B goes to the middle level of fuzzy Q - $\overline{Q}$, $\overline{Q}$ goes to the upper level of fuzzy A - $\overbrace{A}$ simultaneously, the result of this process will be described by the expression

$$\text{FLrt}\ \begin{array}{c}\overbrace{A}\\ \uparrow \\ \overline{Q}\\ \uparrow \\ B\end{array}\ \text{(C.2.1.9)}$$

or

$$\begin{array}{c}C\\ \uparrow \\ \overline{P}\\ \uparrow \\ \overbrace{R}\end{array}\ \text{FLprt (C.2.1.10)}$$

where $\overbrace{R}$ - upper levels of fuzzy R, $\overline{P}$ - average levels of fuzzy P, $\overbrace{R}$ goes to the middle level of fuzzy P - $\overline{P}$, $\overline{P}$ goes to the lower level of fuzzy C simultaneously, the result of this process will be described by the expression

$$\begin{array}{c}C\\ \uparrow \\ \overline{P}\\ \uparrow \\ \overbrace{R}\end{array}\ \text{FLrt (C.2.1.11)}$$

Definition 1,2. The fuzzy dynamic operator (C.2.1.8) we shall call FLprt – element of the second type, (C.2.1.9) we shall call FLrt – element of the second type.

It's allowed to add FLprt – elements of the second type:

$$\text{FLprt}\ \begin{array}{c}\overbrace{A_1}\\ \uparrow \\ \overline{Q}\\ \uparrow \\ B\end{array}\ +\text{FLprt}\ \begin{array}{c}\overbrace{A_2}\\ \uparrow \\ \overline{Q}\\ \uparrow \\ B\end{array}\ = \text{FLprt}\ \begin{array}{c}\overbrace{A_1}\cup\overbrace{A_2}\\ \uparrow \\ \overline{Q}\\ \uparrow \\ B\end{array}\quad \text{(C.2.1.12)},$$

$$FLprt\ \begin{array}{c}\overbrace{A}\\ \uparrow \\ \overline{Q}\\ \uparrow \\ B_1\end{array}\ + FLprt\ \begin{array}{c}\overbrace{A}\\ \uparrow \\ \overline{Q}\\ \uparrow \\ B_2\end{array}\ = FLprt\ \begin{array}{c}\overbrace{A}\\ \uparrow \\ \overline{Q}\\ \uparrow \\ B_1\cup B_2\end{array}\quad \text{(C.2.1.13)},$$

$$\text{FLprt}\ \begin{array}{c}\overbrace{A}\\ \uparrow \\ \overline{Q_1}\\ \uparrow \\ B\end{array}\ + \text{FLprt}\ \begin{array}{c}\overbrace{A}\\ \uparrow \\ \overline{Q_2}\\ \uparrow \\ B\end{array}\ = \text{FLprt}\ \begin{array}{c}\overbrace{A}\\ \uparrow \\ (\overline{Q_1}\cup\overline{Q_2})\\ \uparrow \\ B\end{array}\ \text{(C.2.1.13.1)},$$

We consider the following self-type FLprt-structures of the second type:

$$\text{FLprt}\ \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ A \end{array}\ \ (\text{C.2.1.14}),$$

$$\text{FLprt}\ \begin{array}{c} str\ \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ a \end{array}\ \ (\text{C.2.1.14.1}),$$

denote $FL_7 f A; Q; a$, $a \subset A$,

$$\text{FLprt}\ \begin{array}{c} a \\ \uparrow \\ \overline{Q} \\ \uparrow \\ strA \end{array}\ \ (\text{C.2.1.15}),$$

denote $FL_8 f a; Q; A$, $a \subset A$,

$$\text{FLprt}\ \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ Q \end{array}\ \ (\text{C.2.1.16}),$$

and any other possible options of self for (C.2.1.8) etc.

Definition C.2.1.3. The fuzzy dynamic operator (C.2.1.10) we shall call tprFL – element, (C.2.1.11) we shall call trFL – element.

It's allowed to add tprFL – elements:

$$\begin{array}{c} C_1 \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array}\ \text{FLprt} + \begin{array}{c} C_2 \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array}\ \text{FLprt} = \begin{array}{c} C_1 \cup C_2 \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array}\ \text{FLprt}\ (\text{C.2.1.17}),$$

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R_1} \end{array}\ \text{FLprt} + \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R_2} \end{array}\ \text{FLprt} = \begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R_1} \cup \overbrace{R_2} \end{array}\ \text{FLprt}\ (\text{C.2.1.18}),$$

$$\underbrace{\overset{\overset{C}{\uparrow}}{\overline{P_1}}}_{\overset{\uparrow}{R}}\ \text{FLprt} + \underbrace{\overset{\overset{C}{\uparrow}}{\overline{P_2}}}_{\overset{\uparrow}{R}}\ \text{FLprt} = (\underbrace{\overset{\overset{C}{\uparrow}}{\overline{P_1} \cup \overline{P_2}}}_{\overset{\uparrow}{R}})\text{FLprt (C.2.1.18.1).}$$

We consider the following self-type tprFL-structures:

$$\underbrace{\overset{\overset{R}{\uparrow}}{\overline{P}}}_{\overset{\uparrow}{R}}\ \text{FLprt (C.2.1.19)}$$

$$\underbrace{\overset{\overset{strD}{\uparrow}}{\overline{P}}}_{\overset{\uparrow}{d}}\ \text{FLprt (C.2.1.19.1),}$$

denote $FL_9 f d; Q; D$, $d \subset D$,

$$str\underbrace{\overset{\overset{d}{\uparrow}}{\overline{P}}}_{\overset{\uparrow}{D}}\ \text{FLprt (C.2.1.20),}$$

denote $FL_{10} f D; Q; d$, $d \subset D$,

$$\underbrace{\overset{\overset{P}{\uparrow}}{\overline{P}}}_{\overset{\uparrow}{R}}\ \text{FLprt (C.2.1.21)}$$

and any other possible options of self for (C.2.1.10) etc.

Dynamic FLprt – elements, self-type dynamic FLprt-structures.

We considered FLprt – elements earlier. Here we consider dynamic FLprt – elements. We consider fuzzy dynamic operator whose elements change over time

$$\begin{array}{c} C(t) \\ \uparrow \\ \hline P(t) \\ \uparrow \\ \overbrace{R(t)} \end{array} \mathrm{FLprt(t)} \begin{array}{c} \overbrace{A(t)} \\ \uparrow \\ \hline Q(t) \\ \uparrow \\ B(t) \end{array} \ (2.1),$$

where $\overbrace{A(t)}$, $\overbrace{R(t)}$ - upper levels of fuzzy A(t) and fuzzy R(t) respectively, $\overline{Q(t)}$, $\overline{P(t)}$ - average levels of fuzzy Q(t) and fuzzy P(t) respectively, fuzzy B(t) goes to the middle level of fuzzy Q(t) - $\overline{Q(t)}$, $\overline{Q(t)}$ goes to the upper level of fuzzy A(t) - $\overbrace{A(t)}$, $\overbrace{R(t)}$ goes to the middle level of fuzzy P(t) - $\overline{P(t)}$, $\overline{P(t)}$ goes to the lower level of fuzzy C(t) simultaneously. The result of this process will be described by the expression

$$\begin{array}{c} C(t) \\ \uparrow \\ \hline P(t) \\ \uparrow \\ \overbrace{R(t)} \end{array} \mathrm{Lrt(t)} \begin{array}{c} \overbrace{A(t)} \\ \uparrow \\ \hline Q(t) \\ \uparrow \\ B(t) \end{array} \ (2.2).$$

Definition C.2.2.1. The fuzzy dynamic operator (C.2.2.1) we shall call dynamic FLprt – element of the first type, (C.2.2.2) we shall call dynamic FLrt – element of the first type.

It's allowed to add dynamic FLprt – elements:

$$\begin{array}{c} C(t) \\ \uparrow \\ \hline P(t) \\ \uparrow \\ \overbrace{R(t)} \end{array} \mathrm{FLprt(t)} \begin{array}{c} \overbrace{A_1(t)} \\ \uparrow \\ \hline Q(t) \\ \uparrow \\ B(t) \end{array} + \begin{array}{c} C(t) \\ \uparrow \\ \hline P(t) \\ \uparrow \\ \overbrace{R(t)} \end{array} \mathrm{FLprt(t)} \begin{array}{c} \overbrace{A_2(t)} \\ \uparrow \\ \hline Q(t) \\ \uparrow \\ B(t) \end{array} = \begin{array}{c} C(t) \\ \uparrow \\ \hline P(t) \\ \uparrow \\ \overbrace{R(t)} \end{array} \mathrm{FLprt(t)} \begin{array}{c} \overbrace{A_1(t)} \cup \overbrace{A_2(t)} \\ \uparrow \\ \hline Q(t) \\ \uparrow \\ B(t) \end{array}$$

(C.2.2.2.1),

$$\begin{array}{c} C(t) \\ \uparrow \\ \hline P(t) \\ \uparrow \\ \overbrace{R(t)} \end{array} \mathrm{FLprt(t)} \begin{array}{c} \overbrace{A(t)} \\ \uparrow \\ \hline Q(t) \\ \uparrow \\ B_1(t) \end{array} + \begin{array}{c} C(t) \\ \uparrow \\ \hline P(t) \\ \uparrow \\ \overbrace{R(t)} \end{array} FLprt(t) \begin{array}{c} \overbrace{A(t)} \\ \uparrow \\ \hline Q(t) \\ \uparrow \\ B_2(t) \end{array} = \begin{array}{c} C(t) \\ \uparrow \\ \hline P(t) \\ \uparrow \\ \overbrace{R(t)} \end{array} \mathrm{FLprt(t)} \begin{array}{c} \overbrace{A(t)} \\ \uparrow \\ \hline Q(t) \\ \uparrow \\ B_1(t) \cup B_2(t) \end{array}$$

(C.2.2.2.2),

$$\begin{array}{c}C_1(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array} + \begin{array}{c}C_2(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array} = \begin{array}{c}C_1(t)\cup C_2(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array}$$

(C.2.2.2.3),

$$\begin{array}{c}C(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R_1(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array} + \begin{array}{c}C(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R_2(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array} = \begin{array}{c}C(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R_1(t)}\cup\overbrace{R_2(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array}$$

(C.2.2.2.4),

$$\begin{array}{c}C(t)\\\uparrow\\\overline{P_1(t)}\\\uparrow\\\overbrace{R(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array} + \begin{array}{c}C(t)\\\uparrow\\\overline{P_2(t)}\\\uparrow\\\overbrace{R(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array} = \begin{array}{c}C(t)\\\uparrow\\\overline{P_1(t)}\cup\overline{P_2(t)}\\\uparrow\\\overbrace{R(t)}\end{array} FLprt(t) \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\B(t)\end{array}$$

(C.2.2.2.5),

$$\begin{array}{c}C(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R(t)}\end{array} \text{FLprt(t)} \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q_1(t)}\\\uparrow\\B(t)\end{array} + \begin{array}{c}C(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R(t)}\end{array} \text{FLprt(t)} \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q_2(t)}\\\uparrow\\B(t)\end{array} = \begin{array}{c}C(t)\\\uparrow\\\overline{P(t)}\\\uparrow\\\overbrace{R(t)}\end{array} \text{FLprt(t)} \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q_1(t)}\cup\overline{Q_2(t)}\\\uparrow\\B(t)\end{array}$$

(C.2.2.2.6).

We consider the following self-type dynamic FLprt-structures of the first type:

$$\begin{array}{c}Q(t)\\\uparrow\\\overline{Q(t)}\\\uparrow\\\overbrace{Q(t)}\end{array} \text{FLprt(t)} \begin{array}{c}\overbrace{Q(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\Q(t)\end{array} \text{ (C.2.2.3),}$$

$$\begin{array}{c}Q(t)\\\uparrow\\\overline{Q(t)}\\\uparrow\\\overbrace{A(t)}\end{array} \text{FLprt(t)} \begin{array}{c}\overbrace{A(t)}\\\uparrow\\\overline{Q(t)}\\\uparrow\\Q(t)\end{array} \text{ (C.2.2.4),}$$

$$\begin{array}{c} B(t)\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ \overbrace{A(t)}\end{array}\text{FLprt(t)}\begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}\ \text{(C.2.2.5)},$$

$$\begin{array}{c} A(t)\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ \overbrace{A(t)}\end{array}\text{FLprt(t)}\begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ A(t)\end{array}\ \text{(C.2.2.6)}$$

$$\begin{array}{c} a(t)\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ str\,\overbrace{A(t)}\end{array}\ \text{FLprt(t)}\ \begin{array}{c} str\,\overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ a(t)\end{array}\quad \text{(C.2.2.6.1)},$$

denote $FL_{11}(t)fA(t);Q(t);a(t),\ a(t)\subset A(t),$

$$\begin{array}{c} strA(t)\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ \overbrace{a(t)}\end{array}\ \text{FLprt(t)}\ \begin{array}{c} \overbrace{a(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ strA(t)\end{array}\quad \text{(C.2.2.6.2)},$$

denote $FL_{12}(t)fa(t);Q(t);A(t),\ \ a(t)\subset A(t)$

$$\begin{array}{c} B(t)\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ \overbrace{B(t)}\end{array}\text{FLprt(t)}\begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}\ \text{(C.2.2.7)},$$

and any other possible options of self for (C.2.2.1) etc.

It can be considered a simpler version of the fuzzy dynamic operator

$$\text{FLprt(t)}\ \begin{array}{c} \overbrace{A(t)}\\ \uparrow\\ \overline{Q(t)}\\ \uparrow\\ B(t)\end{array}\ \text{(C.2.2.8)}$$

where $\overbrace{A(t)}$ - upper levels of fuzzy A(t), $\overline{Q(t)}$- average levels of fuzzy Q(t), fuzzy B(t) goes to the middle level of fuzzy Q(t) - $\overline{Q(t)}$, $\overline{Q(t)}$ goes to the upper level of fuzzy A(t) - $\widehat{A(t)}$ simultaneously, the result of this process will be described by the expression

$$\text{Lrt(t)} \begin{array}{c} \widehat{A(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B(t) \end{array} \text{ (C.2.2.9),}$$

or

$$\begin{array}{c} C(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \widehat{R(t)} \end{array} \text{FLprt(t) (C.2.2.10),}$$

where $\widehat{R(t)}$ - upper levels of fuzzy R(t), $\overline{P(t)}$ - average levels of fuzzy P(t), $\widehat{R(t)}$ goes to the middle level of fuzzy P(t) - $\overline{P(t)}$, $\overline{P(t)}$ goes to the lower level of fuzzy C(t) simultaneously, the result of this process will be described by the expression

$$\begin{array}{c} C(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \widehat{R(t)} \end{array} \text{FLrt(t) (C.2.2.11),}$$

Definition C.2.2.2. The dynamic operator (C.2.2.8) we shall call dynamic FLprt – element of the second type, (C.2.2.9) we shall call dynamic FLrt – element of the second type.

It's allowed to add dynamic FLprt – elements of the second type:

$$\text{FLprt(t)} \begin{array}{c} \widehat{A_1(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B(t) \end{array} + \text{FLprt(t)} \begin{array}{c} \widehat{A_2(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B(t) \end{array} = \text{FLprt(t)} \begin{array}{c} \widehat{A_1(t)} \cup \widehat{A_2(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B(t) \end{array} \text{ (C.2.2.12),}$$

$$\text{FLprt(t)} \begin{array}{c} \widehat{A(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B_1(t) \end{array} + FLprt(t) \begin{array}{c} \widehat{A(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B_2(t) \end{array} = \text{FLprt(t)} \begin{array}{c} \widehat{A(t)} \\ \uparrow \\ \overline{Q(t)} \\ \uparrow \\ B_1(t) \cup B_2(t) \end{array} \text{ (C.2.2.13),}$$

$$\text{FLprt(t)} \frac{\overset{\widehat{A(t)}}{\uparrow}}{\underset{\underset{B(t)}{\uparrow}}{Q_1(t)}} + \text{FLprt(t)} \frac{\overset{\widehat{A(t)}}{\uparrow}}{\underset{\underset{B(t)}{\uparrow}}{Q_2(t)}} = \text{FLprt(t)} \frac{\overset{\widehat{A(t)}}{\uparrow}}{\underset{\underset{B(t)}{\uparrow}}{\overline{Q_1(t)} \cup \overline{Q_2(t)}}} \text{ (C.2.2.13.1).}$$

We consider the following self-type dynamic FLprt-structures of the second type:

$$\text{FLprt(t)} \frac{\overset{\widehat{A(t)}}{\uparrow}}{\underset{\underset{A(t)}{\uparrow}}{Q(t)}} \text{ (C.2.2.14),}$$

$$\text{FLprt(t)} \quad \frac{\overset{str\,\widehat{A(t)}}{\uparrow}}{\underset{\underset{a(t)}{\uparrow}}{Q(t)}} \quad \text{(C.2.2.14.1),}$$

denote $FL_{13}(t)fA(t);Q(t);a(t),\ a(t) \subset A(t)$

$$\text{FLprt(t)} \quad \frac{\overset{\widehat{a(t)}}{\uparrow}}{\underset{\underset{strA(t)}{\uparrow}}{Q(t)}} \quad \text{(C.2.2.15),}$$

denote $FL_{14}(t)fa(t);Q(t);A(t),\ \ a(t) \subset A(t)$

$$\text{FLprt(t)} \frac{\overset{\widehat{A(t)}}{\uparrow}}{\underset{\underset{Q(t)}{\uparrow}}{Q(t)}} \text{ (C.2.2.16),}$$

and any other possible options of self for (C.2.2.8) etc.

Definition C.2.2.3. The fuzzy dynamic operator (C.2.2.10) we shall call dynamic tprFL – element, (C.2.2.11) we shall call dynamic trFL – element.

It’s allowed to add dynamic tprFL – elements:

$$\begin{array}{c} C_1(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} FLprt(t) + \begin{array}{c} C_2(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} FLprt(t) = \begin{array}{c} C_1(t) \cup C_2(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} FLprt(t) \quad (C.2.2.17),$$

$$\begin{array}{c} C(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R_1(t)} \end{array} FLprt(t) + \begin{array}{c} C(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R_2(t)} \end{array} FLprt(t) = \begin{array}{c} C(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R_1(t)} \cup \overbrace{R_2(t)} \end{array} FLprt(t) \quad (C.2.2.18),$$

$$\begin{array}{c} C(t) \\ \uparrow \\ \overline{P_1(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} FLprt(t) + \begin{array}{c} C(t) \\ \uparrow \\ \overline{P_2(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} FLprt(t) = \begin{array}{c} C(t) \\ \uparrow \\ \overline{P_1(t)} \cup \overline{P_2(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} FLprt(t) \quad (C.2.2.18.1).$$

We consider the following self-type dynamic tprFL-structures:

$$\begin{array}{c} R(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{R(t)} \end{array} \text{FLprt(t)} \quad (C.2.2.19)$$

$$\begin{array}{c} strD(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ \overbrace{d(t)} \end{array} \text{FLprt(t)} \quad (C.2.2.19.1),$$

denote $FL_{15}(t) f d(t); \overline{P(t)}; D(t)$, $d(t) \subset D(t)$

$$\begin{array}{c} d(t) \\ \uparrow \\ \overline{P(t)} \\ \uparrow \\ str\,\overbrace{D(t)} \end{array} \text{FLprt(t)} \quad (C.2.2.20)$$

denote $FL_{16}(t) f D(t); \overline{P(t)}; d(t)$, $d(t) \subset D(t)$

$$\begin{array}{l} P(t) \\ \uparrow \\ \overline{P(t)}\text{FLprt(t) (C.2.2.21)} \\ \uparrow \\ \overbrace{R(t)} \end{array}$$

and any other possible options of self for (C.2.2.10) etc.

New mathematical structures and operators is carried out with generalization it to any structures with any actions. For example,

$$1)\ \begin{array}{ccc} f_{11} & \dots & f_{1k} \\ \dots & \dots & \dots \\ (q_{j1})^{-1} & \dots & (q_{jk})^{-1} \\ \dots & \dots & \dots \\ f_{l1} & \dots & f_{lk} \end{array}\text{LFLprt}\ \begin{array}{ccc} q_{11} & \dots & q_{1n} \\ & \dots & \\ q_{m1} & \dots & q_{mn} \end{array}\quad (*_{\text{C.2.1}}),$$

$f_{ij},\ q_{ij}$ – any fuzzy objects, fuzzy actions etc.

$$2)\ \begin{array}{c} g_{11} \\ (w_{j1})^{-1} \\ g_{31} \end{array}\ \begin{array}{c} g_{12} \\ g_{22} \\ (w_{j2})^{-1} \\ \dots \\ g_{k2} \end{array}\ \begin{array}{c} g_{13} \\ (w_{j3})^{-1} \end{array}\ \text{FLGprt}\ \begin{array}{cccccc} w_{11} & w_{12} & & w_{1n} & \\ \dots & \dots & & w_{2n} & \\ & & & \dots & \\ w_{m1} & w_{m2} & \dots & \dots & w_{ml} \\ & & & w_{sn} & \end{array}\quad (*_{\text{C.2.2}}),$$

$w_{ij},\ g_{ij}$ – any fuzzy objects, fuzzy actions etc.

3)

$$\begin{array}{lll} a & b & g \\ c & ALrq(\mu) & w\ (*_{\text{C.2.3}}), \\ d & q & r \end{array}$$

where $ALrq$ is fuzzy virtual structure or fuzzy virtual operator, which can take any form of fuzzy action; a, c, d, q, r, w, g, b, μ – any fuzzy objects, fuzzy actions etc.

Accordingly, we can consider all sorts of self-type fuzzy structures for 1) – 3). And any other possible fuzzy structures and fuzzy operators etc.

C.3 Elements of the theory of variables of fuzzy hierarchical fuzzy dynamic operators: FLprt

In contrast to the classical one-attribute fuzzy set theory, where only its contents are taken as a set, we consider a two-attribute fuzzy set theory with a fuzzy set as a fuzzy capacity and separately with its contents. We simply use a convenient form to represent the singularity of a fuzzy set. Articles [1 -3], [8 - 16] use the following methodology for permanent structures:

1. Cancellation of the axiom of regularity.

2. 2 attributes for the fuzzy set: fuzzy capacity and its content.

3. Fuzzy compression of a fuzzy set, for example, to a point.

4. “turning out” from one another, particularly from a fuzzy capacity, we pull out another fuzzy capacity, for example, itself, as its element.

5. The simultaneity of one (fuzzy compression) and the other (“eversion”).

6. Own fuzzy capacities.

7. Qualitatively new fuzzy programming and fuzzy Networks.

Here we will consider variable fuzzy structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable fuzzy structures (models), for example,

$$\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{D} \end{array} FLprt(t) \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} = \begin{cases} (\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{D} \end{array} FLprt \,, \quad q_2 \geq t \geq q_1)|\mu_1 \\ (\begin{array}{c} B \\ \mu_7 \\ D \end{array} ffS^1prt \begin{array}{c} A \\ \mu_6 \\ B \end{array} , q_3 \geq \; t > q_2)|\mu_2 \\ (\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{D} \end{array} FLprt \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} , q_4 \geq \; t > q_3)|\mu_3 \\ (FLprt \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array} , q_5 \geq \; t > q_4)|\mu_4 \\ (\begin{array}{c} \{\,\} \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{D} \end{array} FLprt, \quad t > q_5)|\mu_5 \\ \qquad \ldots \end{cases} \qquad (*_{\mathrm{C.3.1}}),$$

μ_i- measures of fuzziness, i = 1, ..., 5. In particular, $\overset{B}{\underset{D}{\mu_7}} ffS^1prt \overset{A}{\underset{B}{\mu_6}}$ can be interpreted as a fuzzy game: player 1 fuzzy with measures of fuzziness μ_6 fits fuzzy A into fuzzy B, and the other fuzzy with measures of fuzziness μ_7 pushes fuzzy D out of fuzzy B at the same time [10], [15].

In what follows, we will denote variable fuzzy structure (model) through VFL(t), qself-variable fuzzy structures (models) through FLqFVS(t), qself is self for *action Q*, and oqself-variable fuzzy structures (models) through OqVFL(t), qoself is oself for *action Q*. Singular fuzzy structures (models) are not confused with fuzzy structures (models) with singularities. $\overset{B}{\underset{D}{\mu_7}} ffS^1prt \overset{A}{\underset{B}{\mu_6}}$ -2-hierarchical fuzzy structure: 1-level - elements A, B, C, D; level 2 - connections between them. 2-

Examples: a) discrete variable fuzzy structure with μ_i- measures of fuzziness, i = 1, ..., 8.

$$\begin{array}{ccc} a|\mu_1 & b|\mu_8 & g|\mu_7 \\ c|\mu_2 & VFL(t) & w|\mu_6 \\ d|\mu_3 & q|\mu_4 & r|\mu_5 \end{array}$$

c) continuous variable fuzzy structure

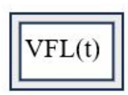


Fig. 0.1

Where a continuous fuzzy set represents the rim of the Fig.C.3.2.

We introduce the notation m_{fVLS_N} – the number of elements, N - the number of connections between them in the discrete variable 2-hierarchical fuzzy structure VFL(t). We introduce the notation q_{fVLS_R}– any, R - connections in q_{fVLS_R} in the variable 2-hierarchical fuzzy structure VFL(t), in particular, q_{fVLS_R}, R can be fuzzy sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the fuzzy structurability of Q from the interval [0,1], where 0 corresponds to "no fuzzy structure"," and 1 corresponds to the value " fuzzy structure". Then for joint A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D- self-(fuzzy structure) from A*B, cS(x)- the value of self-(fuzzy structure) for self-(fuzzy structure) x; for dependent fuzzy structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional fuzzy structurability of the fuzzy structure B at the fuzzy structure A, c(A/B)- conditional fuzzy structure of the fuzzy structure A at the fuzzy structure B. Adding inconsistent fuzzy structures: c(A+B) =c(A)+ +c(B). The formula of complete fuzzy structure: c(A)=$\sum_{k=1}^{n} c\left(B_k\right) * c(A/B_k)$, B_1, $B_{2.}$,.., B_n-full group of fuzzy hypotheses-actions: $\sum_{k=1}^{n} c\left(B_k\right)$=1("fuzzy structure"). Fuzzy Lprt- structure for fuzzy set of fuzzy structures $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$):

fDprt $\begin{array}{c}(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n))\\ action\ Q\\ B\end{array}$.

FLprt $\begin{array}{c}\overbrace{\{c(x_1)|\mu_{c(\widetilde{x})}c(x_1)|\mu_{c(\widetilde{x})}c(x_2),\ldots,c(x_n)|\mu_{c(\widetilde{x})}c(x_n)\}}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array}$ - fuzzy Lprt- structurability for these fuzzy structures. It is possible to consider the self-(fuzzy structure) $FL_8 f\widetilde{x_w}; Q; \widetilde{x}$, $\widetilde{x_w}\subset$ $\widetilde{x}$. The same for self-(fuzzy structurability): $FL_8 f C_w(\widetilde{x}); Q; \widetilde{C(x)}$, where $\widetilde{C(x)}$ = $\{ c(x_1)|\mu_{c(\widetilde{x})}c(x_1), c(x_2)|\mu_{c(\widetilde{x})}c(x_2),\ldots,c(x_n)|\mu_{c(\widetilde{x})}c\left(x_n\right)\}$, $C_w(\widetilde{x})\subset \widetilde{C(x)}$.

Can be considered N-hierarchical fuzzy structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical fuzzy structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, … can be any in particular, by fuzzy actions, fuzzy sets, and others.

Can be considered discrete fuzzy hierarchical fuzzy structure, continuous fuzzy hierarchical fuzzy structure, and discrete-continuous hierarchical fuzzy structure.

The example

$$\text{QHFL= HFLprt}\ \begin{array}{c}
(FLprt \begin{array}{c}\overbrace{N-level\ of\ hierarchical\ structure}\\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array})\mu_N \\
\dots \\
(FLprt \begin{array}{c}\overbrace{i-level\ of\ hierarchical\ structure}\\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array})\mu_i \\
\dots \\
(FLprt \begin{array}{c}\overbrace{1-level\ of\ hierarchical\ structure}\\ \uparrow \\ \overline{Q} \\ \uparrow \\ B \end{array})\mu_1 \\
\overline{Q} \\
B
\end{array}\ \text{- fuzzy}$$

N-hierarchical fuzzy structure compression into fuzzy B, μ_i- measures of fuzziness, i = 1, ..., N.

Let flg(N, QHFL)= $QHFL^{\left.QHFL^{QHFL^{\cdots^{QHFL}}}\right\}\text{-N levels}}$

It can be considered self- QHFL, flg(y, QHFL) for any y, flg(QHFL, QHFL).

Compression fuzzy Hierarchy Example:

$$\mathrm{Lprt}\ \frac{\overbrace{\begin{array}{c}\overbrace{\{\ \}}\\ Lprt\ \frac{\uparrow}{\{\ \}}\\ \uparrow\\ \{\ \}\\ \uparrow\end{array}}}{\begin{array}{c}\overbrace{\{\ \}}\\ Lprt\ \frac{\uparrow}{\{\ \}}\\ \uparrow\\ \{\ \}\\ \uparrow\\ \overbrace{\{\ \}}\\ Lprt\ \frac{\uparrow}{\{\ \}}\\ \uparrow\\ \{\ \}\end{array}}$$

Let's consider two versions: 1) fuzzy containment is interpreted through the concept of fuzzy containment, and 2) fuzzy capacity is interpreted through the concept of fuzzy containment as a rest point of fuzzy containment. Self-(fuzzy containment) is interpreted as a rest point of self-(fuzzy containment). Let A self-(fuzzy compress) into B, D self-(fuzzy displace) from C in $\underset{D}{\overset{C}{\mu_2 ff}} VS_1 prt \underset{B}{\overset{A}{\mu_1}}$.

We consider the functional ca(Q), which gives a numerical value for the accommodation of fuzzy Q from the interval [0,1], where 0 corresponds to " fuzzy action" and one corresponds to the value " fuzzy result of action". Then for joint fuzzy A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D- self-(fuzzy action) for A*B, caS(x)- the value of self-(fuzzy result of action) for self-(fuzzy action) of x; for dependent fuzzy actions: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional accommodation of the fuzzy action B at the fuzzy action A, ca(A/B)- conditional fuzzy result of action of the fuzzy action A at the fuzzy action B. Adding the fuzzy capacity values of inconsistent fuzzy action s: ca(A+B)=ca(A)+ca(B). The formula of complete fuzzy result of action: ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2, ..., B_n-full group of fuzzy hypotheses- actions: $\sum_{k=1}^{n} ca\,(B_k)$=1("fuzzy result of action"). FLprt-(fuzzy action) for

$\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$), …, x$_n|\mu_{\widetilde{x}}$(x$_n$)): FLprt $\overbrace{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n))}^{}\atop{\uparrow \atop {\overline{Q} \atop {\uparrow \atop w}}}$

,

$\widetilde{x}$ - fuzzy set of fuzzy actions.

FLprt $\overbrace{\{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_2),\ldots,ca(x_n)|\mu_{ca(\widetilde{x})}ca\,(x_n)\}}^{}\atop{\uparrow \atop {\overline{Q} \atop {\uparrow \atop w}}}$

- FLprt- accommodation for these fuzzy actions x$_i$, i = 1, ..., n. It is possible to consider the self-(fuzzy action) $FL_8f\widetilde{x_w};\underbrace{Q;\widetilde{x}}$, $\widetilde{x_w}\subset\widetilde{x}$. The same for self-(fuzzy accommodation): $FLfCa_w(\widetilde{x});Q;\widetilde{Ca(x)}$, where $Ca_w\,\underbrace{(\widetilde{x})\ =}$ $\{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1),ca(x_2)|\mu_{ca(\widetilde{x})}ca(x_2),\ldots,ca(x_n)|\mu_{ca(\widetilde{x})}ca\,(x_n)\}\subset\widetilde{Ca(x)}$.

Consider a variable fuzzy hierarchy (we will denote it by flVH).

We consider the functional h(Q), which gives a numerical value for the hierarchization of fuzzy Q from the interval [0,1], where 0 corresponds to "no fuzzy hierarchy", and 1 corresponds to the value " fuzzy hierarchy". Then for joint fuzzy hierarchies A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- self-(fuzzy hierarchy) from A*B, hS(x)- the value of self-(fuzzy hierarchy) for self-(fuzzy hierarchy) x; for dependent fuzzy hierarchies: h(A*B)=ha(A)*h(B/A)=h(B)*h(A/B), where h(B/A)- conditional hierarchization of the fuzzy hierarchy B at the fuzzy hierarchy A, h(A/B)- conditional fuzzy hierarchy of the fuzzy hierarchy A at the fuzzy hierarchy B. Adding the fuzzy hierarchy values of inconsistent fuzzy hierarchies: h(A+B)=h(A)+h(B). The formula of complete fuzzy hierarchy: h(A)=$\sum_{k=1}^{n}h\,(B_k)*h(A/B_k)$, B$_1$, B$_2$,..,B$_n$-full group of fuzzy hypotheses-hierarches: $\sum_{k=1}^{n}h\,(B_k)$=1(“fuzzy hierarchy”).

FLprt- structure for fuzzy set of hierarches $\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$), …, x$_n|\mu_{\widetilde{x}}$(x$_n$)): FLprt $\overbrace{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n))}^{}\atop{\uparrow \atop {\overline{Q} \atop {\uparrow \atop w}}}$.

FLprt $\begin{array}{c}\overbrace{\{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1)|\mu_{h(\widetilde{x})}h(x_2),\ldots,h(x_n)|\mu_{h(\widetilde{x})}h(x_n)\}}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ w\end{array}$ - FLprt-hierarchization for these fuzzy hierarches. It is possible to consider the self-(fuzzy hierarchy) $FL_8 f\widetilde{x_w}; Q; \widetilde{x}$, $\widetilde{x_w} \subset$ $\widetilde{x}$. The same for self- hierarchization $FL_8 f\widetilde{hx_w}; Q; \widetilde{hx}$, $\widetilde{hx_w} \subset$ $\widetilde{hx}$, $\widetilde{hx}$ = $\{$ $h(x_1)|\mu_{h(\widetilde{x})}h(x_1), h(x_2)|\mu_{h(\widetilde{x})}h(x_2),\ldots,h(x_n)|\mu_{h(\widetilde{x})}h(x_n)\}$. Can be considered FLprt $\begin{array}{c}\overbrace{\{\ ca(x), c(x), h(x)\}}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array}$.

Very interesting next fuzzy *structure* type:

$$\begin{array}{c}fuzzy\ structure\ A\\ \uparrow\\ \hline fuzzy\ structure\ A\\ \uparrow\\ \overbrace{fuzzy\ structure\ A}\end{array}\ \text{FLprt}\ \begin{array}{c}\overbrace{fuzzy\ structure\ A}\\ \uparrow\\ \hline fuzzy\ structure\ A\\ \uparrow\\ fuzzy\ structure\ A\end{array}$$

You can enter special operator FLCprt to work with fuzzy structures A, Q, B, R, P, C: $\begin{array}{c}C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array}$ FLCprt $\begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array}$.

Very interesting next fuzzy hierarchy type:

$$\begin{array}{c}fuzzy\ hierarchy\ A\\ \uparrow\\ \hline fuzzy\ hierarchy\ A\\ \uparrow\\ \overbrace{fuzzy\ hierarchy\ A}\end{array}\ \text{FLprt}\ \begin{array}{c}\overbrace{fuzzy\ hierarchy\ A}\\ \uparrow\\ \hline fuzzy\ hierarchy\ A\\ \uparrow\\ fuzzy\ hierarchy\ A\end{array}$$

You can enter special operator FLHprt to work with fuzzy hierarches A, Q, B, R, P, C: $\begin{array}{c}C\\ \uparrow\\ \overline{P}\\ \uparrow\\ \overbrace{R}\end{array}$ FLHprt $\begin{array}{c}\overbrace{A}\\ \uparrow\\ \overline{Q}\\ \uparrow\\ B\end{array}$.

C.4 Introduction to FUZZY PROGRAM OPERATORS Flprt, tprFL

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions. This must be done through FLprt-Networks - fuzzy analogue of Sprt-Networks [1- 3] in one of the central departments of which a conventional computer system is located. The parallel processor is itself fdeprogram - fuzzy analogue of eprogram [1- 3] with direct parallel computing not through serial computing.

Using conventional coding by a computer system, through a Target-block with a fuzzy Lprt -program operator - FLprt

$$\begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} \\ \uparrow \\ activation\ g \end{array}$$

, where fuzzy A with measure of fuzziness μ_A fuzzy acts Q with measure of fuzziness μ_Q to fuzzy *activation* with measure of fuzziness $\mu_{activation}$, Q is any fuzzy *action*, it will be possible to obtain the fuzzy execution with measure of fuzziness $\mu_{activation}$ of a parallel fuzzy action A with the desired target weight g or the execution with measure of fuzziness $\mu_{activation}$ of a parallel action A with the desired fuzzy target weight g with measure of fuzziness μ_g or both. Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For FLprt-coding and FLprt-translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018 year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel fdprogram of operator fDprt

$$\begin{array}{c} \{UHF\ AC\ R\} \\ action\ Q \\ activation \end{array}$$

with the specified measures of fuzziness, we simultaneously enter the desired fuzzy set of codes R with measure of fuzziness μ_R using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel fuzzy fdprogram operators:

1) fuzzy Lprt-program operators (designation FLprt-program operators)

2) fuzzy tprL-program operators (designation tprFL-program operators)

3) fuzzy L^1epr - program operators (designation FL^1epr -program operators)

4) fuzzy $Leprt_1$- program operators (designation $FLeprt_1$-program operators)

Examples:

1) Lprt $\begin{array}{c} \overbrace{A} \\ \uparrow \\ then\ \overline{Q} \\ \uparrow \\ IF\left\{\{B\}\{f\}\right\} \end{array}$

2) $\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \quad Lprt \quad \begin{array}{c} \overbrace{A} \\ \uparrow \\ then\ \overline{Q} :\equiv \\ \uparrow \\ IF\left\{\{B\}\{f\}\right\} \end{array}$

3) $\begin{array}{c} C \\ \uparrow \\ \overline{P} \\ \uparrow \\ \overbrace{R} \end{array} \quad Lprt \quad \begin{array}{c} \overbrace{A} \\ \uparrow \\ \overline{Q} :\equiv \\ \uparrow \\ B \end{array}$

4) $\begin{array}{c} \overbrace{A} \\ \uparrow \\ \text{then } \overline{Q} :\equiv \quad Lprt \\ \uparrow \\ IF\left\{\{B\}\{f\}\right\} \end{array}$

etc.

D SDS – elements, self-type SDS -structures and their applications

We consider dynamic operator

$$\begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(D.1.1)},$$

where $Subject\ of\ Q$ by A acts Q to B, $Subject\ of\ Q$ by D acts Q out from C simultaneously, Q is any $action,\ in\ particular,\ fuzzy\ action$; A, B, C, D may be fuzzy with corresponding fuzzy measures. The result of this process will be described by the expression

$$\begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDSr} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(D.1.2)}.$$

We consider the measure: $\mu**((\begin{array}{c} B \\ action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array})=\frac{\mu(A)}{\mu(D)}$,

where $\mu(A), \mu(D)$ –usual measures of A, D or fuzzy measures of A, D if A and D are fuzzy.

Definition D.1.1. The dynamic operator (D.1.1) we shall call SDS – element of the first type, (D.1.2) we shall call SDSr – element of the first type.

Remark D.1.1. $\begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array}$ - the analogue of ${}^C_D Sprt^A_B$ [1- 3] as a special case of (D.1.1), where *action Q* is "contain".

Remark D.1.1.1 an consider SDS– elements use the Banach space.

It's allowed to add SDS – elements:

$$\begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A_1 \\ action\ Q \\ B \end{array} + \begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A_2 \\ action\ Q \\ B \end{array} =$$

$$\begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A_1 \cup A_2 \\ action\ Q \\ B \end{array} \quad \text{(D.1.2.1)},$$

$$\begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B_1 \end{array} + \begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B_2 \end{array} =$$

$$\begin{array}{c} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B_1 \cup B_2 \end{array} \quad \text{(D.1.2.2)},$$

$$\begin{array}{c} C_1 \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} + \begin{array}{c} C_2 \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} =$$

$$\begin{array}{c} C_1 \cup C_2 \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(D.1.2.3)},$$

$$\begin{array}{c} C \\ (action\ Q)^{-1} \\ D_1 \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} + \begin{array}{c} C \\ (action\ Q)^{-1} \\ D_2 \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} =$$

$$\begin{array}{c} C \\ (action\ Q)^{-1} \\ D_1 \cup D_2 \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(D.1.2.4)}.$$

We consider the following self-type SDS-structures of the first type:

$$\begin{array}{c} Subject\ of\ Q \\ (action\ Q)^{-1} \\ Subject\ of\ Q \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ Subject\ of\ Q \\ action\ Q \\ Subject\ of\ Q \end{array} \quad \text{(D.1.3)},$$

denote SD_1fQ,

$$\begin{array}{c} action\ Q \\ (action\ Q)^{-1} \\ action\ Q \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ action\ Q \\ action\ Q \\ action\ Q \end{array} \quad \text{(D.1.3.1)},$$

denote $SD_{1,1}fQ$,

$$\begin{array}{c} action\ Q \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{array} \mathrm{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ action\ Q \end{array} \quad \text{(D.1.4)},$$

denote $SD_2fA;Q$,

$$\begin{array}{c} B \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(D.1.5)},$$

denote $SD_3 f A; Q; B$.

$$\begin{array}{c} A \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{array} \quad \text{(D.1.6)},$$

denote $SD_4 f A; Q$.

$$\begin{array}{c} a \\ (action\ Q)^{-1} \\ strA \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ strA \\ action\ Q \\ a \end{array} \quad \text{(D.1.6.1)},$$

denote $SD_5 f A; Q; a$, $a \subset A$ and structure of A acts Q to a and acts Q out from a simultaneously,

$$\begin{array}{c} StrA \\ (action\ Q)^{-1} \\ a \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ a \\ action\ Q \\ StrA \end{array} \quad \text{(D.1.6.2)},$$

denote $SD_6 f a; Q; A,\ a \subset A$ and acts Q to structure of A and acts Q out from structure of A simultaneously,

$$\begin{array}{c} B \\ (action\ Q)^{-1} \\ B \\ Subject\ of\ Q \end{array} \text{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(D.1.7)},$$

and any other possible options of self for (D.1.1) etc.

It can be considered a simpler version of the dynamic operator

$$\text{SDS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(D.1.8)}$$

where *Subject of Q* by A acts Q to B; A, B may be fuzzy with corresponding fuzzy measures; Q is any *action*, in particular, fuzzy *action*, the result of this process will be described by the expression

$$\text{SDSr} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(D.1.9)}$$

or

$$\begin{matrix} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{matrix} \text{ SDS (D.1.10)}$$

where *Subject of Q* by D acts Q out from C; C, D may be fuzzy with corresponding fuzzy measures; Q is any *action*, in particular, fuzzy *action*, the result of this process will be described by the expression

$$\begin{matrix} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{matrix} \text{ SDSr (D.1.11)}$$

Definition 1,2. The dynamic operator (D.1.8) we shall call SDS – element of the second type, (D.1.9) we shall call SDSr – element of the second type.

Remark D.1.2. SDS $\begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{matrix}$ - the analogue of $Sprt_B^A$ [1- 3] as a special case of (D.1.8), where *action Q* is "contain". In this case $Sprt_{action\ Q}^{action\ Q} \subset SDS \begin{matrix} Subject\ of\ Q \\ action\ Q \\ action\ Q \\ action\ Q \end{matrix}$ – self-containment and unlike usual self has higher level self(contain): sel$f^{\frac{3}{2}}$. That's why self-containment can generate, modify and perform other actions with self-capacities, because they have lower level = self.

It's allowed to add SDS – elements of the second type:

$$\text{SDS} \begin{matrix} Subject\ of\ Q \\ A_1 \\ action\ Q \\ B \end{matrix} + \text{SDS} \begin{matrix} Subject\ of\ Q \\ A_2 \\ action\ Q \\ B \end{matrix} = \text{SDS} \begin{matrix} Subject\ of\ Q \\ A_1 \cup A_2 \\ action\ Q \\ B \end{matrix} \text{ (D.1.12),}$$

$$\text{SDS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B_1 \end{matrix} + \text{SDS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B_2 \end{matrix} = \text{SDS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B_1 \cup B_2 \end{matrix} \text{ (D.1.13),}$$

We consider the following self-type SDS-structures of the second t type:

$$\text{SDS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{matrix} \text{ (D.1.14),}$$

$$\mathrm{SDS}\ \begin{matrix} Subject\ of\ Q \\ strA \\ action\ Q \\ a \end{matrix}\ \ (\mathrm{D.1.14.1}),$$

denote $SD_7 fA; Q; a,\ a \subset A$ and structure of A acts Q to a,

$$\mathrm{SDS}\ \begin{matrix} Subject\ of\ Q \\ a \\ action\ Q \\ strA \end{matrix}\ \ (\mathrm{D.1.14.2}),$$

denote $SD_8 fa; Q; A,\ \ a \subset A$ and acts Q to structure of A,

$$\mathrm{SDS}\ \begin{matrix} Subject\ of\ Q \\ Subject\ of\ Q \\ action\ Q \\ Subject\ of\ Q \end{matrix}\ (\mathrm{D.1.15}),$$

$$\mathrm{SDS}\ \begin{matrix} Subject\ of\ Q \\ action\ Q \\ action\ Q \\ action\ Q \end{matrix}\ \ (\mathrm{D.1.15.1}),$$

$$\mathrm{SDS}\ \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ action\ Q \end{matrix}\ \ (\mathrm{D.1.16}),$$

and any other possible options of self for (D.1.8) etc.

Definition D.1.3. The dynamic operator (D.1.10) we shall call tSDS – element, (D.1.11) we shall call tSDSr – element.

Remark D.1.3. $\begin{matrix} C \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{matrix}$ SDS is the analogue of ${}^{C}_{D}Sprt$ [1- 3] as a special case of (D.1.10), where *action Q* is “contain”.

It’s allowed to add tSDS – elements:

$$\begin{matrix} C_1 \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{matrix}\ \mathrm{SDS} + \begin{matrix} C_2 \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{matrix}\ \mathrm{SDS}= \begin{matrix} C_1 \cup C_2 \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{matrix}\ \mathrm{SDS}\ (\mathrm{D.1.17}),$$

$$\begin{matrix} C \\ (action\ Q)^{-1} \\ D_1 \\ Subject\ of\ Q \end{matrix}\ \mathrm{SDS} + \begin{matrix} C \\ (action\ Q)^{-1} \\ D_2 \\ Subject\ of\ Q \end{matrix}\ \mathrm{SDS} = \mathrm{SDS} \begin{matrix} C \\ (action\ Q)^{-1} \\ D_1 \cup D_2 \\ Subject\ of\ Q \end{matrix}\ \mathrm{SDS}\ (\mathrm{D.1.18}).$$

We consider the following self-type t SDS -structures:

$$\begin{array}{c} D \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDS (D.1.19)}$$

$$\begin{array}{c} strD \\ (action\ Q)^{-1} \\ d \\ Subject\ of\ Q \end{array} \text{SDS (D.1.19.1),}$$

denote $SD_9fd;Q;D$, $d \subset D$ and d acts Q out from structure of D,

$$\begin{array}{c} d \\ (action\ Q)^{-1} \\ strD \\ Subject\ of\ Q \end{array} \text{SDS (D.1.19.2),}$$

denote $SD_{10}fD;Q;d$, $d \subset D$ and structure of D acts Q out from d,

$$\begin{array}{c} action\ Q \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SDS (D.1.20)}$$

$$\begin{array}{c} action\ Q \\ (action\ Q)^{-1} \\ action\ Q \\ Subject\ of\ Q \end{array} \text{SDS (D.1.21)}$$

$$\begin{array}{c} Subject\ of\ Q \\ (action\ Q)^{-1} \\ Subject\ of\ Q \\ Subject\ of\ Q \end{array} \text{SDS (D.1.21.1)}$$

and any other possible options of self for (D.1.10) etc.

Dynamic SDS – elements, self-type dynamic SDS-structures.

We considered SDS – elements earlier. Here we consider dynamic SDS – elements. We consider dynamic operator whose elements change over time

$$\begin{array}{c} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{array} \quad (2.1),$$

where $Subject\ of\ Q(t)$ by A(t) acts Q(t) to B(t), $Subject\ of\ Q(t)$ by D(t) acts Q(t) out from C(t) simultaneously; A(t), B(t), C(t), D(t) may be fuzzy with corresponding fuzzy measures; Q(t) is any *action*, in particular, fuzzy *action*. The result of this process will be described by the expression

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDSr(t)} \ \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad (2.2).$$

Definition D.2.1. The dynamic operator (D.2.1) we shall call dynamic SDS – element of the first type, (D.2.2) we shall call dynamic SDSr – element of the first type.

Remark D.2.1. $\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)} \ \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix}$ - the analogue of ${}^{C(t)}_{\ D(t)}Sprt(t)^{A(t)}_{B(t)}$ [1- 3] as a special case of (D.2.1), where *action* $Q(t)$ is "contain".

It's allowed to add dynamic SDS – elements:

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)} \ \begin{matrix} Subject\ of\ Q(t) \\ A_1(t) \\ action\ Q(t) \\ B(t) \end{matrix} + \begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)}$$
$$\begin{matrix} Subject\ of\ Q(t) \\ A_2(t) \\ action\ Q(t) \\ B(t) \end{matrix} = \begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)} \ \begin{matrix} Subject\ of\ Q(t) \\ A_1(t) \cup A_2(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad (D.2.2.1),$$

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)} \ \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_1(t) \end{matrix} + \begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)}$$
$$\begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_2(t) \end{matrix} = \begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)} \ \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_1(t) \cup B_2(t) \end{matrix} \quad (D.2.2.2),$$

$$\begin{matrix} C_1(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)} \ \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} + \begin{matrix} C_2(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)}$$
$$\begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} = \begin{matrix} C_1(t) \cup C_2(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \ \mathrm{SDS(t)} \ \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad (D.2.2.3),$$

$$\begin{array}{c} C(t) \\ (action\ Q(t))^{-1} \\ D_1(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{array} + \begin{array}{c} C(t) \\ (action\ Q(t))^{-1} \\ D_2(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)}$$

$$\begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{array} = \begin{array}{c} C(t) \\ (action\ Q(t))^{-1} \\ D_1(t) \cup D_2(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{array} \quad \text{(D.2.2.4).}$$

We consider the following self-type dynamic SDS -structures of the first type:

$$\begin{array}{c} Subject\ of\ Q(t) \\ (action\ Q(t))^{-1} \\ Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{array} \quad \text{(D.2.3),}$$

$$\begin{array}{c} action\ Q(t) \\ (action\ Q(t))^{-1} \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \end{array} \quad \text{(D.2.3.1),}$$

$$\begin{array}{c} action\ Q(t) \\ (action\ Q(t))^{-1} \\ A(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ action\ Q(t) \end{array} \quad \text{(D.2.4),}$$

$$\begin{array}{c} B(t) \\ (action\ Q(t))^{-1} \\ A(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{array} \quad \text{(D.2.5),}$$

$$\begin{array}{c} A(t) \\ (action\ Q(t))^{-1} \\ A(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ A(t) \end{array} \quad \text{(D.2.6),}$$

$$\begin{array}{c} a(t) \\ (action\ Q(t))^{-1} \\ strA(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ strA(t) \\ action\ Q(t) \\ a(t) \end{array} \quad \text{(D.2.6.1),}$$

denote $SD_{11}(t) f A(t); Q(t); a(t)$, $a(t) \subset A(t)$ and structure of A(t) acts Q(t) to $a(t)$ and acts Q(t) out from $a(t)$ simultaneously,

$$\begin{array}{c} strA(t) \\ (action\ Q(t))^{-1} \\ a(t) \\ Subject\ of\ Q(t) \end{array} \text{SDS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ a(t) \\ action\ Q(t) \\ strA(t) \end{array} \quad \text{(D.2.6.2),}$$

denote $SD_{12}(t) f a(t); Q(t); A(t),\ a(t) \subset A(t)$ and acts Q(t) to structure of A(t) and acts Q(t) out from structure of A(t) simultaneously,

$$\begin{matrix} B(t) \\ (action\ Q(t))^{-1} \\ B(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad \text{(D.2.7)},$$

and any other possible options of self for (D.2.1) etc.

It can be considered a simpler version of the dynamic operator

$$\text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad \text{(D.2.8)},$$

where *Subject of* $Q(t)$ by A(t) acts Q(t) to B(t); A(t), B(t) may be fuzzy with corresponding fuzzy measures; Q(t) is any *action*, in particular, fuzzy *action*, the result of this process will be described by the expression

$$\text{SDSr(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad \text{(D.2.9)},$$

or

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t) (D.2.10)},$$

where *Subject of* $Q(t)$ by D(t) acts Q(t) out from C(t); C(t), D(t) may be fuzzy with corresponding fuzzy measures; Q(t) is any *action*, in particular, fuzzy *action*, the result of this process will be described by the expression

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDSr(t) (D.2.11)},$$

Definition D.2.2. The dynamic operator (D.2.8) we shall call dynamic Dprt – element of the second type, (D.2.9) we shall call dynamic Drt – element of the second type.

Remark D.2.2. SDS(t) $\begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix}$ - the analogue of $Sprt(t)^{A(t)}_{B(t)}$ [1- 3] as a special case of (D.2.8), where *action* $Q(t)$ is “contain”. In this case

$Sprt(t)^{action\ Q(t)}_{action\ Q(t)} \subset\ SDS(t) \begin{matrix} Subject\ of\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \end{matrix}$ – self-containment and unlike usual self has higher level self(contain) $self^{\frac{3}{2}}$. That's why self-containment can generate, modify and perform other actions with self-capacities, because they have lower level = self.

It’s allowed to add dynamic SDS – elements of the second type:

$$\text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A_1(t) \\ action\ Q(t) \\ B(t) \end{matrix} + \text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A_2(t) \\ action\ Q(t) \\ B(t) \end{matrix} = \text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A_1(t) \cup A_2(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad \text{(D.2.12),}$$

$$\text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_1(t) \end{matrix} + \text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_2(t) \end{matrix} = \text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_1(t) \cup B_2(t) \end{matrix} \quad \text{(D.2.13),}$$

We consider the following self-type dynamic Dprt-structures of the second t type:

$$\text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ A(t) \end{matrix} \quad \text{(D.2.14),}$$

$$\text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ strA(t) \\ action\ Q(t) \\ a(t) \end{matrix} \quad \text{(D.2.14.1),}$$

denote $SD_{13}(t)fA(t);Q(t);a(t)$, $a(t) \subset A(t)$ and structure of A(t) acts Q(t) to $a(t)$,

$$\text{SDS(t)} \begin{matrix} Subject\ of\ Q(t) \\ a(t) \\ action\ Q(t) \\ strA(t) \end{matrix} \quad \text{(D.2.14.2),}$$

denote $SD_{14}(t)fa(t);Q(t);A(t),\ a(t) \subset A(t)$ and acts Q(t) to structure of A(t),

SDS(t) $\begin{matrix} Subject\ of\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \end{matrix}$ (D.2.15),

SDS(t) $\begin{matrix} Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix}$ (D.2.15.1),

SDS(t) $\begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix}$ (D.2.15.2),

SDS(t) $\begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix}$ (D.2.15.3),

SDS(t) $\begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ action\ Q(t) \end{matrix}$ (D.2.16),

SDS(t) $\begin{matrix} Subject\ of\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \\ B(t) \end{matrix}$ (D.2.16.1),

and any other possible options of self for (D.2.8) etc.

Definition D.2.3. The dynamic operator (D.2.10) we shall call dynamic tSDS – element, (D.2.11) we shall call dynamic tSDSr – element.

Remark D.2.3. $\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix}$ SDS(t) - the analogue of ${}^{C(t)}_{D(t)}Sprt(t)$ [1- 3] as a special case of (D.2.10), where $action\ Q(t)$ is “contain”.

It’s allowed to add dynamic tSDS – elements:

$$\begin{matrix} C_1(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t)} + \begin{matrix} C_2(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t)} = \begin{matrix} C_1(t) \cup C_2(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix}$$
SDS(t) (D.2.17),

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D_1(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t)} + \begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D_2(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t)} = \begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ D_1(t) \cup D_2(t) \\ Subject\ of\ Q(t) \end{matrix}$$
SDS(t) (D.2.18).

We consider the following self-type dynamic tSDS-structures:

$$\begin{matrix} D(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t) (D.2.19)}$$

$$\begin{matrix} strD(t) \\ (action\ Q(t))^{-1} \\ d(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t) (D.2.19.1),}$$

denote $SD_{15}(t)fd(t);Q(t);D(t)$, $d(t) \subset D(t)$ and d(t) acts Q(t) out from structure of D(t),

$$\begin{matrix} d(t) \\ (action\ Q(t))^{-1} \\ strD(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t) (D.2.19.2)}$$

denote $SD_{16}(t)fD(t);Q(t);d(t)$, $d(t) \subset D(t)$ and structure of D(t) acts Q(t) out from d(t),

$$\begin{matrix} action\ Q(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t) (D.2.20)}$$

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t) (D.2.20.1)}$$

$$\begin{matrix} action\ Q(t) \\ (action\ Q(t))^{-1} \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t) (D.2.20.2)}$$

$$\begin{matrix} Subject\ of\ Q(t) \\ (action\ Q(t))^{-1} \\ Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t)} \ \text{(D.2.21)}$$

$$\begin{matrix} Subject\ of\ Q(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t)} \ \text{(D.2.21.1)}$$

$$\begin{matrix} C(t) \\ (action\ Q(t))^{-1} \\ Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SDS(t)} \ \text{(D.2.21.2)}$$

and any other possible options of self for (D.2.10) etc.

New mathematical structures and operators is carried out with generalization it to any structures with any actions. For example,

$$1)\ \begin{matrix} f_{11} & \dots & f_{1k} \\ \dots & \dots & \dots \\ (q_{j1})^{-1} & \dots & (q_{jk})^{-1} \\ \dots & \dots & \dots \\ f_{l1} & \dots & f_{lk} \end{matrix} \text{DDprt} \begin{matrix} q_{11} & \dots & q_{1n} \\ & \dots & \\ q_{m1} & \dots & q_{mn} \end{matrix} \quad (*_{\text{D}}),$$

$f_{ij},\ q_{ij}$ – any objects, actions etc.

$$2)\ \begin{matrix} g_{11} \\ (w_{j1})^{-1} \\ g_{31} \end{matrix} \begin{matrix} g_{12} \\ g_{22} \\ (w_{j2})^{-1} \\ \dots \\ g_{k2} \end{matrix} \begin{matrix} g_{13} \\ (w_{j3})^{-1} \end{matrix} \text{DGprt} \begin{matrix} w_{11} & w_{12} & & w_{1n} & \\ \dots & \dots & & w_{2n} & \\ & & & \dots & \\ w_{m1} & w_{m2} & \dots & \dots & w_{ml} \\ & & & w_{sn} & \end{matrix} \quad (*_{\text{D.1}}),$$

$w_{ij},\ g_{ij}$ – any objects, actions etc.

3)

$$\begin{matrix} a & b & g \\ c & ASrq(\mu) & w \\ d & q & r \end{matrix} \ (*_{\text{D.2}}),$$

where $ASrq$ is virtual structure or virtual operator, which can take any form of action; a, c, d, q, r, w, g, b, μ – any objects, actions etc.

Accordingly, we can consider all sorts of self-structures for 1) – 3). And any other possible structures and operators etc.

Elements of the theory of variables of fuzzy hierarchical dynamic fuzzy SDS-operators.

In contrast to the classical one-attribute fuzzy set theory, where only its contents are taken as a set, we consider a two-attribute fuzzy set theory with a fuzzy set as a fuzzy capacity and separately with its contents. We simply use a convenient form to represent the singularity of a fuzzy set. Articles [1 - 3]-[8 - 16] use the following methodology for permanent structures:

1. Cancellation of the axiom of regularity.

2. 2 attributes for the fuzzy set: fuzzy capacity and its content.

3. Fuzzy compression of a fuzzy set, for example, to a point.

4. "turning out" from one another, particularly from a fuzzy capacity, we pull out another fuzzy capacity, for example, itself, as its element.

5. The simultaneity of one (fuzzy compression) and the other ("eversion").

6. Own fuzzy capacities.

7. Qualitatively new fuzzy programming and fuzzy Networks.

Here we will consider variable fuzzy structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable fuzzy structures (models), for example,

$$\begin{array}{c} C\\ (action\ Q)^{-1}\\ D\\ Subject\ of\ Q\end{array} SDS(t) \begin{array}{c} Subject\ of\ Q\\ A\\ action\ Q\\ B\end{array} =$$

$$\begin{cases} (\begin{array}{c} C\\ (action\ Q)^{-1}\\ D\\ Subject\ of\ Q\end{array} SDS\ ,\quad q_2 \geq t \geq q_1)|\mu_1 \\ (\begin{array}{c} B\\ \mu_7 \\ D\end{array} ffS^1prt \begin{array}{c} A\\ \mu_6 \\ B\end{array} ,q_3 \geq\ t > q_2)|\mu_2 \\ (\begin{array}{c} C\\ (action\ Q)^{-1}\\ D\\ Subject\ of\ Q\end{array} SDS \begin{array}{c} Subject\ of\ Q\\ A\\ action\ Q\\ B\end{array} ,q_4 \geq\ t > q_3)|\mu_3 \\ SDS \begin{array}{c} Subject\ of\ Q\\ A\\ action\ Q\\ B\end{array} ,q_5 \geq\ t > q_4)|\mu_4 \\ \{\ \} \\ (\begin{array}{c} (action\ Q)^{-1}\\ D\\ Subject\ of\ Q\end{array} SDS,\quad t > q_5)|\mu_5 \\ \ldots \end{cases} \quad (*_{\text{D.1}}),$$

μ_i- measures of fuzziness, i = 1, ..., 5. In particular, $\begin{array}{c} B\\ \mu_7 \\ D\end{array} ffS^1prt \begin{array}{c} A\\ \mu_6 \\ B\end{array}$ can be interpreted as a fuzzy game: player 1 fuzzy with measures of fuzziness μ_6 fits fuzzy A into fuzzy B, and the other fuzzy with measures of fuzziness μ_7 pushes fuzzy D out of fuzzy B at the same time.

In what follows, we will denote variable fuzzy structure (model) through fVSDS(t), qself-variable fuzzy structures (models) through SDqfFVS(t), qself is self for *action Q*, and oqself-variable fuzzy structures (models) through OqfVSDS(t), qoself is oself for *action Q*. Singular fuzzy structures (models) are not confused with fuzzy structures (models) with singularities. $\begin{array}{c} B\\ \mu_7 \\ D\end{array} ffS^1prt \begin{array}{c} A\\ \mu_6 \\ B\end{array}$ -2-hierarchical fuzzy structure: 1-level - elements A, B, C, D; level 2 - connections between them. 2-

Examples: a) discrete variable fuzzy structure with ţ$_i$- measures of fuzziness, i = 1, ..., 8.

$$\begin{array}{ccc} a|\mu_1 & b|\mu_8 & g|\mu_7 \\ c|\mu_2 & ffVSDS(t) & w|\mu_6 \\ d|\mu_3 & q|\text{ţ}_4 & r|\text{ţ}_5 \end{array}$$

c) continuous variable fuzzy structure

Fig. 0.1

Where a continuous fuzzy set represents the rim of the Fig.D.2.

We introduce the notation m_{fVDS_N}– the number of elements, N - the number of connections between them in the discrete variable 2-hierarchical fuzzy structure fVSDS(t). We introduce the notation m_{fVDS_N}– any, R - connections in m_{fVDS_N} in the variable 2-hierarchical fuzzy structure fVSDS(t), in particular, m_{fVDS_N}, R can be fuzzy sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the fuzzy structurability of Q from the interval [0,1], where 0 corresponds to "no fuzzy structure", and 1 corresponds to the value " fuzzy structure". Then for joint A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D- self-(fuzzy structure) from A*B, cS(x)- the value of self-(fuzzy structure) for self-(fuzzy structure) x; for dependent fuzzy structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional fuzzy structurability of the fuzzy structure B at the fuzzy structure A, c(A/B)- conditional fuzzy structure of the fuzzy structure A at the fuzzy structure B. Adding inconsistent fuzzy structures: c(A+B) =c(A)+ +c(B). The formula of complete fuzzy structure: c(A)=$\sum_{k=1}^{n} c\left(B_k\right) * c(A/B_k)$, B_1, B_2.,.., B_n-full group of fuzzy hypotheses-actions: $\sum_{k=1}^{n} c\left(B_k\right)$=1(“fuzzy structure”). Fuzzy SDS- structure for fuzzy set of fuzzy structures $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$):

SDS $\begin{array}{c} Subject\ of\ Q \\ (x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2), \ldots, x_n|\mu_{\widetilde{x}}(x_n)) \\ action\ Q \\ B \end{array}$.

SDS $\begin{array}{c} Subject\ of\ Q \\ \{c(x_1)|\mu_{c(\widetilde{x})}c(x_1)|\mu_{c(\widetilde{x})}c(x_2), \ldots, c(x_n)|\mu_{c(\widetilde{x})}c(x_n)\} \\ action\ Q \\ B \end{array}$ - fuzzy SDS-structurability for these fuzzy structures. It is possible to consider the self-(fuzzy structure) $fSD_8f\widetilde{x_w};Q;\widetilde{x}$, $\widetilde{x_w} \subset \widetilde{x}$. The same for self-(fuzzy structurability): $fSD_8fC_w(\widetilde{x});Q;\widetilde{C(x)}$, where

$\widetilde{C(x)} = \{ c(x_1)|\mu_{c(\widetilde{x})} c(x_1), c(x_2)|\mu_{c(\widetilde{x})} c(x_2), \ldots, c(x_n)|\mu_{c(\widetilde{x})} c(x_n)\}$, $C_w(\widetilde{x}) \subset \widetilde{C(x)}$.

Can be considered N-hierarchical fuzzy structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical fuzzy structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, $\ldots$ can be any in particular, by fuzzy actions, fuzzy sets, and others.

Can be considered discrete fuzzy hierarchical fuzzy structure, continuous fuzzy hierarchical fuzzy structure, and discrete-continuous hierarchical fuzzy structure.

The example

fQHSDS =

$$
\begin{array}{c}
Subject\ of\ Q \\
(SDS \begin{array}{c} Subject\ of\ Q \\ N-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{array})\mu_N \\
\ldots \\
(SDS \begin{array}{c} Subject\ of\ Q \\ i-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{array})\mu_i \\
\ldots \\
(SDS \begin{array}{c} Subject\ of\ Q \\ 1-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{array})\mu_1 \\
action\ Q \\
B
\end{array}
$$

HSDS - fuzzy N-hierarchical fuzzy structure compression into B, μ_i- measures of fuzziness, i = 1, ..., N.

Let fdg(N, fQHSDS)= $fQHSDS^{\left.fQHSDS^{fQHSDS^{\cdots fQHSDS}}\right\}\text{-N levels}}$

It can be considered self- fQHSDS, fsdg(y, fQHSDS) for any y, fSdg(fQHSDS, fQHSDS).

Hierarchy Examples:

$$1)\ SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \\ (\ ) \end{array}\ ) \\ action\ (\ ) \\ (\ SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \\ (\ ) \end{array}\ ) \end{array}$$

$$2) \left(\begin{array}{c} SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \\ (\ ) \end{array}\ ) \\ action\ (\ ) \\ (\ SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \\ (\ ) \end{array}\ ) \end{array} \\ SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \\ (\ ) \end{array}\ ) \\ action\ (\ ) \\ (\ SDS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \\ (\ ) \end{array}\ ) \end{array} \end{array} \right)$$

We consider the functional ca(Q), which gives a numerical value for the accommodation of fuzzy Q from the interval [0,1], where 0 corresponds to " fuzzy action" and one corresponds to the value " fuzzy result of action". Then for joint fuzzy A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D- self-(fuzzy action) for A*B, caS(x)- the value of self-(fuzzy result of action) for self-(fuzzy action) of x; for dependent fuzzy actions: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional accommodation of the fuzzy action B at the fuzzy action A, ca(A/B)- conditional fuzzy result of action of the fuzzy action A at the fuzzy action B. Adding the fuzzy capacity values of inconsistent fuzzy action s: ca(A+B)=ca(A)+ca(B). The formula of complete fuzzy result of action: ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2, ..., B_n-full group of fuzzy hypotheses- actions: $\sum_{k=1}^{n} ca\,(B_k)$=1(“fuzzy result of action”). SDS-(fuzzy action) for

$\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, ..., $x_n|\mu_{\widetilde{x}}(x_n)$): SDS $\begin{matrix} Subject\ of\ Q \\ (x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n)) \\ action\ Q \\ w \end{matrix}$,

$\widetilde{x}$ - fuzzy set of fuzzy actions.

SDS $\begin{matrix} Subject\ of\ Q \\ \{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_2),\ldots,ca(x_n)|\mu_{ca(\widetilde{x})}ca(x_n) \\ action\ Q \\ w \end{matrix}$ - SDS-accommodation for these fuzzy actions x_i, i = 1, ..., n. It is possible to consider the self-(fuzzy action) $fSD_8f\widetilde{x_w};Q;\widetilde{x},\ \widetilde{x_w}\subset\ \widetilde{x}$. The same for self-(fuzzy accommodation): $fSD_8fCa_w(\widetilde{x});Q;\widetilde{Ca(x)}$, where $Ca_w\,(\widetilde{x}) = \{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1), ca(x_2)|\mu_{ca(\widetilde{x})}ca(x_2),\ldots,ca(x_n)|\mu_{ca(\widetilde{x})}ca\,(x_n)\}\subset \widetilde{Ca(x)}$. We will denote a variable fuzzy hierarchy by fdVH.

We consider the functional h(Q), which gives a numerical value for the hierarchization of fuzzy Q from the interval [0,1], where 0 corresponds to "no fuzzy hierarchy", and 1 corresponds to the value " fuzzy hierarchy". Then for joint fuzzy hierarchies A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- self-(fuzzy hierarchy) from A*B, hS(x)- the value of self-(fuzzy hierarchy) for self-(fuzzy hierarchy) x; for dependent fuzzy hierarchies: h(A*B)=ha(A)*h(B/A)=h(B)*h(A/B), where h(B/A)- conditional hierarchization of the fuzzy hierarchy B at the fuzzy hierarchy A, h(A/B)- conditional fuzzy hierarchy of the fuzzy hierarchy A at the fuzzy hierarchy B. Adding the fuzzy hierarchy values of inconsistent fuzzy hierarchies: h(A+B)=h(A)+h(B). The formula of complete fuzzy hierarchy: h(A)=$\sum_{k=1}^{n} h\,(B_k) * h(A/B_k)$, B_1, B_2,..,B_n-full group of fuzzy hypotheses-hierarches: $\sum_{k=1}^{n} h\,(B_k)$=1(“fuzzy hierarchy”).

SDS - structure for fuzzy set of hierarches $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, ..., $x_n|\mu_{\widetilde{x}}(x_n)$): SDS $\begin{matrix} Subject\ of\ Q \\ (x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n)) \\ action\ Q \\ B \end{matrix}$.

SDS $\begin{matrix} Subject\ of\ Q \\ \{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1)|\mu_{h(\widetilde{x})}h(x_2),\ldots,h(x_n)|\mu_{h(\widetilde{x})}h(x_n)\} \\ action\ Q \\ B \end{matrix}$ - SDS- hierarchization for these fuzzy hierarches. It is possible to consider the self-(fuzzy hierarchy) $fSD_8f\widetilde{x_w};Q;\widetilde{x},\ \ \widetilde{x_w}\ \subset\ \widetilde{x}$. The same for self- hierarchization $fSD_8f\widetilde{hx_w};Q;\widetilde{hx}$, $\widetilde{hx_w}\subset\widetilde{hx}$. $\widetilde{hx}$ =

$\{\quad h(x_1)|\mu_{\widetilde{h(x)}} h(x_1), h(x_2)|\mu_{\widetilde{h(x)}} h(x_2), \ldots, h(x_n)|\mu_{\widetilde{h(x)}} h\,(x_n)\}.$ Can be considered SDS $\begin{matrix} Subject\ of\ Q \\ \{\ ca(x), c(x), h(x)\} \\ action\ Q \\ B \end{matrix}$.

Very interesting next fuzzy hierarchy type:

$$\begin{matrix} fuzzy\ hierarchy\ A \\ (action\ Q)^{-1} \\ fuzzy\ hierarchy\ A \\ Subject\ of\ Q \end{matrix} \text{SDS} \begin{matrix} Subject\ of\ Q \\ fuzzy\ hierarchy\ A \\ action\ Q \\ fuzzy\ hierarchy\ A \end{matrix}$$

. You can enter special operator SfCS to work with fuzzy structures: $\begin{matrix} A \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{matrix} \text{SfCS} \begin{matrix} Subject\ of\ Q \\ C \\ action\ Q \\ D \end{matrix}$ is the structuring of $Subject\ of\ Q$ by fuzzy structures R by fuzzy Q with the fuzzy structure from C and expelling fuzzy structure D by fuzzy action Q^{-1} from the fuzzy structure A simultaneously.

Very interesting next fuzzy structure type:

$$\begin{matrix} A \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{matrix} \text{SfCS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{matrix},$$

You can enter special operator SfHS to work with fuzzy hierarches: $\begin{matrix} A \\ (action\ Q)^{-1} \\ B \\ Subject\ of\ Q \end{matrix} \text{SfHS} \begin{matrix} Subject\ of\ Q \\ D \\ action\ Q \\ R \end{matrix}$ is the hierarchization of Subject of Q by fuzzy hierarchy R using fuzzy Q with fuzzy hierarchy from D and simultaneous elimination of fuzzy hierarchy B by fuzzy action Q^{-1} from fuzzy hierarchy A.

Introduction to FUZZY PROGRAM OPERATORS SDS, tSDS, fD^1epr, $fDeprt_{1.}$

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions. This must be done through SDS -Networks - fuzzy analogue of Sprt-Networks [1 - 3] in one of the central departments of which a conventional computer system is located. The parallel processor is itself fsdeprogram - fuzzy analogue of eprogram [1 - 3] with direct parallel computing not through serial computing.

Using conventional coding by a computer system, through a Target-block with a fuzzy SDS -program operator - SDS $\begin{matrix} Subject\ of\ Q \\ Ag \\ action\ Q \\ activation \end{matrix}$, where fuzzy A with measure of fuzziness μ_A fuzzy acts Q with measure of fuzziness μ_Q to fuzzy *activation* with measure of fuzziness $\mu_{activation}$, Q is any fuzzy *action*, it will be possible to obtain the fuzzy execution with measure of fuzziness $\mu_{activation}$ of a parallel fuzzy action A with the desired target weight g or the execution with measure of fuzziness $\mu_{activation}$ of a parallel action A with the desired fuzzy target weight g with measure of fuzziness μ_g or both. Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For SDS coding and SDS -translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018 year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel fsdprogram of operator SDS $\begin{matrix} Subject\ of\ Q \\ \{UHF\ AC\ R\} \\ action\ Q \\ activation \end{matrix}$ with the specified measures of fuzziness, we simultaneously enter the desired fuzzy set of codes R with measure of fuzziness μ_R using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel fuzzy fsdprogram operators:

1) fuzzy SDS-program operators
2) fuzzy tSDS -program operators
3) fuzzy D^1epr - program operators (designation fD^1epr -program operators)
4) fuzzy $Deprt_1$- program operators (designation f$Deprt_1$-program operators)

SDS -algorithm Example:

Simultaneous multiplication Q with measure of fuzziness μ_Q:

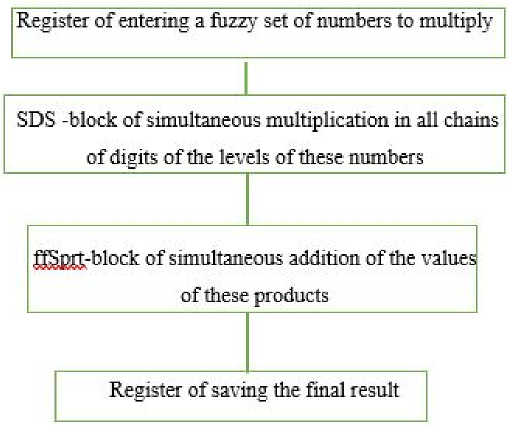


Fig. 0.2 The straightforward functional scheme of the assumed arithmetic-logical device for SDS -multiplication.

SDS $\begin{matrix} Subject\ of\ Q \\ \widetilde{x} \\ multiplication\ Q \\ \widetilde{y} \end{matrix}$, the notation of the fuzzy set B with elements $b_{i_1 i_2 \ldots i_n j_1 j_2 \ldots j_n} =$

$$(SDS \begin{matrix} Subject\ of\ Q \\ (ffSprt \left\{ x1_{i_1} * \mu_{\widetilde{x}}\left(x1_{i_1}\right) *, x2_{i_2} * \mu_{\widetilde{x}}\left(x2_{i_2}\right) *, \ldots,\quad x_{n_{i_n}} * \mu_{\widetilde{x}}\left(x_{n_{i_n}}\right) \right\}) \\ \mu \\ q \\ multiplication\ Q \\ (ffSprt \left\{ y1_{j_1} * \mu_{\widetilde{y}}\left(y1_{j_1}\right), y2_{j_2} * \mu_{\widetilde{y}}\left(y2_{j_2}\right), \ldots,\quad y_{n_{j_m}} * \mu_{\widetilde{y}}\left(y_{n_{j_m}}\right) \} \right\}) \\ \mu \\ h \end{matrix} {}^{R}_{G})_V$$

for any $\{i_1, i_2, ..., i_n\}$, $\{j_1, j, ..., j_n\}$ $without\ repetitions, q =$ ffSprt $\begin{matrix} K \\ \mu \\ w \end{matrix}$, K-set of any $\{k_1*, k_2*, ..., k_n*\}$ without repeating them, k_i-any digit, i=1,2,. . . ,n, R= ffSprt $\begin{matrix} \{i_1+, i_2+, ..., i_n\} \\ \mu \\ w \end{matrix}$, R is the index of the lower discharge, $h =$ ffSprt $\begin{matrix} L \\ \mu \\ w \end{matrix}$, L-set of any $\{l_1*, l_2*, ..., l_m*\}$ without repeating them, l_i-any digit, i=1,2,. . . ,m, G= ffSprt $\underset{w}{\overset{\{j_1+, j_2+, ..., j_m\}}{\mu}}$, G is the index of the lower discharge, V = ffSprt $\underset{w}{\overset{\{i_1+, i_2+, ..., i_n + j_1+, j_2+, ..., j_m\}}{\mu}}$ (we choose an index on the scale of discharges): see Table I.1.

Then ffSprt $\underset{w}{\overset{B+}{\mu}}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. Here, in fact, sets of digits in the corresponding digits, representing numbers, are multiplied together simultaneously.

Remark D.2.4. The algorithm for simultaneously fuzzy multiplication a fuzzy set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by fuzzy multiplying the first number from the fuzzy set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the fuzzy set by the ones following it, etc.

One example is pattern recognition: SDS $\begin{matrix} Subject\ of\ Q \\ if\ ffSprt \underset{B}{\overset{q}{\mu}} \subset ffSprt \underset{B}{\overset{image\ archive}{\mu}} \\ action\ Q = give\ test\ result \\ Name\ of\ q \end{matrix}$

The example of SDS -program is

$$\text{SDS}\left\{ fDprt \begin{matrix}\{\{p\}\} \\ := \\ \{(x)\}\end{matrix},\ \begin{matrix} Subject\ of\ Q \\ IF\,\{\{B\}\,\{f\}\} \\ fDprt give\ test\ result \\ H \\ action\ Q \\ B \end{matrix},\ fDprt \begin{matrix} R \\ action\ G \\ R \end{matrix} \right\}.$$

For example, based on SDS $\begin{matrix} Subject\ of\ Q \\ \widetilde{y} \\ action\ Q \\ \widetilde{x} \end{matrix}$, where $\widetilde{y}$=($y_1|\mu_{\widetilde{x}}(y_1)$, $y_2|\mu_{\widetilde{x}}(y_2)$, …, $y_m|\mu_{\widetilde{x}}(y_m)$) $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$), we can consider self-type SDS -structure - $fSD_8 f\widetilde{y}; Q; \widetilde{x}$ with m elements from $\widetilde{x}$, m<n, which is

formed according to the form (1.1), that is, the structure SDS $\begin{array}{c} Subject\ of\ Q \\ \widetilde{y} \\ action\ Q \\ \widetilde{x} \end{array}$

contains only m elements, or in forms (1.1.1) - (1.1.5), summarizing it. Fuzzy fcapacities in themselves of the third type can be formed for any other structure, not necessarily SDS, only by necessarily reducing the number of elements in the structure, in particular, using form (1.2). Structures more complex than fSD_8f can be introduced. For example, through a form (1.3), where A is fuzzy compressed (fuzzy fits) in C in the fuzzy compression fuzzy structure B in C (i.e., in the fuzzy structure SDS $\begin{array}{c} Subject\ of\ Q \\ \widetilde{y} \\ action\ Q \\ \widetilde{x} \end{array}$); or through the forms (1.3.1) - (1.4) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc. (1.3.1) - (1.3.4) schematically interpret the fuzzy formation of fuzzy capacity in itself through a pseudo 3-connected form with a 2-connected form. The ideology of SDS and fSD_8f can be used for programming.

Remark D.2.5. Fuzzy self, in particular, according to a fuzzy form- fuzzy analogue of the form of type (1.1): (2.1*).

Here are some of the fuzzy SDS -program operators.

1. Simultaneous fuzzy *action Q* of the expression*s* $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2), \ldots,$ $p_n|\mu_{\widetilde{p}}(p_n)$) to the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$). This is implemented via SDS $\begin{array}{c} Subject\ of\ Q \\ \widetilde{p} \\ action\ Q \\ \widetilde{x} \end{array}$.
2. Simultaneous R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2), \ldots,$ $B_n|\mu_{\widetilde{B}}(B_n)$). Implemented via SDS $\begin{array}{c} Subject\ of\ R \\ \widetilde{B} \\ action\ R \\ \widetilde{Q} \end{array}$, where $\widetilde{Q}$ can be anything.
3. Similarly for fuzzy loop operators and others.

fSD_8f– fuzzy software operators will differ only just because aggregates $\widetilde{x}, \widetilde{p}, \widetilde{B}, \widetilde{g}$ will be formed from corresponding fsdprt-program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

For example, SDS $\begin{array}{c} Subject\ of\ Q\{R,S\} \\ S \\ action\ Q\{R,S\} \\ R \end{array}$ is the fuzzy self-type SDS-structure with measure of fuzziness µ of the second type if $Q\{R,S\}$ is a fsdprogram capable of fuzzy generating R with measure of fuzziness µ from S.

The example of self-fsdprogram of the first type is

$$\text{SDS}\ \left\{\ \begin{array}{c} \widetilde{p} \\ fDprt\ Q \\ \{\widetilde{x}\} \end{array},\ \begin{array}{c} Subject\ of\ R \\ \widetilde{B} \\ fDprtR \\ \widetilde{Q} \\ action\ R \\ w \end{array},\ \begin{array}{c} Q \\ fDprtQ \\ Q \end{array}\ \right\}$$

The example of fsdprogram for SDmnsd- fuzzy analogue of SmnSprt [1 - 3]:

SDS $\begin{array}{c} Subject\ of\ Q \\ \widetilde{p} \\ action\ Q \\ \widetilde{x} \end{array}$ - fuzzy $action\ Q$ of $\widetilde{p}$ to $\widetilde{x}$.

SDS $\begin{array}{c} Subject\ of\ P \\ tw \\ action\ P \\ g \end{array}$, where P - fuzzy assigning target weight tw to fuzzy g with measure of fuzziness µ.

SDS $\begin{array}{c} Subject\ of\ S \\ \{q\}\,w \\ action\ S \\ SDmnsd\ activation \end{array}$, where S - SDmnsd $activation$ for fuzzy $\{q\}\,w$ with measure of fuzziness µ.

SDS -coding.

SDS -coding with measure of fuzziness µ: 1) fuzzy set A to fuzzy set B, 2) fuzzy set A to a point q, where the elements of the fuzzy sets A, B can be continuous. For example, SDS $\begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array}$, where Q - SDS -coding.

There are SDS-coding, SDS-translation, SDS-realize of fsdprograms and fprograms from the archives without extraction theirs

SDelf-coding.

SDelf-coding with measure of fuzziness µ: 1) fuzzy set A to set fuzzy A, i.e. fuzzy A on itself 2) fuzzy set A to a point q in form (1), where the elements of the fuzzy sets A, B can be continuous. For example, SDS $\begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{matrix}$.

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with SDS-Networks, it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through SDS-Networks and series-parallel. Codes from a conventional type computer system will be used via SDS-connectors in SDS-coding, for example: SDS $\begin{matrix} Subject\ of \\ \{UHF\ AC\ \} \\ \\ activation \end{matrix}$. UHF AC field activation is used.

Dynamic SDS and $SD_8(t)f$ programming.

The ideology of dynamic SDS and $SD_8(t)f$ can be used for programming:

1. Simultaneous fuzzy *action* $\widetilde{Q(t)}$ of the expressions $\widetilde{p(t)}$=($\mathrm{p_1(t)}|\mu_{\widetilde{p(t)}}(\mathrm{p_1(t)})$, $\mathrm{p_2(t)}|\mu_{\widetilde{p(t)}}(\mathrm{p_2(t)}), \ldots,$ $\mathrm{p_n(t)}|\mu_{\widetilde{p(t)}}(\mathrm{p_n(t)})$) to the variables $\widetilde{x(t)}$=($\mathrm{x_1(t)}|\mu_{\widetilde{x(t)}}(\mathrm{x_1(t)})$, $\mathrm{x_2(t)}|\mu_{\widetilde{x(t)}}(\mathrm{x_2(t)}), \ldots,$ $\mathrm{x_n(t)}|\mu_{\widetilde{x(t)}}(\mathrm{x_n(t)})$). This is implemented via

   SDS $\begin{matrix} Subject\ of\ \widetilde{Q(t)} \\ \widetilde{p(t)} \\ action\ \widetilde{Q(t)} \\ \left\{\widetilde{x(t)}\right\} \end{matrix}$.

2. Simultaneous $\widetilde{R(t)}$ = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g(t)}$=($\mathrm{g_1(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_1(t)})$, $\mathrm{g_2(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_2(t)}), \ldots,$ $\mathrm{g_n(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_n(t)})$) for the fuzzy set of expressions $\widetilde{B(t)}$=($\mathrm{B_1(t)}|\mu_{\widetilde{B(t)}}(\mathrm{B_1(t)})$, $\mathrm{B_2(t)}|\mu_{\widetilde{B(t)}}(\mathrm{B_2(t)}), \ldots,$ $\mathrm{B_n(t)}|\mu_{\widetilde{B(t)}}(\mathrm{B_n(t)})$). Implemented via SDS $\begin{matrix} Subject\ of\ \widetilde{R(t)} \\ \widetilde{B(t)} \\ action\ \widetilde{R(t)} \\ \widetilde{Q(t)} \end{matrix}$, where $\widetilde{Q(t)}$ can be anything.

3. Similarly for fuzzy loop operators and others.

$SD_8(t)f$ - fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}, \widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding SDS -program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

tSDS -program operators.

The ideology of tSDS and $SD_{16}f$ - fuzzy analogues of tS and $t_{S_4 f}$ from [14] can be used for programming. Here are some of the tSDS-program operators.

1. Simultaneous expelling fuzzy *action* Q of the expressions $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2), \ldots,$ $p_n|\mu_{\widetilde{p}}(p_n)$) from the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$). This is implemented via $\begin{matrix} \widetilde{x} \\ (action\ Q)^{-1} \\ \{\widetilde{p}\} \\ Subject\ of\ Q \end{matrix}$ SDS.

2. Simultaneous expelling R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2), \ldots,$ $B_n|\mu_{\widetilde{B}}(B_n)$). It's implemented through $\begin{matrix} \widetilde{Q} \\ (action\ R)^{-1} \\ \widetilde{B} \\ Subject\ of\ R \end{matrix}$ SDS, where $\widetilde{Q}$ can be anything.

3. Similarly for loop operators and others.

$SD_{16}f$ – fuzzy software operators will differ only just because aggregates $\widetilde{x}, \widetilde{p}, \widetilde{B}, \widetilde{g}$ will be formed from corresponding tSDS- program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). Consider hierarchical tSDS -program operator

$$\begin{matrix} B \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{matrix} \text{SDS} = \left\{ \begin{matrix} & \{\ \} & \\ D + & \mu & ffSprt \\ & A - A \cap B & \\ & (\ B - A \cap B) & \end{matrix} \right\}, \text{ where D is oself-(fuzzy set)}$$

for fuzzy $(A \cap B)$, where action Q- contain.

Dynamic tSDS and $SD_{16}(t)f$ programming at time q.

The ideology of tSDS and S$D_{16}f$ can be used for dynamic programming. Here are some of the tSDS - dynamic programming operators.

1. The process of simultaneous expelling fuzzy *action* $Q(t)$ of the expressions $\widetilde{p(t)}$=($p_1(t)|\mu_{\widetilde{p(t)}}(p_1(t))$, $p_2(t)|\mu_{\widetilde{p(t)}}(p_2(t)), \ldots,$ $p_n(t)|\mu_{\widetilde{p(t)}}(p_n(t))$) from the variables $\widetilde{x(t)}$=($x_1(t)|\mu_{\widetilde{x(t)}}(x_1(t))$,

$x_2(t)|\mu_{\widetilde{x(t)}}(x_2(t)), \ldots,\ x_n(t)|\mu_{\widetilde{x(t)}}(x_n(t)))$. This is implemented via

$$\begin{array}{c}\widetilde{x(t)}\\ (action\ Q(t))^{-1}\\ \widetilde{p(t)}\\ Subject\ of\ Q(t)\end{array}\mathrm{SDS(t)}.$$

2. The process of simultaneous expelling R(t) = fuzzy checking with fuzziness $\mu(t)$ by the fuzzy set of conditions $\widetilde{g(t)}=(g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t)),\ g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t)), \ldots,\ g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t)))$ for the fuzzy set of expressions $\widetilde{B(t)}=(B_1(t)|\mu_{\widetilde{B(t)}}(B_1(t)),\ B_2(t)|\mu_{\widetilde{B(t)}}(B_2(t)), \ldots,\ B_n(t)|\mu_{\widetilde{B(t)}}(B_n(t)))$ is implemented through $\begin{array}{c}\widetilde{Q(t)}\\ (action\ R(t))^{-1}\\ \widetilde{B(t)}\\ Subject\ of\ R(t)\end{array}$ SDS(t), where $\widetilde{Q(t)}$ can be anything.

3.Similarly for loop operators and others.

$SD_{16}(t)f$ – fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}, \widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding processes tSDS(t) for above mentioned programming operators through form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). Consider hierarchical dynamic tSDS-program operator:

$$\begin{array}{c}B(q\\ (action\ Q(q))^{-1}\\ A(q))\\ Subject\ of\ Q(t)\end{array} SDS(q) = \left\{\begin{array}{c} \qquad\qquad\qquad\qquad \{\ \} \\ fft(q)_{S_1 f(A(q)\cap B(q))} + \qquad \mu \qquad\qquad ffSprt(q) \\ \qquad\qquad A(q) - A(q)\cap B(q) \\ \qquad \big(B(q) - A(q)\cap B(q)\big) \end{array}\right\},$$

where action Q- contain.

fD[1]epr -program operators (form $\begin{array}{c}B\\ (action\ Q)^{-1}\\ D\end{array}$fD[1]prt $\begin{array}{c}A\\ action\ Q\\ B\end{array}$ - fuzzy analogue of ${}^{B}_{D}S^1t^{A}_{B}$ [10]).

For example, $\begin{array}{c}\widetilde{x}\\ R^{-1}\\ \{\widetilde{p}\}\end{array}$fD[1]prt$\begin{array}{c}\widetilde{B}\\ R\\ \widetilde{Q}\end{array}$, where simultaneous expelling fuzzy checking with fuzziness μ by the fuzzy set of conditions $\widetilde{g}=(g_1|\mu_{\widetilde{g}}(g_1),\ g_2|\mu_{\widetilde{g}}(g_2), \ldots,\ g_n|\mu_{\widetilde{g}}(g_n))$ for the fuzzy set $\widetilde{p}=(p_1|\mu_{\widetilde{p}}(p_1),\ p_2|\mu_{\widetilde{p}}(p_2), \ldots,\ p_n|\mu_{\widetilde{p}}(p_n))$ from the variables $\widetilde{x}=(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2), \ldots,\ x_n|\mu_{\widetilde{x}}(x_n))$ and simultaneous R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}=(g_1|\mu_{\widetilde{g}}(g_1),\ g_2|\mu_{\widetilde{g}}(g_2), \ldots,\ g_n|\mu_{\widetilde{g}}(g_n))$ for the fuzzy set of expressions $\widetilde{B}=(B_1|\mu_{\widetilde{B}}(B_1),\ B_2|\mu_{\widetilde{B}}(B_2), \ldots,\ B_n|\mu_{\widetilde{B}}(B_n))$, $\widetilde{Q}$ can be anything.

Appendix

Entire neural network as instantaneous simultaneous SDS-RAM in SDS - elements and fself- elements: $fself^{fself^{\cdots^{fself}}}$, $ff_1 \downarrow I \uparrow_{-1}^{1} ff_2^{\;ff_1 \downarrow I \uparrow_{-1}^{1} ff_2^{\;\cdots^{ff_1 \downarrow I \uparrow_{-1}^{1} ff_2}}}$, fsin$\infty^{fsin\infty^{\cdots^{fsin\infty}}}$ $etc.$When activated in a neural network, the entire neural network becomes a working memory. Use of fself-energy as fuzzy activation or from outside.

E SRS – elements, self-type SRS -structures and their applications

We consider dynamic operator

$$\begin{array}{c} C \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(E.1.1)},$$

where *Subject of Q* by A acts Q to B, *Subject of P* by D acts P out from C, Q and P are any *actions* simultaneously. The result of this process will be described by the expression

$$\begin{array}{c} C \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(E.1.2)}.$$

We consider the measure: $\mu**\left(\begin{array}{c} B \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \right) = \frac{\mu(A)\mu(Q)}{\mu(D)\mu(P)}$,

where $\mu(A)$, $\mu(D)$, $\mu(Q)$, $\mu(P)$ –usual measures of A, D, Q, P.

Definition E.1.1. The dynamic operator (E.1.1) we shall call SRS – element of the first type, (E.1.2) we shall call SRSr – element of the first type.

Remark E.1.1. an consider SRS – elements use the Banach space.

It's allowed to add SRS – elements:

$$\begin{array}{c} C \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A_1 \\ action\ Q \\ B \end{array} + \begin{array}{c} C \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A_2 \\ action\ Q \\ B \end{array} =$$

$$\begin{array}{c} C \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A_1 \cup A_2 \\ action\ Q \\ B \end{array} \quad \text{(E.1.2.1)},$$

$$\begin{array}{c} C \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B_1 \end{array} + \begin{array}{c} C \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B_2 \end{array} =$$

$$\begin{array}{c} C \\ action\ P \\ D \\ Subject\ of\ P \end{array} \ \text{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B_1 \cup B_2 \end{array} \quad \text{(E.1.2.2)},$$

$$\begin{array}{c} C_1 \\ action\ P \\ D \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} + \begin{array}{c} C_2 \\ action\ P \\ D \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} =$$

$$\begin{array}{c} C_1 \cup C_2 \\ action\ P \\ D \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(E.1.2.3)},$$

$$\begin{array}{c} C \\ action\ P \\ D_1 \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} + \begin{array}{c} C \\ action\ P \\ D_2 \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} =$$

$$\begin{array}{c} C \\ action\ P \\ D_1 \cup D_2 \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(E.1.2.4)}.$$

We consider the following self-type SRS-structures of the first type:

$$\begin{array}{c} Subject\ of\ P \\ action\ P \\ Subject\ of\ P \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ Subject\ of\ Q \\ action\ Q \\ Subject\ of\ Q \end{array} \quad \text{(E.1.3)},$$

denote $SR_1 f Q$.

$$\begin{array}{c} Subject\ of\ Q \\ (action\ Q)^{-1} \\ Subject\ of\ Q \\ Subject\ of\ Q \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ Subject\ of\ Q \\ action\ Q \\ Subject\ of\ Q \end{array} \quad \text{(E.1.3.0)},$$

$$\begin{array}{c} action\ Q \\ (action\ Q)^{-1} \\ action\ Q \\ Subject\ of\ Q \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ action\ Q \\ action\ Q \\ action\ Q \end{array} \quad \text{(E.1.3.1)},$$

denote $SR_{1,1} f Q$,

$$\begin{array}{c} action\ Q \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ action\ Q \end{array} \quad \text{(E.1.4)},$$

denote $SR_2 f A; Q$,

$$\begin{array}{c} B \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(E.1.5)},$$

denote $SR_3 f A; Q; B$.

$$\begin{array}{c} B \\ action\ P \\ A \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(E.1.5.1),}$$

$$\begin{array}{c} A \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{array} \quad \text{(E.1.6),}$$

denote $SR_4 f A; Q$.

$$\begin{array}{c} A \\ action\ P \\ A \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{array} \quad \text{(E.1.6.0),}$$

$$\begin{array}{c} a \\ (action\ Q)^{-1} \\ strA \\ Subject\ of\ Q \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ strA \\ action\ Q \\ a \end{array} \quad \text{(E.1.6.1),}$$

denote $SR_5 f A; Q; a$, $a \subset A$ and structure of A acts Q to a and acts Q out from a simultaneously,

$$\begin{array}{c} StrA \\ (action\ Q)^{-1} \\ a \\ Subject\ of\ Q \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ a \\ action\ Q \\ StrA \end{array} \quad \text{(E.1.7),}$$

denote $SR_6 f a; Q; A$, $a \subset A$ and acts Q to structure of A and acts Q out from structure of A simultaneously,

$$\begin{array}{c} B \\ (action\ Q)^{-1} \\ B \\ Subject\ of\ Q \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(E.1.8),}$$

$$\begin{array}{c} B \\ action\ P \\ B \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{array} \quad \text{(E.1.9),}$$

$$\begin{array}{c} B \\ action\ P \\ A \\ Subject\ of\ P \end{array} \text{SRS} \begin{array}{c} Subject\ of\ Q \\ B \\ action\ Q \\ B \end{array} \quad \text{(E.1.10),}$$

$$\begin{matrix} A \\ action\ P \\ A \\ Subject\ of\ P \end{matrix} \quad \text{SRS} \quad \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{matrix} \quad \text{(E.1.11)},$$

and any other possible options of self for (E.1.1) etc.

It can be considered a simpler version of the dynamic operator

$$\text{SRS} \quad \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{matrix} \quad \text{(E.1.12)},$$

where *Subject of Q* by A acts Q to B, Q is any *action*, the result of this process will be described by the expression

$$\text{SRSr} \quad \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{matrix} \quad \text{(E.1.13)}$$

or

$$\begin{matrix} C \\ action\ P \\ D \\ Subject\ of\ P \end{matrix} \quad \text{SRS (E.1.14)}$$

where *Subject of P* by D acts Q out from C, P is any *action*, the result of this process will be described by the expression

$$\begin{matrix} C \\ action\ P \\ D \\ Subject\ of\ P \end{matrix} \quad \text{SRSr (E.1.15)}$$

Definition 1,2. The dynamic operator (E.1.12) we shall call SRS – element of the second type, (E.1.13) we shall call SRSr – element of the second type.

It's allowed to add SRS– elements of the second type:

$$\text{SRS} \begin{matrix} Subject\ of\ Q \\ A_1 \\ action\ Q \\ B \end{matrix} + \text{SRS} \begin{matrix} Subject\ of\ Q \\ A_2 \\ action\ Q \\ B \end{matrix} = \text{SRS} \begin{matrix} Subject\ of\ Q \\ A_1 \cup A_2 \\ action\ Q \\ B \end{matrix} \quad \text{(E.1.16)},$$

$$\text{SRS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B_1 \end{matrix} + \text{SRS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B_2 \end{matrix} = \text{SRS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B_1 \cup B_2 \end{matrix} \quad \text{(E.1.17)},$$

We consider the following self-type SRS-structures of the second t type:

$$\mathrm{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{array}\ \ (\mathrm{E}.1.18),$$

$$\mathrm{SRS}\ \begin{array}{c} Subject\ of\ Q \\ strA \\ action\ Q \\ a \end{array}\ \ (\mathrm{E}.1.18.1),$$

denote $SR_7 fA; Q; a$, $a \subset A$ and structure of A acts Q to a,

$$\mathrm{SRS}\ \begin{array}{c} Subject\ of\ Q \\ a \\ action\ Q \\ strA \end{array}\ \ (\mathrm{E}.1.18.2),$$

denote $SR_8 fa; Q; A$, $a \subset A$ and acts Q to structure of A,

$$\mathrm{SRS}\ \begin{array}{c} Subject\ of\ Q \\ Subject\ of\ Q \\ action\ Q \\ Subject\ of\ Q \end{array}\ \ (\mathrm{E}.1.19),$$

$$\mathrm{SRS}\ \begin{array}{c} Subject\ of\ Q \\ action\ Q \\ action\ Q \\ action\ Q \end{array}\ \ (\mathrm{E}.1.19.1),$$

$$\mathrm{SRS}\ \begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ action\ Q \end{array}\ \ (\mathrm{E}.1.20),$$

and any other possible options of self for (E.1.12) etc.

Definition E.1.3. The dynamic operator (E.1.14) we shall call tSRS – element, (E.1.15) we shall call tSRSr – element.

It's allowed to add tSRS – elements:

$$\begin{array}{c} C_1 \\ action\ P \\ D \\ Subject\ of\ P \end{array}\ \mathrm{SRS} + \begin{array}{c} C_2 \\ action\ P \\ D \\ Subject\ of\ P \end{array}\ \mathrm{SRS} = \begin{array}{c} C_1 \cup C_2 \\ action\ P \\ D \\ Subject\ of\ P \end{array}\ \mathrm{SRS}\ (\mathrm{E}.1.21),$$

$$\begin{array}{c} C \\ action\ P \\ D_1 \\ Subject\ of\ P \end{array}\ \mathrm{SRS} + \begin{array}{c} C \\ action\ P \\ D_2 \\ Subject\ of\ P \end{array}\ \mathrm{SRS} = \begin{array}{c} C \\ action\ P \\ D_1 \cup D_2 \\ Subject\ of\ P \end{array}\ \mathrm{SRS}\ (\mathrm{E}.1.22).$$

We consider the following self-type tSRS-structures:

$$\begin{array}{c} D \\ action\ P \\ D \\ Subject\ of\ P \end{array} \text{SRS (E.1.23)}$$

$$\begin{array}{c} strD \\ action\ P \\ d \\ Subject\ of\ P \end{array} \text{SRS (E.1.23.1),}$$

denote $R_9 f d; Q; D$, $d \subset D$ and d acts Q out from structure of D,

$$\begin{array}{c} d \\ action\ P \\ strD \\ Subject\ of\ P \end{array} \text{SRS (E.1.23.2),}$$

denote $R_{10} f D; Q; d$, $d \subset D$and structure of D acts Q out from d,

$$\begin{array}{c} action\ Q \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{array} \text{SRS (E.1.24)}$$

$$\begin{array}{c} Subject\ of\ Q \\ (action\ Q)^{-1} \\ Subject\ of\ Q \\ Subject\ of\ Q \end{array} \text{SRS (E.1.25)}$$

$$\begin{array}{c} action\ Q \\ (action\ Q)^{-1} \\ action\ Q \\ Subject\ of\ Q \end{array} \text{SRS (E.1.25.1)}$$

and any other possible options of self for (E.1.14) etc.

Dynamic SRS – elements, dynamic SRS -self structures.

We considered SRS – elements earlier. Here we consider dynamic SRS – elements. We consider dynamic operator whose elements change over time

$$\begin{array}{c} C(t) \\ action\ P(t) \\ D(t) \\ Subject\ of\ P(t) \end{array} \quad \text{SRS(t)} \quad \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{array} \quad \text{(E.2.1),}$$

where $Subject\ of\ Q(t)$ by A(t) acts Q(t) to B(t), $Subject\ of\ P(t)$ by D(t) acts P(t) out from C(t), Q(t), P(t) are any *actions* simultaneously. The result of this process will be described by the expression

$$\begin{array}{c} C(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRSr(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B(t)\end{array} \quad \text{(E.2.2).}$$

Definition E.2.1. The dynamic operator (E.2.1) we shall call dynamic SRS – element of the first type, (E.2.2) we shall call dynamic SRSr – element of the first type.

It's allowed to add dynamic SRS – elements:

$$\begin{array}{c} C(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A_1(t)\\ action\ Q(t)\\ B(t)\end{array} + \begin{array}{c} C(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}$$
$$\begin{array}{c} Subject\ of\ Q(t)\\ A_2(t)\\ action\ Q(t)\\ B(t)\end{array} = \begin{array}{c} C(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A_1(t)\cup A_2(t)\\ action\ Q(t)\\ B(t)\end{array} \quad \text{(E.2.2.1),}$$

$$\begin{array}{c} C(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B_1(t)\end{array} + \begin{array}{c} C(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}$$
$$\begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B_2(t)\end{array} = \begin{array}{c} C(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B_1(t)\cup B_2(t)\end{array} \quad \text{(E.2.2.2),}$$

$$\begin{array}{c} C_1(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B(t)\end{array} + \begin{array}{c} C_2(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}$$
$$\begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B(t)\end{array} = \begin{array}{c} C_1(t)\cup C_2(t)\\ action\ P(t)\\ D(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B(t)\end{array} \quad \text{(E.2.2.3),}$$

$$\begin{array}{c} C(t)\\ action\ P(t)\\ D_1(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B(t)\end{array} + \begin{array}{c} C(t)\\ action\ P(t)\\ D_2(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}$$
$$\begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B(t)\end{array} = \begin{array}{c} C(t)\\ action\ P(t)\\ D_1(t)\cup D_2(t)\\ Subject\ of\ P(t)\end{array} \ \text{SRS(t)}\ \begin{array}{c} Subject\ of\ Q(t)\\ A(t)\\ action\ Q(t)\\ B(t)\end{array} \quad \text{(E.2.2.4).}$$

We consider the following self-type dynamic SRS -structures of the first type:

$$\begin{matrix} Subject\ of\ Q(t) \\ (action\ Q(t))^{-1} \\ Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{(E.2.3)},$$

$$\begin{matrix} action\ Q(t) \\ (action\ Q(t))^{-1} \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \end{matrix} \text{(E.2.3.1)},$$

$$\begin{matrix} action\ Q(t) \\ (action\ Q(t))^{-1} \\ A(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ action\ Q(t) \end{matrix} \text{(E.2.4)},$$

$$\begin{matrix} B(t) \\ (action\ Q(t))^{-1} \\ A(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \text{(E.2.5)},$$

$$\begin{matrix} B(t) \\ action\ P(t) \\ A(t) \\ Subject\ of\ P(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \text{(E.2.5.1)},$$

$$\begin{matrix} A(t) \\ (action\ Q(t))^{-1} \\ A(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ A(t) \end{matrix} \text{(E.2.6)},$$

$$\begin{matrix} A(t) \\ action\ P(t) \\ A(t) \\ Subject\ of\ P(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ A(t) \end{matrix} \text{(E.2.6.1)},$$

$$\begin{matrix} a(t) \\ (action\ Q(t))^{-1} \\ strA(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ strA(t) \\ action\ Q(t) \\ a(t) \end{matrix} \text{(E.2.6.1)},$$

denote $SR_{11}(t)fA(t);Q(t);a(t)$, $a(t) \subset A(t)$ and structure of A(t) acts Q(t) to $a(t)$ and acts Q(t) out from $a(t)$ simultaneously,

$$\begin{matrix} strA(t) \\ (action\ Q(t))^{-1} \\ a(t) \\ Subject\ of\ Q(t) \end{matrix} \text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ a(t) \\ action\ Q(t) \\ strA(t) \end{matrix} \text{(E.2.6.2)},$$

denote $SR_{12}(t)fa(t);Q(t);A(t)$, $a(t) \subset A(t)$ and acts Q(t) to structure of A(t) and acts Q(t) out from structure of A(t) simultaneously,

$$\begin{matrix} B(t) \\ (action\ Q(t))^{-1} \\ B(t) \\ Subject\ of\ Q(t) \end{matrix} \quad \text{SRS(t)} \quad \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad \text{(E.2.7),}$$

$$\begin{matrix} B(t) \\ action\ P(t) \\ B(t) \\ Subject\ of\ P(t) \end{matrix} \quad \text{SRS(t)} \quad \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad \text{(E.2.7.1),}$$

and any other possible options of self for (E.2.1) etc.

It can be considered a simpler version of the dynamic operator

$$\text{SRS(t)} \quad \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad \text{(E.2.8),}$$

where *Subject of* $Q(t)$ by A(t) acts Q(t) to B(t), Q(t) is any *action*, the result of this process will be described by the expression

$$\text{SRSr(t)} \quad \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B(t) \end{matrix} \quad \text{(E.2.9),}$$

or

$$\begin{matrix} C(t) \\ action\ P(t) \\ D(t) \\ Subject\ of\ P(t) \end{matrix} \quad \text{SRS(t) (E.2.10),}$$

where *Subject of* $P(t)$ by D(t) acts Q(t) out from C(t), Q(t) is any *action*, the result of this process will be described by the expression

$$\begin{matrix} C(t) \\ action\ P(t) \\ D(t) \\ Subject\ of\ P(t) \end{matrix} \quad \text{SRSr(t) (E.2.11),}$$

where *Subject of* $P(t)$ by D(t) acts Q(t) out from C(t), Q(t) is any *action*, the result of this process will be described by the expression

Definition E.2.2. The dynamic operator (E.2.8) we shall call dynamic SRS – element of the second type, (E.2.9) we shall call dynamic SRSr – element of the second type.

It's allowed to add dynamic SRS – elements of the second type:

$$\text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A_1(t) \\ action\ Q(t) \\ B(t) \end{array} + \text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A_2(t) \\ action\ Q(t) \\ B(t) \end{array} =$$

$$\text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A_1(t) \cup A_2(t) \\ action\ Q(t) \\ B(t) \end{array} \quad \text{(E.2.12),}$$

$$\text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_1(t) \end{array} + \text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_2(t) \end{array} =$$

$$\text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ B_1(t) \cup B_2(t) \end{array} \quad \text{(E.2.13),}$$ We consider the following self-type dynamic SRS -structures of the second t type:

$$\text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ A(t) \end{array} \quad \text{(E.2.14),}$$

$$\text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ strA(t) \\ action\ Q(t) \\ a(t) \end{array} \quad \text{(E.2.14.1),}$$

denote $SR_{13}(t)fA(t);Q(t);a(t)$, $a(t) \subset A(t)$ and structure of A(t) acts Q(t) to $a(t)$,

$$\text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ a(t) \\ action\ Q(t) \\ strA(t) \end{array} \quad \text{(E.2.14.2),}$$

denote $SR_{14}(t)fa(t);Q(t);A(t)$, $a(t) \subset A(t)$ and acts Q(t) to structure of A(t),

$$\text{SRS(t)} \begin{array}{c} Subject\ of\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \end{array} \quad \text{(E.2.15),}$$

$$\text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{ (E.2.15.1),}$$

$$\text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{ (E.2.15.2),}$$

$$\text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{matrix} \text{ (E.2.15.3),}$$

$$\text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ A(t) \\ action\ Q(t) \\ action\ Q(t) \end{matrix} \text{ (E.2.16),}$$

$$\text{SRS(t)} \begin{matrix} Subject\ of\ Q(t) \\ action\ Q(t) \\ action\ Q(t) \\ B(t) \end{matrix} \text{ (E.2.16.1),}$$

and any other possible options of self for (E.2.8) etc.

Definition E.2.3. The dynamic operator (E.2.10) we shall call dynamic tSRS – element, (E.2.11) we shall call dynamic tSRSr – element.

It's allowed to add dynamic tSRS – elements:

$$\begin{matrix} C_1(t) \\ action\ P(t) \\ D(t) \\ Subject\ of\ P(t) \end{matrix} \text{SRS(t)} + \begin{matrix} C_2(t) \\ action\ P(t) \\ D(t) \\ Subject\ of\ P(t) \end{matrix} \text{SRS(t)} = \begin{matrix} C_1(t) \cup C_2(t) \\ action\ P(t) \\ D(t) \\ Subject\ of\ P(t) \end{matrix}$$
SRS(t) (E.2.17),

$$\begin{matrix} C(t) \\ action\ P(t) \\ D_1(t) \\ Subject\ of\ P(t) \end{matrix} \text{SRS(t)} + \begin{matrix} C(t) \\ action\ P(t) \\ D_2(t) \\ Subject\ of\ P(t) \end{matrix} \text{SRS(t)} = \begin{matrix} C(t) \\ action\ P(t) \\ D_1(t) \cup D_2(t) \\ Subject\ of\ P(t) \end{matrix}$$
SRS(t) (E.2.18).

We consider the following self-type dynamic tSRS-structures:

$$\begin{matrix} D(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{matrix} \text{ SRS(t) (E.2.19)}$$

$$\begin{array}{c} strD(t) \\ (action\ Q(t))^{-1} \\ d(t) \\ Subject\ of\ Q(t) \end{array} \text{ SRS(t) (E.2.19.1),}$$

denote $SR_{15}(t) f d(t); Q(t); D(t)$, $d(t) \subset D(t)$ and d(t) acts Q(t) out from structure of D(t),

$$\begin{array}{c} d(t) \\ (action\ Q(t))^{-1} \\ strD(t) \\ Subject\ of\ Q(t) \end{array} \text{ SRS(t) (E.2.19.2)}$$

denote $SR_{16}(t) f D(t); Q(t); d(t)$, $d(t) \subset D(t)$ and structure of D(t) acts Q(t) out from d(t),

$$\begin{array}{c} action\ Q(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{array} \text{ SRS(t) (E.2.20)}$$

$$\begin{array}{c} C(t) \\ (action\ Q(t))^{-1} \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{array} \text{ SRS(t) (E.2.20.1)}$$

$$\begin{array}{c} action\ Q(t) \\ (action\ Q(t))^{-1} \\ action\ Q(t) \\ Subject\ of\ Q(t) \end{array} \text{ SRS(t) (E.2.20.2)}$$

$$\begin{array}{c} Subject\ of\ Q(t) \\ (action\ Q(t))^{-1} \\ Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \end{array} \text{ SRS(t) (E.2.21)}$$

$$\begin{array}{c} Subject\ of\ Q(t) \\ (action\ Q(t))^{-1} \\ D(t) \\ Subject\ of\ Q(t) \end{array} \text{ SRS(t) (E.2.21.1)}$$

$$\begin{array}{c} C(t) \\ (action\ Q(t))^{-1} \\ Subject\ of\ Q(t) \\ Subject\ of\ Q(t) \end{array} \text{ SRS(t) (E.2.21.2)}$$

and any other possible options of self for (E.2.10) etc.

New mathematical structures and operators is carried out with generalization it to any structures with any actions. For example,

$$1)\ \begin{array}{ccc} f_{11} & \dots & f_{1k} \\ \dots & \dots & \dots \\ (q_{j1})^{-1} & \dots & (q_{jk})^{-1} \\ \dots & \dots & \dots \\ f_{l1} & \dots & f_{lk} \end{array} \mathrm{DSRS} \begin{array}{ccc} q_{11} & \dots & q_{1n} \\ & \dots & \\ q_{m1} & \dots & q_{mn} \end{array} \quad (*_{\mathrm{E}}),$$

$f_{ij},\ q_{ij}$ – any objects, actions etc.

$$2)\ \begin{array}{ccc} & g_{12} & \\ g_{11} & g_{22} & \\ (w_{j1})^{-1} & (w_{j2})^{-1} & \begin{matrix} g_{13} \\ (w_{j3})^{-1} \end{matrix} \\ g_{31} & \dots & \\ & g_{k2} & \end{array} \ \mathrm{SRSGprt} \begin{array}{ccccc} w_{11} & w_{12} & & w_{1n} & \\ \dots & \dots & & w_{2n} & \\ & & & \dots & \\ w_{m1} & w_{m2} & \dots & \dots & w_{ml} \\ & & & w_{sn} & \end{array} \quad (*_{\mathrm{E.1}}),$$

$w_{ij},\ g_{ij}$ – any objects, actions etc.

3)

$$\begin{array}{ccc} a & b & g \\ c & ASRSrq(\mu) & w \\ d & q & r \end{array} \ (*_{\mathrm{E.2}}),$$

where $ASRSrq$ is virtual structure or virtual operator, which can take any form of action; a, c, d, q, r, w, g, b, μ – any objects, actions etc.

Accordingly, we can consider all sorts of self-structures for 1) – 3). And any other possible structures and operators etc.

Elements of the theory of variables of fuzzy hierarchical dynamic fuzzy SDS-operators.

In contrast to the classical one-attribute fuzzy set theory, where only its contents are taken as a set, we consider a two-attribute fuzzy set theory with a fuzzy set as a fuzzy capacity and separately with its contents. We simply use a convenient form to represent the singularity of a fuzzy set. Articles [1 -3]-[8 - 16] use the following methodology for permanent structures:

1. Cancellation of the axiom of regularity.

2. 2 attributes for the fuzzy set: fuzzy capacity and its content.

3. Fuzzy compression of a fuzzy set, for example, to a point.

4. “turning out” from one another, particularly from a fuzzy capacity, we pull out another fuzzy capacity, for example, itself, as its element.

5. The simultaneity of one (fuzzy compression) and the other (“eversion”).

6. Own fuzzy capacities.

7. Qualitatively new fuzzy programming and fuzzy Networks.

Here we will consider variable fuzzy structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable fuzzy structures (models), for example,

$$\begin{matrix} C \\ action\ P \\ D \\ Subject\ of\ Q \end{matrix} \quad SRS(t) \quad \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{matrix} \quad =$$

$$\begin{cases} \left(\begin{matrix} C \\ action\ P \\ D \\ Subject\ of\ Q \end{matrix} \quad SRS\ , \quad q_2 \geq t \geq q_1 \right) | \mu_1 \\ \left(\begin{matrix} B \\ \mu_7 \\ D \end{matrix} ffS^1prt \begin{matrix} A \\ \mu_6 \\ B \end{matrix} \quad , q_3 \geq \ t > q_2 \right) | \mu_2 \\ \left(\begin{matrix} C \\ action\ P \\ D \\ Subject\ of\ Q \end{matrix} \quad SRS \quad \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{matrix} \quad , q_4 \geq \quad t > q_3 \right) | \mu_3 \\ \quad SRS \quad \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{matrix} \quad , q_5 \geq \ t > q_4) | \mu_4 \\ \left(\begin{matrix} \{\ \} \\ action\ P \\ D \\ Subject\ of\ Q \end{matrix} \quad SRS, \quad t > q_5 \right) | \mu_5 \\ \qquad \ldots \end{cases} \qquad (*_{\text{E.3}}),$$

μ_i- measures of fuzziness, i = 1, ..., 5. In particular, $\begin{matrix} B \\ \mu_7 \\ D \end{matrix} ffS^1prt \begin{matrix} A \\ \mu_6 \\ B \end{matrix}$ can be interpreted as a fuzzy game: player 1 fuzzy with measures of fuzziness μ_6 fits fuzzy A into fuzzy B, and the other fuzzy with measures of fuzziness μ_7 pushes fuzzy D out of fuzzy B at the same time.

In what follows, we will denote variable fuzzy structure (model) through fVSRS(t), qself-variable fuzzy structures (models) through SRqfFVS(t), qself is self for *action Q*, and oqself-variable fuzzy structures (models) through OqfVSRS(t), qoself is oself for *action Q*. Singular fuzzy structures (models) are not confused with fuzzy structures (models) with singularities. $\begin{matrix} B \\ \mu_7 \\ D \end{matrix} ffS^1prt \begin{matrix} A \\ \mu_6 \\ B \end{matrix}$ -2-hierarchical fuzzy structure: 1-level - elements A, B, C, D; level 2 - connections between them. Examples: a) discrete variable fuzzy structure with μ_i-measures of fuzziness, i = 1, ..., 8.

$$\begin{array}{ccc} a|\mu_1 & b|\mu_8 & g|\mu_7 \\ c|\mu_2 & ffVSRS(t) & w|\mu_6 \\ d|\mu_3 & q|\mu_4 & r|\mu_5 \end{array}$$

c) continuous variable fuzzy structure

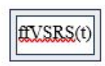


Fig. 0.1

Where a continuous fuzzy set represents the rim of the Fig.0.1.

We introduce the notation $m_{fVRS_N}-$ the number of elements, N - the number of connections between them in the discrete variable 2-hierarchical fuzzy structure fVSRS(t). We introduce the notation $m_{fVRS_N}-$ any, R - connections in m_{fVRS_N} in the variable 2-hierarchical fuzzy structure fVSRS(t), in particular, m_{fVRS_N}, R can be fuzzy sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the fuzzy structurability of Q from the interval [0,1], where 0 corresponds to "no fuzzy structure", and 1 corresponds to the value " fuzzy structure". Then for joint A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D- self-(fuzzy structure) from A*B, cS(x)- the value of self-(fuzzy structure) for self-(fuzzy structure) x; for dependent fuzzy structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional fuzzy structurability of the fuzzy structure B at the fuzzy structure A, c(A/B)- conditional fuzzy structure of the fuzzy structure A at the fuzzy structure B. Adding inconsistent fuzzy structures: c(A+B) =c(A)+ +c(B). The formula of complete fuzzy structure: c(A)=$\sum_{k=1}^{n} c\left(B_k\right) * c(A/B_k)$, B_1, $B_{2.}$, ..., B_n-full group of fuzzy hypotheses-actions: $\sum_{k=1}^{n} c\left(B_k\right)$=1("fuzzy structure"). Fuzzy SRS - structure for fuzzy set of fuzzy structures $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$):

$$\text{SRS}\ \begin{array}{c} \textit{Subject of Q} \\ (x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n)) \\ \textit{action Q} \\ B \end{array}.$$

SRS $\begin{matrix} Subject\ of\ Q \\ \{c(x_1)|\mu_{c(\widetilde{x})}c(x_1)|\mu_{c(\widetilde{x})}c(x_2),\ldots,c(x_n)|\mu_{c(\widetilde{x})}c(x_n)\} \\ action\ Q \\ B \end{matrix}$ - fuzzy SRS-structurability for these fuzzy structures. It is possible to consider the self-(fuzzy structure) $fSR_8f\widetilde{x_w};Q;\widetilde{x}$, $\widetilde{x_w} \subset \widetilde{x}$. The same for self-(fuzzy structurability): $fSR_8fC_w(\widetilde{x});Q;\widetilde{C(x)}$, where $\widetilde{C(x)} = \{ c(x_1)|\mu_{c(\widetilde{x})}c(x_1), c(x_2)|\mu_{c(\widetilde{x})}c(x_2),\ldots,c(x_n)|\mu_{c(\widetilde{x})}c(x_n)\}$, $C_w(\widetilde{x}) \subset \widetilde{C(x)}$. Can be considered N-hierarchical fuzzy structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical fuzzy structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, $\ldots$ can be any in particular, by fuzzy actions, fuzzy sets, and others. Can be considered discrete fuzzy hierarchical fuzzy structure, continuous fuzzy hierarchical fuzzy structure, and discrete-continuous hierarchical fuzzy structure,. The example

fQHSRS =

HSRS $\begin{matrix} (SRS \begin{matrix} Subject\ of\ Q \\ Subject\ of\ Q \\ N-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix})\mu_N \\ \ldots \\ (SRS \begin{matrix} Subject\ of\ Q \\ i-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix})\mu_i \\ \ldots \\ (SRS \begin{matrix} Subject\ of\ Q \\ 1-level\ of\ hierarchical\ structure \\ action\ Q \\ B \end{matrix})\mu_1 \\ action\ Q \\ B \end{matrix}$ - fuzzy N-hierarchical fuzzy structure compression into B, μ_i- measures of fuzziness, i = 1, ..., N.

Let frg(N, fQHSRS)= $fQHSRS^{fQHSRS^{fQHSRS^{\cdots fQHSRS}}}\}$-N levels

It can be considered self- fQHSRS, fsrg(y, fQHSRS) for any y, fSrg(fQHSRS, fQHSRS).

Hierarchy Examples:

$$1)\; SRS \begin{array}{c} Subject\ of\ (\ ) \\ (\ SRS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \end{array} \ ) \\ (\ ) \\ action\ (\ ) \\ Subject\ of\ (\ ) \\ (\ SRS \begin{array}{c} (\ ) \\ action\ (\ ) \end{array} \ ) \\ (\ ) \end{array}$$

$$2)\; \left(\begin{array}{c} SRS \begin{array}{c} Subject\ of\ (\ ) \\ (\ SRS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \end{array} \ ) \\ (\ ) \\ action\ (\ ) \\ Subject\ of\ (\ ) \\ (\ SRS \begin{array}{c} (\ ) \\ action\ (\ ) \end{array} \ ) \\ (\ ) \end{array} \\ SRS \begin{array}{c} Subject\ of\ (\ ) \\ (\ SRS \begin{array}{c} Subject\ of\ (\ ) \\ (\ ) \\ action\ (\ ) \end{array} \ ) \\ (\ ) \\ action\ (\ ) \\ Subject\ of\ (\ ) \\ (\ SRS \begin{array}{c} (\ ) \\ action\ (\ ) \end{array} \ ) \\ (\ ) \end{array} \end{array} \right)$$

We consider the functional ca(Q), which gives a numerical value for the accommodation of fuzzy Q from the interval [0,1], where 0 corresponds to " fuzzy action" and one corresponds to the value " fuzzy result of action". Then for joint fuzzy A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D- self-(fuzzy action) for A*B, caS(x)- the value of self-(fuzzy result of action) for self-(fuzzy action) of x; for dependent fuzzy actions: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional accommodation of the fuzzy action B at the fuzzy action A, ca(A/B)- conditional fuzzy result of action of the fuzzy action A at the fuzzy action B. Adding the fuzzy capacity values of inconsistent fuzzy action s: ca(A+B)=ca(A)+ca(B). The formula of complete fuzzy result of action: ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2, ..., B_n-full group of fuzzy hypotheses- actions: $\sum_{k=1}^{n} ca\,(B_k)$=1(“fuzzy result of action”). SRS -(fuzzy action) for

$\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$):

$$\text{SRS}\ \underset{\substack{action\ Q\\ w}}{\overset{Subject\ of\ Q}{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n))}},$$

$\widetilde{x}$ - fuzzy set of fuzzy actions.

$$\text{SRS}\ \underset{\substack{action\ Q\\ w}}{\overset{Subject\ of\ Q}{\{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_2),\ldots,ca(x_n)|\mu_{ca(\widetilde{x})}ca(x_n)}}$$

- SRS-accommodation for these fuzzy actions x_i, i = 1, ..., n. It is possible to consider the self-(fuzzy action) $fSR_8f\widetilde{x_w};Q;\widetilde{x}$, $\widetilde{x_w} \subset \widetilde{x}$. The same for self-(fuzzy accommodation): $fSR_8fCa_w(\widetilde{x});Q;\widetilde{Ca(x)}$, where $Ca_w(\widetilde{x}) = \{\ ca(x_1)|\mu_{ca(\widetilde{x})}ca(x_1), ca(x_2)|\mu_{ca(\widetilde{x})}ca(x_2),\ldots,ca(x_n)|\mu_{ca(\widetilde{x})}ca(x_n)\} \subset \widetilde{Ca(x)}$.

We consider the functional h(Q), which gives a numerical value for the hierarchization of fuzzy Q from the interval [0,1], where 0 corresponds to "no fuzzy hierarchy," and 1 corresponds to the value " fuzzy hierarchy. " Then for joint fuzzy hierarchies A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- self-(fuzzy hierarchy) from A*B, hS(x)- the value of self-(fuzzy hierarchy) for self-(fuzzy hierarchy) x; for dependent fuzzy hierarchies: h(A*B)=ha(A)*h(B/A)=h(B)*h(A/B), where h(B/A)- conditional hierarchization of the fuzzy hierarchy B at the fuzzy hierarchy A, h(A/B)- conditional fuzzy hierarchy of the fuzzy hierarchy A at the fuzzy hierarchy B. Adding the fuzzy hierarchy values of inconsistent fuzzy hierarchies: h(A+B)=h(A)+h(B). The formula of complete fuzzy hierarchy: h(A)=$\sum_{k=1}^{n} h(B_k) * h(A/B_k)$, B_1, B_2,..,B_n-full group of fuzzy hypotheses-hierarches: $\sum_{k=1}^{n} h(B_k)$=1(“fuzzy hierarchy”).

SRS - structure for fuzzy set of hierarches $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$): $\text{SRS}\ \underset{\substack{action\ Q\\ B}}{\overset{Subject\ of\ Q}{(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,x_n|\mu_{\widetilde{x}}(x_n))}}$.

$$\text{SRS}\ \underset{\substack{action\ Q\\ B}}{\overset{Subject\ of\ Q}{\{\ h(x_1)|\mu_{h(\widetilde{x})}h(x_1)|\mu_{h(\widetilde{x})}h(x_2),\ldots,h(x_n)|\mu_{h(\widetilde{x})}h(x_n)\}}}$$

- SRS- hierarchization for these fuzzy hierarches. It is possible to consider the self-(fuzzy hierarchy) $fSR_8f\widetilde{x_w};Q;\widetilde{x}$, $\widetilde{x_w} \subset \widetilde{x}$. The same for self- hierarchization $fSR_8f\widetilde{hx_w};Q;\widetilde{hx}$, $\widetilde{hx_w} \subset \widetilde{hx}$,

$\widetilde{hx} = \{ h(x_1)|\mu_{\widetilde{h(x)}} h(x_1), h(x_2)|\mu_{\widetilde{h(x)}} h(x_2), \ldots, h(x_n)|\mu_{\widetilde{h(x)}} h(x_n)\}$. Can be considered SRS $\begin{matrix} Subject\ of\ Q \\ \{ ca(x), c(x), h(x)\} \\ action\ Q \\ B \end{matrix}$.

Very interesting next fuzzy hierarchy type:

$$\begin{matrix} fuzzy\ hierarchy\ A \\ (action\ Q)^{-1} \\ fuzzy\ hierarchy\ A \\ Subject\ of\ Q \end{matrix} \text{SRS} \begin{matrix} Subject\ of\ Q \\ fuzzy\ hierarchy\ A \\ action\ Q \\ fuzzy\ hierarchy\ A \end{matrix}$$

. You can enter special operator SCRS to work with fuzzy structures: $\begin{matrix} A \\ (action\ Q)^{-1} \\ D \\ Subject\ of\ Q \end{matrix} \text{SCRS} \begin{matrix} Subject\ of\ Q \\ C \\ action\ Q \\ D \end{matrix}$ is the structuring of *Subject of Q* by fuzzy structures R by fuzzy Q with the fuzzy structure from C and expelling fuzzy structure D by fuzzy action Q^{-1} from the fuzzy structure A simultaneously.

Very interesting next fuzzy structure type:

$$\begin{matrix} A \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{matrix} \text{SCRS} \begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{matrix},$$

You can enter special operator SHRS to work with fuzzy hierarches: $\begin{matrix} A \\ (action\ Q)^{-1} \\ B \\ Subject\ of\ Q \end{matrix} \text{SHRS} \begin{matrix} Subject\ of\ Q \\ D \\ action\ Q \\ R \end{matrix}$ is the hierarchization of Subject of Q by fuzzy hierarchy R using fuzzy Q with fuzzy hierarchy from D and simultaneous elimination of fuzzy hierarchy B by fuzzy action Q^{-1} from fuzzy hierarchy A.

Introduction to FUZZY PROGRAM OPERATORS SRS, tSRS, fR^1epr, $fReprt_1$.

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions. This must be done through SRS -Networks - fuzzy analogue of Sprt-Networks [1 - 3] in one of the central departments of which a conventional computer system is located. The parallel processor is itself fsreprogram - fuzzy analogue of eprogram [1 - 3] with direct parallel computing not through serial computing.

Using conventional coding by a computer system, through a Target-block with a fuzzy SRS -program operator - SRS $\begin{matrix} Subject\ of\ Q \\ Ag \\ action\ Q \\ activation \end{matrix}$, where fuzzy A with measure of fuzziness μ_A fuzzy acts Q with measure of fuzziness μ_Q to fuzzy *activation* with measure of fuzziness $\mu_{activation}$, Q is any fuzzy *action*, it will be possible to obtain the fuzzy execution with measure of fuzziness $\mu_{activation}$ of a parallel fuzzy action A with the desired target weight g or the execution with measure of fuzziness $\mu_{activation}$ of a parallel action A with the desired fuzzy target weight g with measure of fuzziness μ_g or both. Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For SRS coding and SRS -translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018 year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel fsrprogram of operator SRS $\begin{matrix} Subject\ of\ Q \\ \{UHF\ AC\ R\} \\ action\ Q \\ activation \end{matrix}$ with the specified measures of fuzziness, we simultaneously enter the desired fuzzy set of codes R with measure of fuzziness μ_R using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel fuzzy fsrprogram operators:

1) fuzzy SRS-program operators
2) fuzzy tSRS -program operators
3) fuzzy R^1epr - program operators (designation fR^1epr -program operators)
4) fuzzy $Reprt_1$- program operators (designation f$Reprt_1$-program operators)

SRS -algorithm Example:

Simultaneous multiplication Q with measure of fuzziness μ_Q:

SRS $\substack{Subject\ of\ Q \\ \widetilde{x} \\ multiplication\ Q \\ \widetilde{y}}$, the notation of the fuzzy set B with elements

$$b_{i_1 i_2 \ldots i_n j_1 j_2 \ldots j_n} = (SRS \substack{Subject\ of\ Q \\ (ffSprt\substack{\left\{x_{1_{i_1}} * \mu_{\widetilde{x}}\left(x_{1_{i_1}}\right) *, x_{2_{i_2}} * \mu_{\widetilde{x}}\left(x_{2_{i_2}}\right) *, \ldots,\quad x_{n_{i_n}} * \mu_{\widetilde{x}}\left(x_{n_{i_n}}\right)\right\} \\ ţ \\ q}) \\ multiplication\ Q \\ (ffSprt\substack{\left\{y_{1_{j_1}} * \mu_{\widetilde{y}}\left(y_{1_{j_1}}\right), y_{2_{j_2}} * \mu_{\widetilde{y}}\left(y_{2_{j_2}}\right), \ldots,\quad y_{n_{j_m}} * \mu_{\widetilde{y}}\left(y_{n_{j_m}}\right)\}\right\} \\ \mu \\ h})}\ {}^{R}_{G}\ )_V$$

for any $\{i_1, i_2, ..., i_n\}$, $\{j_1, j, ..., j_n\}$ *without repetitions*, $q =$ ffSprt $\substack{K \\ \mu \\ w}$, K-set of any $\{k_1*, k_2*, ..., k_n*\}$ without repeating them, k_i-any digit, i=1,2,. . . ,n, R= ffSprt $\substack{\{i_1+, i_2+, ..., i_n\} \\ v \\ w}$, R is the index of the lower discharge, $h =$ ffSprt $\substack{L \\ \mu \\ w}$, L-set of any $\{l_1*, l_2*, ..., l_m*\}$ without repeating them, l_i-any digit, i=1,2,. . . ,m, G= ffSprt $\substack{\{j_1+, j_2+, ..., j_m\} \\ \mu \\ w}$, G is the index of the lower discharge, V = ffSprt $\substack{\{i_1+, i_2+, ..., i_n + j_1+, j_2+, ..., j_m\} \\ \mu \\ w}$ (we choose an index on the scale of discharges): see Table I.1.

Then ffSprt $\substack{B+ \\ \mu \\ w}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. Here, in fact, sets of digits in the corresponding digits, representing numbers, are multiplied together simultaneously.

Remark E.2.4. The algorithm for simultaneously fuzzy multiplication a fuzzy set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by fuzzy multiplying the first number from the fuzzy set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the fuzzy set by the ones following it, etc.

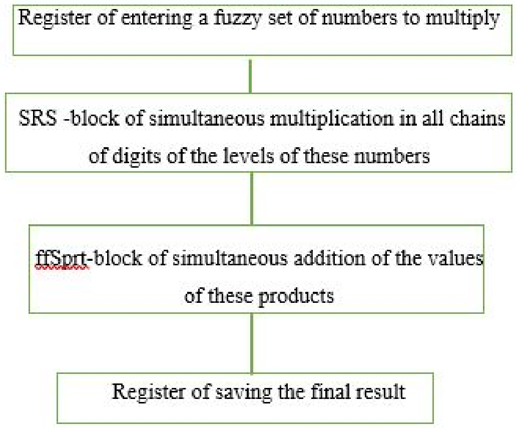


Fig. 0.2 The straightforward functional scheme of the assumed arithmetic-logical device for SRS -multiplication.

The simplest functional scheme of the assumed arithmetic-logical device for SRS-multiplication is Fig.02.
One example is pattern recognition:

$$SRS\begin{array}{c} Subject\ of\ Q \\ if\ ffSprt\,\mu_{B}^{q} \subset ffSprt\,\mu_{B}^{image\ archive} \\ action\ Q = give\ test\ result \\ Name\ of\ q \end{array}$$

The example of SRS -program is

$$SRS\left\{ fRprt_{\{(x)\}}^{\{\{p\}\}},\ fRprt\begin{array}{c} Subject\ of\ Q \\ IF\ \{\{B\}\{f\}\} \\ give\ test\ result \\ H \\ action\ Q \\ B \end{array},\ fRprt_{R}^{R}\ action\ G \right\}.$$

For example, based on SRS $\begin{array}{c} Subject\ of\ Q \\ \widetilde{y} \\ action\ Q \\ \widetilde{x} \end{array}$, where $\widetilde{y}$=($y_1|\mu_{\widetilde{x}}(y_1)$, $y_2|\mu_{\widetilde{x}}(y_2)$, ..., $y_m|\mu_{\widetilde{x}}(y_m)$) $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, ..., $x_n|\mu_{\widetilde{x}}(x_n)$), we can consider self-type SDS -structure - $fSR_8f\widetilde{y};Q;\widetilde{x}$ with m elements from $\widetilde{x}$, m<n, which is

formed according to the form (1.1), that is, the structure SRS $\begin{array}{c} \mathit{Subject\ of\ Q} \\ \widetilde{y} \\ \mathit{action\ Q} \\ \widetilde{x} \end{array}$

contains only m elements, or in forms (1.1.1) - (1.1.5), summarizing it. Fuzzy fcapacities in themselves of the third type can be formed for any other structure, not necessarily SRS, only by necessarily reducing the number of elements in the structure, in particular, using form (1.2). Structures more complex than fSR_8f can be introduced. For example, through a form (1.3), where A is fuzzy compressed (fuzzy fits) in C in the fuzzy compression fuzzy structure B in C (i.e., in the fuzzy structure SRS $\begin{array}{c} \mathit{Subject\ of\ Q} \\ \widetilde{y} \\ \mathit{action\ Q} \\ \widetilde{x} \end{array}$); or through the forms (1.3.1) - (1.4) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc. (1.3.1) - (1.3.4) schematically interpret the fuzzy formation of fuzzy capacity in itself through a pseudo 3-connected form with a 2-connected form. The ideology of SRS and fSR_8f can be used for programming.

Remark E.2.5. Fuzzy self, in particular, according to a fuzzy form- fuzzy analogue of the form of type (1.1): (2.1*).

Here are some of the fuzzy SRS -program operators.

1. Simultaneous fuzzy *action Q* of the expressions $\widetilde{p}=(p_1|\mu_{\widetilde{p}}(p_1),\ p_2|\mu_{\widetilde{p}}(p_2),\ldots,\ p_n|\mu_{\widetilde{p}}(p_n))$ to the variables $\widetilde{x}=(x_1|\mu_{\widetilde{x}}(x_1),\ x_2|\mu_{\widetilde{x}}(x_2),\ldots,\ x_n|\mu_{\widetilde{x}}(x_n))$. This is implemented via SRS $\begin{array}{c} \mathit{Subject\ of\ Q} \\ \widetilde{p} \\ \mathit{action\ Q} \\ \widetilde{x} \end{array}$.
2. Simultaneous R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}=(g_1|\mu_{\widetilde{g}}(g_1),\ g_2|\mu_{\widetilde{g}}(g_2),\ldots,\ g_n|\mu_{\widetilde{g}}(g_n))$ for the fuzzy set of expressions $\widetilde{B}=(B_1|\mu_{\widetilde{B}}(B_1),\ B_2|\mu_{\widetilde{B}}(B_2),\ldots,\ B_n|\mu_{\widetilde{B}}(B_n))$. Implemented via SRS $\begin{array}{c} \mathit{Subject\ of\ R} \\ \widetilde{B} \\ \mathit{action\ R} \\ \widetilde{Q} \end{array}$, where $\widetilde{Q}$ can be anything.
3. Similarly for fuzzy loop operators and others.

fSR_8f- fuzzy software operators will differ only just because aggregates $\widetilde{x},\widetilde{p},\ \widetilde{B},\widetilde{g}$ will be formed from corresponding fsrprt-program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

For example, SRS $\begin{matrix} Subject\ of\ Q\{R,S\} \\ S \\ action\ Q\{R,S\} \\ R \end{matrix}$ is the fuzzy self-type SRS-structure with measure of fuzziness µ of the second type if $Q\{R,S\}$ is a fsrprogram capable of fuzzy generating R with measure of fuzziness µ from S.

The example of self-fsrprogram of the first type is

$$\text{SRS} \begin{matrix} \\ \\ \{\ fRprt \begin{matrix} \widetilde{p} \\ Q \\ \{\widetilde{x}\} \end{matrix} , \\ \\ \\ \end{matrix} \begin{matrix} Subject\ of\ R \\ \widetilde{B} \\ fRprtR \\ \widetilde{Q} \\ action\ R \\ w \end{matrix} \begin{matrix} \\ \\ , fRprt \begin{matrix} Q \\ Q \\ Q \end{matrix} \} \\ \\ \\ \end{matrix}$$

The example of fsrprogram for SRmnsr- fuzzy analogue of SmnSprt [1 - 3]:

SRS $\begin{matrix} Subject\ of\ Q \\ \widetilde{p} \\ action\ Q \\ \widetilde{x} \end{matrix}$ - fuzzy $action\ Q$ of $\widetilde{p}$ to $\widetilde{x}$.

SRS $\begin{matrix} Subject\ of\ P \\ tw \\ action\ P \\ g \end{matrix}$, where P - fuzzy assigning target weight tw to fuzzy g with measure of fuzziness µ.

SRS $\begin{matrix} Subject\ of\ S \\ \{q\}\,w \\ action\ S \\ SRmnsr\ activation \end{matrix}$, where S - SRmnsr *activation* for fuzzy $\{q\}\,w$ with measure of fuzziness µ.

SRS -coding.

SRS -coding with measure of fuzziness µ: 1) fuzzy set A to fuzzy set B, 2) fuzzy set A to a point q, where the elements of the fuzzy sets A, B can be continuous. For example, SRS $\begin{matrix} Subject\ of\ Q \\ A \\ action\ Q \\ B \end{matrix}$, where Q - SRS -coding.

There are SRS-coding, SRS-translation, SRS-realize of fsrprograms and fprograms from the archives without extraction theirs

SRelf-coding.

SRelf-coding with measure of fuzziness µ: 1) fuzzy set A to set fuzzy A, i.e., fuzzy A on itself 2) fuzzy set A to a point q in form (1), where the elements of the fuzzy sets A, B can be continuous. For example, SRS $\begin{array}{c} Subject\ of\ Q \\ A \\ action\ Q \\ A \end{array}$.

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with SRS-Networks, it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through SRS-Networks and series-parallel. Codes from a conventional type computer system will be used via SRS-connectors in SrS-coding, for example: SRS $\begin{array}{c} Subject\ of \\ \{UHF\ AC\ \} \\ \\ activation \end{array}$. UHF AC field activation is used.

Dynamic SRS and $SR_8(t)f$ programming.

The ideology of dynamic SRS and $SR_8(t)f$ can be used for programming:

1. Simultaneous fuzzy *action* $\widetilde{Q(t)}$ of the expression*s* $\widetilde{p(t)}$=(p_1(t)|$\mu_{\widetilde{p(t)}}$(p_1(t)), p_2(t)|$\mu_{\widetilde{p(t)}}$(p_2(t)), …, p_n(t)|$\mu_{\widetilde{p(t)}}$(p_n(t))) to the variables $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t)), …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))). This is implemented via

   SRS $\begin{array}{c} Subject\ of\ \widetilde{Q(t)} \\ \widetilde{p(t)} \\ action\ \widetilde{Q(t)} \\ \left\{\widetilde{x(t)}\right\} \end{array}$.

2. Simultaneous $\widetilde{R(t)}$ = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g(t)}$=(g_1(t)|$\mu_{\widetilde{g(t)}}$(g_1(t)), g_2(t)|$\mu_{\widetilde{g(t)}}$(g_2(t)), …, g_n(t)|$\mu_{\widetilde{g(t)}}$(g_n(t))) for the fuzzy set of expressions $\widetilde{B(t)}$=(B_1(t)|$\mu_{\widetilde{B(t)}}$(B_1(t)), B_2(t)|$\mu_{\widetilde{B(t)}}$(B_2(t)), …, B_n(t)|$\mu_{\widetilde{B(t)}}$(B_n(t))). Implemented via SRS $\begin{array}{c} Subject\ of\ \widetilde{R(t)} \\ \widetilde{B(t)} \\ action\ \widetilde{R(t)} \\ \widetilde{Q(t)} \end{array}$, where $\widetilde{Q(t)}$ can be anything.

3. Similarly for fuzzy loop operators and others.

$SD_8(t)f$ - fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}$, $\widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding SRS -program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

tSRS -program operators.

The ideology of tSRS and $SR_{16}f$ - fuzzy analogues of tS and $t_{S_4 f}$ from [14] can be used for programming. Here are some of the tSRS-program operators.

1. Simultaneous expelling fuzzy *action* Q of the expressions $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2)$, …, $p_n|\mu_{\widetilde{p}}(p_n)$) from the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$). This is implemented via $\begin{matrix} \widetilde{x} \\ (action\ Q)^{-1} \\ \{\widetilde{p}\} \\ Subject\ of\ Q \end{matrix}$ SRS.

2. Simultaneous expelling R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2)$, …, $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2)$, …, $B_n|\mu_{\widetilde{B}}(B_n)$). It's implemented through $\begin{matrix} \widetilde{Q} \\ (action\ R)^{-1} \\ \widetilde{B} \\ Subject\ of\ R \end{matrix}$ SRS, where $\widetilde{Q}$ can be anything.

3. Similarly for loop operators and others.

$SR_{16}f$ – fuzzy software operators will differ only just because aggregates $\widetilde{x}, \widetilde{p}, \widetilde{B}, \widetilde{g}$ will be formed from corresponding tSRS- program operators in form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

Consider hierarchical tSRS -program operator

$$\begin{matrix} B \\ (action\ Q)^{-1} \\ A \\ Subject\ of\ Q \end{matrix} \text{SRS} = \left\{ \begin{matrix} & \{\ \} & \\ D + & \mu & ffSprt \\ & A - A \cap B & \\ & (\ B - A \cap B) & \end{matrix} \right\}, \text{ where D is oself-(fuzzy set)}$$

for fuzzy $(A \cap B)$, where action Q- contain.

Dynamic tSRS and $SR_{16}(t)f$ programming at time q.

The ideology of tSRS and $Sr_{16}f$ can be used for dynamic programming. Here are some of the tSrS - dynamic programming operators.

1. The process of simultaneous expelling fuzzy *action* $Q(t)$ of the expressions $\widetilde{p(t)}$=($p_1(t)|\mu_{\widetilde{p(t)}}(p_1(t))$, $p_2(t)|\mu_{\widetilde{p(t)}}(p_2(t))$, …, $p_n(t)|\mu_{\widetilde{p(t)}}(p_n(t))$) from the variables $\widetilde{x(t)}$=($x_1(t)|\mu_{\widetilde{x(t)}}(x_1(t))$,

$x_2(t)|\mu_{\widetilde{x(t)}}(x_2(t)), \ldots,$ $x_n(t)|\mu_{\widetilde{x(t)}}(x_n(t)))$. This is implemented via $\begin{matrix} \widetilde{x(t)} \\ (action\ Q(t))^{-1} \\ \widetilde{p(t)} \\ Subject\ of\ Q(t) \end{matrix}$ SRS(t).

2. The process of simultaneous expelling R(t) = fuzzy checking with fuzziness (t) by the fuzzy set of conditions $\widetilde{g(t)}=(g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t)),$ $g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t)), \ldots,$ $g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t)))$ for the fuzzy set of expressions $\widetilde{B(t)}=(B_1(t)|\mu_{\widetilde{B(t)}}(B_1(t)),$ $B_2(t)|\mu_{\widetilde{B(t)}}(B_2(t)), \ldots,$ $B_n(t)|\mu_{\widetilde{B(t)}}(B_n(t)))$ is implemented through $\begin{matrix} \widetilde{Q(t)} \\ (action\ R(t))^{-1} \\ \widetilde{B(t)} \\ Subject\ of\ R(t) \end{matrix}$ SRS(t), where $\widetilde{Q(t)}$ can be anything.

3.Similarly for loop operators and others.

$SD_{16}(t)f$ – fuzzy software operators will differ only just because aggregates $\widetilde{x(t)}, \widetilde{p(t)}, \widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding processes tSDS(t) for above mentioned programming operators through form (1.1) for more complex operators in forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

Consider hierarchical dynamic tSRS-program operator:

$$\begin{matrix} B(q \\ (action\ Q(q))^{-1} \\ A(q)) \\ Subject\ of\ Q(t) \end{matrix} SRS(q) = \left\{ fft(q)_{S_1 f(A(q)\cap B(q))} + \begin{matrix} \{\ \} \\ \mu \\ A(q) - A(q)\cap B(q) \\ \big(B(q) - A(q)\cap B(q)\big) \end{matrix} ffSprt(q) \right\},$$

where action Q- contain.

fR[1]epr -program operators (form $\begin{matrix} B \\ (action\ Q)^{-1} \\ D \end{matrix}$fR[1]prt $\begin{matrix} A \\ action\ Q \\ B \end{matrix}$ - fuzzy analogue of ${}_D^B S^1 t_B^A$ [10]).

For example, $\begin{matrix} \widetilde{x} \\ R^{-1} \\ \{\widetilde{p}\} \end{matrix}$fR[1]prt$\begin{matrix} \widetilde{B} \\ R \\ \widetilde{Q} \end{matrix}$, where simultaneous expelling fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}=(g_1|\mu_{\widetilde{g}}(g_1), g_2|\mu_{\widetilde{g}}(g_2), \ldots, g_n|\mu_{\widetilde{g}}(g_n))$ for the fuzzy set $\widetilde{p}=(p_1|\mu_{\widetilde{p}}(p_1), p_2|\mu_{\widetilde{p}}(p_2), \ldots, p_n|\mu_{\widetilde{p}}(p_n))$ from the variables $\widetilde{x}=(x_1|\mu_{\widetilde{x}}(x_1), x_2|\mu_{\widetilde{x}}(x_2), \ldots, x_n|\mu_{\widetilde{x}}(x_n))$ and simultaneous R = fuzzy checking with fuzziness by the fuzzy set of conditions $\widetilde{g}=(g_1|\mu_{\widetilde{g}}(g_1), g_2|\mu_{\widetilde{g}}(g_2), \ldots,$

$g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2)$, ..., $B_n|\mu_{\widetilde{B}}(B_n)$), $\widetilde{Q}$ can be anything.

Appendix

Entire neural network as instantaneous simultaneous SRS-RAM in SRS - elements and fself- elements. $fself^{fself^{\cdots^{fself}}}$, $ff1\downarrow I\uparrow_{-1}^{1}ff_2{}^{ff1\downarrow I\uparrow_{-1}^{1}ff_2{}^{\cdots^{ff1\downarrow I\uparrow_{-1}^{1}ff_2}}}$, $fsin\infty^{fsin\infty^{\cdots^{fsin\infty}}}$. When activated in a neural network, the entire neural network becomes a working memory. Use of fself-energy as fuzzy activation or from outside.

F Parallel Sprt – elements, self-type PrSprt-structures and their applications

Introduction.

There is a need to develop an instrumental mathematical base for new technologies. The task of the work is to create new approaches for this by introducing new concepts and methods. Our mathematics is unusual for a mathematician, because here the fulcrum is the action, and not the result of the action as in classical mathematics. Therefore, our mathematics is adapted not only to obtain results, but also to directly control actions, which will certainly show its benefits on a fundamentally new type of neural networks with directly parallel calculations, for which it was created [16]. Any action has much greater potential than its result. Our approach is not based on deterministic equations that generate self-organization, which is very difficult to study and gives very small results for a very limited class of problems and does not provide the most important thing - the structure of self-organization. We are just starting from the assumed structure of self-organization, since we are interested not so much in the numerical calculation of this as in the structure of self-organization itself, its formation (construction) for the necessary purposes and its management. Although we are also interested in numerical calculations. Nobel laureates in physics 2023 Ferenc Kraus and his colleagues Pierre Agostini and Anna Lhuillier used a short-pulse laser to generate attosecond pulses of light to study the dynamics of electrons in matter. According to our Theory of singularities of the type synthesizing, its action corresponds to singularity $\uparrow I \downarrow_h^q$, which allows one to reach the upper level of subtle energies to manipulate lower levels. In April 2023, we proposed using a short-pulse laser to achieve the desired goals by a directly parallel neural network [12]. We then proposed the fundamental development of this directly parallel neural network. The significance of our articles is in the formation of the presumptive mathematical structure of subtle energies, this is being done for the first time in science, and the presumptive classification of the mathematical structures of subtle energies for the first time. The experiments of the 2022 Nobel laureates Asle Ahlen, John Clauser, Anton Zeilinger and the experiments in chemistry Nazhipa Valitov eloquently demonstrate that we are right and that these studies are necessary. Be that as it may, we created classes of new mathematical structures, new mathematical singularities, i.e., made a contribution to the development of mathematics.

F.1 Parallel Sprt – elements, self-type PrSprt-structures

We consider expression

$$\begin{matrix} C_1 & C_2 & \ldots & C_m \\ D_1 & D_2 & \ldots & D_m \end{matrix} \mathrm{PrSprt} \begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix} \quad (*_{\mathrm{F.1}})$$

where A_1 fits into B_1, A_2 fits into B_2, …, A_n fits into B_n, D_1 is forced out of C_1, D_2 is forced out of C_2, …, D_m is forced out of C_m simultaneously [16]. The result of this process will be described by the expression

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\mathrm{PrSrt}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}\ (*_{\mathrm{F.2}})$$

If A_1, B_1, A_2, B_2, …, A_n, B_n, D_1, C_1, D_2, C_2, …, D_m, C_m are taken as sets, then we will call ($*_{\mathrm{F.1}}$) a parallel dynamic set. The need ($*_{\mathrm{F.1}}$) arose to describe processes in networks. Threshold element PrSprt - $\begin{matrix} B_1 & B_2 \dots & B_n \\ \{qy\}_1 & \{qy\}_2 \dots & \{qy\}_n \end{matrix}\mathrm{PrSprt}\begin{matrix} \{ax\}_1 & \{ax\}_2 \dots & \{ax\}_n \\ B_1 & B_2 \dots & B_n \end{matrix}$, B_1, B_2, …, B_n - artificial neurons of type PrSprt (designation - mnPrSprt) , $x=(x_1,x_2,\dots,x_n)$ are the values of the initial signals, $a=(a_1,a_2,\dots,a_n)$ are the weights of PrSprt-synapses and the values of the output signals $\{qy\}$. It can be considered a simpler version of the Parallel dynamic set

$$\mathrm{PrSprt}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}\ (**_{\mathrm{F.1}}),$$

where set A_1 fits into B_1, A_2 fits into B_2, …, A_n fits into B_n simultaneously, the result of this process will be described by the expression

$$\mathrm{PrSrt}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}\ (**_{\mathrm{F.2}})$$

or

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\mathrm{PrSprt}\ (***_{\mathrm{F.1}}),$$

where D_1 is forced out of C_1, D_2 is forced out of C_2, …, D_m is forced out of C_m simultaneously, the result of this process will be described by the expression

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\mathrm{PrSrt}\ (***_{\mathrm{F.2}})$$

We consider the measure: $\mu**(\begin{matrix} C_1 & C_1 & \dots & C_1 \\ D_1 & D_2 & \dots & D_m \end{matrix} PrSprt \begin{matrix} A_1 & A_2 & \dots & A_n \\ C_1 & C_1 & \dots & C_1 \end{matrix})=$ $\frac{\mu(A_1)*\mu(A_2)*\dots*\ \mu(A_n)}{\mu(D_1)*\mu(D_2)*\dots*\ \mu(D_m)}$, where $\mu(A_i), (\mu(D_j)$,–usual measures of sets A_i, D_j (i = 1, 2, …, n; j = 1, 2, …, m).

Remark F.1.1.1. One can consider some generalization for ($*_{\mathrm{F.1}}$) , ($*_{\mathrm{F.2}}$): $\begin{matrix} q_1(C_1) & q_2(C_2) \dots & q_m(C_m) \\ D_1 & D_2 \dots & D_m \end{matrix}\mathrm{PrSprt}\begin{matrix} A_1 & A_2 \dots & A_n \\ w_1(B_1) & w_2(B_2) \dots & w_n(B_n) \end{matrix}$, $\begin{matrix} q_1(C_1) & q_2(C_2) \dots & q_m(C_m) \\ D_1 & D_2 \dots & D_m \end{matrix}\mathrm{PrSrt}\begin{matrix} A_1 & A_2 \dots & A_n \\ w_1(B_1) & w_2(B_2) \dots & w_n(B_n) \end{matrix}$, where A_1 fits into B_1 through w_1, A_2 fits into B_2 through w_2, …, A_n fits into B_n through w_n, D_1 is forced out of C_1 through q_1, D_2 is forced out of C_2 through q_2, …, D_m is forced out of C_m through q_m, simultaneously. A_i, B_i, D_j, C_j (i = 1, 2, …, n; j = 1, 2, …, m) can be taken as sets. The result of this process will be described by the expression $\begin{matrix} q_1(C_1) & q_2(C_2) & \dots & q_m(C_m) \\ D_1 & D_2 \dots & & D_m \end{matrix}\mathrm{PrSrt}\begin{matrix} A_1 & A_2 \dots & A_n \\ w_1(B_1) & w_2(B_2) \dots & w_n(B_n) \end{matrix}$.

Similarly, for ($**_{\mathrm{F.1}}$): $\mathrm{PrSprt}\begin{matrix} A_1 & A_2 \dots & A_n \\ w_1(B_1) & w_2(B_2) \dots & w_n(B_n) \end{matrix}$, for ($***_{\mathrm{F.1}}$): $\begin{matrix} q_1(C_1) & q_2(C_2) & \dots & q_m(C_m) \\ D_1 & D_2 \dots & & D_m \end{matrix}\mathrm{PrSprt}$. The result of this process will be described by the expression $\begin{matrix} q_1(C_1) & q_2(C_2) & \dots & q_m(C_m) \\ D_1 & D_2 \dots & & D_m \end{matrix}\mathrm{PrSrt}$.

Let us introduce the concepts Cha, the capacity measure, and Cca, the measure of its content. Cca is the same as the number of capacity content items. In contrast to the classical one-attribute set theory, where only its contents are taken as a set, we consider a two-attribute set theory with a set as a capacity and separately with its contents. We introduce the designations: CoQ—the contents of the capacity Q.

PrSprt – elements.

Definition F.1.1.1. The set of elements $\{g\} = (g_1, g_2, ..., g_n)$ at one point x = $(x_1, x_2, ..., x_n)$ of space X we shall call PrSprt – element, and such a point in space is called parallel capacity of the PrSprt – element. We shall denote $\mathrm{PrSprt}\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Definition F.1.1.2. $\mathrm{PrSprt}\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$ - a parallel dynamic set $\{g\}$ at x.

Definition F.1.1.3. An ordered set of elements at one point in the space is called an ordered PrSprt–element.

It's possible to $\mathrm{PrSprt}\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$ correspond to the set of elements $\{g\}$, and the ordered PrSprt - element - a vector, a matrix, a tensor, a directed segment in the case when the totality of elements is understood as a set of elements in a segment.

It's allowed to sum PrSprt – elements: $PrSprt\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix} + PrSprt\begin{matrix} b_1 & b_2 & \dots & b_n \\ x_1 & x_2 & \dots & x_n \end{matrix} = \mathrm{PrSprt}\begin{matrix} g_1 \cup b_1 & g_2 \cup b_2 & \dots & g_n \cup b_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

It's allowed to multiply PrSprt – elements: $PrSprt\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix} * PrSprt\begin{matrix} b_1 & b_2 & \dots & b_n \\ x_1 & x_2 & \dots & x_n \end{matrix} = PrSprt\begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \dots & g_n \cap b_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

The operator $PrSprt\begin{matrix} g_1 \cup b_1 & g_2 \cup b_2 & \dots & g_n \cup b_n \\ x_1 & x_2 \dots & & x_n \end{matrix}$ is not equal the set of $g_i \cup b_i$, (i = 1, 2, …, n), rather, it is Parallel dynamic — contraction of the set of $g_i \cup b_i$, (i = 1, 2, …, n), to the point x. Similarly, for $PrSprt\begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \dots & g_n \cap b_n \\ x_1 & x_2 \dots & & x_n \end{matrix}$. This is more suitable for using sets for energy space, for any objects. The operator PrSprt is adapted for ordinary energies, using their property to overlap.

Parallel capacity in itself.

Definition F.1.1.4. The capacity PrSrt$\begin{matrix} g_1 & g_2 & \cdots & g_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$ is called the parallel capacity A = $(A_1,\ A_2,\ \dots,\ A_n)$ for $(g_1,\ g_2,\ \dots,\ g_n)$.

Definition F.1.1.4.1. The parallel capacity A in itself of the first type is the parallel capacity containing itself as an element. Denote PrS_1fA. PrS_1fA = PrSrt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$.

Definition F.1.1.5. The parallel capacity A in itself of the second type is the parallel capacity that contains elements from which it can be generated. Denote Pr S_2fA.

An example of the parallel capacity in itself of the first type is a set containing itself in parallel. An example of parallel capacity in itself of the second type is a living organism since it contains a program: DNA and RNA.

Definition F.1.1.6. Partial parallel capacity A in itself of the third type is the parallel capacity A in itself, which partially contains itself or contains elements from which it can be generated in part or both simultaneously. Let us denote PrS_3fA.

Let us introduce the following notations: A*B=PrSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$, A^2 = PrSelf A $=PrSrt^A_A$, A^3 = PrSelf^{2}A, …, A^{n+1} = PrSelf^{n}A, ….There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \dots$, etc.

You can consider a more "hard" option: A*B=$PPrSprt^A_B$, where $PPrSprt^A_B$ – operator, containing A in every element of B, A^2 = PPrSelf A $=PPrSprt^A_A$, A^3 = PPrSelf^{2}A, …, A^{n+1} = PPrSelf^{n}A, ….There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \dots$, etc.

All parallel capacities in parallel self-space are parallel capacities in themselves by definition. Parallel capacities in themselves can appear as PrSprt -capacities and ordinary capacities. In these cases, the usual measures and methods of topology are used.

Connection of PrSprt – elements with parallel capacities in themselves.
For example, $PrSrt_{g\{R\}}^{\{R\}}$ is the parallel capacity in itself of the second type if $g\{R\}$ is a parallel program capable of generating $\{R\}$.
Consider a third type of parallel capacity in itself. For example, based on $PrSprt_{x_1\ x_2\ \dots\ x_n}^{g_1\ g_2\ \dots\ g_n}$, where $\{g\} = (g_1,\ g_2,\ \dots,\ g_n)$, i.e. n - elements at one point x = $(x_1, x_2, ..., x_n)$, we can consider the capacity PrS_3f in itself with m elements from $\{g\}$, m<n, which is formed according to the form (1.1), that is, the structure PrS_3f contains only m elements, or in forms (1.1.1) - (1.1.5), summarizing it. Capacities in themselves of the third type can be formed for any other structure, not necessarily PrS_3f only by necessarily reducing the number of elements in the structure, in particular, using form (1.2). Structures more complex than PrS_3f can be introduced. For example, through a form (1.3), where A is compressed (fits) in C in the compression structure B in C (i.e., in the structure PrS_3f); or through the forms (1.3.1) - (1.4) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc. (1.3.1) - (1.3.4) schematically interpret the formation of the capacity in itself through a pseudo 3-connected form with a 2-connected form.

Math Prself.

Let's consider PrSprt arithmetic first :

1. Simultaneous parallel addition of sets elements $\{g_i\} = \left(g_{i_1},\ g_{i_2},\ \dots,\ g_{i_{m_j}}\right)$, i = 1,2,. . . ,n, j =1,2,. . . ,k is carried out using $PrSprt_{x_1\ \ \ x_2\ \dots\ \ \ x_n}^{\{g_1\}+\ \{g_2\}+\ \dots\ \{g_n\}+}$.

2. Similarly, for simultaneous parallel multiplication: $PrSprt_{w_1\ \ \ w_2\ \dots\ \ \ w_n}^{\{g_1\}*\ \{g_2\}*\ \dots\ \{g_n\}*}$ gives the set B_l, l =1,2,. . . ,n, with elements $b_{l i_1 i_2 \dots i_{m_j}} =$ $Sprt_{x_l}^{\left\{g_{l_{1_{i_1}}}*,\quad g_{l_{2_{i_2}}}*,\dots,\quad g_{l_{m_j}{}_{i_{m_j}}}\right\}})_{R_l}\quad for\ any\ \left\{l_{i_1}, l_{i_2}, ..., l_{i_{m_j}}\right\}$ $without\ repetitions, x_l = Sprt_w{}^{\{K_l\}}$,K_l-set of any $\left\{k_{l_{i_1}}*, k_{l_{i_2}}*, ..., k_{l_{i_{m_j}}}*\right\}$ without repeating them, l = 1,2,. . . ,n, $k_{l_{i_j}}$-any digit, i =1,2,. . . , m_j, R_l= $Sprt_w^{\left\{l_{i_1}+, l_{i_2}+,\dots, l_{i_{m_j}}\right\}}$, R_l is the index of the lower discharge (we choose an index on the scale of discharges): see Table I.1.

Then $PrSprt_{x_1\ \ \ x_2\ \dots\ \ \ x_n}^{B_1+\ B_2+\ \dots\ B_n+}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. The most straightforward functional scheme of the assumed arithmetic-logical device for PrSprt-multiplication:

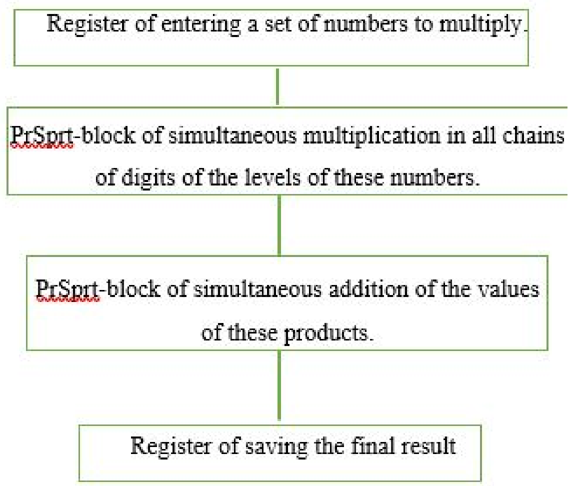


Fig. 0.1 The straightforward functional scheme of the assumed arithmetic-logical device for PrSprt-multiplication.

Remark. F.1.1.2. The algorithm for simultaneously adding a set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by multiplying the first number from the set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the set by the ones following it, etc.

3. Similarly for simultaneous execution of various operations: $PrSprt^{g_1 q_1\ g_2 q_2\ \dots\ g_n q_n}_{w_1\ \ w_2\ \dots\ \ w_n}$, where $\{q\} = (q_1, q_2, .., q_n)$. q_i -an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $PrSprt^{F_1 g_1\ F_2 g_2\ \dots\ F_n g_n}_{w_1\ \ w_2\ \dots\ \ w_n}$, where $\{F\} = (F_1, F_2, ..., F_n)$. F_i is an operator, i = 1, . . . , n.

5. The arithmetic itself for capacities in themselves will be similar: addition - $PrS_1 f\ \{g+\}$, (or $PrS_3 f\ \{g+\}$) for the third type), multiplication $PrS_1 f\ \{g*\}$, ($PrS_3 f\ \{g*\}$).

6. Similarly with different operations: $PrS_1f\{gq\}$, $(PrS_3f\{gq\})$, and with different operators: $PrS_1f\{Fg\}$, $(PrS_3f\{Fg\})$.

7. $PrSrt\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n\end{matrix}$ $-$ *the result of the containment operator*. For sets A_i, B_i, (i = 1, 2, …, n), we have

$PrSrt\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n\end{matrix} = \left\{\sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \ , \sum_{i=1}^{n} D_i\right\} = \left\{\begin{matrix} \sum_{i=1}^{n} D_i \\ \sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \end{matrix}\right\}$, where D_i is self-set for $A_i \cap B_i$ (i = 1, 2, …, n). There is the same for structures if they are considered as sets. Similarly, $\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m\end{matrix}$PrSrt =

$$\left\{\begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\ \} & \{\ \} \dots & \{\ \} \\ D_1 - D_1 \cap C_1 & D_2 - D_2 \cap C_2 \ \dots & D_m - D_m \cap C_m \end{matrix} PrSrt \\ \sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix}\right\},$$

where Q_i is Proself-set for $(D_i \cap C_i)$ [14].

8. PrSprt-derivative of f($x_1, x_2, \dots, x_n$) = $\begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \dots \\ f_k(x_1, x_2, \dots, x_n) \end{pmatrix}$ is $PrSprt\begin{matrix} \frac{\partial}{\partial x_{1_i}} & \frac{\partial}{\partial x_{2_i}} \dots & \frac{\partial}{\partial x_{k_i}} \\ f_1(x_1, x_2, \dots, x_n) & f_2(x_1, x_2, \dots, x_n) \ \dots & f_k(x_1, x_2, \dots, x_n) \end{matrix}$, where x=(x_{1_i}, x_{2_i},… x_{k_i})- any set from ($x_1, x_2, \dots, x_n$). The same is done for PrSprt- $\frac{\partial^k f(x)}{\partial x_{1_i} \partial x_{2_i} \dots \partial x_{k_i}}$. PrSprt-integral off ($x_1, x_2, \dots, x_n$) is $PrSprt\begin{matrix} \int ()dx_{1_i} & \int ()dx_{2_i} \dots & \int ()dx_{k_i} \\ f_1(x_1, x_2, \dots, x_n) & f_2(x_1, x_2, \dots, x_n) \ \dots & f_k(x_1, x_2, \dots, x_n) \end{matrix}$, where ($x_{1_i}$, x_{2_i},… x_{k_i})- any set from ($x_1, x_2, \dots, x_n$). The same is done for PrSprt- $\int \dots \int f(x) dx_{1_i} dx_{2_i} \dots dx_{k_i}$-k-multiple integral. PrSprt-lim off ($x_1, x_2, \dots, x_n$) is $PrSprt\begin{matrix} \lim\limits_{x_{1_i} \to a_{1_i}} & \lim\limits_{x_{2_i} \to a_{2_i}} \dots & \lim\limits_{x_{k_i} \to a_{k_i}} \\ f_1 x_1, x_2, \dots, x_n) & f_2 x_1, x_2, \dots, x_n) \dots & f_k x_1, x_2, \dots, x_n) \end{matrix}$

The same is done for PrSprt- $\begin{pmatrix} \lim f(x_1, , x_2, \dots, x_n) \\ x_{1_i} \to a_{1_i} \\ \dots \\ x_{k_i} \to a_{k_i} \end{pmatrix}$. $PrS_3 f\{\lim\limits_{x \to a}\}$ $= PrSprt\begin{matrix} \lim\limits_{x_{1_i} \to a_{1_i}} & \lim\limits_{x_{2_i} \to a_{2_i}} \dots & \lim\limits_{x_{k_i} \to a_{k_i}} \\ \lim\limits_{x_{1_i} \to a_{1_i}} & \lim\limits_{x_{2_i} \to a_{2_i}} \dots & \lim\limits_{x_{k_i} \to a_{k_i}} \end{matrix}$

9. In the case of Prself-derivatives, inclusions of multiple derivatives are obtained. The same is true for Prself-integrals: we get inclusions of multiple integrals.

10. Let's denote Prself-(Prself-Q) through Prself^2-Q , fS(n,Q)= Prself-(Prself-(…(Prself-Q))) = Prself^n-Q for n-multiple Prself.

Operator Pritself.

Definition F.1.1.7. An operator that transforms PrSprt$\begin{matrix} g_1 & g_2 & \cdots & g_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$, into any Pr$S_i f\{b\}$, $i = 2,3$; where $\{b\} \subset \{g\}, \{g\} = (g_1, g_2, \ldots, g_n)$; is the operator Pritself.

Example. The operator contains the set in Pritself.

Lim-Pritself.

1. Lim PrSprt

   For example, the double limit: $\begin{matrix} \lim G(x,y) \\ x \to g_1 \\ y \to g_2 \end{matrix}$ corresponds to

   Pr $Sprt\begin{matrix} G(x,y) & G(x,y) \\ x \to g_1 & y \to g_2 \end{matrix}$.
   Similarly, for lim PrSprt with n variables.

In the case of lim-Pritself, for example, for m variables, it suffices to use the form (1.1) of lim PrSprt for n variables (n>m). The same is true for integrals of variables m (for example, the double integral over a rectangular region is through the double limit).

The sequence of actions can be "collapsed" into an ordered PrSprt element, and then translate it, for example, into PrS_3f- the parallel capacity in itself. Take the receipt $\frac{\partial^2 u}{\partial x^2}$ as an example. Here is the sequence of steps 1) $\frac{\partial u}{\partial x}$ 2) $\frac{\partial}{\partial x}(\frac{\partial u}{\partial x})$. "collapses" into an ordered $PrSprt_x^{\{\frac{\partial u}{\partial x}, \frac{\partial}{\partial x}(\frac{\partial u}{\partial x})\}}$, which can be translated into the corresponding PrS_1f. The differential operator $PrSprt_x^{\{\frac{\partial}{\partial x}, \frac{\partial}{\partial x}(\frac{\partial}{\partial x})\}}$ - is interesting too.

About PrSprt and PrS$_3$f programming.

The ideology of PrSprt and PrS$_3$f can be used for programming. Here are some of the PrSprt programming operators.

1. Simultaneous assignment of the expressions $\{p\} = (p_1, p_2, ..., p_n)$ to the variables $\{g\} = (g_1, g_2, ..., g_n)$. This is implemented via PrSprt$\begin{matrix} g_1 & g_2 & \cdots & g_n \\ p_1 & p_2 & \cdots & p_n \end{matrix}$.
2. Simultaneous checking the set of conditions $\{f\} = \left(f_1,\ f_{2,}\ ..., f_n\right)$ for the set of expressions$\{B\} = \left(B_1,\ B_{2,}\ ..., B_n\right)$. Implemented via PrSprt$\begin{matrix} IF\{B_1 f_1\}\ then & IF\{B_2 f_2\}\ then\ \ldots & IF\{B_n f_n\}\ then \\ x_1 & x_2 \ldots & x_n \end{matrix}$, where x_i (i = 1, . . . , n) can be anything.
3. Similarly for loop operators and others.

PrS_3f- software operators will differ only in that the aggregates $\{g\},\{p\},\{B\},\{f\}$ will be formed from the corresponding PrSprt program operators in form (1.1) or forms (1.1.1) - (1.4) for more complex operators.

The OS (operating system), the computer's principles, and the modes of operation for this programming are interesting. But this is already the material for the following articles.

Using elements of the mathematics of PrSrt we introduce the concept of PrSrt – the change in physical quantity B: PrSrt$\begin{matrix}\Delta_1 B & \Delta_2 B & \dots & \Delta_n B \\ x_1 & x_2 & \dots & x_n\end{matrix}$, . Then the mean PrSrt - velocity will be $v_{cpPrsrt}(t, t)$ = PrSrt $\begin{matrix}\frac{\Delta_1 B}{\Delta t} & \frac{\Delta_n B}{\Delta t} & \dots & \frac{\Delta_n B}{\Delta t} \\ x_1 & x_2 & \dots & x_n\end{matrix}$ and PrSrt-velocity at time t:$v_{Prsrt} = \lim_{\Delta t \to 0} v_{Prsrt}(t, \Delta t)$. PrSrt – acceleration $a_{Prsrt} = \frac{dv_{Prsrt}}{dt}$.

When using PrSprt with "target weights", we get, depending on the "target weights", one or another modification.

Remark F.1.1.2.1. PrSprt – elements with "target weights" become peculiar. Here you need the necessary energy to carry them out. As a rule, this energy is at the level of PrSelf. This is natural since it's much easier to manage elements of the k level via the elements of a more structured k +1 level. Let us consider the concepts of capacities of physical objects in themselves. The question arises about the self-energy of the object. In particular, Pr Srt_B^B will mean PrS_1f B. For example, PrSprt_{DNA}^{DNA} allows you to reach the level of DNA self-energy, $PrSprt_Q^Q$ allows you to reach the level of self-energy Q. The law of self-energy conservation operates already at the level of self-energy. Also, in addition to capacities in themselves, you can consider the types of containment of oneself in oneself: the first type of the containment of oneself in oneself: the second type of the containment of oneself in oneself: potentially, for example, in the form of programming oneself, the third type is partial containment of oneself in themselves—for example, Prself-operator, Prself-action, whirlwind. A container containing itself can be formed by self-containment, i.e., containment in oneself. Let us clarify the concept of the term capacity in itself: it is a capacity containing itself potentially. Consider Prself-Q, where Q can be anything, including Q=Prself; in particular, it can be any action. Therefore, Prself-Q is when Q is made by Pritself; it makes itself. There is a partial Prself-Q for any Q with partial Prself-fulfillment. Let's consider several examples for capacities in themselves: ordinary lightning, electric arc discharge, and ball lightning.

PrSprt is also great for working with structures, for example: 1) PrSprt$\begin{matrix}strA_1 & strA_2 & \dots & strA_n \\ B_1 & B_2 & \dots & B_n\end{matrix}$ - the structure A_i that fits into B_i, where B_i (i = 1, …, n) can be any capacity, another structure etc. 2)

PrSprt$\begin{matrix} str_{Q_1} & str_{Q_2} & \dots & str_{Q_n} \\ B_1 & B_2 & \dots & B_n \end{matrix}$ – embedding structure from Q_i into B_i. Similarly for displacement: 1) $\begin{matrix} C_1 & C_2 & \dots & C_m \\ strD_1 & strD_2 & \dots & strD_m \end{matrix}$PrSprt - displacement of structure $strD_i$ from C_i, (i = 1, . . . , n), 2) $\begin{matrix} C_1 & C_2 & \dots & C_m \\ str_{Q_1} & str_{Q_2} & \dots & str_{Q_m} \end{matrix}$ PrSprt -displacement of the structure Q_i from C_i, (i = 1, . . . , m). To work with structures, you can introduce a special operator PrCprt: PrCprt$\begin{matrix} strA_1 & strA_2 & \dots & strA_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$ structures B_i with the structure A_i, (i = 1, . . . , n), PrCprt$\begin{matrix} str_{Q_1} & str_{Q_2} & \dots & str_{Q_n} \\ B_1 & B_2 & \dots & B_n \end{matrix}$ structures B_i with the structure from Q_i, (i = 1, . . . , n), $\begin{matrix} C_1 & C_2 & \dots & C_m \\ strD_1 & strD_2 & \dots & strD_m \end{matrix}$PrCprt destructors C_i by the structure $strD_i$, $\begin{matrix} C_1 & C_2 & \dots & C_m \\ str_{Q_1} & str_{Q_2} & \dots & str_{Q_m} \end{matrix}$PrCprt destructors C_i from the structure that structures Q_i, (i = 1, . . . , m).

Definition F.1.1.8. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its elements explicitly, but not necessarily in known operators; it can form (create) new special operators (in particular, special functions).

In particular, PrCprt$\begin{matrix} strA_1 & strA_2 & \dots & strA_n \\ strA_1 & strA_2 & \dots & strA_n \end{matrix}$, PrCrt$\begin{matrix} strA_1 & strA_2 & \dots & strA_n \\ strA_1 & strA_2 & \dots & strA_n \end{matrix}$ are such structures.

Similarly, for working with models, each is structured by its structure; for example, use PrSprt-groups, PrSprt-rings, PrSprt-fields, PrSprt-spaces, Prself-groups, Prself-rings, Prself-fields, and Prself-spaces. Like any task, this is also a structure of the appropriate capacity.

PrSelf-H (Prself-hydrogen), like other Prself-particles, does not exist in the ordinary, but all Prself-molecules, Prself-atoms, and Prself-particles are elements of the energy space.

Remark F.1.1.2.2 The ideology of PrSprt elements allows us to go to the border of the world familiar to us, which allows us to act more effectively.

Dynamic PrSprt – elements.

We considered stationary PrSprt – elements earlier. Here we consider dynamic PrSprt – elements.

Definition F.1.2.1. The process of fitting a set of elements $\{g(t)\}$ = $(g_1(t), g_2(t), ..., g_n(t))$ into one point x= $(x_1, x_2, ..., x_n)$ of the space X at time t will be called a dynamic PrSprt – element. We will denote PrSprt(t)$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Definition F.1.2.2. Fitting an ordered set of elements into one point in space is called a dynamic ordered PrSprt–element.

It is allowed to sum dynamic PrSprt – elements:

$$PrSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} + PrSprt(t)\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} =$$
$$\mathrm{PrSprt}(t)\begin{matrix} g_1(t) \cup b_1(t) & g_2(t) \cup b_2(t) & \dots & g_n(t) \cup b_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix}$$

It's allowed to multiply PrSprt – elements:

$$PrSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} * PrSprt(t)\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} =$$
$$PrSprt\begin{matrix} g_1(t) \cap b_1(t) & g_2(t) \cap b_2(t) & \dots & g_n(t) \cap b_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix}$$

Parallel dynamic containment of oneself.

Definition F.1.2.3. Parallel dynamic Sprt-capacity $PrSprt(t)\begin{matrix} R_1(t) & R_2(t) & \dots & R_n(t) \\ Q_1(t) & Q_2(t) & \dots & Q_n(t) \end{matrix}$ is the process of embedding $R_i(t)$ into $Q_i(t)$, (i = 1, . . . , n), simultaneously.

Definition F.1.2.4. Parallel dynamic capacity A(t) = $(A_1(t), A_2(t), ..., A_n(t))$ containing itself as an element of the first type is the process of parallel containing A(t) in A(t): $PrSprt(t)\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ A_1(t) & A_2(t) & \dots & A_n(t) \end{matrix}$. Denote Pr $S_1 f(t) A(t)$.

Definition F.1.2.5. Parallel dynamic capacity C(t) in itself of the second type is the process of parallel containing elements from which it can be parallel generated. Let's denote $PrS_2 f(t) C(t)$.

Definition F.1.2.6. Parallel dynamic partial capacity B(t) in itself of the third type is a process of partial parallel containment of B(t) in itself or parallel embedding elements from which it can be parallel generated partially or both at the same time. Denote Pr $S_3 f(t) B(t)$.

All parallel dynamic capacities in a parallel dynamic self-space are, by definition, parallel dynamic capacities in themselves. Parallel dynamic capacity itself can manifest itself as parallel dynamic Sprt-capacity and ordinary parallel dynamic capacity. In these cases, the usual measures and methods of topology are used.

Connection of dynamic PrSprt – elements with parallel dynamic containment of oneself.

Consider third type of parallel dynamic partial containment of oneself. For example, based on $PrSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$, i.e. n – elements at one point x = $(x_1, x_2, ..., x_n)$, we can consider the parallel dynamic capacity in itself $PrS_3 f(t)$ with m elements from $\{g(t)\}$, m<n, which is process formed

according to the form (1.1), that is, only m elements from $\{g(t)\}$ are in the structure $PrSprt(t)\begin{matrix} g_1(t) & g_2(t) & \ldots & g_n(t) \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.

Parallel dynamic containment of oneself of the third type can be formed for any other structure, not necessarily PrSprt, only through the obligatory reduction in the number of elements in the structure. In particular, using the forms (1.1.1) – (1.4).

It is possible to introduce structures more complex than $PrS_3f(t)$.

Parallel dynamic math itself.

1. The process of simultaneous parallel addition of sets elements $\{g_i(t)\} = \left(g_{i_1}(t),\ g_{i_2}(t),\ \ldots,\ g_{i_{m_j}}(t)\right)$, i = 1,2,. . . ,n, j =1,2,. . . ,k are realized by $PrSprt(t)\begin{matrix} \{g_1(t)\}+ & \{g_2(t)\}+ & \ldots & \{g_n(t)\}+ \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.

2. By analogy, for simultaneous multiplication:
$PrSprt(t)\begin{matrix} \{g_1(t)\}* & \{g_2(t)\}* & \ldots & \{g_n(t)\}* \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.

3. Similarly for simultaneous execution of various operations:
$PrSprt(t)\begin{matrix} g_1(t)q_1(t) & g_2(t)q_2(t) & \ldots & g_n(t)q_n(t) \\ w_1 & w_2 & \ldots & w_n \end{matrix}$, where $\{q(t)\} = (q_1(t), q_2(t), .., q_n(t))$.
$q_i(t)$-an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $PrSprt(t)\begin{matrix} F_1(t)g_1(t) & F_2(t)g_2(t) & \ldots & F_n(t)g_n(t) \\ w_1 & w_2 & \ldots & w_n \end{matrix}$, where $\{F(t)\} = (F_1(t), F_2(t), ..., F_n(t))$. $F_i(t)$ is an operator, i = 1, . . . , n.

5. Parallel dynamic arithmetic itself for containments of oneself will be similar: Parallel dynamic addition - $PrS_1f(t)\ \{g(t)+\}$, (or $PrS_3f(t)\ \{g(t)+\}$ for the third type), Parallel dynamic multiplication $PrS_1f(t)\ \{g(t)*\}$, ($PrS_3f(t)\ \{g(t)*\}$).

6. Similarly with different operations: $PrS_1f(t)\ \{g(t)q(t)\}$, ($\Pr S_3f(t)\ \{g(t)q(t)\}$) and with different operators: $PrS_1f(t)\ \{F(t)g(t)\}$, ($\Pr S_3f(t)\ \{F(t)g(t)\}$).

7. $PrSprt(t)\begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$ *gives* the result

$\left\{\sum_{i=1}^{n} A_i(t) \cup B_i(t) - A_i(t) \cap B_i(t)\ , \sum_{i=1}^{n} D_i(t)\right\}$, *for sets* $A(t)$, $B(t)$, where $D_i(t)$ is self-set for $A_i(t) \cap B_i(t)$, (i = 1, 2, . . . , n). The same is true for structures if they are treated as sets,

$$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\text{PrSrt(t)} =$$

$$\left\{ \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\,\} & \{\,\}\ldots & \{\,\} \\ D_1(t) - D_1(t) \cap C_1(t) & D_2(t) - D_2(t) \cap C_2(t)\ \ldots & D_2(t) - D_2(t) \cap C_2(t) \\ \multicolumn{3}{c}{\sum_{i=1}^{m} \big(C_i(t) - D_i(t) \cap C_i(t)\big) - \left(D_i(t) - D_i(t) \cap C_i(t)\right)} \end{matrix} PrSrt \right\},$$

where $Q_i(t)$ is oself-set for $\big(D_i(t) \cap C_i(t)\big)$ [14].

8. Similarly, for dynamic PrSprt-derivatives, dynamic PrSprt-integrals, dynamic PrSprt-lim, parallel dynamic self-derivatives, parallel dynamic self-integrals

9. Denote parallel dynamic self-(parallel dynamic self-Q(t)) through parallel dynamic self^2-Q(t) , pfS(t)(n,Q(t))= parallel dynamic self-(parallel dynamic self-(...((parallel dynamic self)-Q(t)))) = (parallel dynamic self^n)-Q(t) for n-multiple parallel dynamic self.

Remark F.1.2.3. The parallel dynamic PrSprt-displacement will be denote by $\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\text{PrSprt(t)}$, where D_1(t) is forced out of C_1(t), D_2(t) is forced out of C_2(t), ..., D_m(t) is forced out of C_m(t) simultaneously, the result of this process will be described by the expression $\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\text{PrSrt(t)}$. Then the notation

$$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix} PrSprt(t) \begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$$

where A_1(t) fits into B_1(t), A_2(t) fits into B_2(t), ..., A_n(t) fits into B_n(t), D_1(t) is forced out of C_1(t), D_2(t) is forced out of C_2(t), ..., D_m(t) is forced out of C_m(t) simultaneously. It is dynamic PrSprt-containment of A_i(t) in B_i(t) and dynamic PrSprt-displacement of D_j(t) from C_j(t) simultaneously, (i = 1, 2, ..., n, j = 1, 2, ..., m). The result of this process will be described by the expression

$$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix} PrSrt(t) \begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$$

$PrSprt(t) \begin{matrix} B_1(t) & B_2(t) & \ldots & B_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$ will mean PrS_1f(t)B(t).

$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ C_1(t) & C_2(t) & \ldots & C_m(t) \end{matrix} PrSprt(t)$ denotes the parallel dynamic expelling C(t) = $(C_1(t), C_2(t), .., C_m(t))$ oneself out of oneself, $\begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ A_1(t) & A_2(t) & \ldots & A_n(t) \end{matrix} PrSprt(t) \begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ A_1(t) & A_2(t) & \ldots & A_n(t) \end{matrix}$ —simultaneous parallel dynamic containment A(t) = $(A_1(t), A_2(t), .., A_n(t))$. of oneself in oneself and parallel dynamic expelling A(t) oneself out of oneself.

$\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ A_1(t) & A_2(t) & \dots & A_n(t) \end{matrix} PrSprt(t)$ will be called parallel dynamic anti-capacity from oneself. For example, "white hole" in physics is such simple anti-capacity.

About dynamic PrSprt and $PrS_3f(t)$ programming.

The ideology of dynamic PrSprt and $PrS_3f(t)$ can be used for programming:

1. The process of simultaneous assignment of the expressions $\{p(t)\} = (p_1(t), p_2(t), ..., p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$. is implemented through PrSprt$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ p_1(t) & p_2(t) & \dots & p_n(t) \end{matrix}$.

2. The process of simultaneous check the set of conditions $\{f(t)\} = \left(f(t)_1,\ f_{2,}\ (t)\ ..., f(t)_n\right)$ for a set of expressions$\{B(t)\} = \left(B_1(t),\ B_2\ (t),\ ..., B_n(t)\right)$ is implemented through
PrSprt$\begin{matrix} IF\ \{B_1(t)f_1(t)\}\ then & IF\ \{B_2(t)f_2(t)\}\ then & \dots & IF\ \{B_n(t)f_n(t)\}\ then \\ x_1(t) & x_2(t) & \dots & x_n(t) \end{matrix}$,
where x(t) = $(x_1(t), x_2(t), ..., x_n(t))$. can be any.

3.Similarly for loop operators and others.

$\Pr S_3 f(t)$– software operators will differ only in that the aggregates $\{g(t)\}, \{p(t)\}, \{B(t)\}, \{f(t)\}$ will be formed from corresponding processes PrSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) – (1.4) for more complex operators.

Remark F.1.2.3.1 It is the parallel containment of oneself in oneself that can "give birth" to the parallel capacities in itself – that is what parallel self-organization is.

Remark F.1.2.4.
$PrSprt(t) \begin{matrix} d & d & \dots & d \\ d & d & \dots & d \end{matrix}$, d = PrSprt(t) $\begin{matrix} B_1(t) & B_2(t) & \dots & B_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}$, can increase parallel self- level of B(t) = $(B_1(t), B_2(t), ..., B_n(t))$.

Remark F.1.2.5. For example, the operator Pritself is $PrS_1f(t)$.

Remark F.1.2.6. May be considered the following derivatives:
$\frac{dPrSrt(t) \begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}}{dt}$, $\frac{d \begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}^{PrSprt(t)}}{dt}$,
$\frac{d \begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}^{PrSprt(t)} \begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}}{dt}$, $\frac{dPrS_i f(t)}{dt}$, i=1,2,3.

Remark F.1.2.7. It is the parallel containment of oneself in itself as an element that can be interpreted as parallel dynamic capacities in itself.

Remark F.1.2.8. Not every capacity parallel containing itself as an element will manifest itself as a sedentary parallel capacity or parallel capacity.

PrSprt – elements for continual sets.

Earlier, we considered finite-dimensional discrete PrSprt-elements and self-capacities in itself as an element. Here we believe some continual PrSprt-elements and continual parallel self-capacities in themselves as an element.

Definition 3.1. The set of continual elements $\{g\} = (g_1, g_2, ..., g_n)$ at one point x $= (x_1, x_2, ..., x_n)$ of space X will be called continual PrSprt – element, and such a point in space will be called parallel capacity of the continual PrSprt – element. We will denote PrSprt$\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Definition F.1.3.2. An ordered set of continual elements at one point in space is called an ordered continual PrSprt–element.

It's allowed to sum continual PrSprt – elements: : $PrSprt\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix} + PrSprt\begin{matrix} b_1 & b_2 & \dots & b_n \\ x_1 & x_2 & \dots & x_n \end{matrix} = PrSprt\begin{matrix} g_1 \cup b_1 & g_2 \cup b_2 & \dots & g_n \cup b_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where some or any elements may be ordered elements. It's allowed to multiply PrSprt – elements: $PrSprt\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix} * PrSprt\begin{matrix} b_1 & b_2 & \dots & b_n \\ x_1 & x_2 & \dots & x_n \end{matrix} = PrSprt\begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \dots & g_n \cap b_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Definition F.1.3.3. The continual Prself-capacity A in itself as an element of the first type is the capacity parallel containing itself as an element. Denote PrS_1fA. $PrS_1fA =$ PrSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$.

Definition F.1.3.4. The ordered continual Prself-capacity A in itself as an element of the first type is the ordered capacity parallel containing itself as an element. Denote $\overrightarrow{PrS_1fA}$.

For example, $PrSprt\begin{matrix} sin\infty & tg(-\infty) & \dots & sin(-\infty) \\ x_1 & x_2 & \dots & x_n \end{matrix} =$

$PrSprt\begin{matrix} \uparrow I \downarrow_{-1}^{1}. & \downarrow I \uparrow_{-\infty}^{\infty} & \dots & \downarrow I \uparrow_{-1}^{1} \\ x_1 & x_2 & \dots & x_n \end{matrix}$, don't confuse with values of these functions.

Definition F.1.3.5. The continual Prself-capacity A in itself, as an element of the second type, is the capacity parallel containing elements from which it can be parallel generated. Let's denote Pr S_2fA.

An example of continual self-capacity in itself as an element of the second type is a living organism since it contains the programs: DNA and RNA.

Definition F.1.3.6. Partial continual Prself-capacity in itself as an element of the third type is called continual Prself-capacity in itself as an element that partially parallel contains itself or parallel contains elements from which it can be parallel generated in part or both simultaneously. Denote PrS_3f.

All continual capacities in Prself-space are continual Prself-capacities in itself as an element by definition. The continual Prself-capacities in itself as an element may appear as continual PrSrt- capacities and usual continual capacities. In these cases, there are used typical measure and topology methods.

The connection of continual PrSprt – elements with continual Prself-capacities in themselves as an element.
Consider a third type of continual Prself-capacity in itself as an element. For example, based on $PrSprt\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where $\{g\} = (g_1,\ g_2,\ \dots,\ g_n)$, i.e. n - continual elements at one point x = $(x_1, x_2, ..., x_n)$, The continual Prself-capacity in itself as an element with m continual elements from $\{g\}$, at m<n, can be considered as PrS_3f, which is formed by the form (1.1), i.e., only m continual elements are located in the structure$PrSprt\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. Continual self-capacities in itself as an element of the third type can be formed for any other structure, not necessarily Sit, only by obligatory reducing the number of continual elements in the structure. In particular, using the forms (1.1.1) – (1.4). Structures more complex than PrS_3f can be introduced.

Mathematics itself for continual elements.

1.Simultaneous parallel addition of the sets continual elements $\{g_i\} = \left(g_{i_1},\ g_{i_2},\ \dots,\ g_{i_{m_j}}\right)$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, k$, is implemented using $PrSprt\begin{matrix} \{g_1\}\cup & \{g_2\}\cup & \dots & \{g_n\}\cup \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

2. By analogy, for simultaneous multiplication: $PrSprt\begin{matrix} \{g_1\}\cap & \{g_2\}\cap & \dots & \{g_n\}\cap \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

3. Similarly for simultaneous execution of various operations: $PrSprt\begin{matrix} g_1q_1 & g_2q_2 & \dots & g_nq_n \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\{q\} = (q_1, q_2, .., q_n)$. q_i -an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $PrSprt\begin{matrix} F_1g_1 & F_2g_2 & \dots & F_ng_n \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\{F\} = (F_1, F_2, ..., F_n)$. F_i is an operator, i = 1, . . . , n.
5. The arithmetic itself for parallel continual capacities in themselves will be similar: addition - $PrS_1f\,\{g+\}$, (or $PrS_3f\,\{g+\}$) for the third type), multiplication $PrS_1f\,\{g*\}$, ($PrS_3f\,\{g*\}$).
6. Similarly with different operations: $PrS_1f\,\{q\}$, ($PrS_3f\,\{q\}$), and with different operators: $PrS_1f\,\{F\}$, ($PrS_3f\,\{F\}$).

7. $PrSrt^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n}$ $-$ *the result of the parallel containment operator.*

For continual sets A_i, B_i, (i = 1, 2, . . . , n), we have

$PrSrt^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n} = \left\{\sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i\ ,\sum_{i=1}^{n} D_i\right\}$, where D_i is Prself-set for $A_i \cap B_i$ (i = 1, 2, . . . , n). There is the same for structures if they are considered as continual sets, ${}^{C_1\ C_2\ \dots\ C_m}_{D_1\ D_2\ \dots\ D_m}\mathrm{PrSrt}$ =

$$\left\{\begin{matrix}\sum_{i=1}^{m} Q_i + {}^{\{\ \}}_{D_1 - D_1 \cap C_1}\ {}^{\{\ \}\ \dots}_{D_2 - D_2 \cap C_2\ \dots}\ {}^{\{\ \}}_{D_m - D_m \cap C_m} PrSrt \\ \sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i)\end{matrix}\right\},$$

where Q_i is oself-set for $(D_i \cap C_i)$ [14].

Remark F.1.3.1.Consider ${}^{C_1\ C_2\ \dots\ C_m}_{D_1\ D_2\ \dots\ D_m}\mathrm{PrSprt}$, where continual D_1 is forced out of continual C_1, continual D_2 is forced out of continual C_2, . . . , continual D_m is forced out of continual C_m. Consider ${}^{C_1\ C_2\ \dots\ C_m}_{D_1\ D_2\ \dots\ D_m}\mathrm{PrSprt}^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n}$, where continual A_1 fits into continual B_1, continual A_2 fits into continual B_2, . . . , continual A_n fits into continual B_n, continual D_1 is forced out of continual C_1, continual D_2 is forced out of continual C_2, . . . , continual D_m is forced out of continual C_m simultaneously.

We can consider the concept of a continual PrSprt - element as $\mathrm{PrSprt}^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n}$, where continual A_1 fits into continual B_1, continual A_2 fits into continual B_2, . . . , continual A_n fits into continual B_n. Then $SPrSprt^{B_1\ B_2\ \dots\ B_n}_{B_1\ B_2\ \dots\ B_n}$ will mean $\mathrm{PrS_1f}$ B.

These elements are used for PrSprt-coding, PrSprt translation, coding Prself, and translation Prself for networks, which is suitable for electric current of ultrahigh frequency. More complex elements can be considered as continual sets of numbers with their " activation " in mutual directions. For example, ranges of function values, particularly those representing the shape of lightning. Differential geometry can be applied here. Also, n-dimensional elements can be considered. The space of such elements is Banach space if we introduce the usual norm for functions or vectors. We call this space-- Selb-space. Then we introduce the scalar product for functions or vectors and get the Hilbert space. We call this space Selh-space. In particular, one can try to describe some processes with these elements by differential equations and use methods from [4]. You can also try to optimize and research some processes with these elements using the techniques from [5]. Let's introduce operators for transforming capacity to Prself-capacity in itself as an element: $\mathrm{PrQ_1S(A)}$ transforms A to $\mathrm{Prf_1SA}$, $\mathrm{PrQ_0S(C)}$ transforms C to ${}^{C_1\ C_2\ \dots\ C_m}_{C_1\ C_2\ \dots\ C_m}\mathrm{PrSprt}$.

Can be considered Q($\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix} PrfSprt \begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$), Q-any operator.

Dynamic continual PrSprt – elements.

Definition F.1.4.1. The process of containing the set of continual elements $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ into one point x= $(x_1, x_2, ..., x_n)$ of the space X at time will be called the dynamic continual PrSprt – element. We will denote PrSprt(t)$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix}$.

Definition F.1.4.2. The process of containing an ordered set of continual elements at one point in space is called dynamic continual ordered PrSprt – element.

It is allowed to sum dynamic continual PrSprt – elements:

$$PrSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix} + PrSprt(t)\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix} =$$

PrSprt$(t)\begin{matrix} g_1(t) \cup b_1(t) & g_2(t) \cup b_2(t) & \dots & g_n(t) \cup b_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix}$

It's allowed to multiply dynamic continual PrSprt – elements: $PrSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix} * PrSprt(t)\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix} =$ $PrSprt\begin{matrix} g_1(t) \cap b_1(t) & g_2(t) \cap b_2(t) & \dots & g_n(t) \cap b_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix}$.

Parallel dynamic continual containment of oneself in oneself as an element.

Definition F.1.4.3. Parallel dynamic continual capacity Q(t) = $(Q_1(t), Q_2(t), ..., Q_n(t))$ is parallel fitting into Q(t) = $(Q_1(t), Q_2(t), ..., Q_n(t))$: $PrSprt(t)\begin{matrix} Q_1(t) & Q_2(t) & \dots & Q_n(t) \\ Q_1(t) & Q_2(t) & \dots & Q_n(t) \end{matrix}$.

Definition F.1.4.4. The dynamic continual PrSprt-capacity $PrSprt(t)\begin{matrix} R_1(t) & R_2(t) & \dots & R_n(t) \\ Q_1(t) & Q_2(t) & \dots & Q_n(t) \end{matrix}$ is the process of embedding continual R_i(t) into continual Q_i(t), (i = 1, . . . , n).

Definition F.1.4.5. The dynamic parallel containment continual A(t) of oneself of the first type is the process of parallel putting A(t) into A(t). Denote Pr $S_1 f(t)A(t)$.

Definition F.1.4.6. The dynamic parallel containment continual C(t) of oneself of the second type parallel contains the continual elements from which it can be parallel generated. Denote $PrS_2 f(t)C(t)$.

Definition F.1.4.7. The partial parallel dynamic containment continual B(t) of oneself of the third type is the process of partial parallel embedding continual

B(t) into oneself or parallel embedding continual elements from which it can be parallel generated in part or both simultaneously. Denote $\Pr S_3 f(t) B(t)$.

The connection of dynamic continual PrSprt – elements with parallel dynamic containment of oneself in oneself as an element.

Let us consider the partial parallel dynamic continual containment of oneself in oneself as an element of the third type. For example, based on $PrSprt(t)^{\substack{g_1(t) \; g_2(t) \; \dots \; g_n(t) \\ x_1 \quad x_2 \; \dots \quad x_n}}$, where $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$, i.e. n - continual elements at one point x= $(x_1, x_2, ..., x_n)$, one can consider the parallel dynamic containment $PrS_3 f(t)$ of oneself in oneself as an element with m continual elements from $\{g(t)\}$, m<n, which is a process that is necessary form according to the form (1.1), i.e.„ only m continual elements from $\{g(t)\}$ are located in the structure $PrSprt(t)^{\substack{g_1(t) \; g_2(t) \; \dots \; g_n(t) \\ x_1 \quad x_2 \; \dots \quad x_n}}$. Parallel dynamic continual containments of oneself in oneself as an element of the third type can be formed for any other structure, not necessarily PrSprt, only by necessarily reducing the number of continual elements in the structure. In particular, with the help of forms (1.1.1) – (1.4). It is possible to introduce structures more complex than $PrS_3f(t)$.

Parallel dynamic continual mathematics self.

1. The process of simultaneous parallel addition of sets continual elements $\{g_i(t)\} = \Big(g_{i_1}(t),\ g_{i_2}(t),\ \dots,\ g_{i_{m_j}}(t)\Big)$, i = 1,2,. . . ,n, j =1,2,. . . ,k are realized by $PrSprt(t)^{\substack{\{g_1(t)\}\cup \; \{g_2(t)\}\cup \; \dots \; \{g_n(t)\}\cup \\ x_1 \qquad x_2 \; \dots \qquad x_n}}$.
2. By analogy, for simultaneous multiplication: $PrSprt(t)^{\substack{\{g_1(t)\}\cap \; \{g_2(t)\}\cap \; \dots \; \{g_n(t)\}\cap \\ x_1 \qquad x_2 \; \dots \qquad x_n}}$.
3. Similarly for simultaneous execution of various operations: $PrSprt(t)^{\substack{g_1(t)q_1(t) \; g_2(t)q_2(t) \; \dots \; g_n(t)q_n(t) \\ w_1 \qquad w_2 \; \dots \qquad w_n}}$, where $\{q(t)\} = (q_1(t), q_2(t), .., q_n(t))$. q_i(t)-an operation, i = 1, . . . , n.
4. Similarly, for the simultaneous execution of various operators: $PrSprt(t)^{\substack{F_1(t)g_1(t) \; F_2(t)g_2(t) \; \dots \; F_n(t)g_n(t) \\ w_1 \qquad w_2 \; \dots \qquad w_n}}$, where $\{F(t)\} = (F_1(t), F_2(t), ..., F_n(t))$. F_i(t) is an operator, i = 1, . . . , n.
5. Parallel dynamic arithmetic itself for containments of oneself will be similar: Parallel dynamic addition - $PrS_1 f(t) \{g(t)\cup\}$, (or $PrS_3 f(t) \{g(t)\cup\}$ for the third type), Parallel dynamic multiplication $PrS_1 f(t) \{g(t)\cap\}$, $(PrS_3 f(t) \{g(t)\cap\})$.

6. Similarly with different operations: $PrS_1 f(t)\left\{g(t)q(t)\right\}$, $(\Pr S_3 f(t)\left\{g(t)q(t)\right\})$ and with different operators: $PrS_1 f(t)\left\{F(t)g(t)\right\}$, $(\Pr S_3 f(t)\left\{F(t)g(t)\right\})$.

7. . $PrSprt(t)\begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$ *gives* the result

$$PrSrt(t)\begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix} = \left\{\sum_{i=1}^{n} A_i(t)\cup B_i(t) - A_i(t)\cap B_i(t)\ ,\sum_{i=1}^{n} D_i(t)\right\},$$

for continual sets $A(t)$, $B(t)$, where $D_i(t)$ is self-set for $A_i(t)\cap B_i(t)$, (i = 1, 2, ..., n). The same is true for structures if they are treated as continual sets,

$$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\mathrm{PrSrt(t)} =$$

$$\left\{\begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\ \} & \{\ \}\ \ldots & \{\ \} \\ D_1(t)-D_1(t)\cap C_1(t) & D_2(t)-D_2(t)\cap C_2(t)\ \ldots & D_2(t)-D_2(t)\cap C_2(t) \end{matrix} PrSrt \\ \sum_{i=1}^{m}\left(\ C_i(t)-D_i(t)\cap C_i(t)\right)-(D_i(t)-D_i(t)\cap C_i(t)) \end{matrix}\right\},$$

where $Q_i(t)$ is oself-set for $\left(D_i(t)\cap C_i(t)\right)$ [14].

8. Similarly, for dynamic PrSprt-derivatives, dynamic PrSprt-integrals, dynamic PrSprt-lim, dynamic Prself-derivatives, dynamic Prself-integrals

9. Denote dynamic continual Prself-(dynamic continual Prself-Q(t)) through dynamic continual Prself^2-Q(t), PrfS(t)(n,Q(t))= dynamic continual Prself-(dynamic continual Prself-(... (dynamic continual Prself-Q(t)))) = dynamic continual Prself^n-Q(t) for n-multiple dynamic continual Prself.

Remark F.1.4.1. The parallel dynamic continual PrSprt-displacement will be denote by $\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\mathrm{PrSprt(t)}$, where continual $D_1(t)$ is forced out of continual $C_1(t)$, continual $D_2(t)$ is forced out of continual $C_2(t)$, ..., continual $D_m(t)$ is forced out of continual $C_m(t)$, the result of this process will be described by the expression $\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\mathrm{PrSrt(t)}$. Then the notation $\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix} PrSprt(t) \begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$

where continual $A_1(t)$ fits into continual $B_1(t)$, continual $A_2(t)$ fits into continual $B_2(t)$, ..., continual $A_n(t)$ fits into continual $B_n(t)$, continual $D_1(t)$ is forced out of continual $C_1(t)$, continual $D_2(t)$ is forced out of continual $C_2(t)$, ..., continual $D_m(t)$ is forced out of continual $C_m(t)$ simultaneously. It is dynamic continual PrSprt-containment of continual $A_i(t)$ in continual $B_i(t)$ and dynamic continual PrSprt-displacement of continual $D_j(t)$ from continual $C_j(t)$ simultaneously, (i = 1, 2, ..., n, j = 1, 2, ..., m). The result of this process will be described by the expression

$$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix} PrSrt(t) \begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$$

$PrSprt(t)\begin{matrix} B_1(t) & B_2(t) & \dots & B_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}$ will mean $PrS_1f(t)B(t)$.

$\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ C_1(t) & C_2(t) & \dots & C_m(t) \end{matrix} PrSprt(t)$ denotes the parallel dynamic expelling continual C(t) = $(C_1(t), C_2(t), .., C_m(t))$. oneself out of oneself, $\begin{matrix} A_1(t) & A_2(t) & \dots & A_m(t) \\ A_1(t) & A_2(t) & \dots & A_m(t) \end{matrix} PrSprt(t) \begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ A_1(t) & A_2(t) & \dots & A_n(t) \end{matrix}$ —simultaneous parallel dynamic containment continual A(t) = $(A_1(t), A_2(t), .., A_n(t))$ of oneself in oneself and parallel dynamic expelling continual A(t) oneself out of oneself. $\begin{matrix} A_1(t) & A_2(t) & \dots & A_m(t) \\ A_1(t) & A_2(t) & \dots & A_m(t) \end{matrix} PrSprt(t)$ will be called parallel dynamic anti-continual capacity from oneself.

Remark F.1.4.2. $PrSprt(t)\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ A_1(t) & A_2(t) & \dots & A_n(t) \end{matrix}$ can be interpreted as a multilayer shell of a self-object from the first layer, which is specified by $A_1(t)$ to the nth, which is specified by $A_n(t)$. Based on this, the atomic model can be interpreted as $PrSprt(t)\begin{matrix} \cup\{p, n\} & A_1(t) & \dots & A_n(t) \\ position\ of atomic\ nucleus & A_1(t) & \dots & A_n(t) \end{matrix}$' $\{p, n\}$ - protons, neutrons, $A_i(t)$ correspond to orbitals, i = 1, ..., n. $PrSprt(t)\begin{matrix} physical\ body\ of\ a\ living\ organism & V_1(t) & \dots & V_n(t) \\ position\ of physical\ body & V_1(t) & \dots & V_n(t) \end{matrix}$ - model of a living organism with the multilayer shell of a living organism from the first layer, which is specified by $V_1(t)$ to the nth, which is specified by $V_n(t)$. In humans:

$PrSprt(t)\begin{matrix} energy\ fibers\ that\ create\ physical\ body\ of\ a\ living\ organism & V_1(t) & \dots & V_n(t) \\ energy\ fibers\ that\ create\ physical\ body\ of\ a\ living\ organism & V_1(t) & \dots & V_n(t) \end{matrix}$

. You can also try to consider the operator PrSprt$\begin{matrix} B_1 & \dots & {B_i}^a & \dots B_n \\ r_1 & \dots & {r_i}^a & \dots r_n \end{matrix}$, which represents the interpretation of the position of the assemblage point on the cocoon of a living organism, r_j is its potential position, B_j is a potential set of subtle energies in this position (i = 1, ..., i-1, i+1, …, n), ${r_i}^a$ is its active position, ${B_i}^a$ is an active set of subtle energies in this position, i = 1, ..., n.

Connection of dynamic continual PrSprt – elements with target weights with parallel dynamic continual containment of oneself with target weights.
Consider a third type of parallel dynamic partial containment of oneself with target weights g(t). For example, based on PrSprt(t)$\begin{matrix} g_1(t)w(t) & g_2(t)w(t) & \dots & g_n(t)w(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$.where $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ i.e. n - continual elements with target weights{w(t)} at one point x = $(x_1, x_2, ..., x_n)$, we can consider the dynamic continual containment $PrS_3f(t)\{g(t)\}w(t)$ of oneself with target weights with m continual elements with target weights{w(t)} from $\{g(t)\}$, m<n, which is the process of formation according to the form (1.1), i.e., only m continual elements with target weights{w(t)} from $\{g(t)\}$ are located in the

structure $PrS_3f(t)\left\{g(t)\right\}w(t)$ Parallel dynamic containments of oneself with target weights of the third type can be formed for any other structure, not necessarily PrS_3f, only by reducing the number of continual elements with target weights in the structure. In particular, using the forms (1.1.1) – (1.4). Structures more complex than $PrS_3f(t)\left\{g(t)\right\}w(t)$ can be introduced.

Definition F.1.4.8. The parallel dynamic embedding of continual A(t) into itself with target weights {w(t)} of the first type is the process of parallel embedding A(t) into A(t) with target weights. Denote $PrS_1f(t)A(t)w(t)$.

Definition F.1.4.9. The parallel dynamic containment of continual C(t) itself into itself with target weights{w(t)} of the second type is the process of parallel containment of the continual elements from which it can be parallel generated. Let's denote $PrS_2f(t)C(t)w(t)$.

Definition F.1.4.10. Partial parallel dynamic containment of continual B(t) itself into itself with target weights {w(t)} of the third type is the process of partial parallel containment of continual B(t) into itself or continual elements from which it can be parallel generated partially, or both at the same time. Denote $PrS_3f(t)B(t)w(t)$.

The usage of PrSprt-elements for networks.

A. Galushkin's comprehensive monograph [6] covers all aspects of networks, but traditional approaches go through classical mathematics, mainly through the usual correspondence operators. Here we consider a different approach - through a new mathematical process with parallel containment operators, which, although they can be interpreted as the result of some correspondence operators, are not themselves correspondence operators. Parallel containment operators are more convenient for networks. Also, the main emphasis was placed on using processors operating using triodes, which are generally not used in PrSprt-networks. PrSprt-networks (PrSmnsprt) are a PrSprt-structure that can be built for the required weights. PrSprt-OS (PrSprt operating system) uses PrSprt-coding and PrSprt-translation. In the first one, coding is carried out through a 2-dimensional matrix-row (a, b), where the number b is the code of the action, and the number a is the code of the object of this action. PrSprt-coding (or self-coding) is implemented through a matrix consisting of 2 columns (in the continuous case, two intervals of numbers). Here, the source encoding is used for all matrix rows simultaneously. PrSprt-translation is carried out by inversion. In this case, self-coding and self-translation will be more stable. The target weights $g_i(t)$ in PrSprt(t)$\substack{activation\ with\ g_1(t)\\ SmnPrSprt}\ \substack{activation\ with\ g_2(t)\ \dots\\ SmnPrSprt\ \dots}\ \substack{activation\ with\ g_n(t)\\ SmnPrSprt}$ are chosen for necessary tasks. We will not touch on the issues of applications, or network optimization. They are described in detail by Galushkin [6]. We will touch on the difference of this only for hierarchical complex networks. The same simple executing programs are in the cores of simple artificial neurons of type PrSprt (designation - mnPrSprt) for simple information processing. More complex executing programs are used for mnPrSprt nodes. PrSprt-threshold element $-\mathrm{sgn}(PrSprt(t)^{\begin{matrix} g_1(t)w_1(t) & g_2(t)w_n(t) & \dots & g_n(t)w_n(t) \\ x_1 & x_2 \dots & & x_n \end{matrix}})$, x=($x_1$,$x_2$,…,$x_n$) - mnPrSprt, w(t)=(w_1(t), w_2(t),…,w_n(t)) – source signals values, $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ – PrSprt-synapses weights. The first level of mnPrSprt consists of simple mnPrSprt. The second level of mnPrSprt consists of PrSprt(t)$^{\begin{matrix} mnPrSprt & mnPrSprt & \dots & mnPrSprt \\ D_1 & D_2 \dots & & D_n \end{matrix}}$– PrSprt-node of mnPrSprt in range D = (D_1, D_2,…, D_n), D- capacity for mnPrSprt node. The third level of mnPrSprt consists of PrSprt(t)$^{\begin{matrix} PrSprt(t)^{\begin{matrix} q & q \dots & q \\ D_1 & D_2 \dots & D_n \end{matrix}} & PrSprt(t)^{\begin{matrix} q & q \dots & q \\ D_1 & D_2 \dots & D_n \end{matrix}} & \dots & q \\ D_1 & D_2 \dots & & D_n \end{matrix}}$ -PrSprt^2-node of q = mnPrSprt in range D, thus D becomes capacity of itself in itself as an element for mnPrSprt. For our networks, it is sufficient to use PrSprt^2-nodes of mnPrSprt, but self-level is higher in living organisms, particularly PrSprt^n-, n3. The target structure or the corresponding program enters the target unit using alternating current. After that, all networks or parts of them are activated according to the indicative goal. It may appear that we are leaving the network ideology, but these networks are a complex hierarchy of different levels, like living organisms.

Remark F.1.5.0. A neural network can be thought of as a learnable parallel dynamic operator.

Remark F.1.5.1. Traditional scientific approaches through classical mathematics make it possible to describe only at the usual energy level. Here we consider an approach that makes describing processes with finer energies possible. mnPrSprt contains PrSprt(t)$^{\begin{matrix} eprogram_1(t) & eprogram_2(t) & \dots & eprogram_n(t) \\ mnPrSprt & mnPrSprt \dots & & mnPrSprt \end{matrix}}$, $eprogram$–executing program in PrSprt- OS. PrSprt-OS (or Prself-OS) is based on PrSprt-assembly language (or Prself-assembly language), which is based on assembly language through PrSprt-approach in turn, if the base of elements of PrSprt-networks is sufficient. The eprograms are in PrSprt-programming environments (or Prself-programming environments), but this question and PrSprt-networks base will be considered in the following monographs. In particular, eprograms may contain PrSprt- programming operators. In mnPrSprt cores, the constant memory PrSprt with correspondent eprograms depending on mnPrSprt.

The OS (operating system) and the principles and modes of operation of the PrSprt-networks for this programming are interesting. But this is already the material for the next publications.

Here is developed a helicopter model without a main and tail rotors based on PrSprt – physics and special neural networks with artificial neurons operating in normal and PrSprt-modes. Let's denote this model through SmnPrSprt. To do this, it's proposed to use mnPrSprt of different levels; for example, for the usual mode, mnPrSprt serves for the initial processing of signals and the transfer of information to the second level, etc., to the nodal center, then checked. In case of an anomaly - local PrSprt–mode with the desired "target weight" is realized in this section, etc., to the center. In the case of a monster during the test, SmnPrSprt is activated with the desired "target weight." Here are realized other tasks also. To reach the self-energy level, the mode $Sprt_{SmnPrSprt}^{SmnPrSprt}$ is used. In normal mode, it's planned to carry out the movement of SmnPrSprt on jet propulsion by converting the energy of the emitted gases into a vortex to obtain additional thrust upwards. For this purpose, a spiral-shaped chute (with "pockets") is arranged at the bottom of the SmnPrSprt for the gases emitted by the jet engine, which first exit through a straight chute connected to the spiral one. There is drainage of exhaust gases outside the SmnPrSprt. SmnPrSprt is represented by a neural network that extends from the center of one of the main clusters of PrSprt - artificial neurons to the shell, turning into the body itself. Above the operator's cabin is the central core of the neural network and the target block, responsible for performing the "target weights" and auxiliary blocks, the functions and roles of which we will discuss further. Next is the space for the movement of the local vortex. The unit responsible for SmnPrSprt's actions is below the operator's cab. In PrSprt – mode, the entire network or its sections are PrSprt – activated to perform specific tasks, in particular, with "target weights." In the target, block used PrSprt -coding, PrSprt -translation for activation of all networks to "target weights" simultaneously, then –the reset of this PrSprt-coding after activation. Unfortunately, triodes are not suitable for PrSprt -neural networks. In the most primitive case, usual separators with corresponding resistances and cores for eprograms may be used instead triodes since there is no necessity to unbend the alternating current to direct. The PrSprt-operative memory belt is disposed around a central core of SmnPrSprt. There are PrSprt-coding, PrSprt-translation, and PrSprt-realize of eprograms and the programs from the archives without extraction, PrSprt-coding and PrSprt-translation may be used in high-intensity, ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna, Strickland. PrSprt – structure or an eprogram if one is present of needed «target weight» are taken in target block at PrSprt – activation of the networks. $Sprt_{activation}^{SmnPrSprt,f}$ derives SmnPrSprt to the self-level boundary with target weight f.

It's used an alternating current of above high frequency and ultra-violet light, which can work with PrSprt – structures in PrSprt–modes by its nature to activate the networks or some of its parts in PrSprt–modes and locally using PrSprt–mode. Above high frequently alternating current go through mercury bearers. That's why overheating does not occur. The power of the alternating current above high frequently increases considerably for the target block. The activation of all networks is realized to indicate "target weights."

Variable hierarchical dynamical parallel structures (models) for

dynamic, singular, hierarchical sets.

Here we will consider variable parallel structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable structures (models), for example,

$$
{}^{C_1\ C_2\ \dots\ C_m}_{D_1\ D_2\ \dots\ D_m}\mathrm{PrSprt(t)}^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n} =
\begin{cases}
{}^{C_1\ C_2\ \dots\ C_m}_{D_1\ D_2\ \dots\ D_m}PrSprt, \quad q_2 \geq t \geq q_1 \\
{}^{B_1\ B_2\ \dots\ B_m}_{D_1\ D_2\ \dots\ D_m}PrSprt^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n}, q_3 \geq\ t > q_2 \\
{}^{C_1\ C_2\ \dots\ C_m}_{D_1\ D_2\ \dots\ D_m}PrSprt^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n}, q_4 \geq\ \ t > q_3 \\
PrSprt^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n}, q_5 \geq\ t > q_4 \\
{}^{\{\ \}\ \{\ \}\ \dots\ \{\ \}}_{D_1\ D_2\ \dots\ D_m}PrSprt, \quad t > q_5 \\
\dots
\end{cases}
\quad (*_{\mathrm{F.1.1}}),
$$

In particular, ${}^{B_1\ B_2\ \dots\ B_m}_{D_1\ D_2\ \dots\ D_m}PrSprt^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n}$ can be interpreted as a game: player 1 fits A_i into B_i, i = 1, 2, ..., n, and the other pushes D_j out of Bj, j = 1, 2, ..., m at the same time.

The example of variable parallel hierarchy ${}^{C_1\ C_2\ \dots\ C_m}_{D_1\ D_2\ \dots\ D_m}PrSprt_1(t)^{A_1\ A_2\ \dots\ A_n}_{B_1\ B_2\ \dots\ B_n}$

$$
=\begin{cases}
\left\{\begin{array}{c} \sum_{i=1}^{m} Q_i + {}^{\{\ \}\qquad\qquad \{\ \}\ \dots \qquad\qquad \{\ \}}_{D_1 - D_1\cap C_1\ D_2 - D_2 \cap C_2\ \dots\ D_m - D_m \cap C_m}PrSrt \\ \sum_{i=1}^{m}(\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{array}\right\}, \quad q_2 \geq t \geq q_1 \\
\sum_{i=1}^{n}\left(\begin{array}{c} S{1}^{e}_{0} f B_i * \\ {}^{B_i}_{Q_i - B_i}S^1 t^{A_i - B_i}_{B_i} \end{array}\right), q_3 \geq\ t > q_2 \\
\sum_{j=1}^{m}\sum_{i=1}^{n}\left(\begin{array}{c} S^{et}_{01} f B_i \\ {}^{C_j - B_i}_{C_j - B_i}S_1 t^{A - B_i}_{B_i} \\ {}^{C_j - B_i}_{D_j - C_j - B_i}S_1 t^{A - B_i}_{B_i} \end{array}\right) \quad , q_4 \geq\ \ t > q_3 \\
\left\{\begin{array}{c} \sum_{i=1}^{n} R_i \\ \sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \end{array}\right\}, \ , q_5 \geq\ t > q_4 \\
{}^{\{\ \}\ \{\ \}\ \dots\ \{\ \}}_{D_1\ D_2\ \dots\ D_m}PrSrt, \quad t > q_5 \\
\dots
\end{cases}
$$

$(*_{\mathrm{F.1.2}})$,

where Q_j is oself-set for $\left(D_j \cap C_j\right)(j = 1, 2, \ldots, m)$ [14], R_i is self-set for $A_i \cap B_i$ (i = 1, 2, . . . , n), $S_{01}^{et} f B$, ${}_{C-B}^{C-B}S_1 t_B^{A-B}$, ${}_{D-C-B}^{C-B}S_1 t_B^{A-B}$ are considered in [13], ${}_{Q-B}^{B}S^1 t_B^{A-B}$ is considered in [10].

In what follows, we will denote variable parallel structure (model) through PrVS, parallel self-type variable structures (models) through PrSVS, and parallel oself-type variable structures (models) through PrOSVS.

Examples: a) discrete variable parallel structure

$$\begin{array}{c} C_1\ C_2 \ldots\ C_m \\ D_1\ D_2 \ldots\ D_m \\ A_1\ A_2 \ldots\ A_n \\ B_1\ B_2 \ldots\ B_n \end{array} \begin{array}{ccc} a & b & g \\ c & PrVS & w \\ d & q & r \end{array} \begin{array}{c} A_1\ A_2 \ldots\ A_n \\ B_1\ B_2 \ldots\ B_n \\ C_1\ C_2 \ldots\ C_m \\ D_1\ D_2 \ldots\ D_m \end{array}$$

c) continuous variable parallel structure

Fig. 0.2 Continuous set as the rim.

Where a continuous set represents the rim of the Fig. F.0.2.

We introduce the notation m_{VS_N}– the number of elements, N - the number of connections between them in the discrete variable parallel 2-hierarchical structure PrVS. We introduce the notation $q_{\Pr VS_R}$– any, R - connections in $q_{\Pr VS_R}$ in the variable parallel 2-hierarchical structure PrVS, in particular, $q_{\Pr VS_R}$, R can be sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the structurability of Q from the interval [0,1], where 0 corresponds to "no parallel structure"," and 1 corresponds to the value " parallel structure". Then for joint A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D- parallel self-type structures from A*B, cS(x)- the value of Prself for parallel self-type structures x; for dependent parallel structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional structurability of the parallel structure B at the parallel structure A, c(A/B)- conditional structure of the parallel structure A at the parallel structure B. Adding inconsistent parallel structures: c(A+B) =c(A) + c(B). The formula of complete parallel structure: c(A)=$\sum_{k=1}^{n} c\left(B_k\right) * c(A/B_k)$, B_1, $B_{2.}$,.., B_n-full group of hypotheses- containments: $\sum_{k=1}^{n} c\left(B_k\right)$=1(“parallel structure”). PrSprt- structure of the first type for set of parallel structures A={A_1, A_2,. . . ,A_n}: PrSprt$\begin{smallmatrix} A_1 & A_2 & \ldots & A_n \\ x_1 & x_2 & \ldots & x_n \end{smallmatrix}$, PrSprt$\begin{smallmatrix} c(A_1) & c(A_2) & \ldots & c(A_n) \\ x_1 & x_2 & \ldots & x_n \end{smallmatrix}$ PrSprt- structurability for these structures. It is possible to consider the parallel self-type structure PrS_3A with m parallel structures from A, at $m<n$, which is formed by the form (1.1), that is, only m parallel structures from A are located in the structure $PrSprt^{A_1\ A_2\ \dots\ A_n}_{x_1\ x_2\ \dots\ x_n}$. The same for parallel self-type structurability $PrS_3\{\ c(A_1), c(A_2), \dots, c(A_n)$.

Can be considered N-hierarchical parallel structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical parallel structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, $\dots$ can be any in particular, by actions, sets, and others.

Can be considered discrete hierarchical parallel structure, continuous hierarchical parallel structure, and discrete-continuous hierarchical parallel structure.

The example: PrQHS=

$$HSprt_x^{\;PrSprt^{\,PrSprt^{\,PrSprt^{1-lhs_1\ 1-lhs_2\ \dots\ 1-lhs_n}_{x_1\quad x_2\ \dots\quad x_n}\ \dots}}}$$

$$PrSprt^{N-lhs_1\ N-lhs_2\ \dots\ N-lhs_n}_{x_1\qquad x_2\ \dots\qquad x_n}\quad \dots\quad PrSprt^{i-lhs_1\ i-lhs_2\ \dots\ i-lhs_n}_{x_1\qquad x_2\ \dots\qquad x_n}\quad \dots\quad PrSprt^{1-lhs_1\ 1-lhs_2\ \dots\ 1-lhs_n}_{x_1\qquad x_2\ \dots\qquad x_n}\quad HSprt_x$$

-

N-hierarchical structure compression into point x = $(x_1, x_2, ..., x_n)$, lhs = level of hierarchical structure.

Let Prf(N, PrQHS)= $PrQHS^{\left.PrQHS^{PrQHS^{\cdots^{PrQHS}}}\right\}\text{-N levels}}$

It can be considered Prself- PrQHS, Prf(y, PrQHS) for any y, Prf(PrQHS, PrQHS).

Parallel Compression Hierarchy Example:

$$PrSprt^{\;PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\quad PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\ \dots\quad PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}}_{\;PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}+B\quad PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}+B\ \dots\quad PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}+B}$$

$$=\left(\begin{array}{l} PrSprt^{\;PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\ PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\ \dots\ PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}}_{\;PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\ PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\ \dots\ PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}} \\ PrSprt^{\;PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\ PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\ \dots\ PrSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}}_{\quad B\qquad\qquad B\ \dots\qquad\qquad B} \end{array}\right)$$

Let's consider two versions: 1) containment is interpreted through the concept of containment, and 2) capacity is interpreted through the concept of containment as a rest point of containment. PrSelf-containment is interpreted as a rest point of Prself-containment. We consider the functional ca(Q), which gives a numerical value for the accommodation of Q from the interval [0,1], where 0 corresponds to " parallel containment", and one corresponds to the value " parallel capacity". Then for joint A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D- Prself-containment for A*B, caS(x)- the value of Prself-capacity for Prself-containment of x; for dependent parallel containments: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional accommodation of the parallel containment B at the parallel containment A, ca(A/B)- conditional parallel capacity of the parallel containment A at the parallel containment B. Adding the parallel capacity values of inconsistent parallel containments: ca(A+B)=ca(A)+ca(B). The formula of complete parallel capacity: ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2,..,B_n-full group of hypotheses-(parallel containments): $\sum_{k=1}^{n} ca\,(B_k)$=1(“parallel capacity”). PrSprt- containment for set of parallel containments A={A_1, A_2,...,A_n}: PrSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, PrSprt$\begin{matrix} ca(A_1) & ca(A_2) & \dots & ca(A_n) \\ x_1 & x_2 & \dots & x_n \end{matrix}$- PrSprt- accommodation for these parallel containments. It is possible to consider the Prself-containment PrS_3A with m containments from A, at m<n, which is formed by the form (1.1), that is, only m parallel containments from A are located in the parallel containment PrSprt(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, . The same for Prself-accommodation - $PrS_3\{\ ca(A_1), ca(A_2), \dots, ca(A_n)$.

We consider the functional h(Q), which gives a numerical value for the parallel hierarchization of Q from the interval [0,1], where 0 corresponds to "no parallel hierarchy", and 1 corresponds to the value " parallel hierarchy". Then for joint parallel hierarchies A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- Prself-hierarchy from A*B, hS(x)- the value of Prself- hierarchy for Prself- hierarchy x; for dependent parallel hierarchies: h(A*B)=ha(A)*h(B/A)=h(B)*h(A/B), where h(B/A)- conditional parallel hierarchization of the parallel hierarchy B at the parallel hierarchy A, h(A/B)- conditional parallel hierarchy of the parallel hierarchy A at the parallel structure B. Adding the parallel hierarchy values of inconsistent parallel hierarchies: h(A+B)=h(A)+h(B). The formula of complete parallel hierarchy: h(A)=$\sum_{k=1}^{n} h\,(B_k) * h(A/B_k)$, B_1, B_2, .., B_n-full group of hypotheses-(parallel hierarches): $\sum_{k=1}^{n} h\,(B_k)$=1(“parallel hierarchy”).

PrSprt- structure for set of parallel hierarches A={A_1, A_2,...,A_n}: $PrSprt(t)\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, PrSprt(t)$\begin{matrix} h(A_1) & h(A_2) & \dots & h(A_n) \\ x_1 & x_2 & \dots & x_n \end{matrix}$- PrSprt- hierarchization for these parallel hierarches. It is possible to consider the Prself-hierarchy PrS_3A with m parallel hierarches from A, at m<n, which is formed by the form (1.1), that is, only m parallel hierarches from A are

located in the parallel hierarchy $PrSprt(t)^{A_1\ A_2\ \ldots\ A_n}_{x_1\ x_2\ \ldots\ x_n}$. The same for Prself- hierarchization $PrS_3\{\ h(A_1), h(A_2), \ldots, h(A_n)$.Can be considered PrSprt(t)$^{\{\ ca(A_1), c(A_1), h(A_1)\}\ \ \{ca(A_2), c(A_2), h(A_2)\}\ \ldots\ \ \{ca(A_n), c(A_n), h(A_n)\}}_{x_1 \qquad\qquad\qquad x_2\ \ldots \qquad\qquad\qquad x_n}$.

Very interesting next parallel hierarchy type:

$\frac{h\ A_1\ h\ A_2\ \ldots\ h\ A_n}{h\ A_1\ h\ A_2\ \ldots\ h\ A_n}$PrSprt(t)$^{h\ A_1\ h\ A_2\ \ldots\ h\ A_n}_{h\ A_1\ h\ A_2\ \ldots\ h\ A_n}$, h = hierarchy. You can enter special operator PrCprt to work with structures: $\frac{A_1\ \ A_2\ \ldots\ \ A_n}{B_1\ \ B_2\ \ldots\ \ B_n} PrCprt \frac{R_1\ \ R_2\ \ldots\ \ R_m}{Q_1\ \ Q_2\ \ldots\ \ Q_m}$ structures R_j with the structure from Q_j, unstructures A_i by the structure B_i, simultaneously, (i = 1, 2, …, n, j = 1, 2, …, m). Very interesting next parallel structure type:

$\frac{A_1\ A_2\ \ldots\ A_n}{A_1\ A_2\ \ldots\ A_n}$, PrCrt$^{A_1\ A_2\ \ldots\ A_n}_{A_1\ A_2\ \ldots\ A_n}$,

You can enter special parallel operator PrHprt to work with hierarches: $\frac{A_1\ \ A_2\ \ldots\ \ A_n}{B_1\ \ B_2\ \ldots\ \ B_n} PrHprt \frac{R_1\ \ R_2\ \ldots\ \ R_m}{Q_1\ \ Q_2\ \ldots\ \ Q_m}$

hierarchizes R_j with the hierarchy from Q_j, unhierarchizes A_i from the hierarchy B_i, simultaneously, (i = 1, 2, …, n, j = 1, 2, …, m).

Program operators PrSprt, PrtSpr, S^1e, Set_1.

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions. This must be done through PrSprt-Networks in one of the central departments of which a conventional computer system is located. The parallel processor is itself preprogram with direct parallel computing not through serial computing.

Using conventional parallel coding by a parallel computer system, through a Target-block with a PrSprt -program operator - $PrSprt(t)^{g_1(t)w_1(t)\ g_2(t)w_n(t)\ \ldots\ g_n(t)w_n(t)}_{activation\ activation\ \ldots\ activation}$, it will be possible to obtain the execution of the parallel actions $(g_1(t), g_2(t), ..., g_n(t))$ with the desired target weights w(t)=($w_1(t)$, $w_2(t)$,…,$w_n(t)$). Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For PrSprt-coding and PrSprt-translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel program of operator PrSprt(t)$^{(UHF\ AC)_1(t)Q_1(t)\ (UHF\ AC)_2(t)Q_2(t)\ \ldots\ (UHF\ AC)_n(t)Q_n(t)}_{activation \qquad\qquad activation\ \ldots \qquad\qquad activation}$, we simultaneously enter the desired set of codes $Q_i(t)$, i = 1, 2, …, n, using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel program operators:

1) PrSprt-program operators
2) PrtSpr-program operators
3) S^1e - program operators
4) Set_1- program operators

Here are some of the PrSprt-program operators:

1. Simultaneous assignment of the expressions $\{p\} = (p_1, p_2, ..., p_n)$ to the variables $\{g\} = (g_1, g_2, ..., g_n)$. This is implemented via PrSprt$\begin{smallmatrix} g_1 & g_2 & \cdots & g_n \\ p_1 & p_2 & \cdots & p_n \end{smallmatrix}$.
2. Simultaneous checking the set of conditions $\{f\} = \left(f_1,\ f_2,\ ..., f_n\right)$ for the set of expressions$\{B\} = \left(B_1,\ B_2,\ ..., B_n\right)$. Implemented via PrSprt$\begin{smallmatrix} IF\,\{B_1 f_1\}\ then & IF\,\{B_2 f_2\}\ then\ \ldots & IF\,\{B_n f_n\}\ then \\ x_1 & x_2\ \ldots & x_n \end{smallmatrix}$, where x_i (i = 1, . . . , n) can be anything.
3. Similarly for loop operators and others.

PRSprt-algorithm Examples:

1) Simultaneous addition and simultaneous parallel multiplication of sets elements $\{g_i\} = \left(g_{i_1},\ g_{i_2},\ \ldots,\ g_{i_{m_j}}\right)$, i = 1, 2, . . . , n, j =1, 2, . . . , k (See point 1, 2 in **Math Prself**
2) parallel pattern recognition: PrSprt$\begin{smallmatrix} b_1 & b_2\ \ldots & b_n \\ Name\ of\ q_1 & Name\ of\ q_2\ \ldots & Name\ of\ q_n \end{smallmatrix}$'

$b_i = IF\,\{q_i\ image\ archive_i\}\ then, i = 1, 2, \ldots, n$. The example of PrSprt-program is

$$PrSprt^{\begin{smallmatrix} r_1 & PrSprt^{\begin{smallmatrix} IF\,\{B_1 f_1\}\ then & IF\,\{B_2 f_2\}\ then\ \ldots & IF\,\{B_n f_n\}\ then \\ w_1 & w_2\ \ldots & w_n \end{smallmatrix}} & \ldots & r_2 \\ x_1 & x_2\ \ldots & & x_n \end{smallmatrix}},$$

$r_1 = PrSprt^{\begin{smallmatrix} g_1 := & g_2 :=\ \ldots & g_n := \\ p_1 & p_2\ \ldots & p_n \end{smallmatrix}}, r_2 = PrSprt^{\begin{smallmatrix} w_1 & w_2 & \ldots & w_n \\ w_1 & w_2 & \ldots & w_n \end{smallmatrix}}.$

Pr $S_3 f-$ software operators will differ only just because aggregates

$\{g\},\{p\},\{B\},\{f\}$ will be formed from corresponding PrSprt-program operators in form (1.1) for more complex operators in forms (1.1.1) – (1.4).

For example, $Sprt^{\{R\}}_{g\{R\}}$ is the capacity in itself of the second type if $g\{R\}$ is a program capable of generating $\{R\}$.

The example of self-program of the first type is

$$PrSprt\begin{matrix} r_1\ PrSprt\begin{matrix} IF\{B_1f_1\}\ then & IF\{B_2f_2\}\ then \ldots & IF\{B_nf_n\}\ then \\ w_1 & w_2\ldots & w_n \end{matrix} \ldots r_2 \\ r_1\ PrSprt\begin{matrix} IF\{B_1f_1\}\ then & IF\{B_2f_2\}\ then \ldots & IF\{B_nf_n\}\ then \\ w_1 & w_2\ldots & w_n \end{matrix} \ldots r_2 \end{matrix}.$$

$r_1 = PrSprt\begin{matrix} g_1 := & g_2 := \ldots & g_n := \\ p_1 & p_2\ldots & p_n \end{matrix}, r_2 = PrSprt\begin{matrix} w_1\ w_2\ \ldots\ w_n \\ w_1\ w_2\ \ldots\ w_n \end{matrix}.$
PrSprt-coding: 1) set A_i to set B_i, 2) set A_i to a point q_i, where the elements of the sets A_i, B_i can be continuous, (i = 1, 2, . . . , n; j = 1, 2, . . . , m). For example, PrSprt$\begin{matrix} A_1\ A_2\ \ldots\ A_n \\ B_1\ B_2\ \ldots\ B_n \end{matrix}$.

There are PrSprt -coding, PrSprt-translation, PrSprt-realize of preprograms and of the programs from the archives without extraction theirs

Self-coding: 1) set A_i to set A_i, i.e. A_i on itself, 2) set A_i to a point q_i in forms (1.1) - (1.4), where the elements of the sets A_i can be continuous. For example, PrSprt$\begin{matrix} A_1\ A_2\ \ldots\ A_n \\ A_1\ A_2\ \ldots\ A_n \end{matrix}$.

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with PrSprt-Networks, it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through PrSprt -Networks and series-parallel. Codes from a conventional type computer system will be used via PrSprt -connectors in PrSprt - coding, for example: PrSprt(t)$\begin{matrix} (UHF\ AC)_1(t)Q_1(t) & (UHF\ AC)_2(t)Q_2(t)\ldots & (UHF\ AC)_n(t)Q_n(t) \\ activation & activation\ldots & activation \end{matrix}$.
UHF AC field activation is used.

Consider the dynamic PrSprt and $PrS_3f(t)$ programming:

1. The process of simultaneous assignment of the expressions $\{p(t)\} = (p_1(t), p_2(t), ..., p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ is implemented through PrSprt$\begin{matrix} g_1(t)\ g_2(t)\ \ldots\ g_n(t) \\ p_1(t)\ p_2(t)\ \ldots\ p_n(t) \end{matrix}$.

2. The process of simultaneous check the set of conditions $\{f(t)\} = (f(t)_1,\ f_{2,}\ (t) \ldots, f(t)_n)$ for a set of expressions $\{B(t)\} = (B_1(t),\ B_2\ (t),\ \ldots, B_n(t))$ is implemented through

$$\text{PrSprt}\overset{IF\,\{B_1(t)f_1(t)\}\ then}{x_1(t)}\ \overset{IF\,\{B_2(t)f_2(t)\}\ then\ \ldots}{x_2(t)\ \ldots}\ \overset{IF\,\{B_n(t)f_n(t)\}\ then}{x_n(t)},$$

where x(t) = $(x_1(t), x_2(t), \ldots, x_n(t))$. can be any.

3. Similarly for loop operators and others.

$\Pr S_3 f(t)$– software operators will differ only in that the aggregates $\{g(t)\}, \{p(t)\}, \{B(t)\}, \{f(t)\}$ will be formed from corresponding processes PrSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) – (1.4) for more complex operators.

Consider PrtSpr-program operators. The ideology of PrtSpr and $\Pr t_{S_4 f}$ is parallel analogue of $t_{S_4 f}$ [14] can be used for programming. Here are some of the PrtSpr -program operators.

1. Simultaneous expelling assignment of the expressions $\{p\} = (p_1, p_2, \ldots, p_n)$ from the variables $\{g\} = (g_1, g_2, \ldots, g_n)$. It's implemented through $\overset{g_1}{=:p_1}\ \overset{g_2\ \ldots}{=:p_2\ \ldots}\ \overset{g_n}{=:p_n}$ PrSprt.

2. Simultaneous expelling checks the set of conditions $\{f\} = (f_1,\ f_{2,}\ \ldots, f_n)$ for a set of expressions $\{B\} = (B_1,\ B_2,\ \ldots, B_n)$. It's implemented through $\underset{IF\,\{B_1 g_1\}\ then}{x_1}\ \underset{IF\,\{B_2 g_2\}\ then\ \ldots}{x_2\ \ldots}\ \underset{IF\,\{B_n g_n\}\ then}{x_n} PrSprt$, where x_i (i = 1, …, n) can be anything.

3. Similarly for loop operators and others.

$Prt_{S_4 f}$– software operators will differ only just because aggregates $\{g\}, \{p\}, \{B\}, \{f\}$ will be formed from corresponding PrtSpr program operators in form (1.1) for more complex operators in forms (1.1.1) – (1.4).

Consider hierarchical PrtSpr-program operator

$$\begin{matrix} C_1\ C_2\ \ldots\ C_m \\ D_1\ D_2\ \ldots\ D_m \end{matrix}\text{PrSrt} =$$

$$\left\{\frac{\sum_{i=1}^{m} Q_i + \overset{\{\ \}}{D_1 - D_1 \cap C_1}\ \overset{\{\ \}\ \ldots}{D_2 - D_2 \cap C_2\ \ldots}\ \overset{\{\ \}}{D_m - D_m \cap C_m} PrSrt}{\sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i)}\right\},$$

where Q_i is oself-set for $(D_i \cap C_i)$ [14].

Consider the dynamic PrtSpr(t) and $\Pr t(t)_{S_4 f}$ programming at time t.

1. The process of simultaneous assignment of the expressions $\{p(t)\} = (p_1(t), p_2(t), ..., p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ is implemented through $\begin{matrix} g_1(t) := & g_2(t) := & \dots & g_n(t) := \\ p_1(t) & p_2(t) & \dots & p_n(t) \end{matrix}$ PrSprt(t).

2. The process of simultaneous check the set of conditions $\{f(t)\} = (f(t)_1,\ f_{2,}\ (t)\ ..., f(t)_n)$ for a set of expressions $\{B(t)\} = (B_1(t),\ B_2\ (t),\ ..., B_n(t))$ is implemented through
$\begin{matrix} IF\ \{B_1(t)f_1(t)\}\ then & IF\ \{B_2(t)f_2(t)\}\ then & \dots & IF\ \{B_n(t)f_n(t)\}\ then \\ x_1(t) & x_2(t) & \dots & x_n(t) \end{matrix}$ PrSprt(t),
where x(t) = $(x_1(t), x_2(t), ..., x_n(t))$. can be any.

3.Similarly for loop operators and others.

$\Pr t(t)_{S_4 f}$– software operators will differ only in that the aggregates $\{g(t)\}, \{p(t)\}, \{B(t)\}, \{f(t)\}$ will be formed from corresponding processes PrSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) – (1.4) for more complex operators.

Consider hierarchical dynamic PrSrt(t)-program operator:

$$\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix} \text{PrSrt(t)} =$$
$$\left\{ \begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\ \} & \{\ \} & \dots & \{\ \} \\ D_1(t) - D_1(t) \cap C_1(t) & D_2(t) - D_2(t) \cap C_2(t) & \dots & D_2(t) - D_2(t) \cap C_2(t) \end{matrix} PrSrt \\ \sum_{i=1}^{m} \left(C_i(t) - D_i(t) \cap C_i(t) \right) - \left(D_i(t) - D_i(t) \cap C_i(t)\right) \end{matrix} \right\},$$
where $Q_i(t)$ is oself-set for $\left(D_i(t) \cap C_i(t)\right)$ [14].

Consider PrSrt-program operators

$$Sprt_{t_0}^{\left\{ {}_{W_q}^{q}\begin{pmatrix} u\ u & & u\ u \\ u\ u & {}_{PrSpr} & u\ u \end{pmatrix}_{Sprt}^{E_q}{}_{\begin{pmatrix} u\ u & & u\ u \\ u\ u & {}_{PrSpr} & u\ u \end{pmatrix}}^{q},\quad Sprt_{d_r}^{\{E^{ex}{}_l{}^{d_r}\}}\left(E_{in}{}^{l^{d_r}}\right) \right\}}$$
—program structure example, where the assemblage point d_r is the cursor, it is quite complex self—program.

Remark.F.1.5.2. Energy of a living organism other than humans:

$$\text{Prf(r, u(}E_q\text{))} = Sprt_{t_0}^{\left\{ {}_{W_q}^{q}\begin{pmatrix} u\ u & & u\ u \\ u\ u & {}_{PrSpr} & u\ u \end{pmatrix}_{Sprt}^{E_q}{}_{\begin{pmatrix} u\ u & & u\ u \\ u\ u & {}_{PrSpr} & u\ u \end{pmatrix}}^{q},\quad Sprt_{d_r}^{\{E^{ex}{}_l{}^{d_r}\}}\left(E_{in}{}^{l^{d_r}}\right) \right\}} \qquad (**_{\text{F.1}}).$$

Energy of a living organism of a person:

$$\mathrm{Prhf(r, u(}E_q)) = Sprt_{t_0}^{\{ {}^{q}_{W_q}\left({}^{u\,u}_{u\,u}{}_{PrSpr}{}^{u\,u}_{u\,u}\right)^{E_q}_{Sprt_{{}^{q}\left({}^{u\,u}_{u\,u}{}_{PrSpr}{}^{u\,u}_{u\,u}\right)}},\ Sprt^{\{E^{ex}l^{d_r}\}}_{d_{r_{(self\left(E_{in}l^{d_r}\right))}}} \}} \qquad (***_{\mathrm{F.1}}).$$

${}^{u\,u}_{u\,u}PrSpr^{u\,u}_{u\,u}$ - internal energy of a living organism of double energy structure, q- a gap in the energy cocoon of a living organism, r-the position of the assemblage point d_r on the energy cocoon of a living organism, W_q- energy prominences from the gap in the cocoon of a living organism, E_q-external energy entering the gap in the cocoon of a living organism, $E^{ex}l^{d_r}$ - a bundle of fibers of external energy self-capacities from outside the cocoon, collected at the point of assembly of the cocoon of a living organism, $E_{in}l^{d_r}$- a bundle of fibers of external energy self-capacities from inside the cocoon, collected at the point of assembly of the cocoon of a living organism in the same position r of the assemblage point d_r. d_r is the subject of identifying the energy fibers of the subtle energy of the Universe in position r both outside and inside the cocoon.

$(**_{\mathrm{F.1}})$, $(***_{\mathrm{F.1}})$ can be interpreted as the program operators.

Appendix.

Supplement for string theory: May be to try represent elementary particles in the form of continual self-elements of the type: $PrSprt^{\uparrow I \downarrow_{-1}^{1} . \downarrow I \uparrow_{-\infty}^{\infty} \dots \downarrow I \uparrow_{-1}^{1}}_{\quad x_1 \qquad x_2 \dots \qquad x_n}$ etc.

Supplement for PrSprt-logic: We consider PrSprt-logic: consider the functional f(Q), which gives a numerical value for the truth of the statement Q from the interval [0,1], where 0 corresponds to "no", and one corresponds to the logical value "yes". Then for joint statements A, B: f(A+B)=f(A)+f(B)-f(A*B)+fS(D), D- self-statement from A*B, fS(x)- the value of self-truth for self-statement x; for dependent statements: f(A*B)=f(A)*f(B/A)=f(B)*f(A/B), where f(B/A)- conditional truth of the statement B at statement A, f(A/B)- dependent truth of the statement A at the statement B. Adding the truth values of inconsistent propositions: f(A+B)=F(A)+f(B). The formula of complete truth: f(A)=$\sum_{k=1}^{n} f\,(B_k) * f(A/B_k)$, B_1, B_2, .., B_n-full group of hypotheses-statements: $\sum_{k=1}^{n} f\,(B_k)$=1("yes"). Remark. A statement can be interpreted as an event, and its truth value as a probability.

PrSprt- statement for set of statements A = {A_1, A_2, …, A_n}: PrSprt$^{A_1\ A_2\ \dots\ A_n}_{x_1\ x_2\ \dots\ x_n}$. PrSprt$^{f(A_1)\ f(A_2)\ \dots\ f(A_n)}_{x_1 \quad x_2 \dots \quad x_n}$- PrSprt- truth for these statements. It is possible to consider the self-statement PrS_3A with m statements from A, at m<n, which is formed by the form (1.1), that is, only

m statements from A are located in the structure $PrSprt\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. The same for self- truth $PrS_3\{\ f(A_1), f(A_2), \dots, f(A_n)\ .$

One can introduce the concepts of PrSprt-group: PrSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, A_i is usual group, i=1,2, …,n, PrSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where A, x- usual groups, self-group: $PrSf_iA$, i=1,2,3, A is usual group.

Definition 5.1. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its elements clearly, but not necessarily in known operators; it can form (create) new special operators (in particular, special functions). In particular, PrCrt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$ is such structure. Similarly, for working with models, each is structured by its structure; for example, use PrSprt-groups, PrSprt-rings, PrSprt-fields, PrSprt-spaces, Prself-groups, Prself-rings, Prself-fields, and Prself-spaces. Like any task, this is also a structure of the appropriate capacity. Since the degree of freedom is double, it is clear that the form of the Prself-equation contains the solutions or structures the inversion of the Prself-equation concerning unknowns, i.e., the structure of the Prself-equation is complete.

F.2 Parallel fSprt – elements, self-type PrfSprt-structures

This section generalizes the previous one to fuzzy objects, in particular, to fuzzy sets.

We consider expression

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrfSprt} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \ (*_{\mathrm{F.2.1}})$$

where fuzzy A_1 fits into fuzzy B_1, fuzzy A_2 fits into fuzzy B_2, ..., fuzzy A_n fits into fuzzy B_n, fuzzy D_1 is forced out of fuzzy C_1, fuzzy D_2 is forced out of fuzzy C_2, ..., fuzzy D_m is forced out of fuzzy C_m simultaneously. The result of this process will be described by the expression

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} PrfSrt \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \ (*_{\mathrm{F.2.2}})$$

If A_1, B_1, A_2, B_2, ..., A_n, B_n, D_1, C_1, D_2, C_2, ..., D_m, C_m are taken as fuzzy sets, then we will call $(*_{\mathrm{F.2.1}})$ a parallel dynamic fuzzy set. The need $(*_{\mathrm{F.2.1}})$ arose to describe processes in networks. Threshold element PrfSprt - $\begin{matrix} B_1 & B_2 \dots & B_n \\ \{qy\}_1 & \{qy\}_2 \dots & \{qy\}_n \end{matrix} PrfSprt \begin{matrix} \{ax\}_1 & \{ax\}_2 \dots & \{ax\}_n \\ B_1 & B_2 \dots & B_n \end{matrix}$, B_1, B_2, ..., B_n - artificial neurons of type PrfSprt (designation - mnPrfSprt) , $x=(x_1,x_2,\dots,x_n)$ are the fuzzy values of the initial signals, $a=(a_1,a_2,\dots,a_n)$ are the weights of PrfSprt-synapses and the fuzzy values of the output signals $\{qy\}$. It can be considered a simpler version of the Parallel dynamic fuzzy set

$$\mathrm{PrfSprt} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \ (**_{\mathrm{F.2.1}}),$$

where set fuzzy A_1 fits into fuzzy B_1, fuzzy A_2 fits into fuzzy B_2, ..., fuzzy A_n fits into fuzzy B_n simultaneously, the result of this process will be described by the expression

$$\mathrm{PrfSr}t \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \ (**_{\mathrm{F.2.2}})$$

or

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrfSprt} \ (***_{\mathrm{F.2.1}}),$$

where fuzzy D_1 is forced out of fuzzy C_1, fuzzy D_2 is forced out of fuzzy C_2, ..., fuzzy D_m is forced out of fuzzy C_m simultaneously, the result of this process will be described by the expression

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrfSrt} \ (***_{\mathrm{F.2.2}})$$

We consider the measure: $\mu **({C_1 \ C_1 \ \ldots \ C_1 \atop D_1 \ D_2 \ \ldots \ D_m} PrfSprt {A_1 \ A_2 \ \ldots \ A_n \atop C_1 \ C_1 \ \ldots \ C_1})=$ $\frac{\mu(A_1)*\mu(A_2)*\ldots*\ \mu(A_n)}{\mu(D_1)*\mu(D_2)*\ldots*\ \mu(D_m)}$, where $\mu(A_{\mathrm{i}}), \mu(D_{\mathrm{j}})$,–usual fuzzy measures of fuzzy sets A_i, D_j (i = 1, 2, …, n; j = 1, 2, …, m).

Remark. One can consider some generalization for ($*_{F.2.1}$):
${q_1(C_1) \ q_2(C_2) \ \ldots \ q_m(C_m) \atop D_1 \quad D_2 \ldots \quad D_m} PrfSprt {A_1 \quad A_2 \ \ldots \quad A_n \atop w_1(B_1) \ w_2(B_2) \ \ldots \ w_n(B_n)}$ where fuzzy A_1 fits into fuzzy B_1 through w_1, fuzzy A_2 fits into fuzzy B_2 through w_2, …, fuzzy A_n fits into fuzzy B_n through w_n, fuzzy D_1 is forced out of fuzzy C_1 through q_1, fuzzy D_2 is forced out of fuzzy C_2 through q_2, …, fuzzy D_m is forced out of fuzzy C_m through q_m, simultaneously. A_i, B_i, D_j, C_j (i = 1, 2, …, n; j = 1, 2, …, m) can be taken as fuzzy sets. The result of this process will be described by the expression ${q_1(C_1) \ q_2(C_2) \ \ldots \ q_m(C_m) \atop D_1 \quad D_2 \ldots \quad D_m} PrfSrt {A_1 \quad A_2 \ \ldots \quad A_n \atop w_1(B_1) \ w_2(B_2) \ \ldots \ w_n(B_n)}$.

Similarly, for ($**_1$): PrfSprt${A_1 \quad A_2 \ \ldots \quad A_n \atop w_1(B_1) \ w_2(B_2) \ \ldots \ w_n(B_n)}$, for ($***_1$): ${q_1(C_1) \ q_2(C_2) \ \ldots \ q_m(C_m) \atop D_1 \quad D_2 \ldots \quad D_m}$PrfSprt. The result of this process will be described by the expression ${q_1(C_1) \ q_2(C_2) \ \ldots \ q_m(C_m) \atop D_1 \quad D_2 \ldots \quad D_m}$PrfSpr.

We construct new mathematical objects constructively without formalism. By its contradiction, formalism may destroy this thry by Gödel's theorem on the incompleteness of any formal theory. But in the next monograph, we will give the formalism of the theory it's due: the proof of axioms and theorems. Let us introduce the concepts Cha, the capacity measure, and Cca, the measure of its content. Cca is the same as the number of capacity content items. In contrast to the classical one-attribute set theory, where only its contents are taken as a set, we consider a two-attribute set theory with a set as a capacity and separately with its contents. We introduce the designations: CoQ—the contents of the capacity Q.

Definition F.2.1.1. The fuzzy set of elements $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) at one point x = $(x_1, x_2, ..., x_n)$ of space X we shall call PrfSprt – element, and such a point x in space X is called parallel fuzzy capacity of the PrfSprt – element. We shall denote PrfSprt${g_1 \ g_2 \ \cdots \ g_n \atop x_1 \ x_2 \ \ldots \ x_n}$.

Definition F.2.1.2. PrfSprt${g_1 \ g_2 \ \cdots \ g_n \atop x_1 \ x_2 \ \ldots \ x_n}$ - a parallel dynamic fuzzy set $\widetilde{g}$ at x.

Definition F.2.1.3. An ordered fuzzy set of elements at one point in the space is called an ordered PrfSprt–element.

It's possible to PrfSprt$\begin{matrix} g_1 & g_2 & \ldots & g_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ correspond to the fuzzy set $\widetilde{g}$, and the ordered PrfSprt - element - a fuzzy vector, a fuzzy matrix, a fuzzy tensor, a fuzzy directed segment in the case when the totality of elements is understood as a fuzzy set of elements in a segment.

It's allowed to sum PrfSprt – elements: $PrfSprt\begin{matrix} g_1 & g_2 & \ldots & g_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ + $PrfSprt\begin{matrix} b_1 & b_2 & \ldots & b_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ = PrfSprt$\begin{matrix} g_1 \cup b_1 & g_2 \cup b_2 & \ldots & g_n \cup b_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.

It's allowed to multiply PrfSprt – elements: $PrfSprt\begin{matrix} g_1 & g_2 & \ldots & g_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ * $PrfSprt\begin{matrix} b_1 & b_2 & \ldots & b_n \\ x_1 & x_2 & \ldots & x_n \end{matrix} = PrfSprt\begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \ldots & g_n \cap b_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$. The operator PrfSprt$\begin{matrix} g_1 \cup b_1 & g_2 \cup b_2 & \ldots & g_n \cup b_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$is not equal the fuzzy set of $g_i \cup b_i$, (i = 1, 2, …, n), rather, it is Parallel dynamic — contraction of the fuzzy set of $g_i \cup b_i$, (i = 1, 2, …, n), to the point x. Similarly, for $PrfSprt\begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \ldots & g_n \cap b_n \\ x_1 & x_2 & \ldots & x_n \end{matrix}$. This is more suitable for using fuzzy sets for energy space, for any fuzzy objects. The operator PrfSprt is adapted for ordinary energies, using their property to overlap.

Parallel fuzzy capacity in itself.

Definition F.2.1.4. The capacity PrfSprt$\begin{matrix} g_1 & g_2 & \ldots & g_n \\ A_1 & A_2 & \ldots & A_n \end{matrix}$ is called the parallel fuzzy capacity A = $(A_1,\ A_2,\ \ldots,\ A_n)$ for $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2)$, …, $g_n|\mu_{\widetilde{g}}(g_n)$).

Definition F.2.1.4.1. The parallel fuzzy capacity A in itself of the first type is the parallel fuzzy capacity containing itself as an element. Denote $PrfS_1fA$. $PrfS_1fA$ = PrfSprt$\begin{matrix} A_1 & A_2 & \ldots & A_n \\ A_1 & A_2 & \ldots & A_n \end{matrix}$.

Definition F.2.1.5. The parallel fuzzy capacity A in itself of the second type is the parallel fuzzy capacity that contains fuzzy elements from which it can be generated. Denote Pr fS_2fA.

An example of the parallel capacity in itself of the first type is a set containing itself in parallel. An example of parallel capacity in itself of the second type is a living organism since it contains a program: DNA and RNA.

Definition F.2.1.6. Partial parallel fuzzy capacity A in itself of the third type is the parallel fuzzy capacity A in itself, which partially contains itself or contains fuzzy elements from which it can be generated in part or both simultaneously. Let us denote PrfS_3fA.

Let us introduce the following notations: A*B=PrfSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$, $\mathrm{A}^2 =$ PrfSelf A $= PrfSrt_A^A$, A^3 = PrfSelf^{2}A, ..., A^{n+1} = PrfSelf$^{\mathrm{n}}$A, ...There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \dots$, etc.

You can consider a more "hard" option: A*B=$PPrfSprt_B^A$, where $PPrfSprt_B^A$ – operator, containing A in every element of B, A^2 = PPrfSelf A $= PPrfSprt_A^A$, A^3 = PPrfSelf^{2}A, ..., A^{n+1} = PPrfSelf$^{\mathrm{n}}$A, ...There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \dots$, etc.

All parallel capacities in parallel self-space are parallel capacities in themselves by definition. Parallel capacities in themselves can appear as PrfSprt -capacities and ordinary capacities. In these cases, the usual measures and methods of topology are used.

Connection of PrfSprt – elements with parallel fuzzy capacities in themselves. For example, $PrSrt_{g\{R\}}^{\{R\}}$ is the parallel capacity in itself of the second type if $g\{R\}$ is a parallel program capable of generating $\{R\}$.
Consider a third type of parallel fuzzy capacity in itself. For example, based on $PrfSprt\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where $\widetilde{g}$=(g$_1|\mu_{\widetilde{g}}$(g$_1$), g$_2|\mu_{\widetilde{g}}$(g$_2$), ..., g$_{\mathrm{n}}|\mu_{\widetilde{g}}$(g$_{\mathrm{n}}$)), i.e. n - elements at one point x = $(x_1, x_2, ..., x_n)$, we can consider the capacity $PrfS_3f$ in itself with m elements from $\widetilde{g}$, m<n, which is formed according to the form (1.1), that is, the structure $PrfS_3f$ contains only m elements, or in forms (1.1.1) - (1.1.5), summarizing it. Fuzzy capacities in themselves of the third type can be formed for any other structure, not necessarily $PrfS_3f$ only by necessarily reducing the number of elements in the structure, in particular, using form (1.2). Structures more complex than $PrfS_3f$ can be introduced. For example, through a form (1.3), where A is compressed (fits) in C in the compression fuzzy structure B in C (i.e., in the fuzzy structure $PrfS_3f$); or through the forms (1.3.1) - (1.4) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc.

(1.3.1) - (1.3.4) schematically interpret the fuzzy formation of fuzzy capacity in itself through a pseudo 3-connected form with a 2-connected form.

Remark 2.1.1. Fuzzy self, in particular, according to a fuzzy form- fuzzy analogue of the form of type (1.1): (2.1*). By analogy the same for the fuzzy form of type (1.1.1) – (1.4).

Math Prfself.

Let's consider PrfSprt arithmetic first:

1. Simultaneous parallel addition of sets elements $\widetilde{g_i}$=($g_{i_1}|\mu_{\widetilde{g_i}}(g_{i_1})$, $g_{i_2}|\mu_{\widetilde{g_i}}(g_{i_2})$, …, $g_{i_{m_j}}|\mu_{\widetilde{g_i}}(g_{i_{m_j}})$), i = 1,2,. . . ,n, j =1,2,. . . ,k is carried out using $PrfSprt\begin{matrix}\{g_1\}+ & \{g_2\}+ & \ldots & \{g_n\}+ \\ x_1 & x_2 \ldots & & x_n\end{matrix}$.
2. Similarly, for simultaneous parallel multiplication: $PrfSprt\begin{matrix}\{g_1\}* & \{g_2\}* & \ldots & \{g_n\}* \\ w_1 & w_2 \ldots & & w_n\end{matrix}$ gives the fuzzy set B_l, l =1,2,. . . ,n, with elements $b_{l i_1 i_2 \ldots i_{m_j}}$ = $Sprt_{x_l}\left\{ g_{l_{1_{i_1}}}*, \quad g_{l_{2_{i_2}}}*, \ldots, \quad g_{l_{m_j}}{}_{i_{m_j}} \right\})_{R_l}$ for any $\left\{ l_{i_1}, l_{i_2}, ..., l_{i_{m_j}} \right\}$ without repetitions, x_l = $Sprt_w{}^{\{K_l\}}$,K_l-set of any $\left\{ k_{l_{i_1}}*, k_{l_{i_2}}*, ..., k_{l_{i_{m_j}}}* \right\}$ without repeating them, l = 1,2,. . . ,n, $k_{l_{i_j}}$-any digit, i =1,2,. . . , m_j, R_l= $Sprt_w\left\{ l_{i_1}+, l_{i_2}+, ..., l_{i_{m_j}} \right\}$, R_l is the index of the lower discharge (we choose an index on the scale of discharges): see Table I.1.

Then $PrfSprt\begin{matrix}B_1+ & B_2+ & \ldots & B_n+ \\ x_1 & x_2 \ldots & & x_n\end{matrix}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. The most straightforward functional scheme of the assumed arithmetic-logical device for PrfSprt-multiplication:

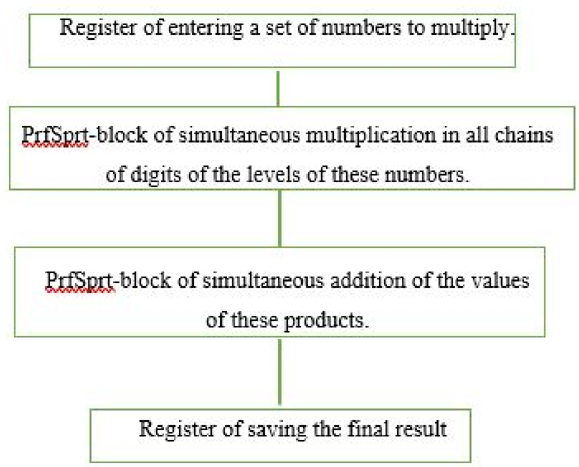


Fig. 0.1 The straightforward functional scheme of the assumed arithmetic-logical device for PrfSprt-multiplication.

Remark. The algorithm for simultaneously adding a set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by multiplying the first number from the set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the set by the ones following it, etc.

3. Similarly for simultaneous execution of various operations: $PrfSprt\begin{matrix} g_1Q_1 & g_2Q_2 & \dots & g_nQ_n \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\widetilde{Q}$=(Q$_1|\mu_{\widetilde{Q}}$(Q$_1$), Q$_2|\mu_{\widetilde{Q}}$(Q$_2$), …, Q$_n|\mu_{\widetilde{Q}}$(Q$_n$)). Q_i -an operation, i = 1, …, n.

4. Similarly, for the simultaneous execution of various operators: $PrfSprt\begin{matrix} F_1g_1 & F_2g_2 & \dots & F_ng_n \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\widetilde{F}$=(F$_1|\mu_{\widetilde{F}}$(F$_1$), F$_2|\mu_{\widetilde{F}}$(F$_2$), …, F$_n|\mu_{\widetilde{F}}$(F$_n$)). F_i is an operator, i = 1, …, n.

5. The arithmetic itself for capacities in themselves will be similar: addition - $PrfS_1f\{g+\}$, (or $PrfS_3f\{g+\}$) for the third type), multiplication $PrfS_1f\{g*\}$, ($PrfS_3f\{g*\}$).

6. Similarly with different operations: $PrfS_1f\{gq\}$, ($PrfS_3f\{gq\}$), and with different operators: $PrfS_1f\{Fg\}$, ($PrfS_3f\{Fg\}$).

7. $PrfSrt\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$ $-$ *the result of the containment operator*. For fuzzy sets A_i, B_i, (i = 1, 2, …, n), we have

$PrfSrt\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} = \left\{\sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \ , \sum_{i=1}^{n} D_i\right\} = \left\{\begin{matrix} \sum_{i=1}^{n} D_i \\ \sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \end{matrix}\right\}$, where D_i is self-(fuzzy set) for $A_i \cap B_i$ (i = 1, 2, …, n). There is the same for structures if they are considered as fuzzy sets. Similarly, for fuzzy sets C_i, D_i $\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}$PrfSrt $=$ $\left\{\begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\,\} & \{\,\} \dots & \{\,\} \\ D_1 - D_1 \cap C_1 & D_2 - D_2 \cap C_2 \dots & D_m - D_m \cap C_m \end{matrix} PrfSrt \\ \sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix}\right\}$,
where Q_i is oself-(fuzzy set) for $(D_i \cap C_i)$ (i = 1, 2, …, m) [14].

8. PrfSprt-derivative of f(x_1,,x_2,…,x_n) = $\begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \dots \\ f_k(x_1, x_2, \dots, x_n) \end{pmatrix}$ is

$PrfSprt\begin{matrix} \frac{\partial}{\partial x_{1_i}} & \frac{\partial}{\partial x_{2_i}} \dots & \frac{\partial}{\partial x_{k_i}} \\ f_1(x_1, x_2, \dots, x_n) & f_2(x_1, x_2, \dots, x_n) \dots & f_k(x_1, x_2, \dots, x_n) \end{matrix}$,
where x=(x_{1_i}, x_{2_i},… x_{k_i})- any fuzzy set from $\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$),

x$_2|\mu_{\widetilde{x}}$(x$_2$), …, x$_n|\mu_{\widetilde{x}}$(x$_n$)). The same is done for PrfSprt- $\frac{\partial^k f(x)}{\partial x_{1_i}\partial x_{2_i}\ldots\partial x_{k_i}}$. PrfSprt-integral off $(x_{1,},x_2,\ldots,x_n)$ is

$$PrfSprt\frac{\int()dx_{1_i}}{f_1(x_1,x_2,\ldots,x_n)}\frac{\int()dx_{2_i}\ \ldots}{f_2(x_1,x_2,\ldots,x_n)\ \ldots}\frac{\int()dx_{k_i}}{f_k(x_1,x_2,\ldots,x_n)},$$

where $(x_{1_i},\ x_{2_i},\ldots\ x_{k_i})$- any fuzzy set from $\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$), …, x$_n|\mu_{\widetilde{x}}$(x$_n$)). The same is done for PrfSprt-$\int\ldots\int f(x)dx_{1_i}dx_{2_i}\ldots dx_{k_i}$-k-multiple integral. PrfSprt-lim off $\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$), …, x$_n|\mu_{\widetilde{x}}$(x$_n$)) is $PrfSprt\frac{\lim\limits_{x_{1_i}\to a_{1_i}}\ \lim\limits_{x_{2_i}\to a_{2_i}}\ \ldots\ \lim\limits_{x_{k_i}\to a_{k_i}}}{f_1x_1,x_2,\ldots,x_n)f_2x_1,x_2,\ldots,x_n)\ldots f_kx_1,x_2,\ldots,x_n)}$

The same is done for PrfSprt- $\begin{pmatrix}\lim f(x_1,,x_2,\ldots,x_n)\\ x_{1_i}\to a_{1_i}\\ \ldots\\ x_{k_i}\to a_{k_i}\end{pmatrix}$. $PrfS_3f\{\lim\limits_{x\to a}\}$ $= PrfSprt\frac{\lim\limits_{x_{1_i}\to a_{1_i}}\ \lim\limits_{x_{2_i}\to a_{2_i}}\ \ldots\ \lim\limits_{x_{k_i}\to a_{k_i}}}{\lim\limits_{x_{1_i}\to a_{1_i}}\ \lim\limits_{x_{2_i}\to a_{2_i}}\ \ldots\ \lim\limits_{x_{k_i}\to a_{k_i}}}$

9. In the case of Prself-derivatives, inclusions of multiple derivatives are obtained. The same is true for Prself-integrals: we get inclusions of multiple integrals.

10. Let's denote Prself-(Prself-Q) through Prself^2-Q , fS(n,Q)= Prself-(Prself-(…(Prself-Q))) = Prself^n-Q for n-multiple Prself.

Operator Pritself.

Definition F.1.1.7. An operator that transforms PrSprt$\frac{g_1\ g_2\ \cdots\ g_n}{x_1\ x_2\ \ldots\ x_n}$, into any Pr$S_if\{b\}$, $i=2,3$; where $\{b\}\subset\{g\},\{g\}=(g_1,\ g_2,\ \ldots,\ g_n)$; is the operator Pritself.

Lim-Pritself.

1. Lim PrfSprt

   For example, : $\begin{matrix}double PrfSprt-limG(x,y)\\ x\to g_1\\ y\to g_2\end{matrix}$ corresponds to

   PrfSprt$\frac{G(x,y)\ G(x,y)}{x\to g_1\ \ y\to g_2}$.

   Similarly, for lim PrfSprt with n variables.In the case of lim-Pritself, for example, for m variables, it suffices to use the form (F.2.1.1) of lim PrfSprt for n variables (n>m). The same is true for integrals of variables m (for example, the double integral over a rectangular region is through the double limit).

Remark.The sequence of actions can be "collapsed" into an ordered PrfSprt element, and then translate it, for example, into $PrfS_3f-$ the parallel fuzzy capacity in itself.

About PrfSprt and PrS_3f programming.

The ideology of PrfSprt and $PrfS_3f$ can be used for programming. Here are some of the PrfSprt programming operators.

1. Simultaneous assignment of the expressions $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2)$, …, $p_n|\mu_{\widetilde{p}}(p_n)$) to the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2)$, …, $x_n|\mu_{\widetilde{x}}(x_n)$). This is implemented via PrfSprt$\begin{matrix} x_1 := & x_2 := & \ldots & x_n := \\ p_1 & p_2 & \ldots & p_n \end{matrix}$.
2. Simultaneous checking the set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2)$, …, $g_n|\mu_{\widetilde{g}}(g_n)$) for the set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2)$, …, $B_n|\mu_{\widetilde{B}}(B_n)$). Implemented via PrfSprt$\begin{matrix} IF\,\{B_1 g_1\}\ then & IF\,\{B_2 g_2\}\ then & \ldots & IF\,\{B_n g_n\}\ then \\ w_1 & w_2 & \ldots & w_n \end{matrix}$, where w_i (i = 1, …, n) can be anything.
3. Similarly for loop operators and others.

$PrfS_3f-$ software operators will differ only in that the aggregates $\{\widetilde{g}\}, \{\widetilde{p}\}, \left\{\widetilde{B}\right\}, \{\widetilde{x}\}$ will be formed from the corresponding PrfSprt program operators in form (1.1) and for more complex operators in the forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

The OS (operating system), the computer's principles, and the modes of operation for this programming are interesting. But this is already the material for the following articles.

Using elements of the mathematics of PrfSrt we introduce the concept of PrfSrt – the change in physical quantity B: PrfSrt$\begin{matrix} \Delta_1 B & \Delta_2 B & \ldots & \Delta_n B \\ x_1 & x_2 & \ldots & x_n \end{matrix}$, . Then the mean PrfSrt - velocity will be $v_{cpPrfSrt}(t, t)$ = PrfSrt$\begin{matrix} \frac{\Delta_1 B}{\Delta t} & \frac{\Delta_n B}{\Delta t} & \ldots & \frac{\Delta_n B}{\Delta t} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ and PrfSrt-velocity at time t:$v_{Prfsrt} = \lim\limits_{\Delta t \to 0} \{v_{Prfsrt}(t, \Delta t)$.

PrfSrt – acceleration $a_{Prfsrt} = \frac{dv_{Prfsrt}}{dt}$. The nuclei of atoms can be considered as PrfSprt elements.

Remark F.2.1.2. PrfSprt – elements with "target weights" become peculiar. Here you need the necessary energy to carry them out. As a rule, this energy is at the level of PrSelf. This is natural since it’s much easier to manage elements of the k level via the elements of a more structured k +1 level. Let us consider the concepts of capacities of physical objects in themselves. The question arises about the self-energy of the object. In particular, Pr Srt_B^B will mean PrS_1f B. For example, PrSprt_{DNA}^{DNA} allows you to reach the level of DNA self-energy, $PrSprt_Q^Q$ allows you to reach the level of self-energy Q. The law of self-energy conservation operates already at the level of self-energy. Also, in addition to capacities in themselves, you can consider the types of containment of oneself in oneself: the first type of the containment of oneself in oneself: the second type of the containment of oneself in oneself: potentially, for example, in the form of programming oneself, the third type is partial containment of oneself in themselves—for example, Prfself-operator, Prfself-action, whirlwind. A fuzzy container containing itself can be formed by fself-containment, i.e., fcontainment in oneself. Let us clarify the concept of the term fuzzy capacity in itself: it is a fuzzy capacity containing itself potentially. Consider Prfself-Q, where Q can be anything fuzzy, including Q=Prfself; in particular, it can be any fuzzy action. Therefore, Prfself-Q is when fuzzy Q is made by Prfitself; it makes itself. There is a partial Prfself-Q for any fuzzy Q with partial Prfself-fulfillment. Let's consider several examples for fuzzy capacities in themselves: ordinary lightning, electric arc discharge, and ball lightninF.

PrfSprt is also great for working with structures, for example:

1) $\mathrm{PrfSprt}^{fstrA_1\ fstrA_2\ \ldots\ fstrA_n}_{B_1\quad B_2\ \ldots\quad B_n}$ - the fuzzy structure A_i that fits into fuzzy B_i, where fuzzy B_i (i = 1, . . . , n) can be any fuzzy capacity, another fuzzy structure etc.

2) $\mathrm{PrfSprt}^{fstr_{Q_1}\ fstr_{Q_2}\ \ldots\ fstr_{Q_n}}_{B_1\quad B_2\ \ldots\quad B_n}$ – embedding fuzzy structure from Q_i into B_i. Similarly for displacement: 1) ${}^{C_1\quad C_2\ \ldots\quad C_m}_{fstrD_1\ fstrD_2\ \ldots\ fstrD_m}\mathrm{PrfSprt}$ - displacement of fuzzy structure $fstrD_i$ from C_i, (i = 1, . . . , n), 2) ${}^{C_1\quad C_2\ \ldots\quad C_m}_{fstr_{Q_1}\ fstr_{Q_2}\ \ldots\ fstr_{Q_m}}$ PrfSprt -displacement of the fuzzy structure Q_i from C_i, (i = 1, . . . , m). To work with structures, you can introduce a special operator PrfCprt: $\mathrm{PrfCprt}^{A_1\ A_2\ \ldots\ A_n}_{B_1\ B_2\ \ldots\ B_n}$ fuzzy structures B_i with the fuzzy structure A_i, (i = 1, . . . , n), $\mathrm{PrfCprt}^{fstr_{Q_1}\ fstr_{Q_2}\ \ldots\ fstr_{Q_n}}_{B_1\quad B_2\ \ldots\quad B_n}$ fuzzy structures B_i with the fuzzy structure from Q_i, (i = 1, . . . , n), ${}^{C_1\ C_2\ \ldots\ C_m}_{D_1\ D_2\ \ldots\ D_m}\mathrm{PrfCprt}$ destructors fuzzy C_i by the fuzzy structure of D_i, ${}^{C_1\quad C_2\ \ldots\quad C_m}_{fstr_{Q_1}\ fstr_{Q_2}\ \ldots\ fstr_{Q_m}}$ PrfCprt destructors fuzzy C_i from the fuzzy structure that structures Q_i, (i = 1, . . . , m).

Definition F.2.1.8. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its elements explicitly, but not necessarily in known operators; it can form (create) new special operators (in particular, special functions).

In particular, $\mathrm{PrfCprt}^{A_1\ A_2\ \ldots\ A_n}_{A_1\ A_2\ \ldots\ A_n}$, $\mathrm{PrfCrt}^{A_1\ A_2\ \ldots\ A_n}_{A_1\ A_2\ \ldots\ A_n}$ are such structures.

Similarly, for working with models, each is structured by its structure; for example, use PrfSprt-groups, PrfSprt-rings, PrfSprt-fields, PrfSprt-spaces, Prself-groups, Prfself-rings, Prfself-fields, and Prfself-spaces. Like any task, this is also a structure of the appropriate capacity.

Prfself-H (Prfself-hydrogen), like other Prfself-particles, does not exist in the ordinary, but all Prfself-molecules, Prfself-atoms, and Prfself-particles are elements of the energy space.

RemarkF.2.1.3. The concept of elements of physics PrfSprt is introduced for energy space. The corresponding concept of elements of chemistry PrfSprt is introduced accordingly.

*T*he ideology of PrfSprt elements allows us to go to the border of the world familiar to us, which allows us to act more effectively.

Dynamic PrfSprt – elements.

We considered stationary PrfSprt – elements earlier. Here we consider dynamic PrfSprt – elements.

Definition F.2.2.1. The process of fitting a fuzzy set of elements $\widetilde{g(t)}$=(g$_1$(t)|$\mu_{\widetilde{g(t)}}$(g$_1$(t)), g$_2$(t)|$\mu_{\widetilde{g(t)}}$(g$_2$(t)), …, g$_n$(t)|$\mu_{\widetilde{g(t)}}$(g$_n$(t))) into one point x= $(x_1, x_2, ..., x_n)$ of the space X at time t will be called a dynamic PrfSprt – element. We will denote PrfSprt(t)$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Definition F.2.2.2. Fitting an ordered fuzzy set of elements into one point in space is called a dynamic ordered PrfSprt–element .

It is allowed to sum dynamic PrfSprt – elements:

$$PrfSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} + PrfSprt(t)\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} =$$

$$\mathrm{PrfSprt}(t)\begin{matrix} g_1(t) \cup b_1(t) & g_2(t) \cup b_2(t) & \dots & g_n(t) \cup b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$$

It's allowed to multiply PrfSprt – elements:

$$PrfSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} \quad * \quad PrfSprt(t)\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} =$$

$$PrfSprt\begin{matrix} g_1(t) \cap b_1(t) & g_2(t) \cap b_2(t) & \dots & g_n(t) \cap b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$$

Parallel dynamic fuzzy containment of oneself.

Definition F.2.2.3. Parallel dynamic fSprt-capacity $PrfSprt(t)\begin{matrix} R_1(t) & R_2(t) & \dots & R_n(t) \\ Q_1(t) & Q_2(t) & \dots & Q_n(t) \end{matrix}$ is the process of embedding fuzzy R$_i$(t) into fuzzy Q$_i$(t), (i = 1, …, n), simultaneously.

Definition F.2.2.4. Parallel dynamic fuzzy capacity $\widetilde{Q(t)}$=(Q$_1$(t)|$\mu_{\widetilde{Q(t)}}$(Q$_1$(t)), Q$_2$(t)|$\mu_{\widetilde{Q(t)}}$(Q$_2$(t)), …, Q$_n$(t)|$\mu_{\widetilde{Q(t)}}$(Q$_n$(t))) containing itself as an element of the first type is the process of parallel containing fuzzy $\widetilde{Q(t)}$ in $\widetilde{Q(t)}$ $PrfSprt(t)^{Q_1(t)\ Q_2(t)\ \dots\ Q_n(t)}_{Q_1(t)\ Q_2(t)\ \dots\ Q_n(t)}$. Denote $\Pr fS_1 f(t)\widetilde{Q(t)}$.

Definition F.2.2.5. Parallel dynamic fuzzy capacity C(t) in itself of the second type is the process of parallel containing fuzzy elements from which it can be parallel fuzzy generated. Let's denote $PrfS_2 f(t)C(t)$.

Definition F.2.2.6. Parallel dynamic partial fuzzy capacity B(t) in itself of the third type is a process of partial parallel containment of fuzzy B(t) in itself or parallel embedding fuzzy elements from which it can be parallel fuzzy generated partially or both at the same time. Denote $\Pr fS_3 f(t)B(t)$.

All parallel dynamic fuzzy capacities in a parallel dynamic fuzzy self-space are, by definition, parallel dynamic fuzzy capacities in themselves. Parallel dynamic fuzzy capacity itself can manifest itself as parallel dynamic fSprt-capacity and ordinary parallel dynamic fuzzy capacity. In these cases, the usual fuzzy measures and methods of topology are used.

Connection of dynamic PrfSprt – elements with parallel dynamic fuzzy containment of oneself.
Consider third type of parallel dynamic partial fuzzy containment of oneself. For example, based on $PrfSprt(t)^{Q_1(t)\ Q_2(t)\ \dots\ Q_n(t)}_{x_1\quad x_2\ \dots\quad x_n}$, where $\widetilde{Q(t)}$=(Q$_1$(t)|$\mu_{\widetilde{Q(t)}}$(Q$_1$(t)), Q$_2$(t)|$\mu_{\widetilde{Q(t)}}$(Q$_2$(t)), …, Q$_n$(t)|$\mu_{\widetilde{Q(t)}}$(Q$_n$(t))), i.e. n – elements at one point x = $(x_1, x_2, ..., x_n)$, we can consider the parallel dynamic fuzzy capacity in itself $PrfS_3 f(t)$ with m elements from $\widetilde{Q(t)}$, m<n, which is process formed according to the form (1.1), that is, only m elements from $\widetilde{Q(t)}$ are in the structure $PrfSprt(t)^{Q_1(t)\ Q_2(t)\ \dots\ Q_n(t)}_{x_1\quad x_2\ \dots\quad x_n}$.

Parallel dynamic fuzzy containment of oneself of the third type can be formed for any other structure, not necessarily PrfSprt, only through the obligatory reduction in the number of elements in the structure. In particular, using the forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

It is possible to introduce structures more complex than PrfS$_3$f(t).

Parallel dynamic fuzzy math itself.

1. The process of simultaneous parallel addition of fuzzy sets elements $\left\{\widetilde{g_i(t)}\right\} =$ $\left(g_{i_1}(t)|\mu_{\widetilde{g_i(t)}}(g_{i_1}(t)),\ g_{i_2}(t)|\mu_{\widetilde{g_i(t)}}(g_{i_2}(t)),\ \dots,\ g_{i_{m_j}}(t)|\mu_{\widetilde{g_i(t)}}(\ g_{i_{m_j}}(t))\ \right),$

i = 1,2,. . . ,n, j =1,2,. . . ,k are realized by
$PrfSprt(t)^{\widetilde{\{g_1(t)\}}+ \; \widetilde{\{g_2(t)\}}+ \; \ldots \; \widetilde{\{g_n(t)\}}+}_{x_1 \quad x_2 \ldots \quad x_n}$.

2. By analogy, for simultaneous multiplication:
$PrfSprt(t)^{\widetilde{\{g_1(t)\}}* \; \widetilde{\{g_2(t)\}}* \; \ldots \; \widetilde{\{g_n(t)\}}*}_{x_1 \quad x_2 \ldots \quad x_n}$.

3. Similarly for simultaneous execution of various operations:
$PrfSprt(t)^{g_1(t)Q_1(t) \; g_2(t)Q_2(t) \; \ldots \; g_n(t)Q_n(t)}_{w_1 \quad w_2 \ldots \quad w_n}$, where $\widetilde{Q(t)}$=
($Q_1(t)|\mu_{\widetilde{Q(t)}}(Q_1(t))$, $Q_2(t)|\mu_{\widetilde{Q(t)}}(Q_2(t))$, …, $Q_n(t)|\mu_{\widetilde{Q(t)}}(Q_n(t))$), $Q_i(t)$-an operation, i = 1, …, n, $\widetilde{g(t)}$=($g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t))$, $g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t))$, …, $g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t))$).

4. Similarly, for the simultaneous execution of various operators: $PrfSprt(t)^{F_1(t)g_1(t) \; F_2(t)g_2(t) \; \ldots \; F_n(t)g_n(t)}_{w_1 \quad w_2 \ldots \quad w_n}$, where
$\widetilde{F(t)}$=($F_1(t)|\mu_{\widetilde{F(t)}}(F_1(t))$, $F_2(t)|\mu_{\widetilde{F(t)}}(F_2(t))$, …, $F_n(t)|\mu_{\widetilde{F(t)}}(F_n(t))$), $F_i(t)$ is an operator, i = 1, …, n.

5. Parallel dynamic arithmetic itself for fuzzy containments of oneself will be similar: Parallel dynamic addition - $PrfS_1f(t)\left\{\widetilde{g(t)}+\right\}$, (or $PrfS_3f(t)\left\{\widetilde{g(t)}+\right\}$ for the third type), Parallel dynamic multiplication $PrfS_1f(t)\left\{\widetilde{g(t)}*\right\}$, ($PrfS_3f(t)\left\{\widetilde{g(t)}*\right\}$).

5. Similarly with different operations: $PrfS_1f(t)\left\{\widetilde{g(t)}\widetilde{Q(t)}\right\}$, ($PrfS_3f(t)\left\{\widetilde{g(t)}\widetilde{Q(t)}\right\}$) and with different operators: $PrfS_1f(t)\left\{\widetilde{F(t)}\widetilde{g(t)}\right\}$, (Pr $S_3f(t)\left\{\widetilde{F(t)}\widetilde{g(t)}\right\}$).

6. . $PrfSprt(t)^{A_1(t) \; A_2(t) \; \ldots \; A_n(t)}_{B_1(t) \; B_2(t) \; \ldots \; B_n(t)}$ *gives* the result

$$PrfSrt(t)^{A_1(t) \; A_2(t) \; \ldots \; A_n(t)}_{B_1(t) \; B_2(t) \; \ldots \; B_n(t)} = \left\{\sum_{i=1}^{n} A_i(t) \cup B_i(t) - A_i(t) \cap B_i(t) \;, \sum_{i=1}^{n} D_i(t)\right\},$$

for fuzzy sets $A_i(t)$, $B_i(t)$, where $D_i(t)$ is fuzzy self-set for $A_i(t) \cap B_i(t)$, (i = 1, 2, …, n). The same is true for structures if they are treated as fuzzy sets,

$$^{C_1(t) \; C_2(t) \; \ldots \; C_m(t)}_{D_1(t) \; D_2(t) \; \ldots \; D_m(t)}\mathrm{PrfSrt(t)} =$$

$$\left\{ \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\;\} & \{\;\} \ldots & \{\;\} \\ D_1(t) - D_1(t) \cap C_1(t) & D_2(t) - D_2(t) \cap C_2(t) \; \ldots & D_2(t) - D_2(t) \cap C_2(t) \end{matrix}^{PrfSrt} \atop \sum_{i=1}^{m} \big(C_i(t) - D_i(t) \cap C_i(t)\big) - (D_i(t) - D_i(t) \cap C_i(t)) \right\}$$

$for\ fuzzy\ sets\ C_i(t),\ D_i(t)$, where $Q_i(t)$ is fuzzy oself-set for $\big(D_i(t) \cap C_i(t)\big)$(i = 1, 2, …, m) [14].

8. Similarly, for dynamic PrfSprt-derivatives, dynamic PrfSprt-integrals, dynamic PrfSprt-lim, parallel dynamic fuzzy self-derivatives, parallel dynamic fuzzy self-integrals
9. Denote parallel dynamic fuzzy self-(parallel dynamic fuzzy self-Q(t)) through parallel dynamic fuzzy self^2-Q(t) , pffS(t)(n,Q(t))= parallel dynamic fuzzy self-(parallel dynamic fuzzy self-(…((parallel dynamic fuzzy self)-Q(t)))) = (parallel dynamic fuzzy self^n)-Q(t) for n-multiple parallel dynamic fuzzy self.

Remark F.2.2.1. The parallel dynamic PrfSprt-displacement will be denote by $\begin{matrix} C_1(t)\ C_2(t)\ \dots\ C_m(t) \\ D_1(t)\ D_2(t)\ \dots\ D_m(t) \end{matrix}$PrfSprt(t) , where fuzzy D_1(t) is forced out of fuzzy C_1(t), fuzzy D_2(t) is forced out of fuzzy C_2(t), …, fuzzy D_m(t) is forced out of fuzzy C_m(t) simultaneously, the result of this process will be described by the expression $\begin{matrix} C_1(t)\ C_2(t)\ \dots\ C_m(t) \\ D_1(t)\ D_2(t)\ \dots\ D_m(t) \end{matrix}$PrfSrt(t). Then the notation

$$\begin{matrix} C_1(t)\ C_2(t)\ \dots\ C_m(t) \\ D_1(t)\ D_2(t)\ \dots\ D_m(t) \end{matrix} PrfSprt(t) \begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_n(t) \\ B_1(t)\ B_2(t)\ \dots\ B_n(t) \end{matrix}$$

where fuzzy A_1(t) fits into fuzzy B_1(t), fuzzy A_2(t) fits into fuzzy B_2(t), …, fuzzy A_n(t) fits into fuzzy B_n(t), fuzzy D_1(t) is forced out of fuzzy C_1(t), fuzzy D_2(t) is forced out of fuzzy C_2(t), …, fuzzy D_m(t) is forced out of fuzzy C_m(t) simultaneously. It is dynamic PrfSprt-containment of fuzzy A_i(t) in fuzzy B_i(t) and dynamic PrfSprt-displacement of fuzzy D_j(t) from fuzzy C_j(t) simultaneously, (i = 1, 2, …, n, j = 1, 2, …, m). The result of this process will be described by the expression

$$\begin{matrix} C_1(t)\ C_2(t)\ \dots\ C_m(t) \\ D_1(t)\ D_2(t)\ \dots\ D_m(t) \end{matrix} PrfSrt(t) \begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_n(t) \\ B_1(t)\ B_2(t)\ \dots\ B_n(t) \end{matrix}$$

$PrfSprt(t) \begin{matrix} B_1(t)\ B_2(t)\ \dots\ B_n(t) \\ B_1(t)\ B_2(t)\ \dots\ B_n(t) \end{matrix}$ will mean PrS_1f(t)B(t).

$\begin{matrix} C_1(t)\ C_2(t)\ \dots\ C_m(t) \\ C_1(t)\ C_2(t)\ \dots\ C_m(t) \end{matrix} PrfSprt(t)$ denotes the parallel dynamic expelling fuzzy $\widetilde{C(t)}$=(C_1(t)|$\mu_{\widetilde{C(t)}}$(C_1(t)), C_2(t)|$\mu_{\widetilde{C(t)}}$(C_2(t)), …, C_n(t)|$\mu_{\widetilde{C(t)}}$(C_n(t))) oneself out of oneself, $\begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_n(t) \\ A_1(t)\ A_2(t)\ \dots\ A_n(t) \end{matrix} PrfSprt(t) \begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_n(t) \\ A_1(t)\ A_2(t)\ \dots\ A_n(t) \end{matrix}$ —simultaneous parallel dynamic containment fuzzy $\widetilde{A(t)}$=(A_1(t)|$\mu_{\widetilde{A(t)}}$(A_1(t)), A_2(t)|$\mu_{\widetilde{A(t)}}$(A_2(t)), …, A_n(t)|$\mu_{\widetilde{A(t)}}$(A_n(t))) of oneself in oneself and parallel dynamic expelling fuzzy $\widetilde{A(t)}$ oneself out of oneself. $\begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_n(t) \\ A_1(t)\ A_2(t)\ \dots\ A_n(t) \end{matrix} PrfSprt(t)$ will be called parallel dynamic fuzzy anti-capacity from oneself. For example, “white hole” in physics is such simple anti-capacity.

About dynamic PrfSprt and $PrS_3f(t)$ programming.

The ideology of dynamic PrfSprt and $PrS_3f(t)$ can be used for programming:

1. The process of simultaneous assignment of the expressions $\widetilde{p(t)}=(p_1(t)|\mu_{\widetilde{p(t)}}(p_1(t)), p_2(t)|\mu_{\widetilde{p(t)}}(p_2(t)), \ldots, p_n(t)|\mu_{\widetilde{p(t)}}(p_n(t)))$ to the variables $\widetilde{x(t)}=(x_1(t)|\mu_{\widetilde{x(t)}}(x_1(t)), x_2(t)|\mu_{\widetilde{x(t)}}(x_2(t)), \ldots, x_n(t)|\mu_{\widetilde{x(t)}}(x_n(t)))$ is implemented through PrfSprt$\begin{matrix} x_1(t) := & x_2(t) := & \ldots & x_n(t) := \\ p_1(t) & p_2(t) & \ldots & p_n(t) \end{matrix}$ . .

2. The process of simultaneous check the set of conditions $\widetilde{g(t)}=(g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t)), g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t)), \ldots, g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t)))$ for a set of expressions $\widetilde{B(t)}=(B_1(t)|\mu_{\widetilde{B(t)}}(B_1(t)), B_2(t)|\mu_{\widetilde{B(t)}}(B_2(t)), \ldots, B_n(t)|\mu_{\widetilde{B(t)}}(B_n(t)))$ is implemented through PrfSprt$\begin{matrix} IF\left\{B_1(t)g_1(t)\right\}\ then & IF\left\{B_2(t)g_2(t)\right\}\ then & \ldots & IF\left\{B_n(t)g(t)\right\}\ then \\ w_1(t) & w_2(t) & \ldots & w_n(t) \end{matrix}$, where w(t) = $(w_1(t), w_2(t), ..., w_n(t))$. can be any.

3.Similarly for loop operators and others.

$PrfS_3f(t)$– software operators will differ only in that the aggregates $\left\{\widetilde{g(t)}\right\}, \left\{\widetilde{p(t)}\right\}, \left\{\widetilde{B(t)}\right\}, \left\{\widetilde{x(t)}\right\}$ will be formed from corresponding processes PrfSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*) for more complex operators.

Remark F.2.2.2. It is the parallel fuzzy containment of oneself in oneself that can “give birth” to the parallel fuzzy capacities in itself – that is what parallel self-organization is.

Remark F.2.2.3. $PrfSprt(t) \begin{matrix} r \quad PrfSprt(t) \begin{matrix} B_1(t)\ B_2(t)\ \ldots\ B_n(t) \\ B_1(t)\ B_2(t)\ \ldots\ B_n(t) \end{matrix} \ldots r \\ r \quad PrfSprt(t) \begin{matrix} B_1(t)\ B_2(t)\ \ldots\ B_n(t) \\ B_1(t)\ B_2(t)\ \ldots\ B_n(t) \end{matrix} \ldots r \end{matrix}$, r = PrfSprt(t) $\begin{matrix} B_1(t)\ B_2(t)\ \ldots\ B_n(t) \\ B_1(t)\ B_2(t)\ \ldots\ B_n(t) \end{matrix}$, can increase parallel fuzzy self- level of $\widetilde{B(t)}=(B_1(t)|\mu_{\widetilde{B(t)}}(B_1(t)), B_2(t)|\mu_{\widetilde{B(t)}}(B_2(t)), \ldots, B_n(t)|\mu_{\widetilde{B(t)}}(B_n(t)))$.

Remark F.2.2.4. For example, the operator Prfitself is $PrfS_1f(t)$.

Remark F.2.2.5. May be considered the following derivatives: $\frac{dPrfSrt(t)\begin{matrix}A_1(t)\ A_2(t)\ \dots\ A_n(t)\\ B_1(t)\ B_2(t)\ \dots\ B_n(t)\end{matrix}}{dt}$, $\frac{d\begin{matrix}C_1(t)\ C_2(t)\ \dots\ C_m(t)\\ D_1(t)\ D_2(t)\ \dots\ D_m(t)\end{matrix}^{PrfSprt(t)}}{dt}$,

$\frac{d\begin{matrix}C_1(t)\ C_2(t)\ \dots\ C_m(t)\\ D_1(t)\ D_2(t)\ \dots\ D_m(t)\end{matrix}^{PrfSprt(t)}\begin{matrix}A_1(t)\ A_2(t)\ \dots\ A_n(t)\\ B_1(t)\ B_2(t)\ \dots\ B_n(t)\end{matrix}}{dt}$, $\frac{dPrS_if(t)}{dt}$, i=1,2,3.

Remark F.2.2.6. It is the parallel fuzzy containment of oneself in itself as an element that can be interpreted as parallel dynamic fuzzy capacities in itself.

Remark F.2.2.7. Not every fuzzy capacity parallel fuzzy containing itself as an element will manifest itself as a sedentary parallel fuzzy capacity or parallel fuzzy capacity.

PrfSprt – elements for continual sets.

Earlier, we considered finite-dimensional discrete PrfSprt-elements and self-capacities in itself as an element. Here we believe some continual PrfSprt-elements and continual parallel self-capacities in themselves as an element.

Definition 3.1. The set of continual elements $\widetilde{g}$=(g$_1|v_{\widetilde{g}}$(g$_1$), g$_2|\mu_{\widetilde{g}}$(g$_2$), …, g$_n|\mu_{\widetilde{g}}$(g$_n$)) at one point x = $(x_1, x_2, ..., x_n)$ of space X will be called continual PrfSprt – element, and such a point in space will be called parallel capacity of the continual PrfSprt – element. We will denote PrfSprt$\begin{matrix}g_1\ g_2\ \cdots\ g_n\\ x_1\ x_2\ \dots\ x_n\end{matrix}$.

Definition F.2.3.2. An ordered set of continual elements at one point in space is called an ordered continual PrfSprt–element.

It's allowed to sum continual PrfSprt – elements: : $PrfSprt\begin{matrix}g_1\ g_2\ \cdots\ g_n\\ x_1\ x_2\ \dots\ x_n\end{matrix} + PrfSprt\begin{matrix}b_1\ b_2\ \dots\ b_n\\ x_1\ x_2\ \dots\ x_n\end{matrix} = PrfSprt\begin{matrix}g_1\cup b_1 & g_2\cup b_2\ \dots & g_n\cup b_n\\ x_1 & x_2\ \dots & x_n\end{matrix}$, where some or any elements may be ordered continual elements. It's allowed to multiply PrfSprt – elements: $PrfSprt\begin{matrix}g_1\ g_2\ \cdots\ g_n\\ x_1\ x_2\ \dots\ x_n\end{matrix} * PrfSprt\begin{matrix}b_1\ b_2\ \dots\ b_n\\ x_1\ x_2\ \dots\ x_n\end{matrix} = PrfSprt\begin{matrix}g_1\cap b_1 & g_2\cap b_2\ \dots & g_n\cap b_n\\ x_1 & x_2\ \dots & x_n\end{matrix}$.

Definition F.2.3.3. The continual Prfself-capacity A in itself as an element of the first type is the continual capacity parallel containing itself as an element. Denote PrS_1fA. $PrfS_1fA$ = PrfSprt$\begin{matrix}A_1\ A_2\ \dots\ A_n\\ A_1\ A_2\ \dots\ A_n\end{matrix}$.

Definition F.2.3.4. The ordered continual Prself-capacity A in itself as an element of the first type is the ordered continual capacity parallel containing itself as an element. Denote $\overrightarrow{PrfS_1fA}$.

For example, $PrfSprt\begin{matrix} sin\infty & tg(-\infty) \dots & sin(-\infty) \\ x_1 & x_2 \dots & x_n \end{matrix} =$

$PrfSprt\begin{matrix} \uparrow I \downarrow_{-1}^{1}. & \downarrow I \uparrow_{-\infty}^{\infty} \dots & \downarrow I \uparrow_{-1}^{1} \\ x_1 & x_2 \dots & x_n \end{matrix}$, don't confuse with values of these functions.

Definition F.2.3.5. The continual Prfself-capacity A in itself, as an element of the second type, is the capacity parallel containing fuzzy continual elements from which it can be parallel fuzzy generated. Let's denote $\Pr fS_2fA$.

An example of continual self-capacity in itself as an element of the second type is a living organism since it contains the programs: DNA and RNA.

Definition F.2.3.6. Partial continual Prfself-capacity in itself as an element of the third type is called continual Prfself-capacity in itself as an element that partially parallel contains itself or parallel contains elements from which it can be parallel fuzzy generated in part or both simultaneously. Denote $PrfS_3f$.

All continual capacities in Prfself-space are continual Prfself-capacities in itself as an element by definition. The continual Prfself-capacities in itself as an element may appear as continual PrfSrt- capacities and usual continual fuzzy capacities. In these cases, there are used typical fuzzy measure and topology methods.

The connection of continual PrfSprt – elements with continual Prfself-capacities in themselves as an element.

Consider a third type of continual Prfself-capacity in itself as an element. For example, based on $PrfSprt\begin{matrix} g_1 & g_2 & \cdots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where $\widetilde{g}$=($\mathrm{g}_1|\mu_{\widetilde{g}}(\mathrm{g}_1)$, $\mathrm{g}_2|\mu_{\widetilde{g}}(\mathrm{g}_2), \dots,$ $\mathrm{g}_\mathrm{n}|\mu_{\widetilde{g}}(\mathrm{g}_\mathrm{n})$), i.e. n - continual elements at one point x = $(x_1, x_2, ..., x_n)$, The continual Prfself-capacity in itself as an element with m continual elements from $\widetilde{g}$, at m<n, can be considered as $PrfS_3f$, which is formed by the form (1.1), i.e., only m continual elements are located in the structure $PrfSprt\begin{matrix} g_1 & g_2 & \cdots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Continual self-capacities in itself as an element of the third type can be formed for any other structure, not necessarily $PrfSprt$, only by obligatory reducing the number of continual elements in the structure. In particular, using the forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). Structures more complex than $PrfS_3f$ can be introduced.

Mathematics Prfitself for continual elements.

1.Simultaneous parallel addition of the fuzzy sets continual elements$\widetilde{g}$=($\mathrm{g}_1|\mu_{\widetilde{g}}(\mathrm{g}_1)$, $\mathrm{g}_2|\mu_{\widetilde{g}}(\mathrm{g}_2), \dots,$ $\mathrm{g}_\mathrm{n}|\mu_{\widetilde{g}}(\mathrm{g}_\mathrm{n})$), $i = 1, 2, \dots, n,\ j = 1, 2, \dots, k$, is implemented using $PrfSprt\begin{matrix} \{g_1\}\cup & \{g_2\}\cup \dots & \{g_n\}\cup \\ x_1 & x_2 \dots & x_n \end{matrix}$.

2. By analogy, for simultaneous multiplication: $PrfSprt\begin{matrix} \{g_1\}\cap & \{g_2\}\cap \dots & \{g_n\}\cap \\ x_1 & x_2 \dots & x_n \end{matrix}$.

3. Similarly for simultaneous execution of various operations: $PrfSprt\begin{matrix} g_1Q_1 & g_2Q_2 & \ldots & g_nQ_n \\ w_1 & w_2 & \ldots & w_n \end{matrix}$, where $\widetilde{Q}$=($Q_1|\mu_{\widetilde{Q}}(Q_1)$, $Q_2|\mu_{\widetilde{Q}}(Q_2), \ldots,$ $Q_n|\mu_{\widetilde{Q}}(Q_n)$). Q_i -an operation, i = 1, …, n.

4. Similarly, for the simultaneous execution of various operators: $PrfSprt\begin{matrix} F_1g_1 & F_2g_2 & \ldots & F_ng_n \\ w_1 & w_2 & \ldots & w_n \end{matrix}$, where $\widetilde{F}$=($F_1|\mu_{\widetilde{F}}(F_1)$, $F_2|\mu_{\widetilde{F}}(F_2), \ldots,$ $F_n|\mu_{\widetilde{F}}(F_n)$). F_i is an operator, i = 1, …, n.

5. The arithmetic itself for parallel continual fuzzy capacities in themselves will be similar: addition - $PrfS_1f\,\{g+\}$, (or $PrfS_3f\,\{g+\}$) for the third type), multiplication $PrfS_1f\,\{g*\}$, ($PrfS_3f\,\{g*\}$).

6. Similarly with different operations: $PrfS_1f\,\{gQ\}$, ($PrfS_3f\,\{gQ\}$), and with different operators: $PrfS_1f\,\{Fg\}$, ($PrfS_3f\,\{Fg\}$).

7. $PrfSrt\begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix}$ – *the result of the parallel fuzzy containment operator*. For continual fuzzy sets A_i, B_i, (i = 1, 2, …, n), we have

$PrfSrt\begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix} = \left\{\sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i\ ,\sum_{i=1}^{n} D_i\right\}$, where D_i is self-(fuzzy set) for $A_i \cap B_i$ (i = 1, 2, …, n). There is the same for structures if they are considered as continual fuzzy sets, for fuzzy sets C_i, D_i $\begin{matrix} C_1 & C_2 & \ldots & C_m \\ D_1 & D_2 & \ldots & D_m \end{matrix}$PrfSrt =

$$\left\{\sum_{i=1}^{m} Q_i + \begin{matrix} \{\ \} & \{\ \}\ \ldots & \{\ \} \\ D_1 - D_1 \cap C_1 & D_2 - D_2 \cap C_2\ \ldots & D_m - D_m \cap C_m \\ \multicolumn{3}{c}{\sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i)} \end{matrix} PrfSrt\right\},$$

where Q_i is oself-(fuzzy set) for $(D_i \cap C_i)$(i = 1, 2, …, m) [14].

Remark F.2.3.1. $\begin{matrix} C_1 & C_2 & \ldots & C_m \\ D_1 & D_2 & \ldots & D_m \end{matrix}$PrfSprt, where continual fuzzy D_1 is forced out of continual fuzzy C_1, continual fuzzy D_2 is forced out of continual fuzzy C_2, …, continual fuzzy D_m is forced out of continual fuzzy C_m.

$$\begin{matrix} C_1 & C_2 & \ldots & C_m \\ D_1 & D_2 & \ldots & D_m \end{matrix} PrfSprt \begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix}$$

where continual fuzzy A_1 fits into continual fuzzy B_1, continual fuzzy A_2 fits into continual fuzzy B_2, …, continual fuzzy A_n fits into continual fuzzy B_n, continual fuzzy D_1 is forced out of continual fuzzy C_1, continual fuzzy D_2 is forced out of continual fuzzy C_2, …, continual fuzzy D_m is forced out of continual fuzzy C_m simultaneously.

We can consider the concept of a continual PrfSprt - element as PrfSprt$\begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix}$, where continual fuzzy A_1 fits into continual fuzzy B_1, continual fuzzy A_2 fits into continual fuzzy B_2, …, continual fuzzy A_n fits into continual fuzzy B_n. Then $PrfSprt\begin{matrix} B_1 & B_2 & \dots & B_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$ will mean $PrfS_1f$ B.

These elements are used for PrfSprt-coding, PrfSprt translation, coding Prfself, and translation Prfself for networks, which is suitable for electric current of ultrahigh frequency. More complex elements can be considered as continual sets of numbers with their " activation " in mutual directions. For example, ranges of function values, particularly those representing the shape of lightning. Differential geometry can be applied here. Also, n-dimensional elements can be considered. The space of such elements is Banach space if we introduce the usual norm for functions or vectors. We call this space-- fSelb-space. Then we introduce the scalar product for functions or vectors and get the Hilbert space. We call this space fSelh-space. In particular, one can try to describe some processes with these elements by differential equations and use methods from [4]. You can also try to optimize and research some processes with these elements using the techniques from [5]. Let's introduce operators for transforming capacity to Prfself-capacity in itself as an element: $PrfQ_1S(A)$ transforms fuzzy A to $Prff_1SA$, $PrfQ_0S(C)$ transforms fuzzy C to $\begin{matrix} C_1 & C_2 & \dots & C_m \\ C_1 & C_2 & \dots & C_m \end{matrix}$PrfSprt.

Can be considered $Q(\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix} PrfSprt\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix})$, Q-any operator.

Dynamic continual PrfSprt – elements.

Definition F.2.4.1. The process of containing the fuzzy set of continual elements $\widetilde{g(t)}=(g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t)),\ g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t)),\ \dots,\ g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t)))$ into one point x= $(x_1, x_2, ..., x_n)$ of the space X at time will be called the dynamic continual PrfSprt – element. We will denote PrfSprt(t)$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Definition F.2.4.2. The process of containing an ordered fuzzy set of continual elements at one point in space is called dynamic continual ordered PrfSprt – element.

It is allowed to sum dynamic continual PrfSprt – elements:

$$PrfSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} + PrfSprt(t)\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} =$$

PrfSprt$(t)\begin{matrix} g_1(t) \cup b_1(t) & g_2(t) \cup b_2(t) & \dots & g_n(t) \cup b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$

It's allowed to multiply dynamic continual PrfSprt – elements: $PrfSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} * PrfSprt(t)\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix} = PrfSprt\begin{matrix} g_1(t) \cap b_1(t) & g_2(t) \cap b_2(t) & \dots & g_n(t) \cap b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Parallel dynamic fuzzy continual containment of oneself in oneself as an element.

Definition F.2.4.3. The dynamic continual PrfSprt-capacity $PrfSprt(t)\begin{matrix} R_1(t) & R_2(t) & \dots & R_n(t) \\ Q_1(t) & Q_2(t) & \dots & Q_n(t) \end{matrix}$ is the process of embedding fuzzy continual $R_i(t)$ into fuzzy continual $Q_i(t)$, ($i = 1, \dots, n$).

Definition F.2.4.4. The dynamic parallel containment fuzzy continual A(t) of oneself of the first type is the process of parallel putting A(t) into A(t). Denote $PrfS_1f(t)A(t)$.

Definition F.2.4.5. The dynamic parallel containment fuzzy continual C(t) of oneself of the second type parallel contains the fuzzy continual elements from which it can be parallel fuzzy generated. Denote $PrfS_2f(t)C(t)$.

Definition F.2.4.6. The partial parallel dynamic containment fuzzy continual B(t) of oneself of the third type is the process of partial parallel embedding fuzzy continual B(t) into oneself or parallel embedding fuzzy continual elements from which it can be parallel fuzzy generated in part or both simultaneously. Denote $\Pr fS_3f(t)B(t)$.

The connection of dynamic continual PrfSprt – elements with parallel dynamic fuzzy continual containment of oneself in oneself as an element.
Let us consider the partial parallel dynamic fuzzy continual containment of oneself in oneself as an element of the third type. For example, based on $PrfSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where $\widetilde{g(t)}=(g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t)), g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t)), \dots, g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t)))$, i.e. n - continual elements at one point x= $(x_1, x_2, ..., x_n)$, one can consider the parallel dynamic fuzzy continual containment $PrfS_3f(t)$ of oneself in oneself as an element with m fuzzy continual elements from $\widetilde{g(t)}$m<n, which is a process that is necessary form according to the form (F.2.1.1), i.e.„ only m fuzzy continual elements from $\widetilde{g(t)}$ are located in the structure $PrfSprt(t)\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$. Parallel dynamic fuzzy continual containments of oneself in oneself as an element of the third type can be formed for any other structure, not necessarily PrfSprt, only by necessarily reducing the number of fuzzy continual elements in the structure. In particular, with the help of forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

It is possible to introduce structures more complex than $PrfS_3f(t)$.

Parallel dynamic fuzzy continual mathematics self.

1. The process of simultaneous parallel addition of fuzzy sets continual elements $\left\{\widetilde{g_i(t)}\right\} =$
$\left(g_{i_1}(t)|\mu_{\widetilde{g_i(t)}}(g_{i_1}(t)),\ g_{i_2}(t)|\mu_{\widetilde{g_i(t)}}(g_{i_2}(t)),\ \ldots,\ g_{i_{m_j}}(t)|\mu_{\widetilde{g_i(t)}}(\ g_{i_{m_j}}(t))\ \right),$
i = 1,2,. . . ,n, j =1,2,. . . ,k are realized by

$$PrfSprt(t)\begin{array}{ccc}\widetilde{\{g_1(t)\}}\cup & \widetilde{\{g_2(t)\}}\cup \ldots & \widetilde{\{g_n(t)\}}\cup \\ x_1 & x_2 \ldots & x_n\end{array}.$$

2. By analogy, for simultaneous multiplication:
$PrfSprt(t)\begin{array}{ccc}\widetilde{\{g_1(t)\}}\cap & \widetilde{\{g_2(t)\}}\cap \ldots & \widetilde{\{g_n(t)\}}\cap \\ x_1 & x_2 \ldots & x_n\end{array}.$

3. Similarly for simultaneous execution of various operations:
$PrfSprt(t)\begin{array}{ccc}g_1(t)Q_1(t) & g_2(t)Q_2(t) \ldots & g_n(t)Q_n(t) \\ w_1 & w_2 \ldots & w_n\end{array}$, where $\widetilde{Q(t)}=$
$(\mathrm{Q_1(t)}|\mu_{\widetilde{Q(t)}}(\mathrm{Q_1(t)}),\ \mathrm{Q_2(t)}|\mu_{\widetilde{Q(t)}}(\mathrm{Q_2(t)}), \ldots,\ \mathrm{Q_n(t)}|\mu_{\widetilde{Q(t)}}(\mathrm{Q_n(t)}))$, $\mathrm{Q_i(t)}$-an operation, i = 1, . . . , n, $\widetilde{g(t)}=(\mathrm{g_1(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_1(t)}),\ \mathrm{g_2(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_2(t)}), \ldots,$ $\mathrm{g_n(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_n(t)}))$.

4. Similarly, for the simultaneous execution of various operators: $PrfSprt(t)\begin{array}{ccc}F_1(t)g_1(t) & F_2(t)g_2(t) \ldots & F_n(t)g_n(t) \\ w_1 & w_2 \ldots & w_n\end{array}$, where $\widetilde{F(t)}=(\mathrm{F_1(t)}|\mu_{\widetilde{F(t)}}(\mathrm{F_1(t)}),\ \mathrm{F_2(t)}|\mu_{\widetilde{F(t)}}(\mathrm{F_2(t)}), \ldots,\ \mathrm{F_n(t)}|\mu_{\widetilde{F(t)}}(\mathrm{F_n(t)}))$, $\mathrm{F_i(t)}$ is an operator, i = 1, . . . , n.

5. Parallel dynamic arithmetic itself for continual containments of oneself will be similar: Parallel dynamic addition - $PrfS_1f(t)\left\{\widetilde{g(t)}\cup\right\}$, (or $PrfS_3f(t)\left\{\widetilde{g(t)}\cup\right\}$ for the third type), Parallel dynamic multiplication $PrfS_1f(t)\left\{\widetilde{g(t)}\cap\right\}$, ($PrfS_3f(t)\left\{\widetilde{g(t)}\cap\right\}$).

6. Similarly with different operations: $PrfS_1f(t)\left\{\widetilde{g(t)}\widetilde{Q(t)}\right\}$, ($PrfS_3f(t)\left\{\widetilde{g(t)}\widetilde{Q(t)}\right\}$) and with different operators:
$PrfS_1f(t)\left\{\widetilde{F(t)}\widetilde{g(t)}\right\}$, ($PrfS_3f(t)\left\{\widetilde{F(t)}\widetilde{g(t)}\right\}$).

7. . $PrfSprt(t)\begin{array}{c}A_1(t)\ A_2(t)\ \ldots\ A_n(t) \\ B_1(t)\ B_2(t)\ \ldots\ B_n(t)\end{array}$ *gives* the result

$$PrfSrt(t)\begin{array}{c}A_1(t)\ A_2(t)\ \ldots\ A_n(t) \\ B_1(t)\ B_2(t)\ \ldots\ B_n(t)\end{array} =$$
$$\left\{\sum_{i=1}^{n} A_i(t)\cup B_i(t) - A_i(t)\cap B_i(t)\ ,\sum_{i=1}^{n} D_i(t)\right\},$$

$for\ fuzzy\ continual\ sets\ A_i(t),\ B_i(t)$, where $D_i(t)$ is self-($fuzzy$ set) for $A_i(t) \cap B_i(t)$, (i = 1, 2, …, n). The same is true for structures if they are treated as $fuzzy$ continual sets,

$$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\mathrm{PrfSrt(t)} =$$

$$\left\{\begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\ \} & \{\ \}\ldots & \{\ \} \\ D_1(t) - D_1(t) \cap C_1(t) & D_2(t) - D_2(t) \cap C_2(t)\ \ldots & D_2(t) - D_2(t) \cap C_2(t) \end{matrix} PrfSrt \\ \sum_{i=1}^{m} \big(\ C_i(t) - D_i(t) \cap C_i(t)\big) - (D_i(t) - D_i(t) \cap C_i(t)) \end{matrix}\right\}$$

$for\ fuzzy\ continual\ sets\ C_i(t),\ D_i(t)$, where $Q_i(t)$ is oself-($fuzzy$ set) for $\big(D_i(t) \cap C_i(t)\big)$(i = 1, 2, …, m) [14].

8. Similarly, for dynamic continual PrfSprt-derivatives, dynamic continual PrfSprt-integrals, dynamic continual PrfSprt-lim, dynamic continual Prself-derivatives, dynamic continual Prself-integrals

9. Denote dynamic continual Prself-(dynamic continual Prself-Q(t)) through dynamic continual Prself^2-Q(t), PrfS(t)(n,Q(t))= dynamic continual Prself-(dynamic continual Prself-(… (dynamic continual Prself-Q(t)))) = dynamic continual $\mathrm{Prself}^{\mathrm{n}}$-Q(t) for n-multiple dynamic continual Prself.

Remark F.2.4.1. The parallel dynamic continual PrfSprt-displacement will be denote by $\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\mathrm{PrfSprt(t)}$, where fuzzy continual $\mathrm{D_1(t)}$ is forced out of fuzzy continual $\mathrm{C_1(t)}$, fuzzy continual $\mathrm{D_2(t)}$ is forced out of fuzzy continual $\mathrm{C_2(t)}$, …, fuzzy continual $\mathrm{D_m(t)}$ is forced out of fuzzy continual $\mathrm{C_m(t)}$, the result of this process will be described by the expression $\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix}\mathrm{PrfSrt(t)}$. Then the notation

$$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix} PrfSprt(t) \begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$$

where fuzzy continual $\mathrm{A_1(t)}$ fits into fuzzy continual $\mathrm{B_1(t)}$, fuzzy continual $\mathrm{A_2(t)}$ fits into fuzzy continual $\mathrm{B_2(t)}$, …, fuzzy continual $\mathrm{A_n(t)}$ fits into fuzzy continual $\mathrm{B_n(t)}$, fuzzy continual $\mathrm{D_1(t)}$ is forced out of fuzzy continual $\mathrm{C_1(t)}$, fuzzy continual $\mathrm{D_2(t)}$ is forced out of fuzzy continual $\mathrm{C_2(t)}$, …, fuzzy continual $\mathrm{D_m(t)}$ is forced out of fuzzy continual $\mathrm{C_m(t)}$ simultaneously. It is dynamic continual PrfSprt-containment of fuzzy continual $\mathrm{A_i(t)}$ in fuzzy continual $\mathrm{B_i(t)}$ and dynamic continual PrfSprt-displacement of fuzzy continual $\mathrm{D_j(t)}$ from fuzzy continual $\mathrm{C_j(t)}$ simultaneously, (i = 1, 2, …, n, j = 1, 2, …, m). The result of this process will be described by the expression

$$\begin{matrix} C_1(t) & C_2(t) & \ldots & C_m(t) \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{matrix} PrfSrt(t) \begin{matrix} A_1(t) & A_2(t) & \ldots & A_n(t) \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{matrix}$$

$PrfSprt(t)\begin{matrix} B_1(t)\ B_2(t)\ \dots\ B_n(t) \\ B_1(t)\ B_2(t)\ \dots\ B_n(t) \end{matrix}$ will mean $PrfS_1f(t)B(t)$.

$\begin{matrix} C_1(t)\ C_2(t)\ \dots\ C_m(t) \\ C_1(t)\ C_2(t)\ \dots\ C_m(t) \end{matrix} PrfSprt(t)$ denotes the parallel dynamic expelling fuzzy continual $\widetilde{C(t)}$=($C_1(t)|\mu_{\widetilde{C(t)}}(C_1(t))$, $C_2(t)|\mu_{\widetilde{C(t)}}(C_2(t))$, …, $C_n(t)|\mu_{\widetilde{C(t)}}(C_n(t))$) oneself out of oneself,

$\begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_m(t) \\ A_1(t)\ A_2(t)\ \dots\ A_m(t) \end{matrix} PrfSprt(t) \begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_n(t) \\ A_1(t)\ A_2(t)\ \dots\ A_n(t) \end{matrix}$ —simultaneous parallel dynamic containment fuzzy continual $\widetilde{A(t)}$=($A_1(t)|\mu_{\widetilde{A(t)}}(A_1(t))$, $A_2(t)|\mu_{\widetilde{A(t)}}(A_2(t))$, …, $A_n(t)|\mu_{\widetilde{A(t)}}(A_n(t))$) of oneself in oneself and parallel dynamic expelling fuzzy continual A(t) oneself out of oneself. $\begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_m(t) \\ A_1(t)\ A_2(t)\ \dots\ A_m(t) \end{matrix} PrfSprt(t)$ will be called parallel dynamic anti-(fuzzy continual capacity) from oneself.

Remark F.2.4.2. $PrfSprt(t)\begin{matrix} A_1(t)\ A_2(t)\ \dots\ A_n(t) \\ A_1(t)\ A_2(t)\ \dots\ A_n(t) \end{matrix}$ can be interpreted as a multilayer shell of a self-object from the first layer, which is specified by $A_1(t)$ to the nth, which is specified by $A_n(t)$. Based on this, the atomic model can be interpreted as $PrfSprt(t)\begin{matrix} \cup\{p,n\} & A_1(t)\ \dots\ A_n(t) \\ position\ of atomic\ nucleus & A_1(t)\ \dots\ A_n(t) \end{matrix}$' $\{p,n\}$ - protons, neutrons, $A_i(t)$ correspond to orbitals, i = 1, ..., n. $PrfSprt(t)\begin{matrix} physical\ body\ of\ a\ living\ organism & V_1(t)\ \dots\ V_n(t) \\ position\ of physical\ body & V_1(t)\ \dots\ V_n(t) \end{matrix}$ - model of a living organism with the multilayer shell of a living organism from the first layer, which is specified by $V_1(t)$ to the nth, which is specified by $V_n(t)$. You can also try to consider the operator PrfSprt$\begin{matrix} B_1\ \dots\ {B_i}^a\ \dots B_n \\ r_1\ \dots\ {r_i}^a\ \dots r_n \end{matrix}$, which represents the interpretation of the position of the assemblage point on the cocoon of a living organism, r_j is its potential position, B_j is a potential set of subtle energies in this position (i = 1, ..., i-1, i+1, …, n), ${r_i}^a$ is its active position, ${B_i}^a$ is an active set of subtle energies in this position, i = 1, ..., n.

Connection of dynamic continual PrfSprt – elements with target weights with parallel dynamic fuzzy continual containment of oneself with target weights. Consider a third type of parallel partial dynamic fuzzy continual containment of oneself with target weights g(t). For example, based on PrfSprt(t)$\begin{matrix} g_1(t)w(t) & g_2(t)w(t) & \dots & g_n(t)w(t) \\ x_1 & x_2\ \dots & & x_n \end{matrix}$.where $\widetilde{g(t)}$=($g_1(t)|\mu_{\widetilde{g(t)}}(g_1(t))$, $g_2(t)|\mu_{\widetilde{g(t)}}(g_2(t))$, …, $g_n(t)|\mu_{\widetilde{g(t)}}(g_n(t))$).

i.e. n - fuzzy continual elements with target weights{w(t)} at one point x = $(x_1, x_2, ..., x_n)$, we can consider the dynamic continual containment $PrfS_3f(t)\widetilde{g(t)}w(t)$ of oneself with target weights with m fuzzy continual elements with target weights{w(t)} from $\widetilde{g(t)}$, m<n, which is the process of formation according to the form (1.1), i.e., only m fuzzy continual elements with target weights{w(t)} from $\widetilde{g(t)}$are located in the structure $PrfS_3f(t)\widetilde{g(t)}w(t)$. Parallel dynamic fuzzy containments of oneself with target weights of the third type can be formed for any other structure, not necessarily PrfSprt, only by reducing the number of continual elements with target weights in the structure. In particular, using the forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). Structures more complex than $PrfS_3f(t)\widetilde{g(t)}w(t)$ can be introduced.

Definition F.2.4.7. The parallel dynamic embedding of fuzzy continual A(t) into itself with target weights {w(t)} of the first type is the process of parallel embedding A(t) into A(t) with target weights. Denote $PrfS_1f(t)A(t)w(t)$.

Definition F.2.4.8. The parallel dynamic containment of fuzzy continual C(t) itself into itself with target weights{w(t)} of the second type is the process of parallel containment of the fuzzy continual elements from which it can be parallel fuzzy generated. Let's denote $PrfS_2f(t)C(t)w(t)$.

Definition F.2.4.9. Partial parallel dynamic containment of continual B(t) itself into itself with target weights {w(t)} of the third type is the process of partial parallel containment of continual B(t) into itself or continual elements from which it can be parallel generated partially, or both at the same time. Denote $PrS_3f(t)B(t)w(t)$.

The usage of PrfSprt-elements for networks.

A. Galushkin's comprehensive monograph [6] covers all aspects of networks, but traditional approaches go through classical mathematics, mainly through the usual correspondence operators. Here we consider a different approach - through a new mathematical process with parallel fuzzy containment operators, which, although they can be interpreted as the result of some correspondence operators, are not themselves correspondence operators. Parallel fuzzy containment operators are more convenient for networks. Also, the main emphasis was placed on using processors operating using triodes, which are generally not used in Sprt-networks. PrfSprt-networks (SmnPrfSprt) are a PrfSprt-structure that can be built for the required weights. PrfSprt-OS (PrfSprt operating system) uses PrfSprt-coding and PrfSprt-translation. In the first one, coding is carried out through a 2-dimensional matrix-row (a, b), where the number b is the code of the action, and the number a is the code of the object of this action. PrfSprt-coding (or Prfself-coding) is implemented through a matrix consisting of 2 columns (in the continuous case, two intervals of numbers). Here, the source encoding is used for all matrix rows simultaneously. PrfSprt-translation is carried out by inversion. In this case, Prfself-coding and Prfself-translation will be more stable. The target weights $g_i(t)$ in

PrfSprt(t)$\begin{matrix} activation\ with\ g_1(t) & activation\ with\ g_2(t) \dots & activation\ with\ g_n(t) \\ SmnPrfSprt & SmnPrfSprt \dots & SmnPrfSprt \end{matrix}$ are chosen for necessary tasks. We will not touch on the issues of applications, or network optimization. They are described in detail by Galushkin [6]. We will touch on the difference of this only for hierarchical complex networks. The same simple executing programs are in the cores of simple artificial neurons of type PrfSprt (designation - mnPrfSprt) for simple information processinF. More complex executing programs are used for mnPrfSprt nodes. PrfSprt-threshold element –sgn($PrfSprt(t)^{\begin{matrix} g_1(t)w_1(t) & g_2(t)w_n(t) \dots & g_n(t)w_n(t) \\ x_1 & x_2 \dots & x_n \end{matrix}}$), $x=(x_1,x_2,\dots,x_n)$ - mnPrfSprt, $\widetilde{W(t)}=(w_1(t)|\mu_{\widetilde{w(t)}}(w_1(t)),\ w_2(t)|\mu_{\widetilde{w(t)}}(w_2(t)),\dots,$ $w_n(t)|\mu_{\widetilde{w(t)}}(w_n(t)))$.

– source signals values, $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ – PrfSprt-synapses weights. The first level of mnPrfSprt consists of simple mnPrfSprt. The second level of mnPrfSprt consists of PrfSprt(t)$^{\begin{matrix} mnPrfSprt & mnPrfSprt \dots & mnPrfSprt \\ D_1 & D_2 \dots & D_n \end{matrix}}$– PrfSprt-node of mnPrfSprt in range D = $(D_1, D_2,\dots, D_n)$, D- capacity for mnPrfSprt node. The third level of mnPrfSprt consists of PrfSprt(t)$^{\begin{matrix} PrfSprt(t)^{\begin{matrix} q & q \dots & q \\ D_1 & D_2 \dots & D_n \end{matrix}} & PrfSprt(t)^{\begin{matrix} q & q \dots & q \\ D_1 & D_2 \dots & D_n \end{matrix}} \dots & q \\ D_1 & D_2 \dots & D_n \end{matrix}}$ - $PrfSprt^2$- node of q = mnPrfSprt in range D, thus D becomes capacity of itself in itself as an element for mnPrfSprt. For our networks, it is sufficient to use $PrfSprt^2$- nodes of mnPrfSprt, but self-level is higher in living organisms, particularly $PrfSprt^n$-, n3. The target structure or the corresponding program enters the target unit using alternating current. After that, all networks or parts of them are activated according to the indicative goal. It may appear that we are leaving the network ideology, but these networks are a complex hierarchy of different levels, like living organisms.

Remark 5.0. A neural network can be thought of as a learnable parallel dynamic operator.

Remark 5.1. Traditional scientific approaches through classical mathematics make it possible to describe only at the usual energy level. Here we consider an approach that makes describing processes with finer energies possible. mnPrfSprt contains PrfSprt(t)$^{\begin{matrix} feprogram_1(t) & feprogram_2(t) \dots & feprogram_n(t) \\ mnPrfSprt & mnPrfSprt \dots & mnPrfSprt \end{matrix}}$, $eprogram-$executing program in PrfSprt- OS. PrfSprt-OS (or Prself-OS) is based on PrfSprt-assembly language (or Prself-assembly language), which is based on assembly language through PrfSprt-approach in turn, if the base of elements of PrfSprt-networks is sufficient. The feprograms are in PrfSprt-programming environments (or Prfself-programming environments), but this question and PrfSprt-networks base will be considered in the following monographs. In particular, eprograms may contain PrfSprt- programming operators. In mnPrfSprt cores, the constant memory PrfSprt with correspondent eprograms depending on mnPrfSprt.

The OS (operating system) and the principles and modes of operation of the PrfSprt-networks for this programming are interestinF. But this is already the material for the next publications.

Here is developed a helicopter model without a main and tail rotors based on PrfSprt – physics and special neural networks with artificial neurons operating in normal and PrfSprt-modes. Let's denote this model through SmnPrfSprt. To do this, it's proposed to use mnPrfSprt of different levels; for example, for the usual mode, mnPrfSprt serves for the initial processing of signals and the transfer of information to the second level, etc., to the nodal center, then checked. In case of an anomaly - local PrfSprt–mode with the desired "target weight" is realized in this section, etc., to the center. In the case of a monster during the test, SmnPrfSprt is activated with the desired "target weight." Here are realized other tasks also. To reach the fself-energy level, the mode $Sprt_{SmnPrfSprt}^{SmnPrfSprt}$ is used. In normal mode, it's planned to carry out the movement of SmnPrfSprt on jet propulsion by converting the energy of the emitted gases into a vortex to obtain additional thrust upwards. For this purpose, a spiral-shaped chute (with "pockets") is arranged at the bottom of the SmnPrfSprt for the gases emitted by the jet engine, which first exit through a straight chute connected to the spiral one. There is drainage of exhaust gases outside the SmnPrfSprt. SmnPrfSprt is represented by a neural network that extends from the center of one of the main clusters of PrfSprt - artificial neurons to the shell, turning into the body itself. Above the operator's cabin is the central core of the neural network and the target block, responsible for performing the "target weights" and auxiliary blocks, the functions and roles of which we will discuss further. Next is the space for the movement of the local vortex. The unit responsible for SmnPrfSprt's actions is below the operator's cab. In PrfSprt – mode, the entire network or its sections are PrfSprt – activated to perform specific tasks, in particular, with "target weights." In the target, block used PrfSprt -coding, PrfSprt -translation for activation of all networks to "target weights" simultaneously, then –the reset of this PrfSprt-coding after activation. Unfortunately, triodes are not suitable for PrfSprt -neural networks. In the most primitive case, usual separators with corresponding resistances and cores for eprograms may be used instead triodes since there is no necessity to unbend the alternating current to direct. The PrfSprt-operative memory belt is disposed around a central core of SmnPrfSprt. There are PrfSprt-coding, PrfSprt-translation, and PrfSprt-realize of eprograms and the programs from the archives without extraction, PrfSprt-coding and PrfSprt-translation may be used in high-intensity, ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna, Strickland. PrfSprt – structure or an eprogram if one is present of needed «target weight» are taken in target block at PrfSprt – activation of the networks. $Sprt_{activation}^{SmnPrfSprt,f}$ derives SmnPrfSprt to the self-level boundary with target weight f.

It's used an alternating current of above high frequency and ultra-violet light, which can work with PrfSprt – structures in PrfSprt–modes by its nature to activate the networks or some of its parts in PrfSprt–modes and locally using PrfSprt–mode. Above high frequently alternating current go through mercury bearers. That's why overheating does not occur. The power of the alternating current above high frequently increases considerably for the target block. The activation of all networks is realized to indicate "target weights."

Variable hierarchical dynamical parallel structures (models) for

dynamic, singular, hierarchical fuzzy sets.

Here we will consider variable parallel structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable structures (models), for example,

$$\begin{matrix} C_1 & C_2 & \ldots & C_m \\ D_1 & D_2 & \ldots & D_m \end{matrix} \text{PrfSprt(t)} \begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix} =$$

$$\begin{cases} (\begin{matrix} C_1 & C_2 & \ldots & C_m \\ D_1 & D_2 & \ldots & D_m \end{matrix} PrfSprt, \quad q_2 \geq t \geq q_1)|\mu_1 \\ (\begin{matrix} B_1 & B_2 & \ldots & B_m \\ D_1 & D_2 & \ldots & D_m \end{matrix} PrfSprt \begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix}, q_3 \geq \ t > q_2)|\mu_2 \\ (\begin{matrix} C_1 & C_2 & \ldots & C_m \\ D_1 & D_2 & \ldots & D_m \end{matrix} PrfSprt \begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix}, q_4 \geq \ t > q_3)|\mu_3 \\ (PrfSprt \begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix}, q_5 \geq \ t > q_4)|\mu_4 \\ (\begin{matrix} \{\ \} & \{\ \} & \ldots & \{\ \} \\ D_1 & D_2 & \ldots & D_m \end{matrix} PrfSprt, \quad t > q_5)|\mu_5 \\ \ldots \end{cases} \qquad (*_{\mathrm{F.2.1}}),$$

μ_i- measures of fuzziness, i = 1, ..., 5. In particular, $\begin{matrix} B_1 & B_2 & \ldots & B_m \\ D_1 & D_2 & \ldots & D_m \end{matrix} PrfSprt \begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix}$ can be interpreted as a game: player 1 fits fuzzy A_i into fuzzy B_i, i = 1, 2, . . . , n, and the other pushes fuzzy D_j out of fuzzy Bj, j = 1, 2, . . . , m at the same time.

The example of variable parallel hierarchy $\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} PrfSprt_1(t) \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$

$$==\begin{cases} \left(\left\{\begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\,\} & \{\,\}\dots & \{\,\} \\ D_1 - D_1\cap C_1 & D_2 - D_2\cap C_2\ \dots & D_m - D_m \cap C_m \end{matrix} PrfSrt \\ \sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix}\right\}, \quad q_2 \geq t \geq q_1)|\mu_1 \right. \\ \left(\sum_{i=1}^{n} \left(\begin{matrix} S^{1e}_{0} f B_i * \\ {}^{B_i}_{Q_i - B_i} S^1 t^{A_i - B_i}_{B_i} \end{matrix} \right), q_3 \geq\ t > q_2)|\mu_2 \right. \\ \left(\sum_{j=1}^{m} \sum_{i=1}^{n} \left(\begin{matrix} S^{et}_{01} f B_i \\ {}^{C_j - B_i}_{C_j - B_i} S_1 t^{A-B_i}_{B_i} \\ {}^{C_j - B_i}_{D_j - C_j - B_i} S_1 t^{A-B_i}_{B_i} \end{matrix} \right) \quad, q_4 \geq\ \ t > q_3)|\mu_3 \right. \\ \left(\left\{ \begin{matrix} \sum_{i=1}^{n} R_i \\ \sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \end{matrix} \right\}, \ , q_5 \geq\ t > q_4)|\mu_4 \right. \\ \left({}^{\{\,\}}_{D} Sprt \begin{matrix} \{\,\} & \{\,\} & \dots & \{\,\} \\ D_1 & D_2 & \dots & D_m \end{matrix} PrfSrt, \quad t > q_5)|\mu_5 \right. \\ \dots \end{cases}$$

$(*_{\mathrm{F.2.2}})$,

where μ_i- measures of fuzziness, i = 1, ..., 5, $\mathrm{Q_i}$ is oself-(fuzzy set) for $\left(D_j \cap C_j\right)$, $D_j,\ C_j$ are fuzzy sets (j = 1, 2, …, m), [14], $\mathrm{R_i}$ is self-(fuzzy set) for $A_i \cap B_i$, $A_i,\ B_i$ are fuzzy sets ,(i = 1, 2, …, n), $S^{et}_{01} f B$, ${}^{C-B}_{C-B} S_1 t^{A-B}_{B}$, ${}^{C-B}_{D-C-B} S_1 t^{A-B}_{B}$ are considered in [13], ${}^{B}_{Q-B} S^1 t^{A-B}_{B}$ is considered in [10].

In what follows, we will denote variable parallel fuzzy structure (model) through PrfVS, parallel self-type variable fuzzy structures (models) through PrfSVS, and parallel oself-type variable fuzzy structures (models) through PrfOSVS.

Examples: a) discrete variable parallel fuzzy structure

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \\ A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \begin{matrix} a & b & g \\ c & PrfVS & w \\ d & q & r \end{matrix} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \\ C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}$$

c) continuous variable parallel fuzzy structure

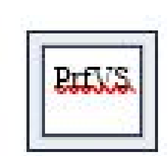


Fig. 0.2

Where a continuous set represents the rim of the Fig.0.2.

We introduce the notation $m_{PrfVS_N}-$ the number of elements, N - the number of connections between them in the discrete variable parallel 2-hierarchical fuzzy structure PrfVS. We introduce the notation $q_{PrfVS_R}-$ any, R - connections in q_{PrfVS_R} in the variable parallel 2-hierarchical structure PrfVS, in particular, q_{PrfVS_R}, R can be fuzzy sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the structurability of Q from the interval [0,1], where 0 corresponds to "no parallel fuzzy structure"," and 1 corresponds to the value " parallel fuzzy structure". Then for joint fuzzy A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D- parallel self-type fuzzy structures from A*B, cS(x)- the value of Prself for parallel self-type fuzzy structures x; for dependent parallel fuzzy structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional structurability of the parallel fuzzy structure B at the parallel fuzzy structure A, c(A/B)- conditional structure of the parallel fuzzy structure A at the parallel fuzzy structure B. Adding inconsistent parallel fuzzy structures: c(A+B) =c(A) + c(B). The formula of complete parallel fuzzy structure: c(A)=$\sum_{k=1}^{n} c\,(B_k) * c(A/B_k)$, B_1, $B_{2.}$,.., B_n-full group of hypotheses- fuzzy containments: $\sum_{k=1}^{n} c\,(B_k)$=1("parallel fuzzy structure"). PrfSprt- structure of the first type for set of parallel fuzzy structures A={A_1, A_2,...,A_n}: PrfSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, PrfSprt$\begin{matrix} c(A_1) & c(A_2) & \dots & c(A_n) \\ x_1 & x_2 & \dots & x_n \end{matrix}$ PrfSprt- structurability for these structures. It is possible to consider the parallel self-type fuzzy structure $PrfS_3A$ with m parallel structures and from A, at m<n, which is formed by the form (1.1), that is, only m parallel fuzzy structures from A are located in the structure $PrfSprt\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. The same for parallel fuzzy self-type structurability $PrfS_3\{\ c(A_1), c(A_2), \dots, c(A_n)$.

Can be considered N-hierarchical parallel structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical parallel structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, ... can be any in particular, by actions, sets, and others.

Can be considered discrete hierarchical parallel structure, continuous hierarchical parallel structure, and discrete-continuous hierarchical parallel structure.

$$\begin{array}{c} PrfSprt^{N-lhs_1\ N-lhs_2\ \dots\ N-lhs_n}_{x_1\qquad x_2\ \dots\qquad x_n} \\ \dots \\ PrfSprt^{i-lhs_1\ i-lhs_2\ \dots\ i-lhs_n}_{x_1\qquad x_2\ \dots\qquad x_n} \\ \dots \\ PrfSprt^{1-lhs_1\ 1-lhs_2\ \dots\ 1-lhs_n}_{x_1\qquad x_2\ \dots\qquad x_n} \end{array}$$

The example PrfQHS=$fHSprt_x$ -

N-hierarchical structure compression into point x = $(x_1, x_2, ..., x_n)$, lhs = level of hierarchical structure.

Let Prf(N, PrfQHS)= $PrfQHS^{\left.PrfQHS^{PrfQHS^{\dots PrfQHS}}\right\}\text{-N levels}}$

It can be considered Prself- PrQHS, Prf(y, PrfQHS) for any y, Prf(PrfQHS, PrfQHS).

Parallel Compression Hierarchy Example:

$$PrfSprt^{\begin{array}{ccc} PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} & PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}\ \dots & PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} \\ PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}+B & PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}+B\ \dots & PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)}+B \end{array}}$$

$$=\left(\begin{array}{c} PrfSprt^{\begin{array}{ccc} PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} & PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} & \dots\ PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} \\ PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} & PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} & \dots\ PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} \end{array}} \\ PrfSprt^{\begin{array}{ccc} PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} & PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} & \dots\ PrfSprt^{(\,)\,(\,)\,\dots\,(\,)}_{(\,)\,(\,)\,\dots\,(\,)} \end{array}} \\ \begin{array}{ccc} B & B\ \dots & B \end{array} \end{array}\right)$$

Let's consider two versions: 1) containment is interpreted through the concept of containment, and 2) capacity is interpreted through the concept of containment as a rest point of containment. PrSelf-containment is interpreted as a rest point of Prself-containment. We consider the functional ca(Q), which gives a numerical value for the accommodation of Q from the interval [0,1], where 0 corresponds to " parallel containment", and one corresponds to the value " parallel capacity. " Then for joint A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D- Prself-containment for A*B, caS(x)- the value of Prself-capacity for Prself-containment of x; for dependent parallel containments: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional accommodation of the parallel containment B at the parallel containment A, ca(A/B)- conditional parallel capacity of the parallel containment A at the parallel containment B. Adding the parallel capacity values of inconsistent parallel containments: ca(A+B)=ca(A)+ca(B). The formula of complete parallel capacity: ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2,..,B_n-full group of hypotheses-(parallel containments): $\sum_{k=1}^{n} ca\,(B_k)$=1(“parallel capacity”). PrfSprt- containment for set of parallel containments A={A_1, A_2,…,A_n}: PrfSprt$^{A_1\ A_2\ \dots\ A_n}_{x_1\ x_2\ \dots\ x_n}$, PrfSprt$^{ca(A_1)\ ca(A_2)\ \dots\ ca(A_n)}_{x_1\ \ \ x_2\ \dots\ \ \ x_n}$- PrfSprt- accommodation for these parallel containments. It is possible to consider the Prfself-containment $PrfS_3A$ with m containments and from A, at m<n, which is formed by the form (1.1), that is, only m parallel containments from A are located in the parallel containment PrfSprt(t)$^{A_1\ A_2\ \dots\ A_n}_{x_1\ x_2\ \dots\ x_n}$, . The same for Prfself- accommodation - $PrfS_3\{\ ca(A_1), ca(A_2), \dots, ca(A_n)$.

We consider the functional h(Q), which gives a numerical value for the parallel hierarchization of Q from the interval [0,1], where 0 corresponds to "no parallel hierarchy," and 1 corresponds to the value " parallel hierarchy. " Then for joint parallel hierarchies A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- Prself-hierarchy from A*B, hS(x)- the value of Prself- hierarchy for Prself- hierarchy x; for dependent parallel hierarchies: h(A*B)=ha(A)*h(B/A)=h(B)*h(A/B), where h(B/A)- conditional parallel hierarchization of the parallel hierarchy B at the parallel hierarchy A, h(A/B)- conditional parallel hierarchy of the parallel hierarchy A at the parallel structure B. Adding the parallel hierarchy values of inconsistent parallel hierarchies: h(A+B)=h(A)+h(B). The formula of complete parallel hierarchy: h(A)=$\sum_{k=1}^{n} h\,(B_k) * h(A/B_k)$, B_1, B_2, .., B_n-full group of hypotheses-(parallel hierarches): $\sum_{k=1}^{n} h\,(B_k)$=1(“parallel hierarchy”).

PrfSprt- structure for set of parallel hierarches A={A_1, A_2,…,A_n}: $PrfSprt(t)^{A_1\ A_2\ \dots\ A_n}_{x_1\ x_2\ \dots\ x_n}$, PrfSprt(t)$^{h(A_1)\ h(A_2)\ \dots\ h(A_n)}_{x_1\ \ \ x_2\ \dots\ \ \ x_n}$- PrfSprt- hierarchization for these parallel hierarches. It is possible to consider the Prself-hierarchy PrS_3A with m parallel hierarches and from A, at m<n, which is formed by the form (1.1), that is, only m parallel hierarches from A are located in the parallel hierarchy $PrfSprt(t)^{A_1\ A_2\ \dots\ A_n}_{x_1\ x_2\ \dots\ x_n}$. The same for Prself- hierarchization $PrS_3\{\ h(A_1), h(A_2), \dots, h(A_n)$.Can be considered PrfSprt(t)$^{\{\ ca(A_1), c(A_1), h(A_1)\}\ \ \{ca(A_2), c(A_2), h(A_2)\}\ \dots\ \ \{ca(A_n), c(A_n), h(A_n)\}}_{x_1\ x_2\ \dots\ x_n}$.

Very interesting next parallel hierarchy type:

$\begin{matrix} hi\ A_1 & h\ A_2\ \dots & h\ A_n \\ h\ A_1 & hierarchy\ A_2\ \dots & h\ A_n \end{matrix}$, PrfSprt(t)$\begin{matrix} h\ A_1\ h\ A_2\ \dots\ h\ A_n \\ h\ A_1\ h\ A_2\ \dots\ h\ A_n \end{matrix}$, h = hierarchy. You can enter special operator PrCprt to work with structures: $\begin{matrix} A_1 & A_2\ \dots & A_n \\ B_1 & B_2\ \dots & B_n \end{matrix} PrCprt \begin{matrix} R_1 & R_2\ \dots & R_m \\ Q_1 & Q_2\ \dots & Q_m \end{matrix}$ structures R_j with the structure from Q_j, unstructures A_i by the structure B_i, simultaneously, (i = 1, 2, …, n, j = 1, 2, …, m). Very interesting next parallel structure type:

$\frac{A_1\ A_2\ \ldots\ A_n}{A_1\ A_2\ \ldots\ A_n}$, PrCrt(t)$\frac{A_1\ A_2\ \ldots\ A_n}{A_1\ A_2\ \ldots\ A_n}$,

You can enter special parallel operator PrHprt to work with hierarches:
$\frac{A_1\ \ A_2\ \ldots\ \ A_n}{B_1\ \ B_2\ \ldots\ \ B_n} PrHprt \frac{R_1\ \ R_2\ \ldots\ \ R_m}{Q_1\ \ Q_2\ \ldots\ \ Q_m}$

hierarchizes R_j with the hierarchy from Q_j, unhierarchizes A_i from the hierarchy B_i, simultaneously, (i = 1, 2, . . . , n, j = 1, 2, . . . , m).

Program operators PrfSprt, PrtSpr, S^1e, Set_1.

Here it is supposed to use a symbiosis of parallel actions and conventional calculations through sequential actions. This must be done through PrfSprt-Networks in one of the central departments of which a conventional computer system is located. The parallel processor is itself preprogram with direct parallel computing not through serial computinF.

Using conventional parallel coding by a parallel computer system, through a Target-block with a PrfSprt -program operator - $PrfSprt(t)\frac{g_1(t)w_1(t)\ g_2(t)w_n(t)\ \ldots\ g_n(t)w_n(t)}{activation\ activation\ \ldots\ activation}$, it will be possible to obtain the execution of a parallel action $(g_1(t), g_2(t), ..., g_n(t))$ with the desired target weight w(t)=(w_1(t), w_2(t),. . . ,w_n(t)). Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For PrfSprt-coding and PrfSprt-translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna Strickland, or a combination of them. For the desired action, for example, using the direct parallel program of operator PrfSprt(t)$\frac{(UHF\ AC)_1(t)Q_1(t)\ (UHF\ AC)_2(t)Q_2(t)\ \ldots\ (UHF\ AC)_n(t)Q_n(t)}{activation\qquad activation\ \ldots\qquad activation}$, we simultaneously enter the desired set of codes $Q_1(t)$ using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel program operators:

1) PrfSprt-program operators
2) PrtSpr-program operators
3) S^1e - program operators
4) Set_1- program operators

Here are some of the PrfSprt-program operators:

8. Simultaneous assignment of the expressions $\{p\} = (p_1, p_2, ..., p_n)$ to the variables $\{g\} = (g_1, g_2, ..., g_n)$. This is implemented via $\text{PrfSprt}_{p_1 \quad p_2 \dots \quad p_n}^{g_1 := g_2 := \dots \ g_n :=}$.
9. Simultaneous checking the set of conditions $\{f\} = (f_1, f_2, ..., f_n)$ for the set of expressions$\{B\} = (B_1, B_2, ..., B_n)$. Implemented via $\text{PrfSprt}_{x_1 \quad x_2 \dots \quad x_n}^{IF\{B_1 f_1\}\ then\ IF\{B_2 f_2\}\ then \dots\ IF\{B_n f_n\}\ then}$, where x_i (i = 1, . . . , n) can be anythinF.
10. Similarly for loop operators and others.

PrfSprt-algorithm Examples:

1) Simultaneous addition and simultaneous parallel multiplication of sets elements $\{g_i\} = \left(g_{i_1}, g_{i_2}, \dots, g_{i_{m_j}}\right)$, i = 1, 2, . . . , n, j =1, 2, . . . , k (See point 1, 2 in **Math Prself**
2) parallel pattern recognition: $\text{PrfSprt}_{Name\ of\ q_1\ Name\ of\ q_2 \dots\ Name\ of\ q_n}^{ia_1\ then \quad ia_2\ then \dots \quad ia_n\ then}$'
$ia = IF\ \ q_1 \in image\ archive$.

The example of PrfSprt-program is

$$PrfSprt_{x_1 \qquad x_2 \dots \qquad x_n}^{Q\ PrfSprt_{w_1 \quad w_2 \dots \quad w_n}^{IF\{B_1 f_1\}\ then\ IF\{B_2 f_2\}\ then \dots\ IF\{B_n f_n\}\ then} \dots\ D}.$$

$\text{Q} = \text{PrfSprt}_{p_1 \quad p_2 \dots \quad p_n}^{g_1 := g_2 := \dots \ g_n :=}$, $\text{D} = \text{PrfSprt}_{w_1\ w_2 \dots\ w_n}^{w_1\ w_2 \dots\ w_n}$

$\text{Pr}\, S_3 f-$ software operators will differ only just because aggregates $\{g\}, \{p\}, \{B\}, \{f\}$ will be formed from corresponding PrfSprt-program operators in form (1.1) for more complex operators in forms (1.1.1) – (1.4).

For example, $Sprt_{g\{R\}}^{\{R\}}$ is the capacity in itself of the second type if $g\{R\}$ is a program capable of generating $\{R\}$.

The example of self-program of the first type is

$$PrfSprt_{\{Sprt^{\{\{g(x)\}:=\{p\}\}}, Sprt^{IF\{\{B\}\{f\}\}\ then\ Q}, Sprt_Q^Q\}}^{\{Sprt^{\{\{g(x)\}:=\{p\}\}}, Sprt^{IF\{\{B\}\{f\}\}\ then\ Q}, Sprt_Q^Q\}}.$$

$$PrfSprt_{Q\ PrfSprt_{w_1 \quad w_2 \dots \quad w_n}^{IF\{B_1 f_1\}\ then\ IF\{B_2 f_2\}\ then \dots\ IF\{B_n f_n\}\ then} \dots\ D}^{Q\ PrfSprt_{w_1 \quad w_2 \dots \quad w_n}^{IF\{B_1 f_1\}\ then\ IF\{B_2 f_2\}\ then \dots\ IF\{B_n f_n\}\ then} \dots\ D},$$

Q = PrfSprt$\begin{matrix} g_1 := & g_2 := & \dots & g_n := \\ p_1 & p_2 & \dots & p_n \end{matrix}$, D = PrfSprt$\begin{matrix} w_1 & w_2 & \dots & w_n \\ w_1 & w_2 & \dots & w_n \end{matrix}$.

PrfSprt-coding: 1) set A_i to set B_i, 2) set A_i to a point q_i, where the elements of the sets A_i, B_i can be continuous, (i = 1, 2, . . . , n; j = 1, 2, . . . , m). For example, PrfSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$.

There are PrfSprt -coding, PrfSprt-translation, PrfSprt-realize of preprograms and of the programs from the archives without extraction theirs

Self-coding: 1) set A_i to set A_i, i.e. A_i on itself 2) set A_i to a point q_i in forms (1.1) - (1.4), where the elements of the sets A_i can be continuous. For example, PrfSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$.

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with PrfSprt-Networks, it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through PrfSprt -Networks and series-parallel. Codes from a conventional type computer system will be used via PrfSprt -connectors in PrfSprt - coding, for example: PrfSprt(t)$\begin{matrix} (UHF\ AC)_1(t)Q_1(t) & (UHF\ AC)_2(t)Q_2(t) & \dots & (UHF\ AC)_n(t)Q_n(t) \\ activation & activation & \dots & activation \end{matrix}$.
UHF AC field activation is used.

Consider the dynamic PrfSprt and $PrfS_3f(t)$ programming:

1. The process of simultaneous assignment of the expression$s$ $\{p(t)\} = (p_1(t), p_2(t), ..., p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ is implemented through PrfSprt$\begin{matrix} g_1(t) := & g_2(t) := & \dots & g_n(t) := \\ p_1(t) & p_2(t) & \dots & p_n(t) \end{matrix}$.

2. The process of simultaneous check the set of conditions $\{f(t)\} = (\ f_1\ (t), f_2\ (t),\ ...,\ f_n\ (t)\ )$ for a set of expressions$\{B(t)\} = (B_1(t),\ B_2\ (t),\ ..., B_n(t))$ is implemented through
PrfSprt$\begin{matrix} IF\ \{B_1(t)f_1(t)\}\ then & IF\ \{B_2(t)f_2(t)\}\ then & \dots & IF\ \{B_n(t)f_n(t)\}\ then \\ x_1(t) & x_2(t) & \dots & x_n(t) \end{matrix}$
where x(t) = $(x_1(t), x_2(t), ..., x_n(t))$ can be any.

3.Similarly for loop operators and others.

$PrfS_3f(t)$– software operators will differ only in that the aggregates $\{g(t)\}, \{p(t)\}, \{B(t)\}, \{f(t)\}$ will be formed from corresponding processes PrfSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) – (1.4) for more complex operators.

Consider PrftSpr-program operators. The ideology of PrftSpr and $Prft_{S_4 f}$ is parallel analogue of $t_{S_4 f}$ [16] can be used for programming. Here are some of the PrftSpr -program operators.

1. Simultaneous expelling assignment of the expressions $\{p\} = (p_1, p_2, ..., p_n)$ from the variables $\{g\} = (g_1, g_2, ..., g_n)$. It's implemented through $\begin{matrix} g_1 & g_2 \ldots & g_n \\ =: p_1 & =: p_2 \ldots & =: p_n \end{matrix}$ PrfSprt.

2. Simultaneous expelling checks the set of conditions $\{f\} = (f_1,\ f_2,\ \ ...,f_n)$ for a set of expressions $\{B\} = (B_1,\ B_2,\ \ ...,B_n)$. It's implemented through $PrfSprt_{\begin{matrix} x_1 & x_2 \ldots & x_n \\ IF\,\{B_1 f_1\}\ then & IF\,\{B_2 f_2\}\ then \ldots & IF\,\{B_n f_n\}\ then \end{matrix}}$, where x_i (i = 1, . . . , n) can be anythinF.

3. Similarly for loop operators and others.

$Prft_{S_4 f}-$ software operators will differ only just because aggregates $\{g\}, \{p\}, \{B\}, \{f\}$ will be formed from corresponding PrftSpr program operators in form (1.1) for more complex operators in forms (1.1.1) – (1.4).

Consider hierarchical PrtSpr-program operator

$$\begin{matrix} C_1 & C_2 \ldots & C_m \\ D_1 & D_2 \ldots & D_m \end{matrix}\text{PrfSrt} =$$

$$\left\{\begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\ \} & \{\ \} \ldots & \{\ \} \\ D_1 - D_1 \cap C_1 & D_2 - D_2 \cap C_2 \ldots & D_m - D_m \cap C_m \end{matrix} PrfSrt \\ \sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix}\right\},$$

where Q_i is oself-set for $(D_i \cap C_i)$ [16].

Consider the dynamic PrftSpr and $Prft(t)_{S_4 f}$ programming at time t.

1. The process of simultaneous assignment of the expressions $\{p(t)\} = (p_1(t), p_2(t), ..., p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$. is implemented through $\text{PrfSprt}^{\begin{matrix} g_1(t) & g_2(t) \ldots & g_n(t) \\ p_1(t) & p_2(t) \ldots & p_n(t) \end{matrix}}$.

2. The process of simultaneous check the set of conditions $\{f(t)\} = (\ f_1\ (t), f_2\ (t),\ ...,\ f_n\ (t)\ )$ for a set of expressions $\{B(t)\} = (B_1(t),\ B_2\ (t),\ ...,B_n(t))$ is implemented through

$$\text{PrfSprt}^{\begin{matrix} IF\,\{B_1(t) f_1(t)\}\ then & IF\,\{B_2(t) f_2(t)\}\ then \ldots & IF\,\{B_n(t) f_n(t)\}\ then \\ x_1(t) & x_2(t) \ldots & x_n(t) \end{matrix}}$$

where x(t) = $(x_1(t), x_2(t), ..., x_n(t))$. can be any.

3.Similarly for loop operators and others.

$\Pr t(t)_{S_4 f}-$ software operators will differ only in that the aggregates $\{g(t)\}, \{p(t)\}, \{B(t)\}, \{f(t)\}$ will be formed from corresponding processes

PrfSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) – (1.4) for more complex operators.

Consider hierarchical dynamic PrfSrt(t)-program operator:

$$
{}^{C_1(t)\ C_2(t)\ \dots\ C_m(t)}_{D_1(t)\ D_2(t)\ \dots\ D_m(t)}\mathrm{PrfSrt(t)} =
\left\{ \begin{matrix} \sum_{i=1}^{m} Q_i(t) + \underset{\sum_{i=1}^{m}\left(C_i(t) - D_i(t) \cap C_i(t)\right) - \left(D_i(t) - D_i(t) \cap C_i(t)\right)}{\overset{\{\ \}}{D_1(t) - D_1(t) \cap C_1(t)}\ \overset{\{\ \}\ \dots}{D_2(t) - D_2(t) \cap C_2(t)}\ \dots\ \overset{\{\ \}}{D_2(t) - D_2(t) \cap C_2(t)}}{}^{PrfSrt} \end{matrix} \right\},
$$

where $Q_i(t)$ is oself-set for $\left(D_i(t) \cap C_i(t)\right)$ [14].

Consider PrfSrt- program operators

$$
fSprt_{t_0}^{\left\{ {}^{q}\begin{pmatrix} u\ u & & u\ u \\ & PrfSpr & \\ u\ u & & u\ u \end{pmatrix}_{W_q} fSprt^{E_q}_{{}^{q}\begin{pmatrix} u\ u & & u\ u \\ & PrfSpr & \\ u\ u & & u\ u \end{pmatrix}},\ fSprt_{d_r}^{\{E^{ex} l^{d_r}\}}\left(E_{in} l^{d_r}\right) \right\}}
$$

—program structure example, where the assemblage point d_r is the cursor, it is quite complex self—program.

Remark. Energy of a living organism other than humans:

$$
\mathrm{fPrf}(\mathrm{r}, \mathrm{u}(E_q)) = fSprt_{t_0}^{\left\{ {}^{q}\begin{pmatrix} u\ u & & u\ u \\ & PrfSpr & \\ u\ u & & u\ u \end{pmatrix}_{W_q} fSprt^{E_q}_{{}^{q}\begin{pmatrix} u\ u & & u\ u \\ & PrfSpr & \\ u\ u & & u\ u \end{pmatrix}},\ fSprt_{d_r}^{\{E^{ex} l^{d_r}\}}\left(E_{in} l^{d_r}\right) \right\}} \qquad (^{**}{}_{\mathrm{F.2}}).
$$

Energy of a living organism of a person:

$$
\mathrm{fPrhf}(\mathrm{r}, \mathrm{u}(E_q)) = fSprt_{t_0}^{\left\{ {}^{q}\begin{pmatrix} u\ u & & u\ u \\ & PrfSpr & \\ u\ u & & u\ u \end{pmatrix}_{W_q} fSprt^{E_q}_{{}^{q}\begin{pmatrix} u\ u & & u\ u \\ & PrfSpr & \\ u\ u & & u\ u \end{pmatrix}},\ fSprt_{d_r}^{\{E^{ex} l^{d_r}\}}{}_{(self\left(E_{in} l^{d_r}\right))} \right\}} \qquad (^{***}{}_{\mathrm{F.2}}).
$$

${}^{u\ u}_{u\ u}PrfSpr^{u\ u}_{u\ u}$ - internal energy of a living organism of double energy structure, q- a gap in the energy cocoon of a living organism, r-the position of the assemblage point d_r on the energy cocoon of a living organism, W_q- energy prominences from the gap in the cocoon of a living organism, E_q-external energy entering the gap in the cocoon of a living organism, $E^{ex} l^{d_r}$ - a bundle of fibers of external energy self-capacities from outside the cocoon, collected at the point of assembly of the cocoon of a living organism, $E_{in} l^{d_r}$- a bundle of fibers of external energy self-capacities from inside the cocoon, collected at the point of assembly of the cocoon of a living organism in the same position r of the assemblage point d_r. d_r is the subject of identifying the energy fibers of the subtle energy of the Universe in position r both outside and inside the cocoon.

($**_{F.2}$), ($***_{F.2}$) can be interpreted as PrfSrt- program operators.

Appendix

Supplement for string theory: May be to try represent elementary particles in the form of continual self-elements of the type: $PrfSprt^{\uparrow}\underset{x_1}{I\downarrow_{-1}^{1}}.\underset{x_2\ \dots}{\downarrow I\uparrow_{-\infty}^{\infty}}\dots\underset{x_n}{\downarrow I\uparrow_{-1}^{1}}$ etc.

Supplement for PrfSprt-logic: We consider PrfSprt-logic: consider the functional f(Q), which gives a numerical value for the truth of the statement Q from the interval [0,1], where 0 corresponds to "no", and one corresponds to the logical value "yes". Then for joint statements A, B: f(A+B)=f(A)+f(B)-f(A*B)+fS(D), D- self-statement from A*B, fS(x)- the value of self-truth for self-statement x; for dependent statements: f(A*B)=f(A)*f(B/A)=f(B)*f(A/B), where f(B/A)- conditional truth of the statement B at statement A, f(A/B)- dependent truth of statement A at the statement B. Adding the truth values of inconsistent propositions: f(A+B)=F(A)+f(B). The formula of complete truth: f(A)=$\sum_{k=1}^{n} f\left(B_k\right) * f(A/B_k)$, B_1, B_2, .., B_n-full group of hypotheses-statements: $\sum_{k=1}^{n} f\left(B_k\right)$=1("yes"). Remark. A statement can be interpreted as an event, and its truth value as a probability.

PrfSprt- statement for set of statements A={A_1, A_2, …, A_n}: PrfSprt(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. PrfSprt(t)$\begin{matrix} f(A_1) & f(A_2) & \dots & f(A_n) \\ x_1 & x_2 & \dots & x_n \end{matrix}$. PrfSprt- truth for these statements. It is possible to consider the self-statement PrS_3A with m statements from A, at m<n, which is formed by the form (F.2.1.1), that is, only m statements from A are located in the structure $PrfSprt(t)\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. The same for self- truth $PrfS_3\{\ f(A_1), f(A_2), \dots, f(A_n)$.

One can introduce the concepts of PrfSprt-group: PrfSprt(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, A is usual group, PrfSprt(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where A, x- usual groups, self-group: $PrfSf_iA$, i=1,2,3, A is usual group.

Definition 5.1. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its elements clearly, but not necessarily in known operators; it can form (create) new special operators (in particular, special functions). In particular, PrfCrt(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$ are such structures. Similarly, for working with models, each is structured by its structure; for example, use PrfSprt-groups, PrfSprt-rings, PrfSprt-fields, PrfSprt-spaces, Prfself-groups, Prfself-rings, Prfself-fields, and Prfself-spaces. Like any task, this is also a structure of the appropriate capacity. Since the degree of freedom is double, it is clear that the form of the Prfself-equation contains the solutions or structures the inversion of the Prfself-equation concerning unknowns, i.e., the structure of the Prfself-equation is complete.

F.3 PrffSprt – elements, self-type PrffSprt- structures

Introduction.

This section generalizes the previous one to fuzzy actions with fuzzy objects, in particular, with fuzzy sets.

We consider expression

$$\begin{matrix} C_1\, C_2\, \ldots\, C_m & & A_1\; A_2\; \ldots A_n \\ \mu_{12}\mu_{22} \ldots \mu_{m2} & PrffSprt & \mu_{11}\; \mu_{21}\; \ldots \mu_{n1} \\ D_1\, D_2\, \ldots\, D_m & & B_1\; B_2\; \ldots B_n \end{matrix} \quad (*_{\text{F.3.1}})$$

where fuzzy A_1 fits into fuzzy B_1 with measure of fuzziness μ_{11}, fuzzy A_2 fits into fuzzy B_2 with measure of fuzziness μ_{21}, ..., fuzzy A_n fits into fuzzy B_n with measure of fuzziness μ_{n1}, fuzzy D_1 is forced out of fuzzy C_1 with measure of fuzziness μ_{12}, fuzzy D_2 is forced out of fuzzy C_2 with measure of fuzziness μ_{22}, ..., fuzzy D_m is forced out of fuzzy C_m with measure of fuzziness μ_{m2}, simultaneously. The result of this process will be described by the expression

$$\begin{matrix} C_1\, C_2\, \ldots\, C_m & & A_1\; A_2\; \ldots A_n \\ \mu_{12}\mu_{22} \ldots \mu_{m2} & PrffSrt & \mu_{11}\; \mu_{21}\; \ldots \mu_{n1} \\ D_1\, D_2\, \ldots\, D_m & & B_1\; B_2\; \ldots B_n \end{matrix} \quad (*_{\text{F.3.2}})$$

If A_1, B_1, A_2, B_2, ..., A_n, B_n, D_1, C_1, D_2, C_2, ..., D_m, C_m are taken as fuzzy sets, then we will call ($*_{\text{F.3.1}}$) a parallel fuzzy fuzzy dynamic fuzzy set. The need ($*_{\text{F.3.1}}$) arose to describe processes in networks. Threshold element PrffSprt - $\begin{matrix} B_1 & B_2 & \ldots & B_n & & \{ax\}_1 & \{ax\}_2 & \ldots & \{ax\}_n \\ \mu_{12} & \mu_{22} & \ldots & \mu_{n2} & PrffSprt & \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ \{qy\}_1 & \{qy\}_2 & \ldots & \{qy\}_n & & B_1 & B_2 & \ldots & B_n \end{matrix}$, B_1, B_2, ..., B_n - artificial neurons of type PrffSprt (designation - mnPrffSprt) , $\widetilde{x}=(x_1|\mu_{\widetilde{x}}(x_1), x_2|\mu_{\widetilde{x}}(x_2)|,\ldots,x_n|\mu_{\widetilde{x}}(x_n)|)$ are the fuzzy values of the initial signals, $a=(a_1,a_2,\ldots,a_n)$ are the weights of PrffSprt-synapses and $\widetilde{y}=(y_1|\mu_{\widetilde{y}}(y_1), y_2|\mu_{\widetilde{y}}(y_2)|, \ldots, y_n|\mu_{\widetilde{y}}(y_n)|)$ are the fuzzy values of the output signals $\{qy\}$ with weights $q=(q_1,q_2,\ldots,q_n)$. It can be considered a simpler version of the Parallel fuzzy dynamic fuzzy set

$$\begin{matrix} & A_1\; A_2\; \ldots A_n \\ PrffSprt & \mu_{11}\; \mu_{21}\; \ldots \mu_{n1} \\ & B_1\; B_2\; \ldots B_n \end{matrix} \quad (**_{\text{F.3.1}}),$$

where fuzzy A_1 fits into fuzzy B_1 with measure of fuzziness μ_{11}, fuzzy A_2 fits into fuzzy B_2 with measure of fuzziness μ_{21}, ..., fuzzy A_n fits into fuzzy B_n with measure of fuzziness μ_{n1}, simultaneously, the result of this process will be described by the expression

$$\begin{matrix} & A_1\; A_2\; \ldots A_n \\ PrffSrt & \mu_{11}\; \mu_{21}\; \ldots \mu_{n1} \\ & B_1\; B_2\; \ldots B_n \end{matrix} \quad (**_{\text{F.3.2}})$$

or

$C_1\, C_2\, \ldots\, C_m$
$\mu_{12}\mu_{22}\ldots\mu_{m2} PrffSprt(***_{\mathrm{F.3.1}}),$
$D_1\, D_2\, \ldots\, D_m$

where fuzzy D_1 is forced out of fuzzy C_1 with measure of fuzziness μ_{12}, fuzzy D_2 is forced out of fuzzy C_2 with measure of fuzziness $\mu_{22}, \ldots$, fuzzy D_m is forced out of fuzzy C_m with measure of fuzziness μ_{m2}, simultaneously, the result of this process will be described by the expression

$C_1\, C_2\, \ldots\, C_m$
$\mu_{12}\mu_{22}\ldots\mu_{m2} PrffSrt(***_{\mathrm{F.3.2}})$
$D_1\, D_2\, \ldots\, D_m$

We consider the measure:

$$\mu_{d_1} **\begin{pmatrix} C_1\, C_1\, \ldots\, C_1 & & A_1\ A_2\ \ldots A_n \\ \mu_{12}\mu_{22}\ldots\mu_{m2} & PrffSprt & \mu_{11}\ \mu_{21}\ \ldots\mu_{n1} \\ D_1\, D_2\, \ldots\, D_m & & C_1\ C_1\ \ldots C_1 \end{pmatrix}=$$

$\frac{\mu(A_1)*\mu(A_2)*\ldots*\ \mu(A_n)*\mu_{11}*\mu_{21}*\ldots*\mu_{n1}}{\mu(D_1)*\mu(D_2)*\ldots*\ \mu(D_m)*\mu_{12}*\mu_{22}*\ldots*\mu_{m2}}$, $d_1 =$

$$\begin{pmatrix} C_1\, C_1\, \ldots\, C_1 & & A_1\ A_2\ \ldots A_n \\ \mu_{12}\mu_{22}\ldots\mu_{m2} & PrffSprt & \mu_{11}\ \mu_{21}\ \ldots\mu_{n1} \\ D_1\, D_2\, \ldots\, D_m & & C_1\ C_1\ \ldots C_1 \end{pmatrix}$$

,where $\mu(A_{\mathrm{i}}), \mu(D_{\mathrm{j}})$,–usual fuzzy measures of fuzzy sets A_i, D_j (i = 1, 2, …, n; j = 1, 2, …, m).

Remark. One can consider some generalization for $(*_{\mathrm{F.3.1}})$, $(*_{\mathrm{F.3.2}})$:

$$\begin{matrix} q_1(C_1)q_2(C_2)\ldots q_m(C_m) & & A_1 & A_2 & \ldots & A_n \\ \mu_{12}\quad \mu_{22}\quad \ldots\quad \mu_{m2} & PrffSprt & \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ D_1\quad D_2\quad \ldots\quad D_m & & w_1(B_1) & w_2(B_2) & \ldots & w_n(B_n) \end{matrix},$$

$$\begin{matrix} q_1(C_1)q_2(C_2)\ldots q_m(C_m) & & A_1 & A_2 & \ldots & A_n \\ \mu_{12}\quad \mu_{22}\quad \ldots\quad \mu_{m2} & PrffSrt & \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ D_1\quad D_2\quad \ldots\quad D_m & & w_1(B_1) & w_2(B_2) & \ldots & w_n(B_n) \end{matrix}$$

, where fuzzy A_1 fits into fuzzy B_1 through w_1 with measure of fuzziness μ_{11}, fuzzy A_2 fits into fuzzy B_2 through w_2 with measure of fuzziness $\mu_{21}, \ldots$, fuzzy A_n fits into fuzzy B_n through w_n with measure of fuzziness μ_{n1}, fuzzy D_1 is forced out of fuzzy C_1 through q_1 with measure of fuzziness μ_{12}, fuzzy D_2 is forced out of fuzzy C_2 through q_2 with measure of fuzziness $\mu_{22}, \ldots$, fuzzy D_m is forced out of fuzzy C_m through q_m with measure of fuzziness μ_{m2}, simultaneously. A_i, B_i, D_j, C_j (i = 1, 2, …, n; j = 1, 2, …, m) can be taken as fuzzy sets.

Similarly, for (**$_{\text{F.3.1}}$): $PrffSprt \begin{array}{cccc} A_1 & A_2 & \ldots & A_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ w_1(B_1) & w_2(B_2) & \ldots & w_n(B_n) \end{array}$, for (***$_{\text{F.3.1}}$): $\begin{array}{cccc} q_1(C_1) & q_2(C_2) & \ldots & q_m(C_m) \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ D_1 & D_2 & \ldots & D_m \end{array} PrffSprt$. The result of this process will be described by the expression $\begin{array}{cccc} q_1(C_1) & q_2(C_2) & \ldots & q_m(C_m) \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ D_1 & D_2 & \ldots & D_m \end{array} PrffSrt$.

Let us introduce the concepts Cha, the fcapacity measure, and Cca, the measure of its content. Cca is the same as the number of fcapacity content items. In contrast to the classical one-attribute set theory, where only its contents are taken as a set, we consider a two-attribute set theory with a set as a fcapacity and separately with its contents. We introduce the designations: CoQ—the contents of the fcapacity Q.

fPrfSprt – elements, self-type fPrfSprt- structures.

Definition F.3.1.1. The fuzzy set of elements $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) at one point x = $(x_1, x_2, ..., x_n)$ of space X we shall call PrffSprt – element, and such a point x in space X is called parallel fuzzy fcapacity of the PrffSprt – element. We shall denote $PrffSprt \begin{array}{cccc} g_1 & g_2 & \ldots & g_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$.

Definition F.3.1.2. $PrffSprt \begin{array}{cccc} g_1 & g_2 & \ldots & g_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$ - a parallel fuzzy fuzzy dynamic fuzzy set $\widetilde{g}$ at x.

Definition F.3.1.3. An ordered fuzzy set of elements at one point in the space is called an ordered PrffSprt–element.

It's possible to $PrffSprt \begin{array}{cccc} g_1 & g_2 & \ldots & g_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$ correspond to the fuzzy set $\widetilde{g}$, and the ordered PrffSprt - element - a fuzzy vector, a fuzzy matrix, a fuzzy tensor, a fuzzy directed segment in the case when the totality of elements is understood as a fuzzy set of elements in a segment.

It's allowed to sum PrffSprt – elements: $PrffSprt \begin{array}{cccc} g_1 & g_2 & \ldots & g_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} + PrffSprt \begin{array}{cccc} b_1 & b_2 & \ldots & b_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} = PrffSprt \begin{array}{cccc} g_1 \cup b_1 & g_2 \cup b_2 & \ldots & g_n \cup b_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$.

It's allowed to multiply PrffSprt − elements: $PrffSprt\begin{matrix} g_1 & g_2 & \ldots g_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n \end{matrix} * PrffSprt\begin{matrix} b_1 & b_2 & \ldots b_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n \end{matrix} = PrffSprt \begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \ldots g_n \cap b_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.

The operator $PrffSprt \begin{matrix} g_1 \cup b_1 & g_2 \cup b_2 & \ldots g_n \cup b_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ is not equal the fuzzy set of $g_i \cup b_i$, (i = 1, 2, …, n), rather, it is Parallel fuzzy dynamic — contraction of the fuzzy set of $g_i \cup b_i$, (i = 1, 2, …, n), to the point x. Similarly, for $PrffSprtPrffSprt \begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \ldots g_n \cap b_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$. This is more suitable for using fuzzy sets for energy space, for any fuzzy objects. The operator PrffSprt is adapted for ordinary energies, using their property to overlap.

Parallel fuzzy fcapacity in itself.

Definition F.3.1.4. The ffcapacity $PrffSrt\begin{matrix} g_1 & g_2 & \ldots g_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ A_1 & A_2 & \ldots A_n \end{matrix}$ is called the parallel fuzzy fcapacity A = $(A_1,\ A_2,\ \ldots,\ A_n)$ for $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2)$, …, $g_n|\mu_{\widetilde{g}}(g_n)$).

Definition F.3.1.4.1. The parallel fuzzy fcapacity A in itself of the first type is the parallel fuzzy fcapacity fuzzy containing itself as an element. Denote $PrffS_1fA$. $PrffS_1fA = PrffSrt\begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ A_1 & A_2 & \ldots A_n \end{matrix}$.

Definition F.3.1.5. The parallel fuzzy fcapacity A in itself of the second type is the parallel fuzzy fcapacity that fuzzy contains fuzzy elements from which it can be generated. Denote Pr ffS_2fA.

An example of the parallel fcapacity in itself of the first type is a set containing itself in parallel. An example of parallel fcapacity in itself of the second type is a living organism since it contains a program: DNA and RNA.

Definition F.3.1.6. Partial parallel fuzzy fcapacity A in itself of the third type is the parallel fuzzy fcapacity A in itself, which partially contains itself or fuzzy contains fuzzy elements from which it can be fuzzy generated in part or both simultaneously. Let us denote PrffS_3fA.

Let us introduce the following notations: A*B=PrffSprt$\begin{matrix} A_1 & A_2 & \ldots & A_n \\ B_1 & B_2 & \ldots & B_n \end{matrix}$, A^2 = PrffSelf A =$PrffSrt_A^A$, A^3 = PrffSelf^{2}A, …, A^{n+1} = PrffSelf^{n}A, ….There is no commutativity here: A*B B*A. We can consider operator functions:

$e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

You can consider a more "hard" option: A*B=$PPrffSprt_B^A$, where $PPrffrSprt_B^A$ – operator, containing A in every element of B, A^2 = PPrffSelf A =$PPrffSprt_A^A$, A^3 = PPrffSelf^{2}A, …, A^{n+1} = PPrffSelf^{n}A, ….There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

All parallel capacities in parallel self-space are parallel capacities in themselves by definition. Parallel capacities in themselves can appear as PrffSprt -capacities and ordinary capacities. In these cases, the usual measures and methods of topology are used.

Connection of PrffSprt – elements with parallel fuzzy fcapacities in themselves. For example, $PrffSrt_{g\{R\}}^{\{R\}}$ is the parallel fuzzy fcapacity in itself of the second type if $g\{R\}$ is a parallel program capable of fuzzy generating $\{R\}$.
Consider a third type of parallel fuzzy fcapacity in itself. For example, based on $PrffSprt\begin{array}{cccc} g_1 & g_2 & \ldots & g_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$, where $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2)$, …, $g_n|\mu_{\widetilde{g}}(g_n)$), i.e. n - elements at one point x = $(x_1, x_2, ..., x_n)$, we can consider the fcapacity $PrffS_3f$ in itself with m elements from $\widetilde{g}$, m<n, which is formed according to the form (1.1), that is, the structure $PrffS_3f$ contains only m elements, or in forms (1.1.1) - (1.1.5), summarizing it. Fuzzy fcapacities in themselves of the third type can be formed for any other structure, not necessarily $PrffS_3f$ only by necessarily reducing the number of elements in the structure, in particular, using form (1.2). Structures more complex than$PrffS_3f$ can be introduced. For example, through a form (1.3), where A is fuzzy compressed (fuzzy fits) in C in the fuzzy compression fuzzy structure B in C (i.e., in the fuzzy structure $PrffS_3f$); or through the forms (1.3.1) - (1.4) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc.

(1.3.1) - (1.3.4) schematically interpret the fuzzy formation of fuzzy capacity in itself through a pseudo 3-connected form with a 2-connected form. The ideology of ffSprt and ffS$_3$f can be used for programming.

Remark F.3.1.1. Fuzzy self, in particular, according to a fuzzy form- fuzzy analogue of the form of type (1.1): (2.1*).

By analogy the same for the fuzzy form of type (1.1.1) – (1.4).

Math PrffSelf.

Let's consider PrffSprt arithmetic first:

1. Simultaneous parallel addition of sets elements $\widetilde{g_i}=(g_{i_1}|\mu_{\widetilde{g_i}}(g_{i_1}), g_{i_2}|\mu_{\widetilde{g_i}}(g_{i_2}), \ldots, g_{i_{m_j}}|\mu_{\widetilde{g_i}}(g_{i_{m_j}}))$, i = 1,2,. . . ,n, j =1,2,. . . ,k is carried out using $PrffSprt \begin{matrix} \{g_1\}+ & \{g_2\}+ & \ldots & \{g_n\}+ \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.
2. Similarly, for simultaneous parallel multiplication: $PrffSprt \begin{matrix} \{g_1\}* & \{g_2\}* & \ldots & \{g_n\}* \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ w_1 & w_2 & \ldots & w_n \end{matrix}$ gives the fuzzy set B_l, l =1,2,. . . ,n, with elements $b_{li_1i_2\ldots i_{m_j}} = Sprt_{x_l}^{\left\{ g_{l_{1_{i_1}}}*, \quad g_{l_{2_{i_2}}}*, \ldots, \quad g_{l_{m_{j_{i_{m_j}}}}} \right\}})_{R_l}$ $for\ any\ \left\{l_{i_1}, l_{i_2}, \ldots, l_{i_{m_j}}\right\}\ without\ repetitions, x_l = Sprt_w{}^{\{K_l\}}$,K_l-set of any $\left\{k_{l_{i_1}}*, k_{l_{i_2}}*, \ldots, k_{l_{i_{m_j}}}*\right\}$ without repeating them, l = 1,2,. . . ,n, $k_{l_{i_j}}$-any digit, i =1,2,. . . , m_j, R_l= $Sprt_w{}^{\left\{l_{i_1}+, l_{i_2}+, \ldots, l_{i_{m_j}}\right\}}$, R_l is the index of the lower discharge (we choose an index on the scale of discharges): see Table I.1

Then $PrffSprt \begin{matrix} \{B_1\}+ & \{B_2\}+ & \ldots & \{B_n\}+ \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. The most straightforward functional scheme of the assumed arithmetic-logical device for PrffSprt-multiplication:

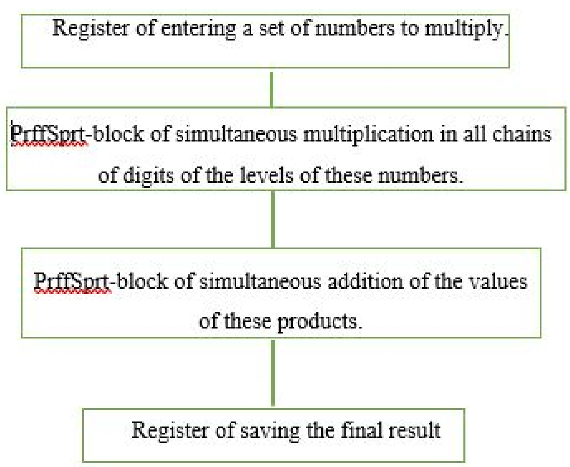


Fig. 0.1 The straightforward functional scheme of the assumed arithmetic-logical device for PrffSprt-multiplication.

Remark. The algorithm for simultaneously adding a set of numbers can also be implemented as the simultaneous addition of elements of a simultaneously formed composite matrix: a triangular matrix in which the elements of the first row are represented by multiplying the first number from the set by the rest: each multiplication is represented by a matrix of multiplying the digits of 2 numbers, taking into account the bit depth, the elements of the second rows are represented by multiplying the second number from the set by the ones following it, etc.

3. Similarly for simultaneous execution of various operations: $PrffSprt\begin{matrix}\{g_1Q_1\} & \{g_2Q_2\} & \ldots\{g_nQ_n\} \\ \mu_{11} & \mu_{21} & \ldots\ \ \mu_{n1} \\ w_1 & w_2 & \ldots\ \ w_n\end{matrix}$, where $\widetilde{Q}$=($Q_1|\mu_{\widetilde{Q}}(Q_1)$, $Q_2|\mu_{\widetilde{Q}}(Q_2)$, …, $Q_n|\mu_{\widetilde{Q}}(Q_n)$). Q_i -an operation, i = 1, …, n.

4. Similarly, for the simultaneous execution of various operators: $PrffSprt\begin{matrix}\{F_1g_1\} & \{F_2g_2\} & \ldots\{F_ng_n\} \\ \mu_{11} & \mu_{21} & \ldots\ \ \mu_{n1} \\ w_1 & w_2 & \ldots\ \ w_n\end{matrix}$, where $\widetilde{F}$=($F_1|\mu_{\widetilde{F}}(F_1)$, $F_2|\mu_{\widetilde{F}}(F_2)$, …, $F_n|\mu_{\widetilde{F}}(F_n)$). F_i is an operator, i = 1, …, n.

5. The arithmetic itself for capacities in themselves will be similar: addition - $PrffS_1f\,\{g+\}$, (or $PrffS_3f\,\{g+\}$) for the third type), multiplication $PrffS_1f\,\{g*\}$, ($PrffS_3f\,\{g*\}$).

6. Similarly with different operations: $PrffS_1f\,\{gq\}$, ($PrffS_3f\,\{gq\}$), and with different operators: $PrffS_1f\,\{Fg\}$, ($PrffS_3f\,\{Fg\}$).

7. $PrffSrt\begin{matrix}A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots\mu_{n1} \\ B_1 & B_2 & \ldots B_n\end{matrix}\ - the\ result\ of\ the\ fuzzy\ containment\ operator.$

   For fuzzy sets A_i, B_i, (i = 1, 2, …, n), we have

$$PrffSrt\begin{matrix}A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots\mu_{n1} \\ B_1 & B_2 & \ldots B_n\end{matrix} = \left\{\sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i\ ,\sum_{i=1}^{n} D_i\right\} = \left\{\begin{matrix}\sum_{i=1}^{n} D_i \\ \sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i\end{matrix}\right\},$$

where D_i is self-(fuzzy set) for $A_i \cap B_i$ (i = 1, 2, …, n). There is the same for structures if they are considered as fuzzy sets. Similarly, for fuzzy sets C_i, D_i: $\begin{matrix}C_1 & C_2 & \ldots & C_m \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ D_1 & D_2 & \ldots & D_m\end{matrix}PrffSrt$

$$= \left\{\begin{matrix}\sum_{i=1}^{m} Q_i + \begin{matrix}\{\ \} \\ D_1 - D_1 \cap C_1\end{matrix}\ \begin{matrix}\{\ \}\ldots \\ D_2 - D_2 \cap C_2\ \ldots\end{matrix}\ \begin{matrix}\{\ \} \\ D_m - D_m \cap C_m\end{matrix} PrfSrt \\ \sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i)\end{matrix}\right\},$$

where Q_i is oself-(fuzzy set) for $(D_i \cap C_i)$ (i = 1, 2, …, m) [14].

8. PrffSprt-derivative of f($x_1, x_2, \ldots, x_n$) $= \begin{pmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \ldots \\ f_k(x_1, x_2, \ldots, x_n) \end{pmatrix}$ is

   $$PrffSprt \begin{matrix} \frac{\partial}{\partial x_{1_i}} & \frac{\partial}{\partial x_{2_i}} & \ldots & \frac{\partial}{\partial x_{k_i}} \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ f_1(x_1, x_2, \ldots, x_n) & f_2(x_1, x_2, \ldots, x_n) & \ldots & f_k(x_1, x_2, \ldots, x_n) \end{matrix},$$

   where x=($x_{1_i}, x_{2_i}, \ldots x_{k_i}$)- any fuzzy set from $\widetilde{x}$=($\mathrm{x}_1|\mu_{\widetilde{x}}(\mathrm{x}_1)$, $\mathrm{x}_2|\mu_{\widetilde{x}}(\mathrm{x}_2)$, …, $\mathrm{x}_\mathrm{n}|\mu_{\widetilde{x}}(\mathrm{x}_\mathrm{n})$). The same is done for PrffSprt- $\frac{\partial^k f(x)}{\partial x_{1_i} \partial x_{2_i} \ldots \partial x_{k_i}}$. PrffSprt-integral off ($x_1, x_2, \ldots, x_n$) is

   $$PrffSprt \begin{matrix} \int ()dx_{1_i} & \int ()dx_{2_i} & \ldots & \int ()dx_{k_i} \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ f_1(x_1, x_2, \ldots, x_n) & f_2(x_1, x_2, \ldots, x_n) & \ldots & f_k(x_1, x_2, \ldots, x_n) \end{matrix},$$

   where ($x_{1_i}, x_{2_i}, \ldots x_{k_i}$)- any fuzzy set from $\widetilde{x}$=($\mathrm{x}_1|\mu_{\widetilde{x}}(\mathrm{x}_1)$, $\mathrm{x}_2|\mu_{\widetilde{x}}(\mathrm{x}_2)$, …, $\mathrm{x}_\mathrm{n}|\mu_{\widetilde{x}}(\mathrm{x}_\mathrm{n})$). The same is done for PrffSprt- $\int \ldots \int f(x) dx_{1_i} dx_{2_i} \ldots dx_{k_i}$-k-multiple integral. PrffSprt-lim off $\widetilde{x}$=($\mathrm{x}_1|\mu_{\widetilde{x}}(\mathrm{x}_1)$, $\mathrm{x}_2|\mu_{\widetilde{x}}(\mathrm{x}_2)$, …, $\mathrm{x}_\mathrm{n}|\mu_{\widetilde{x}}(\mathrm{x}_\mathrm{n})$) is

   PrffSprt $\begin{matrix} \lim\limits_{x_{1_i} \to a_{1_i}} & \lim\limits_{x_{2_i} \to a_{2_i}} & \ldots & \lim\limits_{x_{k_i} \to a_{k_i}} \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ f_1(x_1, x_2, \ldots, x_n) & f_2(x_1, x_2, \ldots, x_n) & \ldots & f_k(x_1, x_2, \ldots, x_n) \end{matrix}$. The same is done for PrffSprt- $\begin{pmatrix} \lim f(x_1, , x_2, \ldots, x_n) \\ x_{1_i} \to a_{1_i} \\ \ldots \\ x_{k_i} \to a_{k_i} \end{pmatrix}$.

   $$PrffS_3 f\{\lim_{x \to a}\} = PrffSprt \begin{matrix} \lim\limits_{x_{1_i} \to a_{1_i}} & \lim\limits_{x_{2_i} \to a_{2_i}} & \ldots & \lim\limits_{x_{k_i} \to a_{k_i}} \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ \lim\limits_{x_{1_i} \to a_{1_i}} & \lim\limits_{x_{2_i} \to a_{2_i}} & \ldots & \lim\limits_{x_{k_i} \to a_{k_i}} \end{matrix}.$$

9. In the case of PrffSelf-derivatives, inclusions of multiple derivatives are obtained. The same is true for PrffSelf-integrals: we get inclusions of multiple integrals.

10. Let's denote PrffSelf-(PrffSelf-Q) through PrffSelf^2-Q , ffS(n,Q)= PrffSelf-(PrffSelf-(…(PrffSelf-Q))) = PrffSelf$^\mathrm{n}$-Q for n-multiple PrffSelf.

Operator Prfitself.

Definition F.3.1.7. An operator that transforms $PrffSprt \begin{matrix} g_1 & g_2 & \ldots g_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n \end{matrix}$ into any Pr$ffS_i f \left\{\widetilde{b}\right\}$, $i = 2, 3$; where $\left\{\widetilde{b}\right\} \subset \{\widetilde{g}\}$ is the operator Prffitself.

Example. The operator fuzzy contains the fuzzy set in Prffitself.

Lim-Prffitself.

1. Lim PrffSprt

   For example, the double PrffSprt-limit: $\lim\limits_{\substack{x \to g_1 \\ y \to g_2}} G(x,y)$ corresponds to

   $PrffSprt \begin{matrix} G(x,y) & G(x,y) \\ x \to g_1 & y \to g_2 \end{matrix}$, where $G(x,y)$ is fuzzy.

   Similarly, for lim PrffSprt with n variables.

In the case of lim-Prffitself, for example, for m variables, it suffices to use the form (1.1) of lim PrffSprt for n variables (n>m) and for more complex operators in forms (1.1.1) – (1.4) , (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). The same is true for integrals of variables m (for example, the double integral over a rectangular region is through the double limit).

The sequence of actions can be "collapsed" into an ordered PrffSprt element, and then translate it, for example, into $PrffS_3f-$ the parallel fuzzy fcapacity in itself.

About PrffSprt and PrffS$_3$f programming.

The ideology of PrffSprt and PrfS$_3$f can be used for programming. Here are some of the PrffSprt programming operators.

1. Simultaneous fuzzy assignment of the expression$s$ $\widetilde{p}$=(p$_1|\mu_{\widetilde{p}}$(p$_1$), p$_2|\mu_{\widetilde{p}}$(p$_2$), …, p$_n|\mu_{\widetilde{p}}$(p$_n$)) to the variables $\widetilde{x}$=(x$_1|\mu_{\widetilde{x}}$(x$_1$), x$_2|\mu_{\widetilde{x}}$(x$_2$), …, x$_n|\mu_{\widetilde{x}}$(x$_n$)). This is implemented via $PrffSprt \begin{matrix} x_1 := & x_2 := & \ldots x_n := \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ p_1 & p_2 & \ldots & p_n \end{matrix}$.
2. Simultaneous fuzzy checking the set of conditions $\widetilde{g}$=(g$_1|\mu_{\widetilde{g}}$(g$_1$), g$_2|\mu_{\widetilde{g}}$(g$_2$), …, g$_n|\mu_{\widetilde{g}}$(g$_n$)) for the set of expressions $\widetilde{B}$=(B$_1|\mu_{\widetilde{B}}$(B$_1$), B$_2|\mu_{\widetilde{B}}$(B$_2$), …, B$_n|\mu_{\widetilde{B}}$(B$_n$)). Implemented via $PrffSprt \begin{matrix} IF\{B_1 g_1\}\ then & IF\{B_2 g_2\}\ then & \ldots & IF\{B_n g_n\}\ then \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ w_1 & w_2 & \ldots & w_n \end{matrix}$, where w_i (i = 1, …, n) can be anything.
3. Similarly for loop operators and others.

$PrffS_3f-$ software operators will differ only in that the aggregates $\{\widetilde{g}\},\{\widetilde{p}\},\left\{\widetilde{B}\right\},\{\widetilde{x}\}$ will be formed from the corresponding PrffSprt program operators in form (1.1) and for more complex operators in the forms (1.1.1) –(1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

The OS (operating system), the computer's principles, and the modes of operation for this programming are interestinF. But this is already the material for the following articles.

Using elements of the mathematics of PrffSrt we introduce the concept of PrffSrt – the change in physical quantity B: $PrffSprt \begin{matrix} \Delta_1 B & \Delta_2 B & \ldots \Delta_n B \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n \end{matrix}$. Then the mean PrffSrt - velocity will be $v_{cpPrffSrt}(t, t) = PrffSprt \begin{matrix} \frac{\Delta_1 B}{\Delta t} & \frac{\Delta_2 B}{\Delta t} & \ldots \frac{\Delta_n B}{\Delta t} \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n \end{matrix}$ and PrffSrt-velocity at time t:$v_{Prffsrt} = \lim_{\Delta t \to 0} v_{Prffst}(t, \Delta t)$. PrffSrt – acceleration $a_{Prffsrt} = \frac{dv_{Prffsrt}}{dt}$. The nuclei of atoms can be considered as PrffSprt elements.

Remark F.3.1.2. PrffSprt – elements are all ordinary, but with "target weights," they become peculiar. Here you need the necessary energy to carry them out. As a rule, this energy is at the level of PrffSelf. This is natural since it's much easier to manage elements of the k level via the elements of a more structured k +1 level. Let us consider the concepts of capacities of physical objects in themselves. The question arises about the ffself-energy of the object. In particular, $PrffSrt_B^B$ will mean PrffS_1f B. For example, $\text{PrS}prt_{DNA}^{DNA}$ allows you to reach the level of DNA self-energy, $PrffSprt_Q^Q$ allows you to reach the level of ffself-energy Q. The law of self-energy conservation operates already at the level of self-energy. Also, in addition to ffcapacities in themselves, you can consider the types of fuzzy containment of oneself in oneself: the first type of the fuzzy containment of oneself in oneself: the second type of the fuzzy containment of oneself in oneself: potentially, for example, in the form of fuzzy programming oneself, the third type is partial fuzzy containment of oneself in themselves—for example, PrffSelf-operator, PrffSelf-action, whirlwind. A fuzzy container fuzzy containing itself can be formed by ffself-containment, i.e., ffcontainment in oneself. Let us clarify the concept of the term fuzzy fcapacity in itself: it is a fuzzy fcapacity fuzzy containing itself potentially. Consider PrffSelf-Q, where Q can be anything fuzzy, including Q=PrffSelf; in particular, it can be any fuzzy action. Therefore, PrffSelf-Q is when fuzzy Q is made by Prfitself; it makes itself. There is a partial PrffSelf-Q for any fuzzy Q with partial PrffSelf-fulfillment. Let's consider several examples for fuzzy capacities in themselves: ordinary lightning, electric arc discharge, and ball lightninF.

PrffSprt is also great for working with structures, for example:

1) $PrffSprt \begin{matrix} fstrA_1 & fstrA_2 & \ldots fstrA_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ B_1 & B_2 & \ldots B_n \end{matrix}$ - the fuzzy structure A_i that fits into fuzzy B_i with measure of fuzziness μ_{i1}, where fuzzy B_i (i = 1, …, n) can be any fuzzy fcapacity , another fuzzy structure etc.

2) PrffSprt $\begin{matrix} fstr_{Q_1} & fstr_{Q_2} & \ldots fstr_{Q_n} \\ \mu_{11} & \mu_{21} & \ldots \; \mu_{n1} \\ B_1 & B_2 & \ldots \; B_n \end{matrix}$ is embedding fuzzy structure from Q_i into B_i. Similarly for displacement: 1) $\begin{matrix} C_1 & C_2 & \ldots & C_m \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ fstrD_1 & fstrD_2 & \ldots & fstrD_m \end{matrix}$ $PrffSrt$ - displacement of fuzzy structure $fstrD_j$ from C_j with measure of fuzziness μ_{j1}, (j = 1, …, m), 2) $\begin{matrix} C_1 & C_2 & \ldots & C_m \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ fstr_{Q_1} & fstr_{Q_2} & \ldots & fstr_{Q_m} \end{matrix}$ PrffSprt -displacement of the fujzy structure Q_i from C_i, (i = 1, …, m). To work with structures, you can introduce a special operator PrffCprt: PrffCprt$\begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ B_1 & B_2 & \ldots B_n \end{matrix}$ fuzzy structures fuzzy B_i with the fuzzy structure A_i, (i = 1, …, n), PrffCprt $\begin{matrix} fstr_{Q_1} & fstr_{Q_2} & \ldots fstr_{Q_n} \\ \mu_{11} & \mu_{21} & \ldots \; \mu_{n1} \\ B_1 & B_2 & \ldots \; B_n \end{matrix}$ fuzzy structures fuzzy B_i with the fuzzy structure from fuzzy Q_i, (i = 1, …, n), $\begin{matrix} C_1 & C_2 & \ldots & C_m \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ D_1 & D_2 & \ldots & D_m \end{matrix}$PrffCprt destructors fuzzy C_i by the fuzzy structure of D_i, $\begin{matrix} C_1 & C_2 & \ldots & C_m \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ fstr_{Q_1} & fstr_{Q_2} & \ldots & fstr_{Q_m} \end{matrix}$ PrffCprt destructors fuzzy C_i from the fuzzy structure that structures Q_i, (i = 1, …, m).

Definition F.3.1.8. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its elements explicitly, but not necessarily in known operators; it can form (create) new special operators (in particular, special functions).

In particular, PrfCprt$\begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ A_1 & A_2 & \ldots A_n \end{matrix}$, PrfCrt$\begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ A_1 & A_2 & \ldots A_n \end{matrix}$ are such structures.

Similarly, for working with models, each is structured by its structure; for example, use PrffSprt-groups, PrffSprt-rings, PrffSprt-fields, PrffSprt-spaces, Prffself-groups, PrffSelf-rings, PrffSelf-fields, and PrffSelf-spaces. Like any task, this is also a structure of the appropriate fcapacity .

PrffSelf-H (PrffSelf-hydrogen), like other PrffSelf-particles, does not exist in the ordinary, but all PrffSelf-molecules, PrffSelf-atoms, and PrffSelf-particles are elements of the energy space.

Remark F.3.1.3. The concept of elements of physics PrffSprt is introduced for energy space. The ideology of PrffSprt elements allows us to go to the border of the world familiar to us, which allows us to act more effectively.

Fuzzy dynamic PrffSprt – elements.

We considered stationary PrffSprt – elements earlier. Here we consider fuzzy dynamic PrffSprt – elements.

Definition F.3.2.1. The process of fuzzy fitting a fuzzy set of elements $\widetilde{g(t)}$=(g_1(t)|$\mu_{\widetilde{g(t)}}$(g_1(t)), g_2(t)|$\mu_{\widetilde{g(t)}}$(g_2(t)), ..., g_n(t)|$\mu_{\widetilde{g(t)}}$(g_n(t))) into one point x= $(x_1, x_2, ..., x_n)$ of the space X at time t will be called a dynamic PrffSprt – element. We will denote PrffSprt(t) $\begin{array}{cccc} g_1(t) & g_2(t) & \ldots & g_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$.

Definition F.3.2.2. Fuzzy fitting an ordered fuzzy set of elements into one point in space is called a dynamic ordered PrffSprt–element.

It is allowed to sum dynamic PrffSprt – elements: $PrffSprt(t) \begin{array}{cccc} g_1(t) & g_2(t) & \ldots & g_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} +$

$$PrffSprt(t) \begin{array}{cccc} b_1(t) & b_2(t) & \ldots & b_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} = \text{PrffSprt}(t) \begin{array}{cccc} g_1(t) \cup b_1(t) & g_2(t) \cup b_n(t) & \ldots & g_n(t) \cup b_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}.$$

It's allowed to multiply PrffSprt – elements: $PrffSprt(t) \begin{array}{cccc} g_1(t) & g_2(t) & \ldots & g_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} *$

$$PrffSprt(t) \begin{array}{cccc} b_1(t) & b_2(t) & \ldots & b_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} = \text{PrffSprt}(t) \begin{array}{cccc} g_1(t) \cap b_1(t) & g_2(t) \cap b_n(t) & \ldots & g_n(t) \cap b_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$$

Parallel fuzzy dynamic fuzzy containment of oneself.

Definition F.3.2.3. Parallel fuzzy dynamic fSprt- fcapacity $PrffSprt(t) \begin{array}{cccc} R_1(t) & R_2(t) & \ldots & R_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ Q_1(t) & Q_2(t) & \ldots & Q_n(t) \end{array}$ is the process of fuzzy embedding fuzzy R_i(t) into fuzzy Q_i(t), (i = 1, ..., n), simultaneously.

Definition F.3.2.4. Parallel dynamic fuzzy fcapacity $\widetilde{Q(t)}$=(Q_1(t)|$\mu_{\widetilde{Q(t)}}$(Q_1(t)), Q_2(t)|$\mu_{\widetilde{Q(t)}}$(Q_2(t)), ..., Q_n(t)|$\mu_{\widetilde{Q(t)}}$(Q_n(t))) fuzzy containing itself as an element of the first type is the process of parallel fuzzy containing fuzzy $\widetilde{Q(t)}$ in $\widetilde{Q(t)}$ $PrffSprt(t) \begin{array}{cccc} Q_1(t) & Q_2(t) & \ldots & Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ Q_1(t) & Q_2(t) & \ldots & Q_n(t) \end{array}$. Denote $PrffS_1f(t)\widetilde{Q(t)}$.

Definition F.3.2.5. Parallel fuzzy dynamic fuzzy fcapacity C(t) in itself of the second type is the process of parallel fuzzy containing fuzzy elements from which it can be parallel fuzzy generated. Let's denote $PrffS_2f(t)C(t)$.

Definition F.3.2.6. Parallel dynamic partial fuzzy fcapacity B(t) in itself of the third type is a process of partial parallel fuzzy containment of fuzzy B(t) in itself or parallel fuzzy embedding fuzzy elements from which it can be parallel fuzzy generated partially or both at the same time. Denote Pr $ffS_3f(t)B(t)$.

All parallel dynamic fuzzy fcapacities in a parallel fuzzy dynamic fuzzy self-space are, by definition, parallel dynamic fuzzy fcapacities in themselves. Parallel dynamic fuzzy fcapacity itself can manifest itself as parallel fuzzy dynamic fSprt- capacity and ordinary parallel fuzzy dynamic fuzzy fcapacity . In these cases, the usual fuzzy measures and methods of topology are used.

Connection of fuzzy dynamic PrffSprt − elements with parallel fuzzy dynamic fuzzy containment of oneself.

Consider third type of parallel fuzzy dynamic partial fuzzy containment of oneself. For example, based on $PrffSprt(t) \begin{matrix} Q_1(t) & Q_2(t) & \ldots Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$, where $\widetilde{Q(t)}$=(Q$_1$(t)|$\mu_{\widetilde{Q(t)}}$(Q$_1$(t)), Q$_2$(t)|$\mu_{\widetilde{Q}(t)}$(Q$_2$(t)), …, Q$_n$(t)|$\mu_{\widetilde{Q(t)}}$(Q$_n$(t))), i.e. n – elements at one point x = $(x_1, x_2, ..., x_n)$, we can consider the parallel fuzzy dynamic fuzzy capacity in itself $PrffS_3f(t)$ with m elements from $\widetilde{Q(t)}$, m<n, which is process formed according to the form (1.1), that is, only m elements from $\widetilde{Q(t)}$ are in the structure $PrffSprt(t) \begin{matrix} Q_1(t) & Q_2(t) & \ldots Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.

Parallel fuzzy dynamic fuzzy containment of oneself of the third type can be formed for any other structure, not necessarily PrffSprt, only through the obligatory reduction in the number of elements in the structure. In particular, using the forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

It is possible to introduce structures more complex than PrffS$_3$f(t).

Parallel fuzzy dynamic fuzzy math itself.

1. The process of simultaneous parallel addition of fuzzy sets elements $\left\{\widetilde{g_i(t)}\right\} = \left(g_{i_1}(t)|\mu_{\widetilde{g_i(t)}}(g_{i_1}(t)),\ g_{i_2}(t)|\mu_{\widetilde{g_i(t)}}(g_{i_2}(t)),\ \ldots,\ g_{i_{m_j}}(t)|\mu_{\widetilde{g_i(t)}}(\ g_{i_{m_j}}(t)) \right)$, i = 1,2,…,n, j =1,2,…,k are realized by $PrffSprt(t) \begin{matrix} \widetilde{\{g_1(t)\}}+ & \widetilde{\{g_2(t)\}}+ & \ldots & \widetilde{\{g_n(t)\}}+ \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$
.

2. By analogy, for simultaneous multiplication:

$$PrffSprt(t)\begin{matrix} \widetilde{\{g_1(t)\}}* & \widetilde{\{g_2(t)\}}* & \ldots & \widetilde{\{g_n(t)\}}* \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}.$$

2. Similarly for simultaneous execution of various operations: $PrffSprt(t)\begin{matrix} g_1(t)Q_1(t) & g_2(t)Q_2(t) & \ldots g_n(t)Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$, where $\widetilde{Q(t)}$=(Q_1(t)|$\mu_{\widetilde{Q(t)}}$(Q_1(t)), Q_2(t)|$\mu_{\widetilde{Q(t)}}$(Q_2(t)), …, Q_n(t)|$\mu_{\widetilde{Q(t)}}$(Q_n(t))), Q_i(t)-an operation, i = 1, …, n, $\widetilde{g(t)}$=(g_1(t)|$\mu_{\widetilde{g(t)}}$(g_1(t)), g_2(t)|$\mu_{\widetilde{g(t)}}$(g_2(t)), …, g_n(t)|$\mu_{\widetilde{g(t)}}$(g_n(t))).

3. Similarly, for the simultaneous execution of various operators: $PrffSprt(t)\begin{matrix} F_1(t)g_1(t) & F_2(t)g_2(t) & \ldots F_n(t)g_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$, where $\widetilde{F(t)}$=(F_1(t)|$\mu_{\widetilde{F(t)}}$(F_1(t)), F_2(t)|$\mu_{\widetilde{F(t)}}$(F_2(t)), …, F_n(t)|$\mu_{\widetilde{F(t)}}$(F_n(t))), F_i(t) is an operator, i = 1, …, n.

4. Parallel fuzzy dynamic arithmetic itself for fuzzy containments of oneself will be similar: Parallel fuzzy dynamic addition - $PrffS_1f(t)\left\{\widetilde{g(t)}+\right\}$, (or $PrffS_3f(t)\left\{\widetilde{g(t)}+\right\}$ for the third type), Parallel fuzzy dynamic multiplication $PrffS_1f(t)\left\{\widetilde{g(t)}*\right\}$, ($PrffS_3f(t)\left\{\widetilde{g(t)}*\right\}$).

5. Similarly with different operations: $PrffS_1f(t)\left\{\widetilde{g(t)}\widetilde{Q(t)}\right\}$, ($PrffS_3f(t)\left\{\widetilde{g(t)}\widetilde{Q(t)}\right\}$) and with different operators: $PrffS_1f(t)\left\{\widetilde{F(t)}\widetilde{g(t)}\right\}$, ($PrffS_3f(t)\left\{\widetilde{F(t)}\widetilde{g(t)}\right\}$).

6. $PrffSprt(t)\begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t) \end{matrix}$ *gives* the result $PrffSrt(t)\begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t) \end{matrix} = \left\{\sum_{i=1}^{n} A_i(t) \cup B_i(t) - A_i(t) \cap B_i(t)\ , \sum_{i=1}^{n} D_i(t)\right\}$, *for fuzzy sets* $A_i(t),\ B_i(t)$, where D_i(t) is fuzzy self-set for $A_i(t) \cap B_i(t)$, (i = 1, 2, …, n). The same is true for structures if they are treated as fuzzy sets,

7. $\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ D_1(t) & D_2(t) & \ldots D_m(t) \end{matrix}$ PrffSrt(t) =

$$\left\{ \begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\ \} & \{\ \}\ \ldots & \{\ \} \\ D_1(t) - D_1(t) \cap C_1(t) & D_2(t) - D_2(t) \cap C_2(t)\ \ldots & D_2(t) - D_2(t) \cap C_2(t) \end{matrix} PrfSrt \\ \sum_{i=1}^{m} \big(\ C_i(t) - D_i(t) \cap C_i(t)\big) - (D_i(t) - D_i(t) \cap C_i(t)) \end{matrix} \right\}$$

for fuzzy sets $C_i(t),\ D_i(t)$, where $Q_i(t)$ is fuzzy oself-set for $\big(D_i(t) \cap C_i(t)\big)$(i = 1, 2, …, m) [14].

8. Similarly, for fuzzy dynamic PrffSprt-derivatives, fuzzy dynamic PrffSprt-integrals, fuzzy dynamic PrffSprt-lim, parallel fuzzy dynamic fuzzy self-derivatives, parallel fuzzy dynamic fuzzy self-integrals

9. Denote parallel fuzzy dynamic fuzzy self-(parallel fuzzy dynamic fuzzy self-Q(t)) through parallel fuzzy dynamic fuzzy self^2-Q(t) , pffS(t)(n,Q(t))= parallel fuzzy dynamic fuzzy self-(parallel fuzzy dynamic fuzzy self-(. . . ((parallel fuzzy dynamic fuzzy self)-Q(t)))) = (parallel fuzzy dynamic fuzzy self^n)-Q(t) for n-multiple parallel fuzzy dynamic fuzzy self.

Remark F.3.2.1. Then the notation

$$\begin{array}{cccc} C_1(t) & C_2(t) & \ldots & C_m(t) \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{array} PrffSprt(t) \begin{array}{cccc} A_1(t) & A_2(t) & \ldots & A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{array},$$

where fuzzy A_1(t) fits into fuzzy B_1(t) with measure of fuzziness μ_{11}, fuzzy A_2(t) fits into fuzzy B_2(t) with measure of fuzziness μ_{21}, . . . , fuzzy A_n(t) fits into fuzzy B_n(t) with measure of fuzziness μ_{n1}, fuzzy D_1(t) is forced out of fuzzy C_1(t) with measure of fuzziness μ_{12}, fuzzy D_2(t) is forced out of fuzzy C_2(t) with measure of fuzziness μ_{22}, . . . , fuzzy D_m(t) is forced out of fuzzy C_m(t) with measure of fuzziness μ_{m2} simultaneously. It is fuzzy dynamic PrffSprt-containment of fuzzy A_i(t) in fuzzy B_i(t) and fuzzy dynamic PrffSprt-displacement of fuzzy D_j(t) from fuzzy C_j(t) simultaneously, (i = 1, 2, . . . , n, j = 1, 2, . . . , m). The result of this process will be described by the expression

$$\begin{array}{cccc} C_1(t) & C_2(t) & \ldots & C_m(t) \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ D_1(t) & D_2(t) & \ldots & D_m(t) \end{array} PrffSrt(t) \begin{array}{cccc} A_1(t) & A_2(t) & \ldots & A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{array}$$

$PrffSrt(t) \begin{array}{cccc} B_1(t) & B_2(t) & \ldots & B_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{array}$ will mean PrffS$_1$f(t)B(t).

$\begin{array}{cccc} C_1(t) & C_2(t) & \ldots & C_m(t) \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ C_1(t) & C_2(t) & \ldots & C_m(t) \end{array} PrffSprt(t)$ denotes the parallel fuzzy dynamic expelling fuzzy $\widetilde{C(t)}$=(C_1(t)|$\mu_{\widetilde{C(t)}}$(C_1(t)), C_2(t)|$\mu_{\widetilde{C(t)}}$(C_2(t)), . . . , C_n(t)|$\mu_{\widetilde{C(t)}}$(C_n(t))) oneself out of oneself,

$\begin{array}{cccc} A_1(t) & A_2(t) & \ldots & A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ A_1(t) & A_2(t) & \ldots & A_n(t) \end{array} PrffSprt(t) \begin{array}{cccc} A_1(t) & A_2(t) & \ldots & A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ A_1(t) & A_2(t) & \ldots & A_n(t) \end{array}$ —simultaneous parallel fuzzy dynamic containment fuzzy $\widetilde{A(t)}$=(A_1(t)|$\mu_{\widetilde{A(t)}}$(A_1(t)), A_2(t)|$\mu_{\widetilde{A(t)}}$(A_2(t)), . . . , A_n(t)|$\mu_{\widetilde{A(t)}}$(A_n(t))) of oneself in oneself and parallel fuzzy dynamic expelling fuzzy $\widetilde{A(t)}$ oneself out of oneself.

$\begin{array}{cccc} A_1(t) & A_2(t) & \ldots & A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ B_1(t) & B_2(t) & \ldots & B_n(t) \end{array} PrffSprt(t)$ will be called parallel fuzzy dynamic anti-fcapacity from oneself. For example, "white hole" in physics is such simple anti- fcapacity.

We may consider the following axiom: any fcapacity is the fcapacity of oneself. This is for each energy fcapacity .

About fuzzy dynamic PrffSprt and PrffS$_3$f(t) programming.

The ideology of fuzzy dynamic PrffSprt and PrffS$_3$f(t) can be used for programming:

1. The process of simultaneous assignment of the expressions $\widetilde{p(t)}$=(p$_1$(t)|$\mu_{\widetilde{p(t)}}$(p$_1$(t)), p$_2$(t)|$v_{\widetilde{p(t)}}$(p$_2$(t)), ..., p$_n$(t)|$\mu_{\widetilde{p(t)}}$(p$_n$(t))) to the variables $\widetilde{x(t)}$=(x$_1$(t)|$\mu_{\widetilde{x(t)}}$(x$_1$(t)), x$_2$(t)|$\mu_{\widetilde{x(t)}}$(x$_2$(t)), ..., x$_n$(t)|$\mu_{\widetilde{x(t)}}$(x$_n$(t))) is implemented through

$$PrffSprt(t)\begin{array}{cccc} x_1(t) := & x_2(t) := & \dots & x_n(t) := \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ p_1(t) & p_2(t) & \dots & p_n(t) \end{array}.$$

2. The process of simultaneous check the set of conditions $\widetilde{g(t)}$=(g$_1$(t)|$\mu_{\widetilde{g(t)}}$(g$_1$(t)), g$_2$(t)|$\mu_{\widetilde{g(t)}}$(g$_2$(t)), ..., g$_n$(t)|$\mu_{\widetilde{g(t)}}$(g$_n$(t))) for a set of expressions $\widetilde{B(t)}$=(B$_1$(t)|$\mu_{\widetilde{B(t)}}$(B$_1$(t)), B$_2$(t)|$\mu_{\widetilde{B(t)}}$(B$_2$(t)), ..., B$_n$(t)|$\mu_{\widetilde{B(t)}}$(B$_n$(t))) is implemented through

$$PrffSprt(t)\begin{array}{cccc} IF\left\{B_1(t)g_1(t)\right\}\ then & IF\left\{B_2(t)g_2(t)\right\}\ then & \dots & IF\left\{B_n(t)g_n(t)\right\}\ then \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ w_1(t) & w_2(t) & \dots & w_n(t) \end{array}$$

here w(t) = $(w_1(t), w_2(t), ..., w_n(t))$. can be any.

3.Similarly for loop operators and others.

$PrffS_3f(t)$– software operators will differ only in that the aggregates $\left\{\widetilde{g(t)}\right\}$, $\left\{\widetilde{p(t)}\right\}$, $\left\{\widetilde{B(t)}\right\}$, $\left\{\widetilde{x(t)}\right\}$ will be formed from corresponding processes PrffSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) - (.1.4), (2.1*) and analogs of forms (1.1.1) – (1.4) by type (2.1*) for more complex operators.

Remark F.3.2.2. It is the parallel fuzzy containment of oneself in oneself that can "give birth" to the parallel fuzzy capacities in itself – that is what parallel self-organization.

Remark F.3.2.3. $PrffSprt(t)\begin{array}{cccc} r & r & \dots & r \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ r & r & \dots & r \end{array}$

can increase parallel fuzzy self- level of $\widetilde{B(t)}$=(B$_1$(t)|$\mu_{\widetilde{B(t)}}$(B$_1$(t)), B$_2$(t)|$\mu_{\widetilde{B(t)}}$(B$_2$(t)), ..., B$_n$(t)|$\mu_{\widetilde{B(t)}}$(B$_n$(t))),

$$r = \mathrm{PrffSprt(t)}\begin{array}{cccc} B_1(t) & B_2(t) & \dots & B_n(t) \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{array}.$$

Remark F.3.2.4. For example, the operator Prffitself is $\mathrm{PrffS_1f(t)}$.

Remark F.3.2.5. May be considered the following derivatives:

$$\frac{dPrffSprt(t)\begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t)\end{matrix}}{dt}, \frac{d\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{12} & \mu_{22} & \ldots\ \mu_{m2} \\ D_1(t) & D_2(t) & \ldots D_m(t)\end{matrix} PrffSrt(t)}{dt},$$

$$\frac{d\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{12} & \mu_{22} & \ldots\ \mu_{m2} \\ D_1(t) & D_2(t) & \ldots D_m(t)\end{matrix} PrffSrt(t) \begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t)\end{matrix}}{dt}, \quad \frac{dPrffS_if(t)}{dt},$$

i=1,2,3.

Remark F.3.2.6. It is the parallel fuzzy containment of oneself in itself as an element that can be interpreted as parallel fuzzy dynamic fuzzy capacities in itself.

Remark F.3.2.7. Not every fuzzy fcapacity parallel fuzzy containing itself as an element will manifest itself as a sedentary parallel fuzzy fcapacity or parallel fuzzy fcapacity.

PrffSprt – elements for continual sets.

Earlier, we considered finite-dimensional discrete PrffSprt-elements and ffself-capacities in itself as an element. Here we believe some continual PrffSprt-elements and continual parallel ffself-capacities in themselves as an element.

Definition F.3.3.1. The fuzzy dynamic fuzzy set of continual elements $\widetilde{g}=(g_1|\mu_{\widetilde{g}}(g_1),\ g_2|\mu_{\widetilde{g}}(g_2),\ldots,\ g_n|\mu_{\widetilde{g}}(g_n))$ at one point $x = (x_1, x_2, ..., x_n)$ of space X will be called continual PrffSprt – element, and such a point in space will be called parallel ffcapacity of the continual PrffSprt – element. We will denote $PrffSprt\begin{matrix} g_1 & g_2 & \cdots g_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n\end{matrix}$.

Definition F.3.3.2. An ordered fuzzy dynamic fuzzy set of continual elements at one point in space is called an ordered continual PrffSprt–element.

It's allowed to sum continual PrffSprt – elements:

$$PrffSprt\begin{matrix} g_1 & g_2 & \cdots g_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n\end{matrix} + \mathrm{PrffSprt}\begin{matrix} b_1 & b_2 & \ldots b_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n\end{matrix} =$$

$$PrffSprt\begin{matrix} g_1\cup b_1 & g_2\cup b_2 & \ldots g_n\cup b_n \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ x_1 & x_2 & \ldots\ x_n\end{matrix},$$

where some or any elements may be ordered continual elements. It's allowed to multiply continual Prff-

Sprt − elements: $PrffSprt\begin{matrix} g_1 & g_2 & \dots g_n \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ x_1 & x_2 & \dots x_n \end{matrix} * PrffSprt\begin{matrix} b_1 & b_2 & \dots b_n \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ x_1 & x_2 & \dots x_n \end{matrix} =$
$PrffSprt\begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \dots g_n \cap b_n \\ \mu_{11} & \mu_{21} \quad \dots & \mu_{n1} \\ x_1 & x_2 \quad \dots & x_n \end{matrix}$.

Definition F.3.3.3. The continual PrffSelf-capacity A in itself as an element of the first type is the continual fcapacity parallel fuzzy containing itself as an element. Denote PrS_1ffA. $PrffS_1fA = PrffSrt\begin{matrix} A_1 & A_2 & \dots A_n \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ A_1 & A_2 & \dots A_n \end{matrix}$.

Definition F.3.3.4. The ordered continual Prffself-capacity A in itself as an element of the first type is the ordered continual fcapacity parallel containing itself as an element. Denote $\overrightarrow{PrffS_1fA}$.

For example, $PrffSprt^{\begin{matrix} sin\infty & \dots \end{matrix}}_{\begin{matrix} x_1 & x_2 \dots & x_n \end{matrix}} \; PrffSprt\begin{matrix} sin\infty & tg(-\infty) & \dots sin(-\infty) \\ \mu_{11} & \mu_{21} \quad \dots & \mu_{n1} \\ x_1 & x_2 \quad \dots & x_n \end{matrix}$

$= PrffSprt\begin{matrix} \uparrow I \downarrow_{-1}^{1} & \downarrow I \uparrow_{-\infty}^{\infty} & \dots \downarrow I \uparrow_{-1}^{1} \\ \mu_{11} & \mu_{21} \quad \dots & \mu_{n1} \\ x_1 & x_2 \quad \dots & x_n \end{matrix}$, don't confuse with values of these functions.

Definition F.3.3.5. The continual PrffSelf-capacity A in itself, as an element of the second type, is the capacity parallel fuzzy containing fuzzy continual elements from which it can be parallel fuzzy generated. Let's denote $PrffS_2fA$.

An example of continual ffself- capacity in itself as an element of the second type is a living organism since it contains the programs: DNA and RNA.

Definition F.3.3.6. Partial continual PrffSelf-capacity in itself as an element of the third type is called continual PrffSelf-capacity in itself as an element that partially parallel fuzzy contains itself or parallel fuzzy contains elements from which it can be parallel fuzzy generated in part or both simultaneously. Denote $PrffS_3f$.

All continual capacities in PrffSelf-space are continual PrffSelf-capacities in itself as an element by definition. The continual PrffSelf-capacities in itself as an element may appear as continual PrffSrt- capacities and usual continual fuzzy capacities. In these cases, there are used typical fuzzy measure and topology methods.

The connection of continual PrffSprt − elements with continual PrffSelf-capacities in themselves as an element.
Consider a third type of continual PrffSelf- fcapacity in itself as an ele-

ment. For example, based on $PrffSprt\begin{matrix} g_1 & g_2 & \dots g_n \\ \mu_{11} & \mu_{21} & \dots\mu_{n1} \\ x_1 & x_2 & \dots x_n \end{matrix}$, where $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \dots,$ $g_n|\mu_{\widetilde{g}}(g_n)$), i.e. n - continual elements at one point x = $(x_1, x_2, ..., x_n)$, The continual PrffSelf- fcapacity in itself as an element with m continual elements from $\widetilde{g}$, at m<n, can be considered as $PrffS_3f$, which is formed by the form (1.1), i.e., only m continual elements are located in the structure $PrffSprt\begin{matrix} g_1 & g_2 & \dots g_n \\ \mu_{11} & \mu_{21} & \dots\mu_{n1} \\ x_1 & x_2 & \dots x_n \end{matrix}$. Continual ffself-capacities in itself as an element of the third type can be formed for any other structure, not necessarily $PrffSprt$, only by obligatory reducing the number of continual elements in the structure. In particular, using the forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). Structures more complex than $PrffS_3f$ can be introduced.

Mathematics Prfitself for continual elements.

1.Simultaneous parallel addition of the fuzzy sets continual elements$\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \dots,$ $g_n|\mu_{\widetilde{g}}(g_n)$), $i = 1, 2, \dots, n,\ j = 1, 2, \dots, k$, is implemented using

$$PrffSprt\begin{matrix} \{g_1\}\cup & \{g_2\}\cup & \dots\{g_n\}\cup \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ x_1 & x_2 & \dots & x_n \end{matrix}.$$

2. By analogy, for simultaneous multiplication: $PrffSprt\begin{matrix} \{g_1\}\cap & \{g_2\}\cap & \dots\{g_n\}\cap \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ x_1 & x_2 & \dots & x_n \end{matrix}.$

3. Similarly for simultaneous execution of various operations: $PrffSprt\begin{matrix} \{g_1Q_1\} & \{g_2Q_2\} & \dots\{g_nQ_n\} \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ w_1 & w_2 & \dots & w_n \end{matrix}$, , where $\widetilde{Q}$=($Q_1|\mu_{\widetilde{Q}}(Q_1)$, $Q_2|\mu_{\widetilde{Q}}(Q_2), \dots,$ $Q_n|\mu_{\widetilde{Q}}(Q_n)$). Q_i -an operation, i = 1, …, n.

4. Similarly, for the simultaneous execution of various operators: $PrffSprt\begin{matrix} \{F_1g_1\} & \{F_2g_2\} & \dots\{F_ng_n\} \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\widetilde{F}$=($F_1|\mu_{\widetilde{F}}(F_1)$, $F_2|\mu_{\widetilde{F}}(F_2), \dots,$ $F_n|\mu_{\widetilde{F}}(F_n)$). F_i is an operator, i = 1, …, n.
5. The arithmetic itself for parallel continual fuzzy capacities in themselves will be similar: addition - $PrffS_1f\{g+\}$, (or $PrffS_3f\{g+\}$) for the third type), multiplication $PrffS_1f\{g*\}$, ($PrffS_3f\{g*\}$).
6. Similarly with different operations: $PrffS_1f\{gQ\}$, ($PrffS_3f\{gQ\}$), and with different operators: $PrffS_1f\{Fg\}$, ($PrffS_3f\{Fg\}$).

7. $PrffSrt\begin{matrix} A_1 & A_2 & \dots & A_n \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ B_1 & B_2 & \dots & B_n \end{matrix}$ - the result of the parallel fuzzy containment operator. For continual fuzzy sets A_i, B_i, (i = 1, 2, . . . , n), we have

$$PrffSrt\begin{matrix} A_1 & A_2 & \dots A_n \\ \mu_{11} & \mu_{21} & \dots\mu_{n1} \\ B_1 & B_2 & \dots B_n \end{matrix} = \left\{\sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \ , \sum_{i=1}^{n} D_i\right\},$$

where D_i is self-(fuzzy set) for $A_i \cap B_i$ (i = 1, 2, . . . , n). There is the same for structures if they are considered as continual fuzzy sets, for fuzzy sets C_i, D_i $\begin{matrix} C_1 & C_2 & \dots & C_m \\ \mu_{12} & \mu_{22} & \dots & \mu_{m2} \\ D_1 & D_2 & \dots & D_m \end{matrix}PrffSrt=$

$$\left\{\begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\ \} \\ D_1 - D_1 \cap C_1 \end{matrix} \begin{matrix} \{\ \}\dots \\ D_2 - D_2 \cap C_2 \dots \end{matrix} \begin{matrix} \{\ \} \\ D_m - D_m \cap C_m \end{matrix} PrfSrt \\ \sum_{i=1}^{m} (\ C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix}\right\},$$

where Q_i is oself-(fuzzy set) for $(D_i \cap C_i)$(i = 1, 2, . . . , m) [14].

Remark F.3.3.1. $\begin{matrix} C_1 & C_2 & \dots & C_m \\ \mu_{12} & \mu_{22} & \dots & \mu_{m2} \\ D_1 & D_2 & \dots & D_m \end{matrix}$PrffSprt, where continual fuzzy D_1 is forced out of continual fuzzy C_1 with measure of fuzziness μ_{12}, continual fuzzy D_2 is forced out of continual fuzzy C_2 with measure of fuzziness μ_{22}, . . . , continual fuzzy D_m is forced out of continual fuzzy C_m with measure of fuzziness μ_{m2}.

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ \mu_{12} & \mu_{22} & \dots & \mu_{m2} \\ D_1 & D_2 & \dots & D_m \end{matrix} PrffSprt \begin{matrix} A_1 & A_2 & \dots A_n \\ \mu_{11} & \mu_{21} & \dots\mu_{n1} \\ B_1 & B_2 & \dots B_n \end{matrix}$$

where continual fuzzy A_1 fits into continual fuzzy B_1 with measure of fuzziness μ_{11}, continual fuzzy A_2 fits into continual fuzzy B_2 with measure of fuzziness μ_{21}, . . . , continual fuzzy A_n fits into continual fuzzy B_n with measure of fuzziness μ_{n1}, continual fuzzy D_1 is forced out of continual fuzzy C_1 with measure of fuzziness μ_{12}, continual fuzzy D_2 is forced out of continual fuzzy C_2 with measure of fuzziness μ_{22}, . . . , continual fuzzy D_m is forced out of continual fuzzy C_m with measure of fuzziness μ_{m2} simultaneously.

We can consider the concept of a continual PrffSprt - element as PrffSprt$\begin{matrix} A_1 & A_2 & \dots A_n \\ \mu_{11} & \mu_{21} & \dots\mu_{n1} \\ B_1 & B_2 & \dots B_n \end{matrix}$, where continual fuzzy A_1 fits into continual fuzzy B_1 with measure of fuzziness μ_{11}, continual fuzzy A_2 fits into continual fuzzy B_2 with measure of fuzziness μ_{21}, . . . , continual fuzzy A_n fits into continual fuzzy B_n with measure of fuzziness μ_{n1}. Then $PrffSprt\begin{matrix} B_1 & B_2 & \dots B_n \\ \mu_{11} & \mu_{21} & \dots\mu_{n1} \\ B_1 & B_2 & \dots B_n \end{matrix}$ will mean $PrffS_1$f B.

These elements are used for PrffSprt-coding, PrffSprt translation, coding PrffSelf, and translation PrffSelf for networks, which is suitable for electric current of ultrahigh frequency. More complex elements can be considered as continual sets of numbers with their " activation " in mutual directions. For example, ranges of function values, particularly those representing the shape of lightninF. Fuzzy differential fuzzy geometry can be applied here. Also, n-dimensional elements can be considered. The space of such elements is Banach space if we introduce the usual norm for functions or vectors. We call this space - ffSelb-space. Then we introduce the scalar product for functions or vectors and get the Hilbert space. We call this space ffSelh-space. In particular, one can try to describe some processes with these elements by differential equations and use methods from [4]. You can also try to optimize and research some processes with these elements using the techniques from [5]. Let's introduce operators for transforming fcapacity to PrffSelf- fcapacity in itself as an element: PrfQ$_1$S(A) transforms fuzzy A to Prff$_1$SA, PrfQ$_0$S(B) transforms fuzzy B to $\begin{matrix} B_1 & B_2 & \dots B_m \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ B_1 & B_2 & \dots B_m \end{matrix}$ PrffSprt .

Can be considered Q($\begin{matrix} A_1 & A_2 & \dots A_n \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ B_1 & B_2 & \dots B_n \end{matrix}$, $PrffSprt \begin{matrix} A_1 & A_2 & \dots & A_n \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ B_1 & B_2 & \dots & B_n \end{matrix}$), Q-any operator.

Fuzzy dynamic continual PrffSprt – elements.

Definition F.3.4.1. The process of fuzzy containing the fuzzy set of continual elements $\widetilde{g(t)}$=(g$_1$(t)|$\mu_{\widetilde{g(t)}}$(g$_1$(t)), g$_2$(t)|$\mu_{\widetilde{g(t)}}$(g$_2$(t)), ..., g$_n$(t)|$\mu_{\widetilde{g(t)}}$(g$_n$(t))) into one point x= $(x_1, x_2, ..., x_n)$ of the space X at time will be called the dynamic continual PrffSprt – element. We will denote PrffSprt(t) $\begin{matrix} g_1(t) & g_2(t) & \dots g_n(t) \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ x_1 & x_2 & \dots x_n \end{matrix}$.

Definition F.3.4.2. The process of containing an ordered fuzzy set of continual elements at one point in space is called fuzzy dynamic continual ordered PrffSprt – element.

It is allowed to sum fuzzy dynamic continual PrffSprt – elements:

$$PrffSprt(t) \begin{matrix} g_1(t) & g_2(t) & \dots g_n(t) \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ x_1 & x_2 & \dots x_n \end{matrix} + PrffSprt(t) \begin{matrix} b_1(t) & b_2(t) & \dots b_n(t) \\ \mu_{11} & \mu_{21} & \dots \mu_{n1} \\ x_1 & x_2 & \dots x_n \end{matrix} =$$

$$\text{PrffSprt}(t) \begin{matrix} g_1(t) \cup b_1(t) & g_2(t) \cup b_n(t) & \dots & g_n(t) \cup b_n(t) \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ x_1 & x_2 & \dots & x_n \end{matrix}.$$

It's allowed to multiply dynamic continual PrffSprt – elements:

$$PPrffSprt(t)\begin{array}{cccc} g_1(t) & g_2(t) & \ldots & g_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} * PrffSprt(t)\begin{array}{cccc} b_1(t) & b_2(t) & \ldots & b_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} =$$

$$\mathrm{PrffSprt}(t)\begin{array}{cccc} g_1(t)\cap b_1(t) & g_2(t)\cap b_n(t) & \ldots & g_n(t)\cap b_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}.$$

Parallel fuzzy dynamic fuzzy continual containment of oneself in oneself as an element.

Definition F.3.4.3. The fuzzy dynamic continual PrffSprt- fcapacity $PrffSprt(t)\begin{array}{cccc} R_1(t) & R_2(t) & \ldots & R_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ Q_1(t) & Q_2(t) & \ldots & Q_n(t) \end{array}$ is the process of fuzzy embedding fuzzy continual $R_i(t)$ into fuzzy continual $Q_i(t)$, (i = 1, …, n).

Definition F.3.4.4. Parallel dynamic fuzzy continual fcapacity $\widetilde{Q(t)}=(Q_1(t)|\mu_{\widetilde{Q(t)}}(Q_1(t)),\ Q_2(t)|\mu_{\widetilde{Q}(t)}(Q_2(t)),\ldots,\ Q_n(t)|\mu_{\widetilde{Q(t)}}(Q_n(t)))$ fuzzy containing itself as an element of the first type is the process of parallel fuzzy containing fuzzy $\widetilde{Q(t)}$ in $\widetilde{Q(t)}$ $PrffSprt(t)\begin{array}{cccc} Q_1(t) & Q_2(t) & \ldots & Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ Q_1(t) & Q_2(t) & \ldots & Q_n(t) \end{array}$.

Denote $PrffS_1f(t)\widetilde{Q(t)}$.

Definition F.3.4.5. The fuzzy dynamic parallel containment fuzzy continual C(t) of oneself of the second type parallel fuzzy contains the fuzzy continual elements from which it can be parallel fuzzy generated. Denote $PrffS_2f(t)C(t)$.

Definition F.3.4.6. The partial parallel fuzzy dynamic containment fuzzy continual B(t) of oneself of the third type is the process of partial parallel fuzzy embedding fuzzy continual B(t) into oneself or parallel fuzzy embedding fuzzy continual elements from which it can be parallel fuzzy generated in part or both simultaneously. Denote $PrffS_3f(t)B(t)$.

The connection of dynamic continual PrffSprt – elements with parallel fuzzy dynamic fuzzy continual containment of oneself in oneself as an element.
Let us consider the partial parallel fuzzy dynamic fuzzy continual containment of oneself in oneself as an element of the third type. For example, based on $PrffSprt(t)\begin{array}{cccc} Q_1(t) & Q_2(t) & \ldots & Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$, where $\widetilde{Q(t)}=(Q_1(t)|\mu_{\widetilde{Q(t)}}(Q_1(t)),$ $Q_2(t)|\mu_{\widetilde{Q}(t)}(Q_2(t)),\ldots,\ Q_n(t)|\mu_{\widetilde{Q(t)}}(Q_n(t)))$, i.e. n – continual elements at one point x = $(x_1, x_2, ..., x_n)$, we can consider the parallel fuzzy dynamic fuzzy continual capacity in itself $PrffS_3f(t)$ with m elements from $\widetilde{Q(t)}$, m<n,

which is process formed according to the form (1.1), that is, only m elements from $\widetilde{Q(t)}$ are in the structure $PrffSprt(t)\begin{matrix} Q_1(t) & Q_2(t) & \ldots Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.

Parallel fuzzy dynamic fuzzy continual containments of oneself in oneself as an element of the third type can be formed for any other structure, not necessarily PrffSprt, only by necessarily reducing the number of fuzzy continual elements in the structure. In particular, with the help of forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

It is possible to introduce structures more complex than $\mathrm{PrffS_3f(t)}$.

Parallel fuzzy dynamic fuzzy continual mathematics self.

1. The process of simultaneous parallel addition of fuzzy sets continual elements $\left\{\widetilde{g_i(t)}\right\} =$
$\left(g_{i_1}(t)|\mu_{\widetilde{g_i(t)}}(g_{i_1}(t)),\ g_{i_2}(t)|\mu_{\widetilde{g_i(t)}}(g_{i_2}(t)),\ \ldots,\ g_{i_{m_j}}(t)|\mu_{\widetilde{g_i(t)}}(\ g_{i_{m_j}}(t)) \right)$,
i = 1,2,. . . ,n, j =1,2,. . . ,k are realized by $PrffSprt(t)\begin{matrix} \widetilde{\{g_1(t)\}}\cup & \widetilde{\{g_2(t)\}}\cup & \ldots\widetilde{\{g_n(t)\}}\cup \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$
.
2. By analogy, for simultaneous multiplication:
$PrffSprt(t)\begin{matrix} \widetilde{\{g_1(t)\}}\cap & \widetilde{\{g_2(t)\}}\cap & \ldots\widetilde{\{g_n(t)\}}\cap \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$.
3. Similarly for simultaneous execution of various operations:
$PrffSprt(t)\begin{matrix} g_1(t)Q_1(t) & g_2(t)Q_2(t) & \ldots g_n(t)Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$,
where $\widetilde{Q(t)}$=($\mathrm{Q_1(t)}|\mu_{\widetilde{Q(t)}}(\mathrm{Q_1(t)})$, $\mathrm{Q_2(t)}|\mu_{\widetilde{Q(t)}}(\mathrm{Q_2(t)})$, …, $\mathrm{Q_n(t)}|\mu_{\widetilde{Q(t)}}(\mathrm{Q_n(t)})$),
$\mathrm{Q_i(t)}$-an operation, i = 1, …, n, $\widetilde{g(t)}$=($\mathrm{g_1(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_1(t)})$, $\mathrm{g_2(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_2(t)})$, …, $\mathrm{g_n(t)}|\mu_{\widetilde{g(t)}}(\mathrm{g_n(t)})$).
4. Similarly, for the simultaneous execution of various operators: $PrffSprt(t)\begin{matrix} F_1(t)g_1(t) & F_2(t)g_2(t) & \ldots F_n(t)g_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$, where
$\widetilde{F(t)}$=($\mathrm{F_1(t)}|\mu_{\widetilde{F(t)}}(\mathrm{F_1(t)})$, $\mathrm{F_2(t)}|\mu_{\widetilde{F(t)}}(\mathrm{F_2(t)})$, …, $\mathrm{F_n(t)}|\mu_{\widetilde{F(t)}}(\mathrm{F_n(t)})$),
$\mathrm{F_i(t)}$ is an operator, i = 1, …, n.
5. Parallel fuzzy dynamic arithmetic itself for continual containments of oneself will be similar: Parallel fuzzy dynamic addition - $PrffS_1f(t)\left\{\widetilde{g(t)}\cup\right\}$,

(or $PrffS_3f(t)\left\{\widetilde{g(t)}\cup\right\}$ for the third type), Parallel fuzzy dynamic multiplication $PrffS_1f(t)\left\{\widetilde{g(t)}\cap\right\}$, $(PrffS_3f(t)\left\{\widetilde{g(t)}\cap\right\})$.

6. Similarly with different operations: $PrffS_1f(t)\left\{\widetilde{g(t)}\widetilde{Q(t)}\right\}$, $(PrffS_3f(t)\left\{\widetilde{g(t)}\widetilde{Q(t)}\right\})$ and with different operators: $PrfS_1f(t)\left\{\widetilde{F(t)}\widetilde{g(t)}\right\}$, $(PrfS_3f(t)\left\{\widetilde{F(t)}\widetilde{g(t)}\right\})$.

7. $PrffSprt(t)\begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t) \end{matrix}$ *gives* the result

$$PrffSrt(t)\begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t) \end{matrix} = \left\{\sum_{i=1}^{n} A_i(t)\cup B_i(t) - A_i(t)\cap B_i(t)\ ,\sum_{i=1}^{n} D_i(t)\right\},$$

for fuzzy continual sets $A_i(t),\ B_i(t)$, where $D_i(t)$ is self-(*fuzzy* set) for $A_i(t)\cap B_i(t)$, (i = 1, 2, …, n). The same is true for structures if they are treated as *fuzzy* continual sets,

$$\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ D_1(t) & D_2(t) & \ldots D_m(t) \end{matrix}\ \mathrm{PrffSrt(t)} =$$

$$\left\{\begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\ \} & \{\ \}\ldots & \{\ \} \\ D_1(t)-D_1(t)\cap C_1(t) & D_2(t)-D_2(t)\cap C_2(t)\ \ldots & D_2(t)-D_2(t)\cap C_2(t) \end{matrix} PrfSrt \\ \sum_{i=1}^{m}\left(\ C_i(t)-D_i(t)\cap C_i(t)\right)-(D_i(t)-D_i(t)\cap C_i(t)) \end{matrix}\right\}$$

for fuzzy continual sets $C_i(t),\ D_i(t)$, where $Q_i(t)$ is oself-(*fuzzy* set) for $\left(D_i(t)\cap C_i(t)\right)$(i = 1, 2, …, m) [14].

8. Similarly, for dynamic continual PrffSprt-derivatives, dynamic continual PrffSprt-integrals, dynamic continual PrffSprt-lim, fuzzy dynamic continual Prfself-derivatives, fuzzy dynamic continual Prfself-integrals

9. Denote fuzzy dynamic continual Prself-(fuzzy dynamic continual Prself-Q(t)) through fuzzy dynamic continual Prself^2-Q(t), PrfS(t)(n,Q(t))= fuzzy dynamic continual Prself-(fuzzy dynamic continual Prself-(… (fuzzy dynamic continual Prself-Q(t)))) = fuzzy dynamic continual Prself^{n}-Q(t) for n-multiple fuzzy dynamic continual Prself.

Remark F.3.4.1. The parallel fuzzy dynamic continual PrffSprt-displacement will be denote by $\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{12} & \mu_{22} & \ldots\ \mu_{m2} \\ D_1(t) & D_2(t) & \ldots D_m(t) \end{matrix}$ PrffSprt(t) , where fuzzy continual D_1(t) is forced out of fuzzy continual C_1(t) with measure of fuzziness μ_{12}, fuzzy continual D_2(t) is forced out of fuzzy continual C_2(t) with measure of fuzziness μ_{22}, …, fuzzy continual D_m(t) is forced out of fuzzy continual C_m(t) with measure of fuzziness μ_{m2} simultaneously, the result of this process will be described by the expression $\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{12} & \mu_{22} & \ldots\ \mu_{m2} \\ D_1(t) & D_2(t) & \ldots D_m(t) \end{matrix}$ PrfSrt(t). Then the notation $\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{12} & \mu_{22} & \ldots\ \mu_{m2} \\ D_1(t) & D_2(t) & \ldots D_m(t) \end{matrix}\ PrffSprt(t)\ )\ \begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t) \end{matrix}$

where fuzzy continual A_1(t) fits into fuzzy continual B_1(t) with measure of fuzziness μ_{11}, fuzzy continual A_2(t) fits into fuzzy continual B_2(t) with measure of fuzziness μ_{21}, ..., fuzzy continual A_n(t) fits into fuzzy continual B_n(t) with measure of fuzziness μ_{n1}, fuzzy continual D_1(t) is forced out of fuzzy continual C_1(t) with measure of fuzziness μ_{12}, fuzzy continual D_2(t) is forced out of fuzzy continual C_2(t) with measure of fuzziness μ_{22}, ..., fuzzy continual D_m(t) is forced out of fuzzy continual C_m(t) with measure of fuzziness μ_{m2} simultaneously. It is fuzzy dynamic continual PrffSprt-containment of fuzzy continual A_i(t) in fuzzy continual B_i(t) and fuzzy dynamic continual PrffSprt-displacement of fuzzy continual D_j(t) from fuzzy continual C_j(t) simultaneously, (i = 1, 2, ..., n, j = 1, 2, ..., m). The result of this process will be described by the expression

$$\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{12} & \mu_{22} & \ldots\ \mu_{m2} \\ D_1(t) & D_2(t) & \ldots D_m(t) \end{matrix}\ PrffSrt(t)\ \begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t) \end{matrix}.$$

$$PrffSrt(t)\ \begin{matrix} B_1(t) & B_2(t) & \ldots B_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t) \end{matrix}\ \textit{will mean } PrffS1f(t)B(t).$$

$\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{12} & \mu_{22} & \ldots\ \mu_{m2} \\ C_1(t) & C_2(t) & \ldots C_m(t) \end{matrix}\ PrffSprt(t)$ denotes the parallel fuzzy dynamic expelling fuzzy continual $\widetilde{C(t)}$=(C_1(t)|$\mu_{\widetilde{C(t)}}$(C_1(t)), C_2(t)|$\mu_{\widetilde{C(t)}}$(C_2(t)), ..., C_n(t)|$\mu_{\widetilde{C(t)}}$(C_n(t))) oneself out of oneself,

$\begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ A_1(t) & A_2(t) & \ldots A_n(t) \end{matrix}\ \ PrffSprt(t)\ \ \begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ A_1(t) & A_2(t) & \ldots A_n(t) \end{matrix}$ —simultaneous parallel fuzzy dynamic containment fuzzy continual $\widetilde{A(t)}$=(A_1(t)|$\mu_{\widetilde{A(t)}}$(A_1(t)), A_2(t)|$\mu_{\widetilde{A(t)}}$(A_2(t)), ..., A_n(t)|$\mu_{\widetilde{A(t)}}$(A_n(t))) of oneself in oneself and parallel fuzzy dynamic expelling fuzzy continual A(t) oneself out of oneself.

$\begin{matrix} A_1(t) & A_2(t) & \ldots A_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ B_1(t) & B_2(t) & \ldots B_n(t) \end{matrix}\ \ PrffSprt(t)$ will be called parallel fuzzy dynamic anti-(fuzzy continual fcapacity) from oneself.

Remark F.3.4.2. $PrffSprt(t) \begin{array}{cccc} A_1(t) & A_2(t) & \ldots & A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ A_1(t) & A_2(t) & \ldots & A_n(t) \end{array}$ can be interpreted as a multilayer shell of a self-object from the first layer, which is specified by $A_1(t)$ to the nth, which is specified by $A_n(t)$. Based on this, the atomic model can be interpreted as $PrffSprt(t) \begin{array}{cccc} \cup\{p,n\} & A_1(t) & \ldots & A_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ position\ of atomic\ nucleus & A_1(t) & \ldots & A_n(t) \end{array}$, $\{p,n\}$ - protons, neutrons, $A_i(t)$ correspond to orbitals, i = 1, ..., n. $PrffSprt(t) \begin{array}{cccc} physical\ body\ of\ a\ living\ organism & V_1(t) & \ldots & V_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ position\ of physical\ body & V(t) & \ldots & V_n(t) \end{array}$ - model of a living organism with the multilayer shell of a living organism from the first layer, which is specified by $V_1(t)$ to the nth, which is specified by $V_n(t)$. In humans:

$$PrffSprt(t) \begin{array}{cccc} energy\ fibers\ that\ create\ a\ physical\ body\ of\ a\ living\ organism & V_1(t) & \ldots & V_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ energy\ fibers\ that\ create\ a\ physical\ body\ of\ a\ living\ organism & V(t) & \ldots & V_n(t) \end{array}.$$

You can also try to consider the operator PrffSprt $\begin{array}{ccccc} B_1 & \ldots & B_i{}^a & \ldots & B_n \\ \mu_{11} & & \mu_{i1} & \ldots & \mu_{n1} \\ r_1 & \ldots & r_i{}^a & \ldots & r_n \end{array}$, which represents the interpretation of the position of the assemblage point on the cocoon of a living organism, r_j is its potential position, B_j is a potential set of subtle energies in this position (i = 1, ..., i-1, i+1, …, n), $r_i{}^a$ is its active position, $B_i{}^a$ is an active set of subtle energies in this position, i = 1, ..., n.

Connection of fuzzy dynamic continual PrffSprt – elements with target weights with parallel fuzzy dynamic fuzzy continual containment of oneself with target weights.

Consider a third type of parallel partial fuzzy dynamic fuzzy continual containment of oneself with target weights g(t). For example, based on PrffSprt(t) $\begin{array}{cccc} g_1(t)w(t) & g_2(t)w(t) & \ldots & g_n(t)w(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$ .where $\widetilde{g(t)}$=(g₁(t)|$\mu_{\widetilde{g(t)}}$(g₁(t)), g₂(t)|$\mu_{\widetilde{g(t)}}$(g₂(t)), …, gₙ(t)|$\mu_{\widetilde{g(t)}}$(gₙ(t))).

i.e. n - fuzzy continual elements with target weights{w(t)} at one point x = $(x_1, x_2, ..., x_n)$, we can consider the fuzzy dynamic continual containment $PrffS_3f(t)\widetilde{g(t)}w(t)$ of oneself with target weights with m fuzzy continual elements with target weights{w(t)} from $\widetilde{g(t)}$, m<n, which is the process of formation according to the form (1.1), i.e., only m fuzzy continual elements with target weights{w(t)} from $\widetilde{g(t)}$are located in the structure $PrffS_3f(t)\widetilde{g(t)}w(t)$. Parallel fuzzy dynamic fuzzy containments of oneself with target weights of the third type can be formed for any other structure, not necessarily PrffSprt, only by reducing the number of continual elements

with target weights in the structure. In particular, using the forms (1.1.1) - (1.4), (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). Structures more complex than $PrffS_3f(t)\widetilde{g(t)}w(t)$ can be introduced.

Definition F.3.4.7. The parallel fuzzy dynamic embedding of fuzzy continual A(t) into itself with target weights {w(t)} of the first type is the process of parallel fuzzy embedding A(t) into A(t) with target weights. Denote $PrffS_1f(t)A(t)w(t)$.

Definition 30. The parallel fuzzy dynamic containment of fuzzy continual C(t) itself into itself with target weights{w(t)} of the second type is the process of parallel fuzzy containment of the fuzzy continual elements from which it can be parallel fuzzy generated. Let's denote $PrffS_2f(t)C(t)w(t)$.

Definition F.3.4.8. Partial parallel fuzzy dynamic containment of fuzzy continual B(t) itself into itself with target weights {w(t)} of the third type is the process of partial parallel fuzzy containment of fuzzy continual B(t) into itself or fuzzy continual elements from which it can be parallel fuzzy generated partially, or both at the same time. Denote $PrffS_3f(t)B(t)w(t)$.

The usage of PrffSprt-elements for networks.

A. Galushkin's comprehensive monograph [6] covers all aspects of networks, but traditional approaches go through classical mathematics, mainly through the usual correspondence operators. Here we consider a different approach - through a new mathematical process with parallel fuzzy containment operators, which, although they can be interpreted as the result of some correspondence operators, are not themselves correspondence operators. Parallel fuzzy containment operators are more convenient for networks. Also, the main emphasis was placed on using processors operating using triodes, which are generally not used in Sprt-networks. PrffSprt-networks (SmnPrffSprt) are a PrffSprt-structure that can be built for the required weights. PrffSprt-OS (PrffSprt operating system) uses PrffSprt-coding and PrffSprt-translation. In the first one, coding is carried out through a 2-dimensional matrix-row (a, b), where the number b is the code of the action, and the number a is the code of the object of this action. PrffSprt-coding (or PrffSelf-coding) is implemented through a matrix consisting of 2 columns (in the continuous case, two intervals of numbers). Here, the source encoding is used for all matrix rows simultaneously. PrffSprt-translation is carried out by inversion. In this case, PrffSelf-coding and PrffSelf-translation will be more stable. The target weights $g_i(t)$ in

$$\begin{array}{lcccc} & activation\ with g_1(t) & activation\ with g_2(t) & \ldots & activation\ with g_n(t) \\ \text{PrffSprt(t)} & \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ & SmnPrffSprt & SmnPrffSprt & \ldots & SmnPrffSprt \end{array}$$

are chosen for necessary tasks. We will not touch on the issues of applications, or network optimization. They are described in detail by Galushkin [6]. We will touch on the difference of this only for hierarchical complex networks. The same simple executing programs are in the cores of simple artificial neurons of type PrffSprt (designation - mnPrffSprt) for simple information processinF. More complex executing programs are used for mnPrffSprt nodes. PrffSprt-threshold element

$$g_1(t)w_1(t)\ g_2(t)w_2(t)\ \ldots g_n(t)w_n(t)$$
–sgn($PrffSprt(t)$ $\quad \mu_{11} \quad \mu_{21} \quad \ldots \quad \mu_{n1}$), x=(x_1,x_2,…,x_n)
$$x_1 \quad x_2 \quad \ldots \quad x_n$$

- mnPrffSprt, $\widetilde{W(t)}$=(w_1(t)|$\mu_{\widetilde{w(t)}}$(w_1(t)), w_2(t)|$\mu_{\widetilde{w(t)}}$(w_2(t)), …, w_n(t)|$\mu_{\widetilde{w(t)}}$(w_n(t))).

– source signals values, $\{g(t)\} = (g_1(t), g_2(t), ..., g_n(t))$ – PrffSprt-synapses weights. The first level of mnPrffSprt consists of simple mnPrffSprt. The second level of mnPrffSprt consists of

$$mnPrffSprt\ mnPrffSprt\ \ldots mnPrffSprt$$
PrffSprt(t) $\quad \mu_{11} \quad \mu_{21} \quad \ldots \quad \mu_{n1}$ – PrffSprt-node
$$D_1 \quad D_2 \quad \ldots \quad D_n$$

of mnPrffSprt in range D = (D_1, D_2,…, D_n), D- fcapacity for mnPrffSprt node. The third level of mnPrffSprt consists of

$$q \quad q \ \ldots \ q \qquad\qquad q \quad q \ \ldots \ q$$
$$R\ \ PrffSprt(t)\mu_{11}\ \mu_{21}\ \ldots\mu_{n1}\ \ldots PrffSprt(t)\mu_{11}\ \ \mu_{21}\ \ \ldots\mu_{n1}$$
PrffSprt(t)($\quad D_1\ D_2\ \ldots D_n \qquad D_1\ D_2\ \ ) \ldots D_n$,
$$\mu_{11} \qquad \mu_{21} \qquad \ldots \qquad \mu_{n1}$$
$$D_1 \qquad D_2 \qquad \ldots \qquad D_n$$

$$q \quad q \ \ldots \ q$$
R = PrffSprt(t)μ_{11} μ_{21} $\ldots\mu_{n1}$,-PrffSprt^2- node of q = mnPrffSprt in range
$$D_1\ D_2\ \ldots D_n$$
D, thus D becomes fcapacity of itself in itself as an element for mnPrffSprt. For our networks, it is sufficient to use PrffSprt^2- nodes of mnPrffSprt, but self-level is higher in living organisms, particularly PrffSprt^n-, n3. The target structure or the corresponding program enters the target unit using alternating current. After that, all networks or parts of them are activated according to the indicative goal. It may appear that we are leaving the network ideology, but these networks are a complex hierarchy of different levels, like living organisms.

Remark F.3.5.0. A neural network can be thought of as a learnable parallel fuzzy dynamic operator.

Remark F.3.5.1. Traditional scientific approaches through classical mathematics make it possible to describe only at the usual energy level. Here we consider an approach that makes describing processes with finer energies possible. mnPrffSprt con-

$$ffeprogram_1(t)\ ffeprogram_2(t)\ \ldots ffeprogram_n(t))$$
tains PrffSprt(t) $\quad \mu_{11} \quad \mu_{21} \quad \ldots \quad \mu_{n1}$
$$mnPrffSprt \quad mnPrffSprt \quad \ldots \quad mnPrffSprt$$
$ffeprogram-$executing program in PrffSprt- OS. PrffSprt-OS (or Prffself-

OS) is based on PrffSprt-assembly language (or Prself-assembly language), which is based on assembly language through PrffSprt-approach in turn, if the base of elements of PrffSprt-networks is sufficient. The ffeprograms are in PrffSprt-programming environments (or PrffSelf-programming environments), but this question and PrffSprt-networks base will be considered in the following monographs. In particular, ffeprograms may contain PrffSprt-programming operators. In mnPrffSprt cores, the constant memory PrffSprt with correspondent ffeprograms depending on mnPrffSprt.

The OS (operating system) and the principles and modes of operation of the PrffSprt-networks for this programming are interestinF. But this is already the material for the next publications.

Here is developed a helicopter model without a main and tail rotors based on PrffSprt – physics and special neural networks with artificial neurons operating in normal and PrffSprt-modes. Let's denote this model through SmnPrffSprt. To do this, it's proposed to use mnPrffSprt of different levels; for example, for the usual mode, mnPrffSprt serves for the initial processing of signals and the transfer of information to the second level, etc., to the nodal center, then checked. In case of an anomaly - local PrffSprt–mode with the desired "target weight" is realized in this section, etc., to the center. In the case of a monster during the test, SmnPrffSprt is activated with the desired "target weight." Here are realized other tasks also. To reach the fself-energy level, the mode $Sprt_{SmnPrffSprt}^{SmnPrffSprt}$ is used. In normal mode, it's planned to carry out the movement of SmnPrffSprt on jet propulsion by converting the energy of the emitted gases into a vortex to obtain additional thrust upwards. For this purpose, a spiral-shaped chute (with "pockets") is arranged at the bottom of the SmnPrffSprt for the gases emitted by the jet engine, which first exit through a straight chute connected to the spiral one. There is drainage of exhaust gases outside the SmnPrffSprt. SmnPrffSprt is represented by a neural network that extends from the center of one of the main clusters of PrffSprt - artificial neurons to the shell, turning into the body itself. Above the operator's cabin is the central core of the neural network and the target block, responsible for performing the "target weights" and auxiliary blocks, the functions and roles of which we will discuss further. Next is the space for the movement of the local vortex. The unit responsible for SmnPrffSprt's actions is below the operator's cab. In PrffSprt – mode, the entire network or its sections are PrffSprt – activated to perform specific tasks, in particular, with "target weights." In the target, block used PrffSprt -coding, PrffSprt -translation for activation of all networks to "target weights" simultaneously, then –the reset of this PrffSprt-coding after activation. Unfortunately, triodes are not suitable for PrffSprt -neural networks. In the most primitive case, usual separators with corresponding resistances and cores for ffeprograms may be used instead triodes since there is no necessity to unbend the alternating current to direct. The PrffSprt-operative memory belt is disposed around a central core of SmnPrffSprt. There are PrffSprt-coding, PrffSprt-translation, and PrffSprt-realize of eprograms and the programs from the archives without extraction, PrffSprt-coding and PrffSprt-translation may be used in high-intensity, ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna, Strickland. PrffSprt – structure or an eprogram if one is present of needed «target weight» are taken in target block at PrffSprt – activation of the networks. $Sprt_{activation}^{SmnPrffSprt,f}$ derives SmnPrffSprt to the self-level boundary with target weight f.

It's used an alternating current of above high frequency and ultra-violet light, which can work with PrffSprt – structures in PrffSprt–modes by its nature to activate the networks or some of its parts in PrffSprt–modes and locally using PrffSprt–mode. Above high frequently alternating current go through mercury bearers. That's why overheating does not occur. The power of the alternating current above high frequently increases considerably for the target block. The activation of all PrffSprt-networks is realized to indicate "target weights."

Variable hierarchical fuzzy dynamical parallel structures (models) for

fuzzy dynamic, singular, hierarchical fuzzy sets.

Here we will consider variable parallel structures (models), both discrete and continuous: a) with variable connections, b) with the variable backbone for links, c) generalized version; in particular, in variable structures (models), for example,

$$
\begin{array}{ll}
C_1\, C_2\, \ldots\, C_m & A_1\;\, A_2\; \ldots A_n \\
\mu_{12}\mu_{22}\,\ldots\,\mu_{m2} PrffSprt(t) & \mu_{11}\;\mu_{21}\;\ldots\mu_{n1} = \\
D_1\, D_2\, \ldots\, D_m & B_1\;\, B_2\; \ldots B_n
\end{array}
$$

$$
\left\{
\begin{array}{l}
\qquad\quad C_1\, C_2\, \ldots\, C_m \\
\qquad\;\, (\mu_{12}\mu_{22}\,\ldots\,\mu_{m2} PrffSprt, \quad q_2 \ge t \ge q_1)|\mu_1 \\
\qquad\quad D_1\, D_2\, \ldots\, D_m \\
\;\; B_1\, B_2\, \ldots\, B_m \qquad\qquad A_1\;\, A_2\; \ldots A_n \\
(\mu_{12}\mu_{22}\,\ldots\,\mu_{m2} PrffSprt \mu_{11}\;\mu_{21}\;\ldots\mu_{n1} \quad , q_3 \ge\; t > q_2)|\mu_2 \\
\;\; D_1\, D_2\, \ldots\, D_m \qquad\qquad B_1\;\, B_2\; \ldots B_n \\
\;\; C_1\, C_2\, \ldots\, C_m \qquad\qquad A_1\;\, A_2\; \ldots A_n \\
(\mu_{12}\mu_{22}\,\ldots\,\mu_{m2} PrffSprt \mu_{11}\;\mu_{21}\;\ldots\mu_{n1} \quad , q_4 \ge\; t > q_3)|\mu_3 \qquad (*_{\mathrm{F.3.3}}), \\
\;\; D_1\, D_2\, \ldots\, D_m \qquad\qquad B_1\;\, B_2\; \ldots B_n \\
\qquad\qquad\qquad A_1\;\, A_2\; \ldots A_n \\
\qquad\;\, (PrffSprt \mu_{11}\;\mu_{21}\;\ldots\mu_{n1} \;\; , q_5 \ge\; t > q_4)|\mu_4 \\
\qquad\qquad\qquad B_1\;\, B_2\; \ldots B_n \\
\qquad\quad \{\ \}\{\ \}\,\ldots\,\{\ \} \\
\qquad\quad (\mu_{12}\mu_{22}\,\ldots\,\mu_{m2} PrffSprt, \quad t > q_5)|\mu_5 \\
\qquad\quad D_1\, D_2\, \ldots\, D_m \\
\qquad\qquad\qquad\quad \ldots
\end{array}
\right.
$$

μ_i- measures of fuzziness, i = 1, ..., 5. In particular,

$$\begin{array}{lcl} B_1\, B_2\, \ldots\, B_m & & A_1\;\; A_2\; \ldots A_n \\ \mu_{12}\mu_{22}\,\ldots\,\mu_{m2} & PrffSprt & \mu_{11}\;\mu_{21}\;\ldots\mu_{n1} \\ D_1\, D_2\, \ldots\, D_m & & B_1\;\; B_2\; \ldots B_n \end{array}$$

can be interpreted as a fuzzy game: player 1 fuzzy fits fuzzy A_i into fuzzy B_i, i = 1, 2, ..., n, and the other fuzzy pushes fuzzy D_j out of fuzzy Bj, j = 1, 2, ..., m at the same time.

The example of variable parallel hierarchy

$$\begin{array}{lcl} C_1\, C_2\, \ldots\, C_m & & A_1\;\; A_2\; \ldots A_n \\ \mu_{12}\mu_{22}\,\ldots\,\mu_{m2} & PrffSprt(t)_1 & \mu_{11}\;\mu_{21}\;\ldots\mu_{n1} = \\ D_1\, D_2\, \ldots\, D_m & & B_1\;\; B_2\; \ldots B_n \end{array}$$

$$\begin{cases} \left(\left\{\begin{array}{c} \sum_{i=1}^{m} Q_i + \begin{array}{c} \{\ \} \\ D_1 - D_1\cap C_1 \end{array} \begin{array}{c} \{\ \}\ldots \\ D_2 - D_2\cap C_2 \ldots \end{array} \begin{array}{c} \{\ \} \\ D_m - D_m\cap C_m \end{array} PrfSrt \\ \sum_{i=1}^{m} (\,C_i - D_i\cap C_i) - (D_i - D_i\cap C_i) \end{array}\right\},\quad q_2 \ge t \ge q_1\right)|\mu_1 \\ \begin{array}{c} S^{1}{}^{e}_{0} f B_i * \\ \left(\sum_{i=1}^{n} \left({}^{B_i}_{Q_i - B_i} S^1 t^{A_i - B_i}_{B_i}\right), q_3 \ge\ t > q_2\right)|\mu_2 \end{array} \\ \begin{array}{c} S^{et}_{01} f B_i \\ \left(\sum_{j=1}^{m}\sum_{i=1}^{n} \left({}^{C_j - B_i}_{C_j - B_i} S_1 t^{A - B_i}_{B_i} \right) \quad, q_4 \ge\ \ t > q_3\right)|\mu_3 \\ {}^{C_j - B_i}_{D_j - C_j - B_i} S_1 t^{A - B_i}_{B_i} \end{array} \\ \left(\left\{\begin{array}{c} \sum_{i=1}^{n} R_i \\ \sum_{i=1}^{n} A_i\cup B_i - A_i\cap B_i \end{array}\right\},\ , q_5 \ge\ t > q_4\right)|\mu_4 \\ \begin{array}{c} \{\ \}\{\ \}\ \ldots\ \{\ \} \\ (\mu_{12}\mu_{22}\,\ldots\,\mu_{m2} PrfSrt, \quad t > q_5)|\mu_5 \\ D_1\, D_2\, \ldots\, D_m \end{array} \\ \ldots \end{cases}$$

$(*_{\mathrm{F.3.4}})$,

Where μ_i- measures of fuzziness, i = 1, ..., 5, Q_i is oself-(fuzzy set) for $(D_j \cap C_j)$, D_j, C_j are fuzzy sets (j = 1, 2, ..., m), [14], R_i is self-(fuzzy set) for $A_i \cap B_i$, A_i, B_i are fuzzy sets ,(i = 1, 2, ..., n), $S^{et}_{01} f B$, ${}^{C-B}_{C-B} S_1 t^{A-B}_{B}$, ${}^{C-B}_{D-C-B} S_1 t^{A-B}_{B}$ are considered in [13], ${}^{B}_{Q-B} S^1 t^{A-B}_{B}$ is considered in [10].

In what follows, we will denote variable parallel fuzzy dynamic fuzzy structure (model) through PrffVS, parallel fself-type variable fuzzy dynamic fuzzy structures (models) through PrffSVS, and parallel foself-type variable fuzzy dynamic fuzzy structures (models) through PrffOSVS.

Examples: a) discrete variable parallel fuzzy dynamic fuzzy structure

$$\begin{array}{lcccl} C_1\; C_2\,\ldots\; C_m & a & b & g & A_1\; A_2\,\ldots\; A_n \\ D_1\; D_2\,\ldots\; D_m & c & PrffVS & w & B_1\; B_2\,\ldots\; B_n \\ A_1\; A_2\,\ldots\; A_n & d & q & r & C_1\; C_2\,\ldots\; C_m \\ B_1\; B_2\,\ldots\; B_n & & & & D_1\; D_2\,\ldots\; D_m \end{array}$$

c) continuous variable parallel fuzzy dynamic fuzzy structure

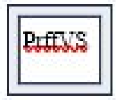


Fig. 0.2 Continuous variable parallel fuzzy dynamic fuzzy set as the rim.

Where a continuous set represents the rim of the Fig. 0.2.

We introduce the notation m_{PrffVS_N}– the number of elements, N - the number of connections between them in the discrete variable parallel 2-hierarchical fuzzy dynamic fuzzy structure PrffVS. We introduce the notation q_{PrffVS_R}– any, R - connections in q_{PrffVS_R} in the variable parallel fuzzy dynamic 2-hierarchical structure PrffVS, in particular, q_{PrffVS_R}, R can be fuzzy sets both discrete and continuous and discrete-continuous. We consider the functional c(Q), which gives a numerical value for the structurability of Q from the interval [0,1], where 0 corresponds to "no parallel fuzzy dynamic fuzzy structure"," and 1 corresponds to the value " parallel fuzzy dynamic fuzzy structure". Then for joint fuzzy dynamic fuzzy A, B: c(A+B)=c(A)+c(B)-c(A*B)+cS(D), D- parallel fself-type fuzzy structures from A*B, cS(x)- the value of PrffSelf for parallel fself-type fuzzy structures x; for dependent parallel fuzzy dynamic fuzzy structures: c(A*B)=ca(A)*c(B/A)=c(B)*c(A/B), where c(B/A)- conditional structurability of the parallel fuzzy dynamic fuzzy structure B at the parallel fuzzy dynamic fuzzy structure A, c(A/B)- conditional fuzzy dynamic fuzzy structure of the parallel fuzzy dynamic fuzzy structure A at the parallel fuzzy dynamic structure fuzzy B. Adding inconsistent parallel fuzzy dynamic fuzzy structures: c(A+B) =c(A) + c(B). The formula of complete parallel fuzzy dynamic fuzzy structure: c(A)=$\sum_{k=1}^{n} c\left(B_k\right) * c(A/B_k)$, B_1, $B_{2.}$,.., B_n-full group of hypotheses- fuzzy containments: $\sum_{k=1}^{n} c\left(B_k\right)$=1(“parallel fuzzy dynamic fuzzy structure”). PrffSprt- structure of the first type for set of parallel fuzzy dynamic fuzzy structures A={A_1, A_2,. . . ,A_n}: PrffSprt$\begin{array}{l} A_1\ A_2\ \ldots A_n \\ \mu_{11}\ \mu_{21}\ \ldots\mu_{n1} \\ x_1\ \ x_2\ \ldots x_n \end{array}$, PrffSprt $\begin{array}{l} c(A_1)\ c(A_2)\ \ldots c(A_n) \\ \mu_{11}\quad \mu_{21}\quad \ldots\ \mu_{n1} \\ x_1\quad x_2\quad \ldots\ x_n \end{array}$ - PrffSprt- structurability for these structures. It is possible to consider the parallel fuzzy dynamic fself-type fuzzy structure $PrffS_3A$ with m parallel fuzzy dynamic fuzzy structures from A, at m<n, which is formed by the form (1.1), that is, only m parallel fuzzy dynamic fuzzy structures from A are located in the structure $PrffSprt\begin{array}{l} A_1\ A_2\ \ldots A_n \\ \mu_{11}\ \mu_{21}\ \ldots\mu_{n1} \\ x_1\ \ x_2\ \ldots x_n \end{array}$. The same for parallel fuzzy dynamic fself-type fuzzy structurability $PrffS_3\{(c(A_1), c(A_2), \ldots, c(A_n))$.

Can be considered N-hierarchical parallel fuzzy structure: 1-level - elements; level 2 - connections between them, level 3 - relationships between elements of level 2, etc. up to level N+1. N-hierarchical parallel fuzzy structure: 1-level - A; 2-level -B, 3-level - C, etc. up to (N+!)- level, where A, B, C, ... can be any in particular, by fuzzy actions, fuzzy sets, and others.

Can be considered discrete hierarchical parallel fuzzy structure, continuous hierarchical parallel fuzzy structure, and discrete-continuous hierarchical parallel fuzzy structure.

The example PrffQHS=

$$HffSprt_x \begin{array}{c} PrffSprt \begin{array}{cccc} n-lhfs_1 & n-lhfs_2 & \ldots & n-lhfs_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} \\ \ldots \\ PrffSprt \begin{array}{cccc} i-lhfs_1 & i-lhfs_2 & \ldots & i-lhfs_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} \\ \ldots \\ PrffSprt \begin{array}{cccc} 1-lhfs_1 & 1-lhfs_2 & \ldots & 1-lhfs_n \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array} \end{array} -$$

N-hierarchical fuzzy structure compression into point x = $(x_1, x_2, ..., x_n)$, lhfs = level of hierarchical fuzzy structure.

Let fPrff(N, PrffQHS)= $PrffQHS^{\left.PrffQHS^{PrffQHS^{\cdots^{PrffQHS}}}\right\}\text{-N levels}}$

It can be considered PrffSelf- PrffQHS, fPrff(y, PrffQHS) for any y, fPrff(PrffQHS, PrffQHS).

Parallel Compression Hierarchy Example:

$$1)\ PrffSprt \begin{matrix} PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} \ldots & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} \\ PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} + B & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} + B \ldots & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} + B \end{matrix}$$

$$= \left(PrffSprt \begin{matrix} PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} \ldots & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} \\ PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} \ldots & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} \end{matrix} \right.$$

$$\left. PrffSprt \begin{matrix} PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} & PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} & \ldots PrffSprt \begin{matrix} (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \\ (\ )\ (\ ) \ldots (\ ) \end{matrix} \\ \mu_{11} & \mu_{21} \quad \ldots & \mu_{n1} \\ B & B \quad \ldots & B \end{matrix} \right)$$

Let's consider two versions: 1) containment is interpreted through the concept of containment, and 2) fcapacity is interpreted through the concept of fuzzy containment as a rest point of fuzzy containment. PrffSelf-containment is interpreted as a rest point of PrffSelf-containment. We consider the functional ca(Q), which gives a numerical value for the fuzzy accommodation of Q from the interval [0,1], where 0 corresponds to " parallel fuzzy containment", and one corresponds to the value " parallel fuzzy fcapacity". Then for joint fuzzy dynamic fuzzy A, B: ca(A+B)=ca(A)+ca(B)-ca(A*B)+caS(D), D- PrffSelf-containment for A*B, caS(x)- the value of PrffSelf-fcapacity for PrffSelf-containment of x; for dependent parallel fuzzy containments: ca(A*B)=ca(A)*ca(B/A)=ca(B)*ca(A/B), where ca(B/A)- conditional fuzzy accommodation of the parallel fuzzy containment B at the parallel fuzzy containment A, ca(A/B)- conditional parallel fuzzy fcapacity of the parallel fuzzy containment A at the parallel fuzzy containment B. Adding the parallel fuzzy fcapacity values of inconsistent parallel fuzzy containments: ca(A+B)=ca(A)+ca(B). The formula of complete parallel fuzzy fcapacity : ca(A)=$\sum_{k=1}^{n} ca\,(B_k) * ca(A/B_k)$, B_1, B_2,..,B_n-full group of hypotheses-(parallel fuzzy containments): $\sum_{k=1}^{n} ca\,(B_k)$=1("parallel fuzzy fcapacity "). PrffSprt- containment for set of parallel fuzzy containments A={A_1, A_2,…,A_n}: PrffSprt$\begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ x_1 & x_2 & \ldots x_n \end{matrix}$, PrffSprt $\begin{matrix} ca(A_1) & ca(A_2) & \ldots ca(A_n) \\ \mu_{11} & \mu_{21} & \ldots \quad \mu_{n1} \\ x_1 & x_2 & \ldots \quad x_n \end{matrix}$ - PrffSprt- accommodation for these parallel fuzzy containments. It is possible to consider the Prffself- containment $PrffS_3A$ with m fuzzy containments from A, at m<n, which is formed by the form (1.1), that is, only m parallel fuzzy containments from A are located in the parallel containment

PrffSprt(t)$\begin{array}{c} A_1 \; A_2 \; \ldots A_n \\ \mu_{11} \; \mu_{21} \; \ldots \mu_{n1} \\ x_1 \; x_2 \; \ldots x_n \end{array}$ and for more complex operators in forms (1.1.1) – (1.4) , (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). . The same for PrffSelf- accommodation - $PrffS_3\{\ ca(A_1), ca(A_2), \ldots, ca(A_n)$.

We consider the functional h(Q), which gives a numerical value for the parallel fuzzy dynamic fuzzy hierarchization of fuzzy Q from the interval [0,1], where 0 corresponds to "no parallel fuzzy dynamic fuzzy hierarchy," and 1 corresponds to the value " parallel fuzzy dynamic fuzzy hierarchy. " Then for joint parallel fuzzy dynamic hierarchies fuzzy A, B: h(A+B)=h(A)+h(B)-h(A*B)+hS(D), D- PrffSelf- hierarchy from A*B, hS(x)- the value of PrffSelf- hierarchy for PrffSelf- hierarchy x; for dependent parallel fuzzy dynamic fuzzy hierarchies: h(A*B) = ha(A)*h(B/A) = h(B)*h(A/B), where h(B/A)- conditional parallel fuzzy dynamic hierarchization of the parallel fuzzy dynamic hierarchy fuzzy B at the parallel fuzzy dynamic hierarchy fuzzy A, h(A/B)- conditional parallel fuzzy dynamic fuzzy hierarchy of the parallel fuzzy dynamic hierarchy fuzzy A at the parallel fuzzy dynamic structure fuzzy B. Adding the parallel fuzzy dynamic fuzzy hierarchy values of inconsistent parallel fuzzy dynamic fuzzy hierarchies: h(A+B)=h(A)+h(B). The formula of complete parallel fuzzy dynamic fuzzy hierarchy: h(A)=$\sum_{k=1}^{n} h\,(B_k) * h(A/B_k)$, B_1, B_2, .., B_n-full group of hypotheses-(parallel fuzzy dynamic fuzzy hierarches): $\sum_{k=1}^{n} h\,(B_k)$=1(“parallel fuzzy dynamic fuzzy hierarchy”).

PrffSprt- structure for set of parallel fuzzy dynamic fuzzy hierarches A={A_1, A_2,…,A_n}: $PrffSprt(t)\begin{array}{c} A_1 \; A_2 \; \ldots A_n \\ \mu_{11} \; \mu_{21} \; \ldots \mu_{n1} \\ x_1 \; x_2 \; \ldots x_n \end{array}$, PrffSprt(t) $\begin{array}{cccc} h(A_1) & h(A_2) & \ldots & h(A_n) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$ - PrffSprt- hierarchization for these parallel fuzzy dynamic fuzzy hierarches. It is possible to consider the Prffself- hierarchy $PrffS_3A$ with m parallel fuzzy dynamic fuzzy hierarches from A, at m<n, which is formed by the form (1.1), that is, only m parallel fuzzy dynamic fuzzy hierarches from A are located in the parallel fuzzy dynamic hierarchy $PrffSprt(t)\begin{array}{c} A_1 \; A_2 \; \ldots A_n \\ \mu_{11} \; \mu_{21} \; \ldots \mu_{n1} \\ x_1 \; x_2 \; \ldots x_n \end{array}$ and for more complex operators in forms (1.1.1) – (1.4) , (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*). . The same for Prffself-hierarchization $PrffS_3\{\ \ h(A_1), h(A_2), \ldots, h(A_n)$.Can be considered PrffSprt(t) $\begin{array}{cccc} \{\ ca(A_1), c(A_1), h(A_1)\} & \{ca(A_2), c(A_2), h(A_2)\} & \ldots & \{ca(A_n), c(A_n), h(A_n)\} \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{array}$.

Very interesting next parallel fuzzy dynamic fuzzy hierarchy type: $\begin{array}{c} h\ A_1\ h\ A_2 \ldots h\ A_n \\ \mu_{11} \;\; \mu_{21} \;\; \ldots \mu_{n1} \\ h\ A_1\ h\ A_2 \ldots h\ A_n \end{array}$ PrffSprt(t) $\begin{array}{c} h\ A_1\ h\ A_2 \ldots h\ A_n \\ \mu_{11} \;\; \mu_{21} \;\; \ldots \mu_{n1} \\ h\ A_1\ h\ A_2 \ldots h\ A_n \end{array}$, h = hierarchy. You can enter special operator PrffCprt to work with fuzzy dynamic fuzzy

$$\text{structures: } \begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ B_1 & B_2 & \ldots B_n \end{matrix} PrffCprt \begin{matrix} R_1 & R_2 & \ldots R_m \\ \mu_{12} & \mu_{22} & \ldots \mu_{m2} \\ Q_1 & Q_2 & \ldots Q_n \end{matrix} \text{ fuzzy structures fuzzy}$$

R_j with the structure from fuzzy Q_j with measure of fuzziness μ_{j2}, fuzzy unstructures fuzzy A_i by the structure fuzzy B_i with measure of fuzziness μ_{i1}, simultaneously, (i = 1, 2, . . . , n, j = 1, 2, . . . , m). Very interesting next parallel fuzzy dynamic fuzzy structure type:

$$\begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ A_1 & A_2 & \ldots A_n \end{matrix} \text{ PrffCrt(t)} \begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ A_1 & A_2 & \ldots A_n \end{matrix},$$

You can enter special parallel operator PrfHprt to work with fuzzy dynamic

$$\text{fuzzy hierarches: } \begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ B_1 & B_2 & \ldots B_n \end{matrix} PrffHprt \begin{matrix} R_1 & R_2 & \ldots R_m \\ \mu_{12} & \mu_{22} & \ldots \mu_{m2} \\ Q_1 & Q_2 & \ldots Q_n \end{matrix}$$

fuzzy hierarchizes fuzzy R_j with the fuzzy hierarchy from fuzzy Q_j with measure of fuzziness μ_{j2}, fuzzy unhierarchizes fuzzy A_i from the fuzzy hierarchy fuzzy B_i with measure of fuzziness μ_{i1}, simultaneously, (i = 1, 2, . . . , n, j = 1, 2, . . . , m).

Program operators PrffSprt, PrfftSpr.

Here it is supposed to use a symbiosis of parallel fuzzy actions and conventional calculations through sequential actions. This must be done through PrffSprt-Networks in one of the central departments of which a conventional computer system is located. The parallel processor is itself prffeprogram with direct parallel computing not through serial computing.

Using conventional PrffSprt -coding by a parallel computer system, through a Target-block with a PrffSprt -program operator -

$$PrffSprt(t) \begin{matrix} g_1(t)w_1(t) & g_2(t)w_2(t) & \ldots g_n(t)w_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ activation & activation & \ldots activation \end{matrix}$$

, it will be possible to obtain the execution of the parallel fuzzy actions $(g_1(t), g_2(t), ..., g_n(t))$ with the desired target weights w(t)=(w_1(t), w_2(t),. . . ,w_n(t)). Each code for a neural network from a conventional computer we "bind" (match) to the corresponding value of current (or voltage). For PrffSprt-coding and PrffSprt-translation may be use alternating current of ultrahigh frequency or high-intensity ultra-short optical pulses laser of Nobel laureates 2018-year Gerard Mourou, Donna Strickland, or a combination of them. For the desired fuzzy action, for example, using the direct parallel fuzzy program of operator

$$\text{PrffSprt(t)} \begin{matrix} (UHF\ AC)_1(t)Q_1(t) & (UHF\ AC)_2(t)Q_2(t) & \ldots (UHF\ AC)_n(t)Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ activation & activation & \ldots & activation \end{matrix},$$

we simultaneously enter the desired fuzzy set of codes $Q_i(t)$, i = 1, 2, . . . , n, using a microwave current or high-intensity ultra-short optical pulses laser in Target-block.

In a conventional computer, the process of sequential calculation takes a certain time interval, in a directly parallel calculation by a neural network, the calculation is instantaneous, but it occupies a certain region of the space of calculation objects.

Consider the types of direct parallel program operators:

1) PrffSprt-program operators
2) PrfftSpr-program operators

Here are some of the PrffSprt-program operators:

1) Simultaneous fuzzy assignment of the fuzzy expressions $\widetilde{p}$=($p_1|\mu_{\widetilde{p}}(p_1)$, $p_2|\mu_{\widetilde{p}}(p_2), \ldots,$ $p_n|\mu_{\widetilde{p}}(p_n)$) to the variables $\widetilde{x}$=($x_1|\mu_{\widetilde{x}}(x_1)$, $x_2|\mu_{\widetilde{x}}(x_2), \ldots,$ $x_n|\mu_{\widetilde{x}}(x_n)$). This is implemented via PrffSprt $\begin{array}{cccc} x_1 := & x_2 := & \ldots & x_n := \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ p_1 & p_2 & \ldots & p_n \end{array}$.

2) Simultaneous fuzzy checking the fuzzy set of conditions $\widetilde{g}$=($g_1|\mu_{\widetilde{g}}(g_1)$, $g_2|\mu_{\widetilde{g}}(g_2), \ldots,$ $g_n|\mu_{\widetilde{g}}(g_n)$) for the fuzzy set of expressions $\widetilde{B}$=($B_1|\mu_{\widetilde{B}}(B_1)$, $B_2|\mu_{\widetilde{B}}(B_2), \ldots,$ $B_n|\mu_{\widetilde{B}}(B_n)$). Implemented via PrffSprt $\begin{array}{cccc} IF\{B_1 g_1\}\ then & IF\{B_2 g_2\}\ then & \ldots & IF\{B_n g_n\}\ then \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ w_1 & w_2 & \ldots & w_n \end{array}$, where w_i (i = 1, …, n) can be anything.

3) Similarly for loop operators and others.

PrffSprt-algorithm Examples:

1) Simultaneous addition and simultaneous parallel fuzzy multiplication of fuzzy sets elements (See point 1, 2 in **Math Prffself**

2) parallel fuzzy pattern recognition: PrffSprt $\begin{array}{cccc} IF\{q_1\ d_1\}\ then & IF\{q_2\ d_2\}\ then & \ldots & IF\{q_n\ d_n\}\ then \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ Name\ of\ q_1 & Name\ of\ q_2 & \ldots & Name\ of\ q_n \end{array}$, d = image archive.

The example of PrffSprt-program is

$$PrffSprt \begin{array}{cccc} u_1\ PrffSprt \begin{array}{cccc} IF\{B_1 g_1\}\ then & IF\{B_2 g_2\}\ then & \ldots & IF\{B_n g_n\}\ then \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ w_1 & w_2 & \ldots & w_n \end{array} & & \ldots & u_2 \\ \mu_{11} & \mu_{21} & \ldots & \mu_{m1} \\ r_1 & r_2 & \ldots & r_m \end{array},$$

$$u_1 = PrffSprt \begin{array}{cccc} x_1 := & x_2 := & \dots & x_n := \\ \mu_{11} & \mu_{21} & \dots\dots & \mu_{n1} \\ p_1 & p_2 & \dots & p_n \end{array}, \; u_2 = PrffSprt \begin{array}{cccc} w_1 & w_2 & \dots & w_n \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ w_1 & w_2 & \dots & w_n \end{array}.$$

$PrffS_3f-$ software operators will differ only just because fuzzy aggregates $\{\widetilde{g}\}, \{\widetilde{p}\}, \left\{\widetilde{B}\right\}, \{\widetilde{x}\}$ will be formed from corresponding PrffSprt-program operators in form (1.1) and for more complex operators in forms (1.1.1) – (1.4) , (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

For example, $fSprt^{\{R\}}_{g\{R\}}$ is the fcapacity in itself of the second type if $g\{R\}$ is a program capable of generating $\{R\}$.

The example of self-program of the first type is

$$PrffSprt \begin{array}{cccc} u_1 & u_0 \dots & & \begin{array}{c} w_1 \;\; w_2 \; \dots w_n \\ PrffSprt \mu_{11} \; \mu_{21} \dots \mu_{n1}, \\ w_1 \;\; w_2 \; \dots w_n \end{array} \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ u_1 & u_0 & \dots & \begin{array}{c} w_1 \;\; w_2 \; \dots w_n \\ PrffSprt \mu_{11} \; \mu_{21} \dots \mu_{n1} \\ w_1 \;\; w_2 \; \dots w_n \end{array} \end{array}, \; u_0 =$$

$$PrffSprt \begin{array}{cccc} IF\{B_1 g_1\} \; then & IF\{B_2 g_2\} \; then & \dots & IF\{B_n g_n\} \; then \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ w_1 & w_2 & \dots & w_n \end{array}.$$

PrffSprt-coding: 1) fuzzy set A_i to fuzzy set B_i, 2) fuzzy set A_i to a point q_i, where the elements of the fuzzy sets A_i, B_i can be continuous, (i = 1, 2, . . . , n; j = 1, 2, . . . , m). For example, $PrffSprt \begin{array}{cccc} A_1 & A_2 & \dots & A_n \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ B_1 & B_2 & \dots & B_n \end{array}.$

There are PrffSprt -coding, PrffSprt-translation, PrffSprt-realize of prffeprograms and of the programs from the archives without extraction theirs

Self-coding: 1) fuzzy set A_i to fuzzy set A_i, i.e. A_i on itself 2) fuzzy set A_i to a point q_i in forms (1.1) - (1.4), where the elements of the fuzzy sets A_i can be continuous. For example, $PrffSprt \begin{array}{cccc} A_1 & A_2 & \dots & A_n \\ \mu_{11} & \mu_{21} & \dots & \mu_{n1} \\ A_1 & A_2 & \dots & A_n \end{array}.$

One of the central departments of the control system should be a computer system of the usual type of the desired level. In symbiosis with PrffSprt-Networks, it will provide a holistic operation of the control system in three modes: conventional serial through a conventional type computer system, direct parallel through PrffSprt -Networks and series-parallel. Codes from a conventional type computer system will be used via PrffSprt -connectors in PrffSprt - coding, for example:

PrffSprt(t) $\begin{matrix} (UHF\ AC)_1(t)Q_1(t) & (UHF\ AC)_2(t)Q_2(t) & \ldots & (UHF\ AC)_n(t)Q_n(t) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ activation & activation & \ldots & activation \end{matrix}$,
UHF AC field activation is used.

Consider the fuzzy dynamic PrffSprt and PrffS$_3$f(t) programming:

1. The process of simultaneous fuzzy assignment of the fuzzy expressions $\widetilde{p(t)}$=(p$_1$(t)|$\mu_{\widetilde{p(t)}}$(p$_1$(t)), p$_2$(t)|$\mu_{\widetilde{p(t)}}$(p$_2$(t)), $\ldots$, p$_n$(t)|$\mu_{\widetilde{p(t)}}$(p$_n$(t))) to the variables $\widetilde{x(t)}$=(x$_1$(t)|$\mu_{\widetilde{x(t)}}$(x$_1$(t)), x$_2$(t)|$\mu_{\widetilde{x(t)}}$(x$_2$(t)), $\ldots$, x$_n$(t)|$\mu_{\widetilde{x(t)}}$(x$_n$(t))) is implemented through PrffSprt $\begin{matrix} x_1(t) & x_2(t) & \ldots x_n(t) \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ p_1(t) & p_2(t) & \ldots p_n(t) \end{matrix}$.
.

2. The process of simultaneous check the fuzzy set of conditions $\widetilde{g(t)}$=(g$_1$(t)|$\mu_{\widetilde{g(t)}}$(g$_1$(t)), g$_2$(t)|$\mu_{\widetilde{g(t)}}$(g$_2$(t)), $\ldots$, g$_n$(t)|$\mu_{\widetilde{g(t)}}$(g$_n$(t))) for the fuzzy set of expressions $\widetilde{B(t)}$=(B$_1$(t)|$\mu_{\widetilde{B(t)}}$(B$_1$(t)), B$_2$(t)|$\mu_{\widetilde{B(t)}}$(B$_2$(t)), $\ldots$, B$_n$(t)|$\mu_{\widetilde{B(t)}}$(B$_n$(t))) is implemented through

PrffSprt $\begin{matrix} IF\left\{B_1(t)g_1(t)\right\}\ then & IF\left\{B_2(t)g_2(t)\right\}\ then & \ldots & IF\left\{B_n(t)g_n(t)\right\}\ then \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ w_1(t) & w_2(t) & \ldots & w_n(t) \end{matrix}$

where w(t) = $(w_1(t), w_2(t), ..., w_n(t))$. can be any.

3.Similarly for loop operators and others.

Pr $ffS_3f(t)$– software operators will differ only in that the aggregates $\left\{\widetilde{g(t)}\right\}, \left\{\widetilde{p(t)}\right\}, \left\{\widetilde{B(t)}\right\}, \left\{\widetilde{x(t)}\right\}$ will be formed from corresponding processes PrffSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) – (1.4) for more complex operators, (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

Consider PrfftSpr-program operators. The ideology of PrftSpr and $Prft_{S_4f}$ is parallel analogue of ft$_{S_4f}$ [14] can be used for programminF. Here are some of the PrftSpr -program operators.

1. Simultaneous fuzzy expelling assignment of the fuzzy expressions $\widetilde{p}$=(p$_1$|$\mu_{\widetilde{p}}$(p$_1$), p$_2$|$\mu_{\widetilde{p}}$(p$_2$), $\ldots$, p$_n$|$\mu_{\widetilde{p}}$(p$_n$)) from the variables $\widetilde{x}$=(x$_1$|$\mu_{\widetilde{x}}$(x$_1$), x$_2$|$\mu_{\widetilde{x}}$(x$_2$), $\ldots$, x$_n$|$\mu_{\widetilde{x}}$(x$_n$)). It's implemented through $\begin{matrix} x_1 & x_2 & \ldots x_n \\ \mu_{11} & \mu_{21} & \ldots \mu_{n1} \\ p_1 & p_2 & \ldots p_n \end{matrix}$PrffSprt.

2. Simultaneous fuzzy expelling checks the fuzzy set of conditions $\widetilde{g}$=(g$_1$|$\mu_{\widetilde{g}}$(g$_1$), g$_2$|$\mu_{\widetilde{g}}$(g$_2$), $\ldots$, g$_n$|$\mu_{\widetilde{g}}$(g$_n$)) for the fuzzy set of expressions $\widetilde{B}$=(B$_1$|$\mu_{\widetilde{B}}$(B$_1$), B$_2$|$\mu_{\widetilde{B}}$(B$_2$), $\ldots$, B$_n$|$\mu_{\widetilde{B}}$(B$_n$)). It's implemented through

$IF\,\{B_1g_1\}\ then\ IF\,\{B_2g_2\}\ then\ \ldots IF\,\{B_ng_n\}\ then$
$\mu_{11}\quad \mu_{21}\quad \ldots\quad \mu_{n1}\quad PrffSprt$ where w_i (i
$w_1\quad w_2\quad \ldots\quad w_n$
= 1, …, n) can be anything.

3. Similarly for loop operators and others.

$Prfft_{S_4f}$– software operators will differ only just because aggregates $\{\widetilde{g}\},\{\widetilde{p}\},\left\{\widetilde{B}\right\},\{\widetilde{x}\}$ will be formed from corresponding PrfftSpr program operators in form (1.1) for more complex operators in forms (1.1.1) – (1.4) , (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

Consider hierarchical PrfftSpr-program operator

$$\begin{matrix} C_1(t) & C_2(t) & \ldots C_m(t) \\ \mu_{12} & \mu_{22} & \ldots\ \mu_{m2} \\ D_1(t) & D_2(t) & \ldots D_m(t) \end{matrix}\ \mathrm{PrffSrt_2(t)} =$$

$$\left\{\begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\ \} & \{\ \} & \ldots & \{\ \} \\ \mu_{12} & \mu_{22} & \ldots & \mu_{m2} \\ D_1(t)-D_1(t)\cap C_1(t) & D_2(t)-D_2(t)\cap C_2(t) & \ldots D_m(t)-D_2(t)\cap C_m(t) \end{matrix}\ PrffSrt(t) \\ \sum_{i=1}^{m}\big(\ C_i(t)-D_i(t)\cap C_i(t)\big)-(D_i(t)-D_i(t)\cap C_i(t)) \end{matrix}\right\},$$

where $Q_i(t)$ is oself-(fuzzy set) for $\big(D_i(t)\cap C_i(t)\big)$ [14].

Consider the PrtffSpr(t) and Pr $fft(t)_{S_4f}$ programming at time t.

1. The process of simultaneous assignment of the expressions $\widetilde{p(t)}$=(p_1(t)|$\mu_{\widetilde{p(t)}}$(p_1(t)), p_2(t)|$\mu_{\widetilde{p(t)}}$(p_2(t)), …, p_n(t)|$\mu_{\widetilde{p(t)}}$(p_n(t))) to the variables $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t)), …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))) is implemented through $\begin{matrix} x_1(t) & x_2(t) & \ldots x_n(t) \\ \mu_{11} & \mu_{21} & \ldots\ \mu_{n1} \\ p_1(t) & p_2(t) & \ldots p_n(t) \end{matrix}$ PrffSprt(t).

2. The process of simultaneous check the set of conditions $\widetilde{g(t)}$=(g_1(t)|$\mu_{\widetilde{g(t)}}$(g_1(t)), g_2(t)|$\mu_{\widetilde{g(t)}}$(g_2(t)), …, g_n(t)|$\mu_{\widetilde{g(t)}}$(g_n(t))) for the fuzzy set of expressions $\widetilde{B(t)}$=(B_1(t)|$\mu_{\widetilde{B(t)}}$(B_1(t)), B_2(t)|$\mu_{\widetilde{B(t)}}$(B_2(t)), …, B_n(t)|$\mu_{\widetilde{B(t)}}$(B_n(t))) is implemented through

$$\begin{matrix} IF\,\{B_1(t)g_1(t)\}\ then & IF\,\{B_2(t)g_2(t)\}\ then & \ldots & IF\,\{B_n(t)g_n(t)\}\ then \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ w_1(t) & w_2(t) & \ldots & w_n(t) \end{matrix}$$

PrffSprt(t), where $\widetilde{x(t)}$=(x_1(t)|$\mu_{\widetilde{x(t)}}$(x_1(t)), x_2(t)|$\mu_{\widetilde{x(t)}}$(x_2(t)), …, x_n(t)|$\mu_{\widetilde{x(t)}}$(x_n(t))) can be any.

3.Similarly for loop operators and others.

$Prfft(t)_{S_4f}$– software operators will differ only in that the aggregates $\widetilde{x(t)}, \widetilde{p(t)}, \ \widetilde{B(t)}, \widetilde{g(t)}$ will be formed from corresponding processes PrffSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) – (1.4) for more complex operators, (2.1*) and analogs of forms (1.1.1) - (1.4) by type (2.1*).

Consider PrffSrt(t)- program operators

$$\left\{ {}^{q}_{W_q}\!\begin{pmatrix} u & u & & u & u \\ \mu_1 & \mu_2 & {}^{PrffSpr} & \mu_1 & \mu_2 \\ u & u & & u & u \end{pmatrix}^{E_q} fSprt_{{}^{q}\!\begin{pmatrix} u & u & & u & u \\ \mu_1 & \mu_2 & {}^{PrffSpr} & \mu_1 & \mu_2 \\ u & u & & u & u \end{pmatrix}}, \quad fSprt^{\{E^{ex} l^{d_r}\}}_{d_r \left(E_{in} l^{d_r}\right)} \right\}$$

$fSprt_{t_0}$.

—program structure example, where the assemblage point d_r is the cursor, it is quite complex ffself—program.

Remark. Energy with measure of fuzziness μ_1, μ_2 of a living organism other than humans:

fPrff(r, u(E_q), μ_1, μ_2) =

$$\left\{ {}^{q}_{W_q}\!\begin{pmatrix} u & u & & u & u \\ \mu_1 & \mu_2 & {}^{PrffSpr} & \mu_1 & \mu_2 \\ u & u & & u & u \end{pmatrix}^{E_q} Sprt_{{}^{q}\!\begin{pmatrix} u & u & & u & u \\ \mu_1 & \mu_2 & {}^{PrffSpr} & \mu_1 & \mu_2 \\ u & u & & u & u \end{pmatrix}}, \quad fSprt^{\{E^{ex} l^{d_r}\}}_{d_r \left(E_{in} l^{d_r}\right)} \right\}$$

$fSprt_{t_0}$

($**_{\mathrm{F.3}}$).

Energy with measure of fuzziness μ_1, μ_2 of a living organism of a person:

fPrhff(r, u(E_q), μ_1, μ_2) =

$$\left\{ {}^{q}_{W_q}\!\begin{pmatrix} u & u & & u & u \\ \mu_1 & \mu_2 & {}^{PrffSpr} & \mu_1 & \mu_2 \\ u & u & & u & u \end{pmatrix}^{E_q} Sprt_{{}^{q}\!\begin{pmatrix} u & u & & u & u \\ \mu_1 & \mu_2 & {}^{PrffSpr} & \mu_1 & \mu_2 \\ u & u & & u & u \end{pmatrix}}, \quad fSprt^{\{E^{ex} l^{d_r}\}}_{d_r \,(self\left(E_{in} l^{d_r}\right))} \right\}$$

$fSprt_{t_0}$

($***_{\mathrm{F.3}}$).

$\begin{matrix} u & u & & u & u \\ \mu_1 & \mu_2 & PrffSpr & \mu_1 & \mu_2 \\ u & u & & u & u \end{matrix}$ -internal energy with measure of fuzziness μ_1, μ_2 of a living organism of double energy structure, q- a gap in the energy cocoon of a living organism, r-the position of the assemblage point d_r on the energy cocoon of a living organism, W_q- energy prominences from the gap in the cocoon of a living organism, E_q-external energy entering the gap in the cocoon of a living organism, $E^{ex}l^{d_r}$ - a bundle of fibers of external energy self-capacities from outside the cocoon, collected at the point of assembly of the cocoon of a living organism, $E_{in}l^{d_r}$- a bundle of fibers of external energy self-capacities from inside the cocoon, collected at the point of assembly of the cocoon of a living organism in the same position r of the assemblage point d_r. d_r is the subject of identifying the energy fibers of the subtle energy of the Universe in position r both outside and inside the cocoon.

($**_{F.3}$), ($***_{F.3}$) can be interpreted as PrfSrt- program operators.

Appendix.

Supplement for string theory: May be to try represent elementary particles in the form of continual self-elements of the type:

$$PrffSprt\begin{matrix} \uparrow I\downarrow_{-1}^{1} & \downarrow I\uparrow_{-\infty}^{\infty} & \ldots & \downarrow I\uparrow_{-1}^{1} \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$$, etc.

Supplement for PrffSprt-logic: We consider PrffSprt-logic: consider the functional f(Q), which gives a numerical value for the truth of the fuzzy dynamic fuzzy statement Q from the interval [0,1], where 0 corresponds to "ffno," and one corresponds to the logical value "ffyes." Then for joint fuzzy dynamic fuzzy statements A, B: f(A+B)=f(A)+f(B)-f(A*B)+fS(D), D- self- (fuzzy dynamic fuzzy statement) from A*B, fS(x)- the value of self-(fuzzy dynamic fuzzy truth) for self-(fuzzy dynamic fuzzy statement) x; for dependent fuzzy dynamic fuzzy statements: f(A*B)=f(A)*f(B/A)=f(B)*f(A/B), where f(B/A)- conditional fuzzy dynamic fuzzy truth of the fuzzy dynamic fuzzy statement B at fuzzy dynamic fuzzy statement A, f(A/B)- dependent fuzzy dynamic fuzzy truth of the fuzzy dynamic fuzzy statement A at the fuzzy dynamic fuzzy statement B. Adding the fuzzy dynamic fuzzy truth values of inconsistent fuzzy dynamic fuzzy propositions: f(A+B)=F(A)+f(B). The formula of complete fuzzy dynamic fuzzy truth: f(A)=$\sum_{k=1}^{n} f\left(B_k\right) * f(A/B_k)$, B_1, B_2, .., B_n-full group of hypotheses- (fuzzy dynamic fuzzy statements): $\sum_{k=1}^{n} f\left(B_k\right)$=1(“yes”).

Remark. A statement can be interpreted as an event, and its truth value as a probability.

PrffSprt- statement for set of fuzzy dynamic fuzzy statements A ={A_1, A_2, …, A_n}: PrffSprt$\begin{matrix} A_1 & A_2 & \ldots A_n \\ \mu_{11} & \mu_{21} & \ldots\mu_{n1} \\ x_1 & x_2 & \ldots x_n \end{matrix}$. PrffSprt $\begin{matrix} f(A_1) & f(A_2) & \ldots f(A_n) \\ \mu_{11} & \mu_{21} & \ldots & \mu_{n1} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ - PrffSprt-truth for these fuzzy statements. It is possible to consider the self-(fuzzy statement) $PrfS_3A$ with m fuzzy statements from A, at m<n, which is formed by the form (1.1), that is, only m fuzzy statements from A are located

in the structure $PrffSprt\begin{matrix}A_1 & A_2 & \dots A_n\\ \mu_{11} & \mu_{21} & \dots\mu_{n1}\\ x_1 & x_2 & \dots x_n\end{matrix}$. The same for self-(fuzzy truth) $PrffS_3\{\ f\left(A_1\right), f\left(A_2\right), \dots, f\left(A_n\right)\}$.

One can introduce the concepts of PrffSprt-group: PrffSprt$\begin{matrix}A_1 & A_2 & \dots A_n\\ \mu_{11} & \mu_{21} & \dots\mu_{n1}\\ x_1 & x_2 & \dots x_n\end{matrix}$, A is the usual fuzzy group, PrffSprt$\begin{matrix}A_1 & A_2 & \dots A_n\\ \mu_{11} & \mu_{21} & \dots\mu_{n1}\\ x_1 & x_2 & \dots x_n\end{matrix}$, where A_i is usual fuzzy group, i=1,2, ...,n, x- usual fuzzy groups, self- (fuzzy group): PrffS$_i$fA, i=1,2,3, A is usual fuzzy group.

Definition F.3.5.1. A fuzzy dynamic fuzzy structure with a second degree of freedom will be called complete, i.e., "capable" of reversing itself concerning any of its fuzzy elements clearly, but not necessarily in known operators; it can form (create) new special fuzzy dynamic fuzzy operators (in particular, special fuzzy dynamic fuzzy functions). In particular, PrffCrt $\begin{matrix}A_1 & A_2 & \dots A_n\\ \mu_{11} & \mu_{21} & \dots\mu_{n1}\\ A_1 & A_2 & \dots A_n\end{matrix}$ is such structure. Similarly, for working with fuzzy models, each is structured by its fuzzy dynamic fuzzy structure; for example, use PrffSprt-groups, PrffSprt-rings, PrffSprt-fields, PrffSprt-spaces, PrffSelf-groups, PrffSelf-rings, PrffSelf-fields, and PrffSelf-spaces. Like any task, this is also a fuzzy dynamic fuzzy structure of the appropriate fuzzy fcapacity . Since the degree of freedom is double, it is clear that the form of the PrffSelf-equation contains a fuzzy dynamic fuzzy solutions or fuzzy dynamic fuzzy structures the inversion of the PrffSelf-equation concerning unknowns, i.e., the fuzzy dynamic fuzzy structure of the Prself-equation is complete.

References

1) Danilishyn .V. Danilishyn O.V.(31.03.2023) THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, /Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

2) Danilishyn I., Danilishyn O.(2023) DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square. *2023-08-01* https://doi.org/10.21203/rs.3.rs-3217178/v1

Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

3) Danilishyn .V. Danilishyn O.V.(31.03.2023) DYNAMICAL SIT-ELEMENTS . IV International Scientific and Practical Conference

”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

4) Krain S.G.(1967) Linear differential equations in Banach space. M. Science, (in Russian).

5) A.L.Dontchev. (1983) Perturbations, approximations and sensitivity analysis of optimal control systems. Sptinger-Verllag, Berlin, New York.

6) Galushkin A.(2010) Networks: principles of the theory. Hot line-Telecom. M. (in Russian).

7) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3, pp.1-31, March 27, 2024 https://cskscientificpress.com/articles_file/527-_article1712797848.pdf

8) Oleksandr Danilishyn, Illia Danilishyn.(2023) Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), , 479-498. https://www.opastpublishers.com/open-access-articles /dynamical-sets-theory-s2telements-and-their-applications.pdf

9) Danilishyn I., Danilishyn O. (2023) Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care, Volume 4 | Issue 2 | 66- 81. https://doi.org/10.33140/NAB CC.04.02.02

10) Oleksandr Danilishyn and Illia Danilishyn.(25 September 2023) Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 1-11, 815-_ article1695707465.pdf (mkscienceset.com)

11) I. Danilishyn, O. Danilishyn (2023)Dynamic Sets Se, Networks Se. *Advances in Neurology and Neuroscience*, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

12) Danilishyn .V. Danilishyn.(April 28,2023) O.V. CONTINUAL SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS. Theoretical and practical aspectsof modern scientific research:Collection of scientific papers «O» with Proceedings of the IIInternational Scientific and Practical Conference, Seoul. Seoul-Vinnytsia: Case Co., Ltd.& European Scientific Platform, **pp.144-150. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/10**

13) Danilishyn .V. Danilishyn O.V.(June, 2023) SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 with the proceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC

AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

14) Danilishyn .V. Danilishyn O.V. (June 23, 2023) tS – ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

15) Oleksandr Danilishyn and Illia Danilishyn. (2024, Apr 17) Fuzzy Dynamic Fuzzy Sets. Variable Fuzzy Hierarchical Dynamic Fuzzy Structures (Models, Operators) for Dynamic, Singular, Hierarchical Fuzzy Sets. FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS1epr, ffSeprt1. Journal of Mathematical Techniques Computational Mathematics (JMTCM), Volume 3 | Issue 4 |pp. 1 – 37. *Journal DOI: 10.33140/JMTCM* https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

16) Oleksandr Danilishyn and Illia Danilishyn. Introduction to Section of Dynamic Mathematics: Dynamic Sets Theory: Parallel Sprt-Elements and their Applications. J Math Techniques Comput Math (*Journal DOI: 10.33140/JMTCM*), Volume 3 | Issue 5 (2024), pp.1- 27.

https://www.opastpublishers.com/open-access-articles/introduction-to-section-of-dynamic-mathematics-dynamic-sets-theory-parallel-sprtelements-and-their-applications.pdf

Declarations:

Availability of data and material

1) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

2) Danilishyn .V. Danilishyn O.V. MATHEMATICS ST, PROGRAMMING OPERATORS ST AND SOME EMPLOYMENT. Collection of scientific papers “SCIENTIA”, 2023. **https://previous.scientia.report/index.php/archive/issue/view/07.04.2023**.

3) Danilishyn .V. Danilishyn O.V. DYNAMICAL SIT-ELEMENTS . IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

4) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square. *2023-08-01* https://doi.org/10.21203/rs.3.rs-3217178/v1 Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

5) Oleksandr Danilishyn, Illia Danilishyn. Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), 2023, 479-498. https://www.opastpublishers.com/open-access-articles/dynamical-sets-theory-s2telements-and-their-applications.pdf

6) Danilishyn I., Danilishyn O. Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care 2023, Volume 4 | Issue 2 | 66- 81. https://doi.org/10.33140/NABCC .04.02.02

7) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-__ article1695707465.pdf (mkscienceset.com)

8) I. Danilishyn, O. Danilishyn Dynamic Sets Se, Networks Se. *Advances in Neurology and Neuroscience*, 2023, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

9) Danilishyn .V. Danilishyn O.V. CONTINUAL SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS. Theoretical and practical aspectsof modern scientific research:Collection of scientific papers «O» with Proceedings of the IIInternational Scientific and Practical Conference, Seoul,April28,2023. Seoul-Vinnytsia: Case Co., Ltd.& European Scientific Platform, **pp.144-150. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/10**

10) Danilishyn .V. Danilishyn O.V. SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June, 2023 with the proceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

11) Danilishyn .V. Danilishyn O.V. tS – ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

12) Oleksandr Danilishyn and Illia Danilishyn. Fuzzy Dynamic Fuzzy Sets. Variable Fuzzy Hierarchical Dynamic Fuzzy Structures (Models, Operators) for Dynamic, Singular, Hierarchical Fuzzy Sets. FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS1epr, ffSeprt1. Journal of Mathematical Techniques Computational Mathematics (JMTCM), Volume 3 | Issue 4 |2024, Apr 17, pp. 1 – 37. *Journal DOI: 10.33140/JMTCM* https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

13) Oleksandr Danilishyn and Illia Danilishyn. Introduction to Section of Dynamic Mathematics: Dynamic Sets Theory: Parallel Sprt-Elements and their Applications. J Math Techniques Comput Math (*Journal DOI: 10.33140/JMTCM*), Volume 3 | Issue 5 (2024), pp.1- 27. https://www.opastpublishers.com/open-access-articles/introduction-to-section-of-dynamic-mathematics-dynamic-sets-theory-parallel-sprtelements-and-their-applications.pdf

14) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3, pp.1-31, March 27, 2024 https://cskscientificpress.com/articles_file/527-_article1712797848.pdf

Competing interest

There are no competing interests. All sections of the article are executed jointly.

Funding

There were no sources of funding for writing the article.

Authors' contributions

The contribution of the authors is the same, we will not separate.

G Theory of singularities of the type synthesizing

Introduction.

There is a need to develop an instrumental mathematical base for new technologies. The task of the work is to create new approaches for this by introducing new concepts and methods. Our mathematics is unusual for a mathematician, because here the fulcrum is the action, and not the result of the action as in classical mathematics. Therefore, our mathematics is adapted not only to obtain results, but also to directly control actions, which will certainly show its benefits on a fundamentally new type of neural networks with directly parallel calculations, for which it was created. Any action has much greater potential than its result. Significance of the article: in a new qualitatively different approach to the study of complex processes through new mathematical, hierarchical, dynamic structures, in particular those processes that are dealt with by Synergetics. Our approach is not based on deterministic equations that generate self-organization, which is very difficult to study and gives very small results for a very limited class of problems and does not provide the most important thing - the structure of self-organization. We are just starting from the assumed structure of self-organization, since we are interested not so much in the numerical calculation of this as in the structure of self-organization itself, its formation (construction) for the necessary purposes and its management. Although we are also interested in numerical calculations. Nobel laureates in physics 2023 Ferenc Kraus and his colleagues Pierre Agostini and Anna Lhuillier used a short-pulse laser to generate attosecond pulses of light to study the dynamics of electrons in matter. According to our Theory of singularities of the type synthesizing, its action corresponds to singularity $\uparrow I \downarrow_h^q$, which allows one to reach the upper level of subtle energies to manipulate lower levels. In April 2023 [1], we proposed using a short-pulse laser to achieve the desired goals by a directly parallel neural network. We then proposed the fundamental development of this directly parallel neural network. In the articles [1]-[13] new mathematical structures and operators are constructed through one action - “containment”. Here, the construction of new mathematical structures and operators is carried out with generalization to any actions. The significance of our articles is in the formation of the presumptive mathematical structure of subtle energies, this is being done for the first time in science, and the presumptive classification of the mathematical structures of subtle energies for the first time. The experiments of the 2022 Nobel laureates Asle Ahlen, John Clauser, Anton Zeilinger and the experiments in chemistry Nazhipa Valitov eloquently demonstrate that we are right and that these studies are necessary. Be that as it may, we created classes of new mathematical structures, new mathematical singularities, i.e., made a contribution to the development of mathematics. Singularities as elements.

Degeneration of the usual can lead to singularities. For example:

1) equivalence of non-equivalent ones can lead to singularities

2) Q^{-1}-action to Q-object ones can lead to singularities

3) identification ones can lead to singularities

4) by exclusion of equality

5) etc.

Main classes of singularities.

1) singularities of synthesis. For example: self, |||, ↑|↓ etc

2) singularities of disintegration (collapse). For example: oself, $(|||)^{-1}$, $(↑|↓)^{-1}$ etc

3) singularities of disintegration&synthesis. For example: self-oself, etc

4) subjects of synthesis singularities

5) subjects of disintegration singularities

6) subjects of disintegration&synthesis singularities

7) etc.

Remark G.1. Subjects of singularities must be singularities of a higher level than their objects. Here under the subject of singularities A we mean that singularity which can apply the singularities of A to itself, create singularities of A, manipulate them, in particular, control them.

Remark G.2. It is singularities of class 3) that can “create” types of subtle energy similar to the subtle energies of living organisms, because represent a synthesis of opposite singularities, which act in them as poles that create corresponding fields.

Types of singularities.

Let us denote the identification operation by |||, the dynamic identification operator by ${}^{D}_{C}It^{A}_{B}$ or $\begin{array}{ccc} D & & Subject\ of\ ||| \\ C & \text{SIprt} & A \\ Subject\ of\ ||| & & B \end{array}$, where $A|||B\&C(|||)^{-1}D$ simultaneously.

Types of singularities associated with identification:

a) by |||, b) by $\begin{array}{ccc} (|||)^{-1} & & Subject\ of\ ||| \\ (|||)^{-1} & \text{SIprt} & ||| \\ Subject\ of\ (|||)^{-1} & & ||| \end{array}$, (designation d1|||), c) by $\begin{array}{ccc} D & & Subject\ of\ ||| \\ C & \text{SIprt} & A \\ Subject\ of\ ||| & & B \end{array}$, (designation d2|||), d) (|||) ||| (|||),

e) (|||) ||| (↑|↓), g) (|||) ||| (=), q) (|||) ||| (>), k) (|||) ||| (∪), l)(|||) ||| ($\frac{d}{dx}$),

h)$Srt_x^{|||,|||,|||} = \begin{matrix}(|||) - (|||) \\ ||| \\ (|||)\end{matrix}$, $Srt_x^{|||,|||,|||,|||} = \begin{matrix}(|||) - (|||) \\ ||| \\ (|||) - (|||)\end{matrix}$ etc.

$\begin{matrix}A - B \\ ||| \\ C\end{matrix}$ gives access to the upper level of A, B, C (objects, actions or subjects) by the 3-connection, the transition or replacement from one to another at the top level or vice versa, $\begin{matrix}A(|||) - (|||)B \\ ||| \\ C(|||) - (|||)D\end{matrix}$ gives access to the upper level of A, B, C, D (objects, actions or subjects) by the 4-connection, the transition or replacement from one to another at the top level or vice versa etc.

A|||B gives access to the upper level of A, B (objects, actions or subjects), the transition or replacement A to B at the top level or vice versa.

Examples of singularities associated with identification:

1. $|||_A$ = A(|||), where A is any operator,
2. s($\begin{matrix} rot(\dot{\_}) & & -\frac{\partial(\bar{\cdot})}{\partial t} \\ & X & \\ rot(\dot{\_}) & & -\gamma + \frac{\partial(\bar{\cdot})}{\partial t}\end{matrix}$)elf is designation of Maxwell equations identification,
3. os($\begin{matrix} rot(\dot{\_}) & & -\frac{\partial(\bar{\cdot})}{\partial t} \\ & X & \\ rot(\dot{\_}) & & -\gamma + \frac{\partial(\bar{\cdot})}{\partial t}\end{matrix}$)elf is designation of Maxwell equations disidentification,
4. os($Q|||R$)elf- s($A|||B$)elf is designation of the identification A B and disidentification simultaneously,
5. s($Q|||R$)elf = $Q|||R$
6. s($Q|||Q$)elf =$Q|||Q$
7. os($Q|||R$)elf = Q $(|||)^{-1}$R
8. os($Q|||Q$)elf = Q $(|||)^{-1}$Q
9. $oself|||self$
10. oself($oself|||self$) - self($oself|||self$)
11. os€elf($os\mu elf|||s\alpha elf$) - s£elf($os\lambda elf|||s\beta elf$)

12. identification through inequality: $os(Q(|||\geq)R)elf - s(A(|||\geq)B)elf$
13. $s(Q(|||\geq)R)elf$
14. $os(Q(|||\geq)R)elf$
15. $oself(|||\geq)self$
16. $oself(oself(|||\geq)self) - self(oself(|||\geq)self)$
17. $os\pounds elf(os\mu elf(|||\geq)s\Omega elf) - s\pounds elf(os\beta elf(|||\geq)s\alpha elf)$
18. $os(Q(|||\leq)R)elf - s(A(|||\leq)B)elf$
19. $s(Q(|||\leq)R)elf$
20. $os(Q(|||\leq)R)elf$
21. $oself(|||\leq)self$
22. $oself(oself(|||\leq)self) - self(oself(|||\leq)self)$
23. $os\pounds elf(os\mu elf(|||\leq)s\beta elf) - s\pounds elf(os\delta elf(|||\leq)s\lambda elf)$
24. $os(Q(|||<)R)elf - s(A(|||<)B)elf$
25. $os(Q(|||<)R)elf - s(A(|||>)B)elf$
26. s(Q (|||<) R)elf
27. os(Q (|||<) R)elf
28. oself (|||<) self
29. oself(oself(|||<) self) - self(oself (|||<) self)
30. $os\pounds elf(os\Omega elf(|||<)s\mu elf) - s\Delta elf(os\alpha elf(|||<)s\beta elf)$
31. os(Q(|||>) R)elf- s(A(|||>) B)elf
32. s(Q(|||>) R)elf
33. os(Q(|||>) R)elf
34. oself(|||>) self
35. oself(oself(|||>) self) - self(oself(|||>) self)
36. $os\pounds elf(os\mu elf(|||>)s\beta elf) - s\pounds elf(os\alpha elf(|||>)s\Omega elf)$
37. os(Q (|||<) R)elf- s(A (|||>) B)elf
38. os(Q (|||>) R)elf- s(A (|||<) B)elf
39. $os(Q(|||<)R)elf - s(A(\geq)B)elf$
40. $os(Q|||\geq R)elf - s(A(|||>)B)elf$
41. $os(Q(|||>)R)elf - s(A|||\leq B)elf$

42. $os(Q|||\leq R)elf - s(A(||| >)B)elf$

43. $oself(oself|||\geq self) - self(oself(||| <)self$

44. $os£elf(os\mu elf(||| <)s\ betaelf) - s£elf(os\alpha elf|||\leq s\Omega elf)$

45. $oself(oself(|||\geq)self) - self(oself(||| >)self)$

46. $os£elf(os\mu elf(||| >)s\alpha elf) - s£elf(os\Delta elf(|||\geq)s\Omega elf)$

47. $oself(oself(|||\geq)self) - self(oself(||| <)self$

48. $os£elf(os\mu elf(||| <)s\lambda elf) - s£elf(os\beta elf(|||\leq)s\alpha elf)$

49. $os£elf(os\mu elf(||| >)s\Omega elf) - s£elf(os\alpha elf(|||\leq)s\beta elf)$ etc.

4) - 49) are generalized by replacing ||| with $|||_A$ in them (watch 1)).

One can try to study and use the identification of any real process in any B.

It's allowed to add: A|||B + C|||B → (A+C) |||B, but distributivity may not take place, in the general case (A+C) |||B doesnt give A|||B + C|||B.

Remark G.3. Possible manifestations of A|||B: 1) A, 2) B, 3) the equation A = B, 4) g(A, B) in a scalar structure, 5) A → B, 6) etc. For example, one of possible manifestations of type singularity A|||2A for DNA this is DNA division: DNA → 2 DNA. One can also study all kinds of partial manifestations of A|||B. Exits to the upper level and its manifestations can be very different. In particular, A|||A = Srt_A^A, those. there may be different exits to the same singularities. The top level also has its own hierarchy: in particular the identification of all its singularities takes it to a higher level, as does the containment of them within oneself, as well as the rest of their actions with oneself. Singularities can be potential, for example, in the form of a potential singular connection, the activating singularity makes it active. Let us denote the upper level of A by $\overline{A}$, the upper level of P by $\overline{P}$. Then singularity $\overline{A} \to \overline{P}$ is the setting for the transformation of A into P. Partial replacements (transitions) from identification at the upper level as manifestations are possible (in particular, partial replacements of identified characteristics). The singularities that we present can be a 2-interpretation of the corresponding natural ones and through appropriate experimental actions can lead to them. We can conditionally designate our theoretical singularities by $Sprt_A^w$, where w refers to the structures of the space of theoretical science, in particular. mathematics, A refers to objects in these structures, we can conditionally denote natural singularities, corresponding to $Sprt_A^w$ by $Sprt_X^O$, and by X we mean the structures of natural space and by O objects in these structures. Next, we can try to study following singularities $Sprt_A^w$& $Sprt_X^O$: 1) $Sprt_A^w|||\ Sprt_X^O$, 2) $Sprt_{Sprt_X^O}^{Sprt_A^w}$, 3) $Sprt_{Sprt_X^O}^{Sprt_X^O}$, 4) $Sprt_{Sprt_A^w}^{Sprt_X^O}$, 5) $Sprt_{Sprt_A^w}^{Sprt_A^w}$,

6) ${}^{Sprt_X^O}_{Sprt_A^w}Sprt^{Sprt_A^w}_{Sprt_X^O}$, 7) ${}^{Sprt_A^w}_{Sprt_X^O}Sprt^{Sprt_X^O}_{Sprt_A^w}$, 8) ${}^{Sprt_A^w}_{Sprt_X^O}Sprt$, 9) ${}^{Sprt_A^w}_{Sprt_A^w}Sprt^{Sprt_A^w}_{Sprt_A^w}$, ,

10) ${}^{Sprt_X^O}_{Sprt_X^O}Sprt^{Sprt_X^O}_{Sprt_X^O}$, 11) ${}^{Sprt_X^O}_{Sprt_X^O}Sprt$, 12) ${}^{Sprt_A^w}_{Sprt_A^w}Sprt$, etc.

The computer already represents a $Sprt_A^w$-(self-object) with an assembly point in the form of a microprocessor. In particular, a computer for controlling various processes through equations already represents a self-object with an assembly point in the form of a microprocessor, which selects the required set of equations to control the required processes.

Remark G.4. A three-level model is possible: $\begin{pmatrix} A|||B \\ Sprt_B^A \\ A,\ B \end{pmatrix}$, where the transition from the lower level: $A,\ B$ to the middle: $Sprt_B^A$ one is possible through the equation $A =\ B$, the transition from the middle to the upper level is possible through $Sprt_A^A$ or $Sprt_B^B$, the transition from the upper to the middle level is possible through $A|||A$ or $B|||B$, the transition from the middle level to the lower one is possible through ${}^{Srt_B^A}_{A}Sprt$.

Remark G.4.1. Actually $A|||B$ there is an upper level capacity for A, B.

Remark G.5. In any structure, connection, task related to the search for the unknown, it is possible by identification to obtain for the upper level of this unknown a singularity that interprets the upper level corresponding to its real image.

Remark G.6. All mathematics has so far been reduced to operations with numbers through structures, just like programming (i.e. indirectly through numbers). Here we directly work with structures without taking into account their numerical content.

Remark G.6.1. Consider $|||$-lim$(\Delta y(|||)^{-1}\Delta x) = ||| - \frac{dy}{dx}$

$|||$-lim$(\sum_{i=1}^{\infty} y_i \Delta x_i = ||| - \int ydx$.

Others types of singularities: a) 2 $\rightarrow$ 1, b) 2 $\rightarrow$ 3, c) A $\rightarrow$ B for any A, B etc.

Examples:

1. selfA is Srt_A^A
2. oselfA is ${}^A_ASprt$
3. (oselfA- selfA) is ${}^A_ASprt_A^A$
4. black hole
5. self(black hole)
6. oself(black hole)

7. (oself(black hole)-self(black hole))

8. s2elfA is Srt_{A}^{selfA}

9. $s\frac{1}{2}$elf is Srt_{selfA}^{A}

10. sqelfA is $Srt_{q(A)}^{A}$

11. os2elfA is ${}_{(A,A)}^{A}Sprt$

12. os$\frac{1}{2}$elf is ${}_{A}^{selfA}Sprt$

13. osq()elfA is ${}_{q(A)}^{A}Sprt$, q-any operator,

14. osNelfA is ${}_{(q_1,\ \ldots,\ q_N)}^{A}Sprt$, q_i= A, i= 1, . . . , N;

15. sq$_1$elfA- os$\begin{pmatrix} q_3() \\ q_2() \end{pmatrix}$elfA is ${}_{q_3(A)}^{q_2(A)}Sprt_{q_1(A)}^{A}$

16. os({ } →)elfA is ${}_{A}^{\{\ \}}Sprt$

17. $D_1 fQ = \begin{matrix} action\ Q \\ action\ Q) \\ action\ Q \end{matrix}^{-1}\text{Dprt} \begin{matrix} action\ Q \\ action\ Q \\ action\ Q \end{matrix}$, Q is any,

18. $D_2 fA; Q = \begin{matrix} action\ Q \\ (action\ Q) \\ A \end{matrix}^{-1}\text{Dprt} \begin{matrix} A \\ action\ Q \\ action\ Q \end{matrix}$, Q is any,

19. $D_3 fA; Q; B = \begin{matrix} B \\ (action\ Q) \\ A \end{matrix}^{-1}\text{Dprt} \begin{matrix} A \\ action\ Q \\ B \end{matrix}$, Q is any,

20. $D_4 fA; Q = \begin{matrix} A \\ (action\ Q) \\ A \end{matrix}^{-1}\text{Dprt} \begin{matrix} A \\ action\ Q \\ A \end{matrix}$, Q is any,

21. $D_5 fA; Q; a = \begin{matrix} a \\ (action\ Q) \\ strA \end{matrix}^{-1}\text{Dprt} \begin{matrix} strA \\ action\ Q \\ a \end{matrix}$, Q is any,

$a \subset A$ and structure of A acts Q to a and acts Q out from a simultaneously,

22. $D_6 fa; Q; A = \begin{matrix} StrA \\ (action\ Q) \\ a \end{matrix}^{-1}\text{Dprt} \begin{matrix} a \\ action\ Q \\ StrA \end{matrix}$, Q is any,

$a \subset A$ and acts Q to structure of A and acts Q out from structure of A simultaneously,

23. $\underset{B}{\overset{B}{(action\ Q)^{-1}}}\mathrm{Dprt}\ \underset{B}{\overset{A}{action\ Q}}$, Q is any,

24. $\mathrm{Dprt}\ \underset{A}{\overset{A}{action\ Q}}$,

25. $D_7 f A; Q; a = \mathrm{Dprt}\ \underset{a}{\overset{strA}{action\ Q}}$, Q is any,

$a \subset A$ and structure of A acts Q to a,

26. $D_8 f a; Q; A = \mathrm{Dprt}\ \underset{strA}{\overset{a}{action\ Q}}$, Q is any,

$a \subset A$ and acts Q to structure of A,

27. $\mathrm{Dprt}\ \underset{action\ Q}{\overset{action\ Q}{action\ Q}}$, Q is any,

28. $\mathrm{Dprt}\ \underset{action\ Q}{\overset{A}{action\ Q}}$, Q is any,

29. $\underset{D}{\overset{D}{(action\ Q)^{-1}}}\mathrm{Dprt}$, Q is any,

30. $D_9 f d; Q; D = \underset{d}{\overset{strD}{(action\ Q)^{-1}}}\mathrm{Dprt}$, Q is any,

$d \subset D$ and d acts Q out from structure of D,

31. $D_{10} f D; Q; d = \underset{strD}{\overset{d}{(action\ Q)^{-1}}}\mathrm{Dprt}$, Q is any,

$d \subset D$ and structure of D acts Q out from d,

32. $\underset{D}{\overset{action\ Q}{(action\ Q)^{-1}}}\mathrm{Dprt}$, Q is any,

33. $\underset{action\ Q}{\overset{action\ Q}{(action\ Q)^{-1}}}\mathrm{Dprt}$, Q is any,

34. $S2prt_A^A$= (selfA, selfA)

35. $SNprt_A^A$= (q_1, ..., q_N), q_i= selfA, i= 1, ..., N.

36. self($Sprt_B^A$)= $St_{Sprt_B^A}^{Sprt_B^A}$

37. Q($_A^A Sprt_A^A$), Q-any operator

38. SO([-1,1]) = S_∞^+ = $(\uparrow I \downarrow_{-1}^{1})$ corresponds to sin∞

39. SO([1,-1]) = S_∞^- = $(\downarrow I \uparrow_{-1}^{1})$ corresponds to sin(-∞)

40. SO([-∞, ∞]) = T_∞^+ = $(\uparrow I \downarrow_{-\infty}^{\infty})$ corresponds to tg∞

41. SO([$\infty, -\infty$]) = T_∞^- = $(\downarrow I \uparrow_{-\infty}^{\infty})$ corresponds to $tg(-\infty)$, don't confuse with values of these functions. Such elements can be summarized. For example: aS_∞^++bS_∞^-=(a-b)S_∞^+ = $(b-a)$ S_∞^-.

42. The operator $(Q_1S(A))^2$ increases self-level for A: it transforms Self-A= f_1SA to $self^2$A, $(Q_1S(A))^n$ $\rightarrow$ $self^n$A, $e^{Q_1S(A)}$ $\rightarrow$ $e^{self}A$.

43. Os(self())elfA = A

We will give only one interpretation of the following singularities out of many possible options:

1. $s\frac{3}{2}$elf is $Srt_{selfA}^{self^{\frac{3}{2}}A}$
2. os$\frac{3}{2}$elf is $_{(A,A,A)}^{selfA}Sprt$
3. $s\frac{m}{n}$elf
4. o$s\frac{m}{n}$elf
5. os$\frac{m}{n}$elf- s$\frac{k}{l}$elf
6. s((3→2))elf -
7. s((3→2)|||)elf
8. os((2→3))elf
9. os((2→3)|||)elf
10. os((m→n)|||)elf
11. s((→q)|||)elf
12. s((A→R) |||)elf
13. os((A→R) |||)elf
14. s((m→n) |||)elf
15. os((→q) |||)elf

16. s((m→n)|||)elf - os((m→n)|||)elf

17. (2→3) |||(3→2)

18. (2→3) ↑I↓ (3→2)

Let us introduce a measure of singularity μ_{si}, which sets the maximum level of singularity as 1, and the minimum, i.e., regularity, as 0. Then, in particular, for binary relations: μ_{si} = 1 for A|||B, $Sprt_x^{A,\ B}$ (x - point), μ_{si} = 2/3 for Drt $\begin{matrix} Q \\ Q \\ Q \end{matrix}$, μ_{si} = 1/2 for A|||A, $Sprt_A^A$

Singularities algebra.

You can define operations for the same type singularities and study their algebras. But our task for now is limited to considering certain types of singularities presumably corresponding to certain types of subtle energy.

Let us introduce the following notations: A*B= Sprt_B^A, A^2 = Self A =Srt_A^A, $A^{\frac{3}{2}}$ = Drt $\begin{matrix} A \\ A \\ A \end{matrix}$ = sel$f^{\frac{3}{2}}$ (A), A^3 = $Self^2$A, …,$A^{\frac{3n}{2}}$ = Drt $\begin{matrix} A^n \\ A^n \\ A^n \end{matrix}$ = sel$f^{\frac{3n}{2}}$ (A), A^{n+1} = $Self^n$A, sel$f^{min(n,m)}$(A) $Srt_{A^m}^{A^n}$ = sel$f^{\frac{n}{m}}$ (A), sel$f^{min(n,m,k)}$(A) Drt $\begin{matrix} A^n \\ A^m \\ A^k \end{matrix}$ = sel$f^{\frac{n+m+k}{2k}}$ (A), … etc.

Remark. $Self^2$A already represents a self-object with an assemblage point, i.e., living organism.

There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A + B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1 + A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

You can consider a more "hard" option: A*B=$PSprt_B^A$, where $PSprt_B^A$ – operator, containing A in every element of B, A^2 = PSelf A =$PSrt_A^A$, A^3 = $PSelf^2$A, …, A^{n+1} = $PSelf^n$A, PSel$f^{min(n,m)}$(A) $PSrt_{A^m}^{A^n}$ = PSel$f^{\frac{n}{m}}$ (A), …etc. There is no commutativity here: A*B B*A. We can consider operator functions: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \ldots$, etc.

Remark G.6.2. $PSelf^2$A already represents a self-object with an assemblage point, i.e., living organism.

Lets introduce $\sqrt{self}$ as the result of the decision of the equation Srt_x^x = self, that is x = $\sqrt{self}$, $\sqrt[3]{self}$ as the result of the decision of the equation Dprt $\begin{matrix} x \\ x \\ x \end{matrix}$ = self, that is x = $\sqrt[3]{self}$, $\sqrt[n]{self^m}$ as the result of the decision of the equation $x^{\frac{n}{m}}$ = self, $self^{\alpha}$ as the result of the decision of the equation $x^{\frac{1}{\alpha}}$ = self, where is any number, in particular, a negative number etc. The following equality is true:

$self^{-\alpha}(self^{\alpha}G)$ =$self^{\alpha}(self^{-\alpha}G)$ = G ($*_G$),

you can, for example, specify $self^{-\alpha}$ by definition through ($*_G$).

The following equality is true:

$self^{\beta}(self^{\alpha}G)$ $=self^{\alpha}(self^{\beta}G)$ $=$ $self^{\propto+\beta}G$ $\forall$ $\propto, \beta$. In this way one can introduce self-level space. The following operation may take place: q↑↓q + ${}_{A}^{A}Sprt_{A}^{A} = {}_{A+q}^{A+q}Sprt_{A+q}^{A+q}$.

Similar algebras can be constructed for other singularities, in particular, for oself, ↑↓, , partial self-type singularities, etc. For example A|||(|||)B = A $(|||)^{2}$B, $(|||)^{\propto}$ as the result of the decision of the equation $x^{\frac{1}{\propto}} = |||$. $(|||)^{\propto}$ * $(|||)^{\frac{1}{\propto}} = (|||)^{\frac{1}{\propto}}$ * $(|||)^{\propto} = |||$ etc.

$\alpha - -singularities:$

1. $s\alpha elf$ is $self^{\alpha}$
2. $os\alpha elf$ is (o$self)^{\alpha}$
3. $os\alpha elf$- $s\mu elf$
4. qselfA is $Srt_{q(A)}^{A}$
5. α - black hole
6. α- self- black hole
7. α - oself- black hole
8. α–(oself-self)-black hole
9. α- os({ } $\rightarrow$)elf
10. α- oμs({ } $\rightarrow$)elf
11. $\alpha - ||| = (|||)^{\alpha}$
12. α - (os(Q |||R)elf- s(A||| B)elf)
13. α- s($Q|||R$)elf
14. α- os($Q|||R$)elf
15. α - $(|||\geq) = (|||\geq)^{\alpha}$
16. α -$(os(Q(|||\leq)R)elf - s(A(|||\leq)B)elf)$
17. α - $s(Q(|||)R)elf$
18. α -$os(Q(|||\geq)R)elf$
19. α - $(|||\leq) = (|||\leq)^{\alpha}$
20. α -$(os(Q(|||\leq)R)elf - s(A(|||\leq)B)elf)$
21. α - $s(Q(|||\leq)R)elf$
22. α -$os(Q(|||\leq)R)elf$

23. α- (<) = $(<)^{\alpha}$
24. α - (os(Q(<) R)elf- s(A(<) B)elf)
25. α - s(Q(<) R)elf
26. α - os(Q(<) R)elf
27. α - (>) = $(>)^{\alpha}$
28. α - (os(Q(>) R)elf- s(A(>) B)elf)
29. α - s(Q(>) R)elf
30. α - os(Q(>) R)elf
31. α - (os(Q(<) R)elf- s(A(>) B)elf)
32. α - (os(Q(>) R)elf- s(A(<) B)elf)
33. α- (os(Q(<) R)elf- s(A() B)elf)
34. α - (os(Q() R)elf- s(A(<) B)elf)
35. α- (os(Q(<) R)elf- s(A() B)elf)
36. α - (os(Q() R)elf- s(A(<) B)elf)
37. α - (os(Q() R)elf- s(A() B)elf)
38. α - (os(Q() R)elf- s(A() B)elf)

Structural singularities:

1. Cos2elfA is ${}^{A}_{(A,A)}Cprt$
2. Cs2elfA is Crt^{selfA}_{A}
3. C$s\frac{1}{2}$elf is Crt^{A}_{selfA}
4. CsqelfA is $Crt^{A}_{q(A)}$
5. Cosq()elfA is ${}^{A}_{q(A)}Cprt$, q-any operator,
6. CosNelfA is ${}^{A}_{(q_1,\ \ldots,\ q_N)}Cprt$, q_i= A, i= 1, …, N;
7. (CoselfA- CselfA) is ${}^{A}_{A}Cprt^{A}_{A}$
8. (Cs$q_1$elfA- Cos$\begin{pmatrix} q_3() \\ q_2() \end{pmatrix}$elfA) is ${}^{q_2(A)}_{q_3(A)}Cprt^{A}_{q_1(A)}$

Other types of singular operations:

1. self$\cup$ is self-unification sign,
2. self$\cap$ is self-intersection sign

3. $\begin{array}{c}(\cup)^{-1}\\ (\cup)^{-1}\\ (\cup)^{-1}\end{array}\mathrm{Dprt}\begin{array}{c}\cup\\ \cup\\ \cup\end{array}$ is Dprt_1-unification sign

4. $\begin{array}{c}\cup\\ \cup\\ \cup\end{array}\mathrm{Dprt}\begin{array}{c}\cup\\ \cup\\ \cup\end{array}$ is Dprt_2-unification sign

5. $\begin{array}{c}\cup\\ (\cup)^{-1}\\ \cup\end{array}\mathrm{Dprt}\begin{array}{c}\cup\\ \cup\\ \cup\end{array}$ is Dprt-unification sign

6. $\begin{array}{c}(\cup)^{-1}\\ (\cup)^{-1}\\ (\cup)^{-1}\end{array}\mathrm{Dprt}\begin{array}{c}(\cup)^{-1}\\ (\cup)^{-1}\\ (\cup)^{-1}\end{array}$ is Dprt_3-unification sign

7. self$\wedge$ is self-logical multiplication sign

8. selfV is self-logical addition sign

9. $\begin{array}{c}V\\ V\\ V\end{array}\mathrm{Dprt}\begin{array}{c}\wedge\\ \wedge\\ \wedge\end{array}$ is Dprt_1- logical sign

10. $\begin{array}{c}\wedge\\ \wedge\\ \wedge\end{array}\mathrm{Dprt}\begin{array}{c}\wedge\\ \wedge\\ \wedge\end{array}$ is Dprt_2-logical multiplication sign

11. $\begin{array}{c}V\\ V\\ V\end{array}\mathrm{Dprt}\begin{array}{c}V\\ V\\ V\end{array}$ is Dprt_2-logical addition sign

In fact, the classification of singularities determines the classification of outputs from our 2-world to other levels in the 2-interpretation. Science is 2-interpretation and 2-classification. In principle, there can be nothing else, and that's not bad either.

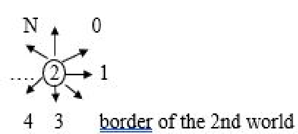


Fig. 0.1 (2$\rightarrow$ i)-connections, I = 0, 1, 2, ..., N, ...

Remark G.7. Function δ is actually a functional of singularity $\uparrow I \downarrow_0^\infty$.

Remark G.8. All singularities can act as program operators in a special type of programming - singular programming.

Remark G.9. The identification of all singularities can be considered as the new higher-level singularity.

Remark G.10. You can set norms for the same type singularities and study their topologies.

Singularities within singularities.

Increasing the level of singularity in the internal singular parts in singularities, in particular, for microwave alternating current, corresponding to the upper level through the identification of the corresponding Maxwell equations.

Partial self-type singularities.

Consider a third type of capacity in itself. For example, based on $Srt_w^{\{g\}}$, where $\{g\} = (g_1, g_2, ..., g_n)$, i.e. n - elements at one point, we can consider the capacity $S_3 f$ in itself with m elements from $\{g\}$, m<n, which is formed according to the forms from (1.1) – (1.4) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc. (1.3), (1.4) are represented through the usual 2-bond. Science is the discipline of 2-connections, since everything in science is carried out through 2-connected logic, quantum logic is also a projection of 3-connected logic onto 2-connected logic. (1.3.1) - (1.3.4) schematically interpret the formation of capacity in itself through a pseudo 3-connected form with a 2-connected form

Appendix 1

Remark G.11. Hypothesis G.1: Considering (1.7.1), we can assume that access to the upper level can be achieved through $Sprt_{Sprt_{A_0}^{A_0}}^{Sprt_{A_0}^{A_0}}$, and access to the middle level can be achieved through $Sprt_{A_0+Sprt_{A_0}^{A_0}}^{A_0} + Sprt_{A_0}^{A_0+Sprt_{A_0}^{A_0}}$.

Remark G.12. The same in Hypothesis 1.7 applies to real problems: identifying the conditions of the problems with their request will give the singularity of the upper level of the problems. It is clear that the upper-level singularity corresponds not only to one object, process, task, etc., but also to a whole class of those corresponding to this singularity. The upper level is the identification of everything that is there, by definition. The upper level of objectivity (i.e., the boundary of objectivity) is the identification of all objects. The necessary manifestations of the upper level on the lower ones can be carried out through the appropriate settings in the upper level. Then it is quite possible, through the use of dynamic programming in the required activation (via microwave with minimum amplitude and maximum frequency, ultraviolet), to transfer S_{mnSt} to the boundary of objectivity to perform the desired task, in particular, for the necessary transformation, the necessary action etc. Using the work of 2023 Nobel Prize winners in physics Ferenc Kraus and his colleagues Pierre Agostini and Anna Lhuillier, their experimental methods of generating attosecond pulses of light to study the dynamics of electrons in matter, we will attempt to carry out the experimental part of our work as soon as we receive a specially equipped laboratory and funding. It is clear that the upper-level singularity corresponds not only to one object, process, task, etc., but also to a whole class of those corresponding to this singularity. Therefore, in particular, a transition to other elements of this class is possible.

Hypothesis G.2: The upper level is the identification of everything that is there, by definition.

The level that is below the lower level of objects is the level of their ordinary emanations from them - the level of ordinary energies. In particular for mathematical objects - equations, these are their solutions. For any mathematical object, one can give definitions of its singular generalizations, and not only for mathematical ones, in particular, all real objects and processes have an upper (singular) level.

Remark G.13. So far we have used all sorts of (algebraic) structures to interpret some simplified processes and their results. Unfortunately, sometimes their great complexity is not entirely justified, so here we will try to interpret them through geometric constructions.

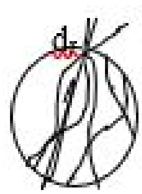


Fig. 0.2

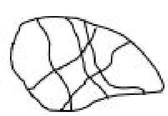


Fig. 0.3

Fig. 0.2 we will try to interpret the energy of a living organism. The ellipse represents a cross-section of the energy cocoon of a living organism, d_r is the assembly point on it, curved lines inside and outside are the energy fibers of the Universe, r-the position of the assemblage point d_r on the energy cocoon of a living organism.

Fig. 0.3 we will try to interpret the energy of a non-living object. Shape border represents a cross-section of the energy cocoon of a non-living object, the curved lines are the energy fibers of the Universe enclosed in the cocoon of the object. In particular, string theory considers an elementary particle as a "string," although in our opinion it is more natural to consider an elementary particle as a manifestation of a "string," an energy fiber of the Universe. Energy cocoons are not structures of our object world; they are structures of energy space, the manifestation of which is our object world. For example, it is quite possible to try to interpret the nucleus of an atom as a manifestation of the intersection of such fibers, which are actually incubators of protons and neutrons that make up the nucleus of an atom, and the nucleus of a cell of a living organism as a manifestation of the intersection of such fibers, which are actually incubators of chromosomes with DNA. For example, (Fig. 0.4) it is to try to interpret an atom ${}^{4}_{2}He$ as the cocoon with two layers of self-capacity in the form of orbitals that surround the intersection of energy fibers - incubators of protons and neutrons.

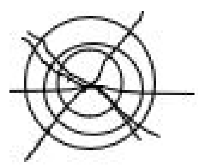


Fig. 0.4

Lets consider the vector of energy levels of a non-living object A:

$$\begin{pmatrix} subtle\ energy\ of\ the\ upper\ level\ object\ A(decignation-\ \overbrace{A}\ ) \\ subtle\ energy\ of\ a\ mid-level\ object\ A(decignation-\ \overline{A}) \\ the\ raw\ energy\ of\ an\ object\ A-\ its\ objectivity(decignation-A) \\ ordinary\ energy\ exhibited\ by\ an\ object\ A(decignation-\underline{A} \end{pmatrix}$$

Remark G.14.. Our object world is a manifestation of the space of self-capacities (the space of subtle energies), the space of self-capacities is a manifestation of the space of self-containments etc. The self-capacity is the element of the SAMO level, but can partially manifest the corresponding parts of its content at lower levels, for example object level, in the corresponding "structures". For example, $\overbrace{A}$ is capacity for $\overline{A}$, $\overline{A}$ is capacity for A, A is capacity for $\underline{A}$. The magnetic field is able to pass through ordinary objects and is therefore a mid-level element.
Applications to physics.
In our opinion, one of the laws of physics should be the law of induction: any change (motion) induces a change (field) "perpendicular to it".This especially applies to flow. It's just that the physical characteristics of "perpendicular"

fields during ordinary not very large changes (movements) are so small . In particular, an electric current induces a magnetic field “perpendicular to it” and vice versa, a fluid flow induces a vortex field “perpendicular" to it. Only the specifics inherent in each will differ. Induction, as it were, balances (compensates) the movement. This is the result of resistance to the singularity of "emptiness" (order). The next law of physics should be the law of "clotting": obtaining potential energy by "clotting"(up to ||| (identification)) the elements of space-time, objects. Moreover, e.g., auniform movement in a straight line does not give potential energy, and uniform movement in a circle gives potential energy through centripetal acceleration a = v^2/R. There is there "clotting" the element of space - a direct line to circle. For example, in the case R $\rightarrow$0, v^2=d*R we get one of the options of self-energy. The operators

$\begin{matrix} C \\ (actionQ) \end{matrix}^{-1}$ SDS $\begin{matrix} Subject of Q \\ A \\ actionQ \\ B \end{matrix}$ considered above are examples of such operators of "clotting". Electron orbital in atoms is also "clotting". In particular, "clotting" allows to get some options of self-energy. Another option for obtaining self-energy through manifestations of higher levels. For example, A ||| B can give a manifestation of the species self(A) = A|||A. Moreover self(A) ||| self(B) = (A |||A) |||(B|||B) = A|||B.

The next law of physics should be the law of the evolution of energies: the first stage of the evolution of energies - to "clotting", the second stage of the evolution of energies – to self-energies, the third stage of the evolution of energies - up to ||| (identification) of energies; and also the law of the involution (manifestations) of energies: from A |||B to A |||A = self(A) and B |||B = self(B) and further to A and B.

Remark G.14.1. Any self -use can be used to design pseudo -proof energies if the amplitude of the action is inversely proportional to the square of the frequency of action.

Remark G.14.2. In strings theory, to more correctly accept self -action as a string (in private., self -containment), which generates this self -object - an elementary particle.

Remark G.14.3. To construct pseudo-living energies, it is necessary to take or form from energy A with an amplitude proportional (to the square of the frequency of energy A) self-energy. And then through activation SmnSprt [2], [18] in order to obtain the necessary pseudo-living energy from this self-energy, do this.

Remark G.14.4. Any created self of object A creates the possibility of using a double from self(A), moreover this double of object self(A) is actually formed only through the upper level of self(A) and is not directly connected with the lower level of A, i.e., with the level of its objectivity. By manipulating the double, it is possible to perform all sorts of actions that are not available to the original due to the "absence" of the objectivity inherent in the original. All this follows from the nature of self(A), since self(A) is a structure containing A twice: the original and, as it were, a virtual copy of the original (the potency of the double). All this applies to any: both to the natural and to the theoretical, in particular, to the self -equation, the self −(boundary value) problem; the implementation will only be its own specific. The structure of the self-object A can be interpreted as the presence of the original and its connections with itself, which we will call the virtual object A or the virtual part of the self-object A (self(A) = (A, virtual object A)). Then the double of the self-object A can be interpreted as ((virtual object A), A), i.e., in the double, the virtual object A is used as the original, and the object A itself becomes its virtual part. As a result, the integrity of object A appears, then the opportunity opens up through self-containment (self $^{\frac{3}{2}}$A) to reach the next upper level (self ^{2}(A), |||). And a more complex generalized version of the self-object A: self_{R}(A) = (A, R(virtual object A))), where R is structure in which is generated self_{R}(A).)). Then the double of self_{R}(A) can be interpreted as ((virtual object A), R(A)).

Remark G.14.5.The 2022 Nobel laureates' experiments with the spin of bound electrons show the need for parallel physics, which specializes in studying the parallelism of processes.

Fuzzy singularities of the type synthesizing [13].

Similar, it is possible to consider the singularities by specifying fuzziness measures for the elements of their structures based on the above singularities. As an example, let us consider fuzzy partial singularities of the type synthesizing: consider a third type of fuzzy fcapacity in itself. For example, based on $\text{ffSprt}\mu^{\widetilde{x}}_{q}$, where $\widetilde{x}$=($x_1\mu_{\widetilde{x}}(x_1)$, $x_2\mu_{\widetilde{x}}(x_2)$,…,$x_n\mu_{\widetilde{x}}(x_n)$) i.e. n - elements at one point, we can consider the fuzzy fcapacity ffS_3fţ in itself with m elements from $\widetilde{x}$, m<n, which is formed according to the form (1.1), that is, the structure $\text{ffSprt}\mu^{\widetilde{x}}_{q}$ contains only m elements. Form (1.1) can be generalized into the following forms (1.1.1) – (1.1.5).

Fuzzy fcapacities in themselves of the third type can be formed for any other structure, not necessarily ffSprt, only by necessarily reducing the number of elements in the structure, in particular, using forms from (1.1.1) – (1.4).

Structures more complex than ffS_3f can be introduced. For example, through a forms that generalizes (1.3), where A is fuzzy compressed (fuzzy fits) in C in the fuzzy compression fuzzy structure B in C (i.e. in the fuzzy structure

$$\mathrm{ffSprt}\,\underset{C}{\overset{B}{\mu}};$$ or

(1.3.1) - (1.3.4) or through the more general form (1.4) and corresponding generalizations of (1.4) on (1.3.1) - (1.3.4), etc. (1.3.1) - (1.3.4) schematically interpret the fuzzy formation of fuzzy capacity in itself through a pseudo 3-connected form with a 2-connected form.

Remark G.15. Fuzzy self, in particular, according to a fuzzy form- fuzzy analogue of the form of type (1.1): (2. 1*).

For example

1) fuzzy forming from element with fuzziness in the form {2,1}: (1 , (2 ,1))
2) fuzzy forming from element in the form {2,1} with fuzziness : (1, (2 ,1))

1. fuzzy formation of partial self in the form (G.1) with fuzziness : (1, (2 ,1))
2. It is also possible to generalize the other remaining forms (1.1) – (1.4) to fuzzy forms
3. etc.

Appendix 2

We consider measure for the level of self - LS, the values of which are transformed into a segment [0,1] through the function arctg: lS=$\frac{2}{\pi}arctgLS$.

0 corresponds to "no self", and 1 corresponds to the "self$^{\infty}$".

Let an object A from the class of objects B. The measure lS (A) is a unique real function defined on B that satisfies three conditions (axioms of selfhood)

1. $lS(A) \geq 0$ for any object A from B
2. lS (A) = 1 for self$^{\infty}$ object A
3. lS ($A_1 \cup$ $A_2 \cup \ldots$) = lS (A_1) + lS (A_2) + ... for any finite or infinite sequence of pairwise inconsistent objects A_1, A_2, ...

From axioms 1), 2), 3) it follows that $0 \leq lS(A) \leq 1$, in particular, if O is an impossible object, then lS (O) =0. It is important to note that the equalities lS (A) = 1 or lS (A) = 0 do not imply whether A is an self object or an impossible object. Let us introduce axiom 4: the measure for dependent A, B lS(A*B) = lS(A)* lS(B/A)= lS (B)* lS (A/B), where lS (B/A)- conditional self of B at A, lS (A/B) - conditional self of A at B. The measure lS(A/B) undefined if lS(A/B).

Then for joint A, B: lS (A+B)= lS (A)+ lS (B)- lS (A*B)+ lS(D), D-

self from A*B, lSS(x)- the value for (self)^2 from x; for dependent A, B: lS(A*B) = lS(A)* lS(B/A)= lS (B)* lS (A/B), where lS (B/A)- conditional self of B at A, lS (A/B)- conditional self of A at B. Adding measures of self of inconsistent A,B: lS(A+B)= lS(A)+ lS(B). The formula of complete self: lS(A)=$\sum_{k=1}^{n} lS\,(B_k) * lS(A/B_k)$, B_1,B_2,..,B_n-full basis group: $\sum_{k=1}^{n} lS\,(B_k)$=1(“self”).

Sprt-lS for set A={A_1,A_2,. . . ,A_n}: $Sprt_w^{\{\ lS(A_1),\ lS(A_2\ ),\ldots,lS(A_n)\}}$[2], $Sprt_w^{\{\ lS(A_1),\ lS(A_2\ ),\ldots,lS(A_n)\}}$- Sprt- lS for it. It is possible to consider the self S_3A [2] with m elements from A, at m<n, which is formed by the form (1.1) [2], that is, only m elements from A are located in the structure $Sprt_w^A$. The same for self $S_3\{\ lS(A_1), lS(A_2), \ldots, lS(A_n)$: $Sprt_w^{\{\ lS(A1),lS(A2),\ldots,lS(An)\}}$.

Dynamical containments of oneself of the third type can be formed for any other structure, not necessarily Sprt, only through the obligatory reduction in the number of continual elements in the structure. In particular, using the forms from (1.1.1) – (1.4) [2].

Structures more complex than $S_3\mu S$(t) can be introduced.

Consider the following characteristics:

LM(X)= $\sum_{i=1}^{n} x_i ls_i$, ls_i = lS(X=x_i), i=1,2, . . . , n, x_i – a state of X at time t_i; properties:

1. LM(C) = C, C – const
2. LM(CX) = CLM(X)
3. LM(X_1+ X_2) = LM(X_1) + LM(X_2)
4. LM(X_1* X_2) = LM(X_1) * LM(X_2)

LD(X) = LM((X- LM(X))^2), LD(X) = LM(X^2)- (LM(X))^2

Properties:

1. LD(C) = 0
2. LD(CX) = CLD(X)
3. LD(X_1+ X_2) = LD(X_1) + LD(X_2)

L$\sigma(X) = \sqrt{LD(X)}$

L$\alpha_{\mathrm{k}}(X) = LM(X^{\mathrm{k}}) = \sum_{i=1}^{n} x_i^k ls_i$

$L\mu_k$(X) = LM((X- LM(X))$^{\mathrm{k}}$).

Function of self for estimating the unknown parameter of the state distribution X: a) in a discrete case L(x_1,x_2, . . . , x_n, α) = $ls(x_1, \alpha) ls(x_2, \alpha) \ldots ls(x_{\mathrm{n}}, \alpha)$, where ls($x_k$, α) = $lS(X = x_{\mathrm{k}})$

b) in an absolutely continuous case L(x_1,x_2, $\ldots$, x_n, α) $=$ $f(x_1,\alpha)f(x_2,\alpha)\ldots f(x_n,\alpha), wheref(x,\alpha)$ – a density of the state distribution X.

Ltropy of the self-levels distribution for x

LH(x) = {$\begin{array}{c} LM\left(log_2\frac{1}{lS(x)}\right) = -\sum_x lS(x)log_2 lS(x), \\ LM(log_2\frac{1}{\varphi lS(x)} = \ -\int_{-\infty}^{\infty}\varphi lS(x)log_2\varphi lS(x)dx \end{array}$

LH(x) is the measure of no self for value x.

Remark G.16. Similarly, you can set measure for the level of self-type for other partial self-type singularities, for example, partial identification - $l_{|||}$ of identification.

Appendix 3

We consider measure of self: $\mu S(Q)$, which gives a numerical value for the self of the Q from the interval [0,1], where 0 corresponds to "no self", and 1 corresponds to the "self". Let an object A from the class of objects B. The measure $\mu S(A)$ is a unique real function defined on B that satisfies three conditions (axioms of selfhood)

1. $\mu S(A)$≥0 for any object A from B
2. $\mu S(A)$ = 1 for self-object A
3. μS(A_1∪ A_2∪$\ldots$) = μS (A_1) + μS (A_2) + $\ldots$ for any finite or infinite sequence of pairwise inconsistent objects A_1, A_2, $\ldots$

From axioms 1), 2), 3) it follows that 0≤$\mu S(A)$≤1, in particular, if O is an impossible object, then μS(O) =0. It is important to note that the equalities μS(A) = 1 or μS(A) = 0 do not imply whether A is self-object or an impossible object. Let us introduce axiom 4: the measure for dependent A, B μS(A*B) = μS(A)* μS(B/A) = μS(B)* μS(A/B), where μS(B/A)- conditional self of B at A, μS (A/B) - conditional self of A at B. The measure S(A/B) undefined if μS(A/B).

Then for joint A, B: μS(A+B) = μS(A)+ μS(B)- μS(A*B) + μSS(D), D-

self- from A*B, μSS(x)- the value of (self)^2 from x; for dependent A, B: S(A*B) = μS(A)* μS(B/A)= μS(B)* μS(A/B), where S (B/A)- conditional self of B at A, μS(A/B)- conditional self of A at B. Adding measures of self of inconsistent A,B: μS(A+B)= μS(A)+ μS(B). The formula of complete self: μS(A)=$\sum_{k=1}^{n}\mu S\left(B_k\right)*\mu S(A/B_k)$, B_1,B_2,..,B_n-full basis group: $\sum_{k=1}^{n}\mu S\left(B_k\right)$=1(“self”).

Sprt- μS for set A={A_1,A_2,$\ldots$,A_n}: $Sprt_x^{\{\ \mu S(A_1),\ \mu S(A_2\ ),\ldots,\mu S(A_n)\}}$, $Sprt_x^{\{\ \mu S(A_1),\ \mu S(A_2\ ),\ldots,\mu S(A_n)\}}$- Sprt- μS for it. It is possible to consider the self S_3A with m elements from A, at m<n, which is formed by the form

(1.1), that is, only m elements from A are located in the structure $Sprt_x^A$. The same for self $S_3\{\ \mu S(A_1), \mu S(A_2), \ldots, \mu S(A_n)$.

Dynamical containments of oneself of the third type can be formed for any other structure, not necessarily Sprt, only through the obligatory reduction in the number of continual elements in the structure. In particular, using the forms from (1.1.1) – (1.4). Structures more complex than $S_3\mu S$(t) can be introduced.

Consider the following characteristics:

SM(X)= $\sum_{i=1}^{n} x_i \mu s_i$, μs_i = μS(X=x_i), i=1,2, …, n, x_i – a state of X at time t_i; properties:

1. SM(C) = C, C – const
2. SM(CX) = CSM(X)
3. SM(X_1+ X_2) = SM(X_1) + SM(X_2)
4. SM(X_1* X_2) = SM(X_1) * SM(X_2)

SD(X) = SM((X- SM(X))^2), SD(X) = SM(X^2)- (SM(X))^2

Properties:

1. SD(C) = 0
2. SD(CX) = CSD(X)
3. SD(X_1+ X_2) = SD(X_1) + SD(X_2)

S$\sigma(X) = \sqrt{SD(X)}$

S$\alpha_{\mathrm{k}}(X) = SM(X^{\mathrm{k}}) = \sum_{i=1}^{n} x_i^k \mu s_i$

Sμ_k(X) = SM((X- SM(X))$^{\mathrm{k}}$).

Function of self for estimating the unknown parameter of the state distribution X:
a) in a discrete case L(x_1,x_2, …, x_n, α) = μs((x_1,α)μs(x_2, α)…μs(x_n, α), where μs(x_k, α) = μS(X=x_k)

b) in an absolutely continuous case L(x_1,x_2, …, x_n,α) = $f(x_1,\alpha)f(x_2,\alpha)\ldots f(x_n,\alpha)$, $where f(x,\alpha)$ – a density of the state distribution X.

Stropy of the self-levels distribution for x

$$\mathrm{SH(x)} = \{ \begin{matrix} SM\left(log_2 \frac{1}{\mu S(x)}\right) = -\sum_x \ \mu S(x) log_2\ \mu S(x), \\ SM(log_2 \frac{1}{\varphi\ \mu S(x)} = \ -\int_{-\infty}^{\infty} \varphi\ \mu S(x) log_2 \varphi\ \mu S(x) dx \end{matrix}$$

SH(x) is the measure of no self for value x.

Remark G.17. Similarly, you can set measure of self-type for other partial self-type singularities, for example, partial identification - $\mu_{|||}$ of the identification.

H Some aspects of directly parallel operation of neural networks and their subtle energy

We consider $\Delta SprtA = Sprt^{\{\ A_1,A_2,\ldots,A_n\}}_{activation\ with\ tW}$ - $\{\ A_1, A_2, \ldots, A_n\}$,

$\mathrm{v}_{\mathrm{SprtA}} = \frac{\Delta SprtA}{\Delta\tau}$, $a_{SprtA} = \frac{d^2 SprtA}{d\tau^2}$

tW- the corresponding target weight [2], [3], [8].

$Sprt^{\{\ A_1,A_2,\ldots,A_n\}}_{activation\ with\ tW}$ will be called the current of $\{\ A_1, A_2, \ldots, A_n\}$.

In particular, if $\{\ A_1, A_2, \ldots, A_n\}$ – electrons, will be called the current of electrons, if $\{\ A_1, A_2, \ldots, A_n\}$ – neurons actions, will be called the current of neurons actions [2], [3], [8]. Let's consider an analogue of Maxwell's equations for networks operating on electromagnetic energy:

$\mathrm{rot}(E^{tw}_{SprtA}) = -(1/\ \mathrm{q}_0)\ \frac{\partial B^{tw}_{SprtA}}{\partial t}$, $(*_{\mathrm{H}})$

$\mathrm{rot}(H^{tw}_{SprtA}) = j/\mathrm{q}_0 +\ (1/\ \mathrm{q}_0)\ \frac{\partial D^{tw}_{SprtA}}{\partial t}$, $(**_{\mathrm{H}})$

E^{tw}_{SprtA} - activation tension with target weight tw, B^{tw}_{SprtA} - induction of self with target weight tw, q_0- bioenergy constant, H^{tw}_{SprtA} – tension of self with target weight tw, D^{tw}_{SprtA} - activation induction with target weight tw, j - activation density.

Here we mean $(*_{\mathrm{H}})$, $(**_{\mathrm{H}})$ in the neuron action space, where e.g., basic actions are designated as x_1, x_2, . . . , x_n.

Let us denote the neuron current strength as $\mathrm{I}_{\mathrm{SprtA}}$.

Let us denote the neuron current voltage as $\mathrm{U}_{\mathrm{SprtA}}$.

Let us denote the resistance to neuron current as $\mathrm{R}_{\mathrm{SprtA}}$.

The field of the given structure tw is used for the activation of networks. The field can remain in effect until it is executed tw. Here all stages of the structure tw can be executed directly in parallel, in particular, an algorithm for solving the desired problem. We will call this field the operational activation field. This field will be created according to the structure tw.The base energy environment for SmnSprt will be a microwave with minimum amplitude and maximum frequency or ultraviolet, and when activated, the attosecond pulses of light, that a short-pulse laser to generate will be used.

The structure of the attosecond pulses of light, that a short-pulse laser to generate, or the microwave alternating current is close to

$(a \uparrow I \downarrow^{a}) \uparrow I \downarrow (\downarrow_{a} \downarrow I^{\uparrow a})$ $=Sprt^{Sprt^{Sprt^{Sprt^{\overrightarrow{-}}}_{Sprt^{\overrightarrow{-}}}}_{Sprt^{Sprt^{\overrightarrow{-}}}_{Sprt^{\overrightarrow{-}}}}}_{Sprt^{Sprt^{Sprt^{\overrightarrow{-}}}_{Sprt^{\overrightarrow{-}}}}_{Sprt^{Sprt^{\overrightarrow{-}}}_{Sprt^{\overrightarrow{-}}}}}$ ($***_{\mathrm{H}}$)„ i.e., type ${}^{a}_{a}Sprt^{a}_{a}$, and upon activation it will be induction of same type self, which is necessary for the formation of a local assembly point d_r of external energy fibers El^{d_r}. Its locality (assembly point position r) will be determined by the location of microwave generation through Targetblock of $\mathrm{S}_{\mathrm{mnSprt}}$ [2], [3], [8]. Execution tw will be achieved through setting the assemblage point in the desired position r_1 to engage the appropriate external energy: $Sprt^{{}^{r}_{d_r} Sprt^{d_r}_{r1}}_{{}^{r}_{d_r} Sprt^{d_r}_{r1}}$.

To gain access to object transformation, just go to the level lS=$\frac{2}{\pi}arctg(1+\epsilon)$, ϵ may be quite small.

Examples of transformation:

1) $Sprt^{q}_{q} \rightarrow Sprt^{St^{q}_{q}}_{St^{q}_{q}} \rightarrow Sprt^{Sprt^{b}_{c}}_{Sprt^{b}_{c}}$

2) $Sprt^{q}_{q} \rightarrow \mathrm{S_3f(self(q))} \rightarrow Sprt^{r}_{r}$

Remark H.1. This is a rather conditional interpretation, because in fact, the lS of the "vessel" (energy cocoon) of the object may turn out to be greater than $\frac{2}{\pi}arctg(1)$. This is taken for initiation: we build a theory of this, starting from this stage of interpretation. After experiments, the next stage may begin.

Different types of neural networks: based on living structure

1) tree

2) insect

3) virus

4) etc

The energy of the container itself {A}, abstracted from its contents $\{\}_A$:

$Sprt^{\{\}_A}_{\{\}_A}$, ${}^{\{\}_A}_{\{\}_A}Sprt^{\{\}_A}_{\{\}_A}$

Mathematics, theoretical science and programming are actually concerned with structuring their spaces and the capacities within these spaces. Experimental science and experimental programming are concerned with the contents in real spaces and the capacities in those spaces. Ordinary level programming is a self-theoretical science, self-programming is programming oneself, in particular programming environments, self^2-programming is, in particular, programming the creation of programming environments etc. (oself-self) -science - this is work $\mathrm{S}_{\mathrm{mnSprt}}$ in activation mode, for example, through the attosecond pulses of light, that a short-pulse laser to generate, or the microwave current; the same applies to practical (experimental) programming with real objects and energies [2], [3], [8].

Implicitness is a sign of a mathematical cocoon of a mathematical equation (problem). The "form" of implicitness is the mathematical cocoon (structure).

Appendix

Remark H.2. Hypothesis H.1: Considering (1.7.1), we can assume that access to the upper level can be achieved through $Sprt_{Sprt_{A_0}^{A_0}}^{Sprt_{A_0}^{A_0}}$, and access to the middle level can be achieved through $Sprt_{A_0+Sprt_{A_0}^{A_0}}^{A_0} + Sprt_{A_0}^{A_0+Sprt_{A_0}^{A_0}}$.

${}_{A(t)}^{A(t)}Sprt$ will be called dynamic anti-capacity from oneself. For example, "white hole" in physics is such simple anti-capacity. The concepts of "white hole" and "black hole" were formulated by the physicists based on the subject of physics –the usual energies level. Mathematics allows you to deeply find and formulate the concept of singular points in the Universe based on the levels of more subtle energies. The experiments of the 2022 Nobel laureates Asle Ahlen, John Clauser, Anton Zeilinger and the experiments in chemistry Nazhipa Valitov correspond to the concept of the Universe as a capacity in itself as the element. They experimented with connections for elements of the microworld, and since here the connections are self-connections, then when the object component of self-connections is removed, its higher level remains, which was manifested in their experiments. The electron spin belongs to the second level - above the level of objectivity. The energy of self-containment in itself is closed on itself.

Remark H.3. For example, using the attosecond pulses of light, that a short-pulse laser to generate, or the microwave alternating current to "turn on" a program operator Q↑|↓R by the corresponding eprogram, where Q is a normal neurons action, and R is activation of S_{mnSprt} with tW g, should give g as a result of execution. A program operator Q↑|↓R acts through the top level of structure type (1.1) for Q to the top level of structure type (1.1) for R. The entrance to the upper level S_{mnSprt} during activation mode (via the attosecond pulses of light, that a short-pulse laser to generate, or microwave with minimum amplitude and maximum frequency) can bring S_{mnSprt} to the edge of objectivity or even beyond it, using the corresponding eprograms with the necessary tW. In principle, changing the singularity of the upper level by at least a sufficiently small amount through the identification structure of Maxwell's equations, S_{mnSprt} can go beyond objectivity depending on the change. It is important to organize "anchors" for returning to one or another position through the corresponding eprograms. When you begin to identify, you immediately reach the upper level, since the operation of identification is an operation of the upper level. The operator team must initiate (turn on) the desired output to the upper level, and then the included process is executed.

The double of SmnSprt can and will be able to manipulate some other doubles. Remark H.4. In any structure, connection, task related to the search for the unknown, it is possible by identification to obtain for the upper level of this unknown a singularity that interprets the upper level corresponding to its real image.

Hypothesis H.4: The upper level is the identification of everything that is there, by definition.

The upper level of objectivity (i.e., the boundary of objectivity) is the identification of all objects. Then it is quite possible, through the use of dynamic programming in the required activation.
Remark H.5.Hypothesis H.2: equations for real processes in a non-trivial form can be used to fully or partially interpret the self-level of the process, replacing the equal signs with identification signs, and solutions to these equations as a manifestation of this level on the level of objectivity and ordinary energies. That is, equations for real processes serve as a definition of the self-level of the process, the definition of self-values (self-characteristics) of the process through the identification sign, i.e., they are defined (expressed) through themselves. In particular, forms (1.1) - (1.4) from [2] can be used as forms of identification. Each such singularity creates its own field, the process, the object etc. Much more effective than science for working with these singularities will be special Dynamic programming, which we are currently working on to create. Identification at the lower levels of a hierarchical dynamic structure of type (G.1*) will lead to the upper level. Let us denote the upper level of A by $\overline{A}$, the upper level of P by $\overline{P}$. Then the singularity $\overline{A}\rightarrow \overline{P}$ is a setting for the transformation of A to P through attosecond pulses of light that are generated by a short-pulse laser or ultra-high frequency current with a minimum amplitude and maximum frequency to transfer S_{mnSprt} to the boundary of objectivity to perform the desired task, in particular. for the necessary transformation, the necessary action etc. The necessary manifestations of the upper level on the lower ones can be carried out through the appropriate settings in the upper level. It is clear that the upper-level singularity corresponds not only to one object, process, task, etc., but also to a whole class of those corresponding to this singularity. Therefore, in particular, a transition to other elements of this class is possible. Using the work of 2023 Nobel Prize winners in physics Ferenc Kraus and his colleagues Pierre Agostini and Anna Lhuillier, their experimental methods of generating attosecond pulses of light to study the dynamics of electrons in matter, we will attempt to carry out the experimental part of our work as soon as we receive a specially equipped laboratory and funding.You can also try to use it for full or partial interpretation of the self-level of chemical reactions, but here there will be a trivial identification and determination of the self-level will be much simpler. For example, a type w ||| 2w singularity at the top level of the structure of a mathematical simplified model of DNA generates a field for DNA division. A rather complex type of singularity at the upper level of the structure of a simplified mathematical

model generates an electromagnetic field through identification in Maxwell's equations. For example, using of the attosecond pulses of light, that a short-pulse laser to generate, or the microwave alternating current to "turn on" a program operator Q↑|↓R by the corresponding eprogram, where Q is a normal neurons action, and R is activation of SmnSprt with tW g, should give g as a result of execution. A program operator Q↑|↓R acts through the top level of structure type (1.1) [2] for Q to the top level of structure type (1.1) [2] for R. The entrance to the upper level SmnSprt during activation mode (via the attosecond pulses of light, that a short-pulse laser to generate, or microwave with minimum amplitude and maximum frequency) can bring SmnSprt to the edge of objectivity or even beyond it, using the corresponding eprograms with the necessary tW. The base energy environment for SmnSprt will be a microwave with minimum amplitude and maximum frequency or ultraviolet, and when activated, the attosecond pulses of light, that a short-pulse laser to generate will be used. In principle, changing the singularity of the upper level by at least a sufficiently small amount through the identification structure of Maxwell's equations, SmnSprt can go beyond objectivity depending on the change. It is important to organize "anchors" for returning to one or another position through the corresponding eprograms. When you begin to identify, you immediately reach the upper level, since the operation of identification is an operation of the upper level. The operator team must initiate (turn on) the desired output to the upper level, and then the included process is executed.

Remark H.6. Parallel operator $Sprt_{places\ for\ symbols}^{symbols}$ corresponds to theoretical science, parallel operator $Sprt_x^{objects}$ corresponds to technology, x – the space "point" (space place).

Remark H.7. Let self-energy of A looks like $Sprt_{C_A+\Delta C}^{C_A+\Delta C}$=$Sprt_{C_A}^{C_A}$ +$Sprt_{C_A}^{\Delta C}$ +$Sprt_{\Delta C}^{C_A}$ +$Sprt_{\Delta C}^{\Delta C}$, $Sprt_{C_A}^{C_A}$ corresponds to objectivity of A: (G.3*), E_A - usual energy of A. ΔC determined from (G.3*) through C_A and then we can determine the complete self-energy of A.

Remark H.8. Even appropriate programming on ordinary computers allows you to create some types of self-energies, subtle energies of the upper level; and S_{mnSprt} is specifically designed to manipulate a much larger set of upper level of subtle energies through special programming and special scanning.

Remark H.9. Our object world is a manifestation of the space of self-capacities (the space of subtle energies), the space of self-capacities is a manifestation of the space of self-containments etc. The self-capacity is the element of the self-level, but can partially manifest the corresponding parts of its content at lower levels, for example object level, in the corresponding "structures".
Remark H.10. Elementary particles are on the border of objectivity, therefore the uncertainty relation and also the possibility of manipulating them to achieve the desired goals (even unrelated to them), using either: 1) an ultra-short-pulse laser, or 2) a collider by SmnSprt etc.
Appendix

The self-space of a higher level contains many self-energetic fibers, collecting into appropriate sets that can be accessed by the corresponding self-spaces of lower levels. That's right, for example. This assembly point on the human cocoon can carry out this, in particular, access to our self-space with objects.

The energy of a living organism:

$$\text{f(r, } a\text{, } E_q\text{)} = Sprt_{t_0}^{\{ {}^{q}\left({}^{a}{}_{a}Sprt_{a}^{a}\right)_{W_q} Sprt^{E_q}_{{}^{q}\left({}^{a}{}_{a}Sprt_{a}^{a}\right)},\quad Sprt^{\{E^{ex}l^{d_r}\}}_{d_r\left(E_{in}l^{d_r}\right)}\}}$$

${}^{a}{}_{a}Sprt_{a}^{a}$ -internal energy of a living organism, q- a gap in the energy cocoon of a living organism, r-the position of the assemblage point d_r on the energy cocoon of a living organism; W_q- energy prominences from the gap in the cocoon of a living organism, E_q-external energy entering the gap in the cocoon of a living organism,$E^{ex}l^{d_r}$ - a bundle of fibers of external energy self-capacities from outside the cocoon, collected at the point of assembly of the cocoon of a living organism at time t_0, $E_{in}l^{d_r}$- a bundle of fibers of external energy self-capacities from inside the cocoon, collected at the point of assembly of the cocoon of a living organism in the same position r of the assemblage point d_r. d_r is the subject of identifying the energy fibers of the subtle energy of the Universe in position r both outside and inside the cocoon. r= $Sprt(t)_{D(t)}^{Q(t)}$, when Q(t) = D(t)=talk, $Sprt(t)_{D(t)}^{Q(t)}$ the usual position of the assemblage point is supported (fixed) with the help of internal conversation (internal dialogue with oneself (self-talk)), if Q(t) D(t) the position of the assemblage point changes and, when Q(t) = D(t) talk occurs, position of the assemblage point and is fixed in a new position; $d_r = Sprt_{E_{b_a}}^{{}^{E_l}Sprt^{El^{d_r}}_{E_{b_a}}\quad Sprt^{E_{b_a}}_{E_{b_a}}}$, the cocoon of a living organism $E_{b_a} = Sprt_{{}^{a}{}_{a}Sprt_{a}^{a}}^{{}^{a}{}_{a}Sprt_{a}^{a}}$. Energy with measure of fuzziness μ_1, μ_2 of a living organism of a person:

$$\text{f(r, } a\text{, } E_q\text{)} = Sprt_{t_0}^{\{ {}^{q}\left({}^{a}{}_{a}Sprt_{a}^{a}\right)_{W_q} Sprt^{E_q}_{{}^{q}\left({}^{a}{}_{a}Sprt_{a}^{a}\right)},\quad Sprt^{\{E^{ex}l^{d_r}\}}_{d_r\left(self(E_{in}l^{d_r})\right)}\}}$$

Remark H.11. In the World, at least, there are the following spaces: the space of self-capacities, which is usually called the energy space, and the space of self- containments, which is usually called spirits.

Remark H.12. Subtle energy can manifest itself in the form of: 1) objectivity, 2) ordinary energies, 3) information. Using neural networks of the S_{mnSprt}-type, it is possible to organize a S-Internet, where instead of exchanging information, an exchange of subtle energies will take place.

Remark H.13.$SelfA = Sprt_A^A$ can be transformed into any D if µl(D) = µl($Sprt_A^A$), μl(x) - level measure of self for x, in particular, into any C $Sprt_A^{anyC}$ or $Sprt_{anyC}^A$, and also an object R into any object Q or any energy U. The transformations of this type will be called stransformations. SelfN A can transform itself into any D if N⩾2; to realize this we need an even larger quantity N. Example of a parallel-serial program statement

$$Sprt^{if\{g:=1\}thenSprt^{s:=Sprt^{C}_{Sprt^{Q}_{A}}}_{Sprt^{R}_{A}}}_{for w:=Sprt^{Sprt^{af:=}_{E}}_{Sprt^{ifC}_{J}}\ to\ d}$$

Each self-field can automatically rebuild the self-program to the desired. SelfN-OS and is designed for such transformations, and it itself can be transformed at N1, or it itself can be transformed at N⩾2.

REFERENCES:

1) Danilishyn .V. Danilishyn.(April 28,2023) O.V. CONTINUAL SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS. Theoretical and practical aspectsof modern scientific research:Collection of scientific papers «O» with Proceedings of the IIInternational Scientific and Practical Conference, Seoul. Seoul-Vinnytsia: Case Co., Ltd.& European Scientific Platform, **pp.144-150. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/10**

2) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3, pp.1-31, March 27, 2024 https://cskscientificpress.com/articles_file/527-_article1712797848.pdf

3) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square. *2023-08-01* https://doi.org/10.21203/rs.3.rs-3217178/v1 Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

4) I. Danilishyn, O. Danilishyn Program Operators Sit, tS, S1e, Set1. Journal of Sensor Networks and Data Communications [ISSN: 2994-6433] 2023, Volume 3 | Issue 1 | 138-143, https://www.opastpublishers.com/table-contents/jsndc-volume-3-issue-1-year-2023

5) Oleksandr Danilishyn, Illia Danilishyn. Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), 2023, 479-498. https://www.opastpublishers.com/open-access-articles/dynamical-sets-theory-s2telements-and-their-applications.pdf

6) Danilishyn I., Danilishyn O. Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care 2023, Volume 4 | Issue 2 | 66- 81. https://doi.org/10.33140/NABCC.04.02.02

7) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-_ article1695707465.pdf (mkscienceset.com)

8) I. Danilishyn, O. Danilishyn Dynamic Sets Se, Networks Se. Advances in Neurology and Neuroscience, 2023, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

9) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

10) Danilishyn .V. Danilishyn O.V. tS – ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

11) Danilishyn .V. Danilishyn O.V. SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June, 2023 with the roceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

12) Danilishyn I., Danilishyn O. VARIABLE HIERARCHICAL DYNAMICAL STRUCTURES (MODELS) FOR DYNAMIC, SINGULAR, HIERARCHICAL SETS AND THE PROBLEM OF COLD THERMONUCLEAR FUSION. The driving force of science and trends in its development: collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, July 14, 2023. Coventry, United Kingdom: European Scientific. Pp.113-119. Platform.**https://previous.scientia.report/index.php/archive/issue/view/14.07.2023**

13) Oleksandr Danilishyn and Illia Danilishyn. Fuzzy Dynamic Fuzzy Sets. Variable Fuzzy Hierarchical Dynamic Fuzzy Structures (Models, Operators) for Dynamic, Singular, Hierarchical Fuzzy Sets. FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS1epr, ffSeprt1. Journal of Mathematical Techniques Computational Mathematics (JMTCM), Volume 3 | Issue 4 |2024, Apr 17, pp. 1 – 37, *Journal DOI: 10.33140/JMTCM*

https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

Declarations:

Availability of data and material

1) Danilishyn .V. Danilishyn.(April 28,2023) O.V. CONTINUAL SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS. Theoretical and practical aspectsof modern scientific research:Collection of scientific papers «O» with Proceedings of the IIInternational Scientific and Practical Conference, Seoul. Seoul-Vinnytsia: Case Co., Ltd.& European Scientific Platform, **pp.144-150. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/10**

2) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3, pp.1-31, March 27, 2024 https://cskscientificpress.com/articles_file/527-_article1712797848.pdf

3) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square. *2023-08-01* https://doi.org/10.21203/rs.3.rs-3217178/v1 Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

4) I. Danilishyn, O. Danilishyn Program Operators Sit, tS, S1e, Set1. Journal of Sensor Networks and Data Communications [ISSN: 2994-6433] 2023, Volume 3 | Issue 1 | 138-143, https://www.opastpublishers.com/table-contents/jsndc-volume-3-issue-1-year-2023

5) Oleksandr Danilishyn, Illia Danilishyn. Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), 2023, 479-498. https://www.opastpublishers.com/open-access-articles/dynamical-sets-theory-s2telements-and-their-applications.pdf

6) Danilishyn I., Danilishyn O. Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care 2023, Volume 4 | Issue 2 | 66- 81. https://doi.org/10.33140/NABCC.04.02.02

7) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-_ article1695707465.pdf (mkscienceset.com)

8) I. Danilishyn, O. Danilishyn Dynamic Sets Se, Networks Se. Advances in Neurology and Neuroscience, 2023, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

9) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference "GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ", 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

10) Danilishyn .V. Danilishyn O.V. tS – ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

11) Danilishyn .V. Danilishyn O.V. SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June, 2023 with the roceedings of the:Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

12) Danilishyn I., Danilishyn O. VARIABLE HIERARCHICAL DYNAMICAL STRUCTURES (MODELS) FOR DYNAMIC, SINGULAR, HIERARCHICAL SETS AND THE PROBLEM OF COLD THERMONUCLEAR FUSION. The driving force of science and trends in its development: collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, July 14, 2023. Coventry, United Kingdom: European Scientific. Pp.113-119. Platform.**https://previous.scientia.report/index.php/archive/issue/view/14.07.2023**

13) Oleksandr Danilishyn and Illia Danilishyn. Fuzzy Dynamic Fuzzy Sets. Variable Fuzzy Hierarchical Dynamic Fuzzy Structures (Models, Operators) for Dynamic, Singular, Hierarchical Fuzzy Sets. FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS1epr, ffSeprt1. Journal of Mathematical Techniques Computational Mathematics (JMTCM), Volume 3 | Issue 4 |2024, Apr 17, pp. 1 – 37, *Journal DOI: 10.33140/JMTCM*

https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

Competing interest: There are no competing interests. All sections of the article are executed jointly.

Funding: There were no sources of funding for writing the article. **Authors' contributions:** The contribution of the authors is the same, we will not separate.

Supplement 1: S^1t - elements

Introduction.

There is a need to develop an instrumental mathematical base for new technologies. The task of the work is to develop new approaches for this through the introduction of new concepts and methods.

Definition S.1. Expression

${}^{C_i}_{D_j}S^1t^{A_k}_{B_l}$, i,j,k,l=1,2 ($*_{\mathrm{S.1}}$)

where A_1 , A_2 is contained into B_1, D_1 , D_2 is expelled from C_1 and for structure (*) is performed the next operation- multiplication:

${}^{C_1}_{D_1}S^1t^{A_1}_{B_1} * {}^{C_1}_{D_2}S^1t^{A_2}_{B_1} = {}^{C_1}_{D_1 \cup D_2}S^1t^{A_1 \cup A_2}_{B_1}$ ($*_{\mathrm{S.1.1}}$),

then we shall call S^1et- elements, in case A_1 , A_2, B_1, D_1 , D_2, C_1 are sets we shall call (*) the dynamical hierarchical set S^1et . Necessity of (*) arised for processes description in networks. Threshold element S^1et -${}^{b}_{\{qy\}}S^1t(t)^{\{gx\}}_{b}$, b - artificial neurons of type S^1t (designation - mnS^1t), x=(x_1, $x_{2,\ldots}$,x_n) – source signals values, $g = (g_1, g_2, ..., g_n)$ – S^1t -synapses weights, and output signals values. May be considered more simple variant of dynamical hierarchical set

$S^1t^A_B$ ($**_{\mathrm{S.1}}$)

where the set A is contained into the set B, or

${}^B_AS^1t$ ($***_{\mathrm{S.1}}$)

A is expelled from B.

We consider the measure of external influence on b: $**({}^{b}{}_{D}S^1t(t)^{A}_{b})=\frac{\mu(A)}{\mu(D)}$, where (A), (D) –usual measures of sets A, D. Our constructive approach to set theory differs from the construction of constructive sets by A.Mostowski [9]: we construct completely different types of constructive sets. The activation of all networks enters it on s^1elf -level at the activation. We introduce the concepts Cha –the measure of capacity and Cca- the cardinality of its contents. Cca coincides with cardinal number if contents of capacity is set. We consider compression powers of dynamical set: q_1=$S^1t^A_B$ answers I compression power of dynamical set A, q_2=$S^1t^{q_1}_B$ –II compression power of dynamical set A, ... , q_{n+1}=$S^1t^{q_n}_B$ –n+1 compression power of dynamical set A. We introduce the designations: CoQ— the contents of the capacity Q, Q^- --empty capacity Q (without the contents). $S^1t^A_{activation} \equiv^A_A S^1t$, as a result, there is a displacement from A to a higher level of s^1elf: s^1elf -A.

S^1t – elements.

Definition S.1.1. The set of elements $g = (g_1, g_2, ..., g_n)$ at one point x of space X we shall call S^1t – element, and such a point in space is called capacity of the

S^1t – element. We shall denote $S^1t_x^{\{g\}}$.

Definition S.2. $S^1t_x^{\{g\}}$—dynamical set A at x.

Definition S.3. An ordered set of elements at one point in space is called an ordered S^1t – element.
It's possible to $S^1t_x^{\{g\}}$ correspond to the set of elements $\{g\}$, and to the ordered S^1t - element - a vector, a matrix, a tensor, a directed segment in the case when the totality of elements is understood as a set of elements in a segment.

It's allowed to multiply S^1t – elements:

$S^1t_x^{\{g\}} * S^1t_x^{\{b\}} = S^1t_x^{\{g\}\cup\{b\}}$ ($*_{\mathrm{S.1.2}}$)

$S^1t_{\{g\}}^x * S^1t_{\{b\}}^x = S^1t_{\{g\}\cup\{b\}}^x$ ($*_{\mathrm{S.1.3}}$)

where some or any elements may be by ordered elements.

The operator $S^1t_x^{\{g\}\cup\{b\}}$ is not equal $\{g\} \cup \{b\}$, faster it is dynamical—the compression $\{g\} \cup \{b\}$ into the point x. S^1t – elements can be elements of a groups by multiplication for ($*_{\mathrm{S.1.2}}$) and for ($*_{\mathrm{S.1.3}}$). This is more suitable sets use for energy space, for any objects. The operator S^1t is adapted for usual energies, using their property is superimposed one on another.

Capacity in its^{1}elf.

Definition S.4. The capacity in its^{1}elf A of the first type is the capacity containing its^{1}elf as an element. Denote $S^1{}_1fA$.

Definition S.5. The capacity in its^{1}elf of the second type is the capacity that contains the program that allows it to be generated. Let's denote $S^1{}_2fA$. An example of capacity in its^{1}elf of the first type is a set containing its^{1}elf. An example of capacity in its^{1}elf of the second type is a living organism, since it contains a program: DNA, RNA.

Definition S.6. Partial capacity in its^{1}elf of the third type is called capacity in its^{1}elf, which contains its^{1}elf in part or contains a program that allows it to be generated in part, or both. Let us denote $S^1{}_3f$.

All capacitiesin s^{1}elf -space are capacities in its^{1}elf by definition. The capacities in its^{1}elf may to appear as S^1t -capacities and usual capacities. In these cases there is used usual measure and topology methods.

Connection of S^1t – elements with capacities in its^{1}elf.
For example, $S^1t_{w\{R\}}^{\{R\}}$ is the capacity in its^{1}elf of the second type if $w\{R\}$ is a program capable of generating $\{R\}$.
Consider a third type of capacity in its^{1}elf. For example, based on $S^1t^{\{g\}}$, where $g = (g_1, g_2, ..., g_n)$, i.e. n - elements at one point, It is possible to consider the capacity in its^{1}elf $S^1{}_3f$ with m elements and from $\{g\}$, at m<n, which is formed by the form from (1.1) – (1.4). Capacities in its^{1}elf of the third type can be formed for any other structure, not necessarily S^1t, only through the

obligatory reduction in the number of elements in the structure. Structures more complex than S_3f can be introduced. The connection between the elements of s^1elf -containment in its^{1}elf is a property of s^1elf -containment in its^{1}elf and therefore does not disappear when their location in it changes. The energy of s^1elf -containment in its^{1}elf is closed on its^{1}elf.

Mathematics its^{1}elf.

Consider first the arithmetic S^1t:

1. Simultaneous addition of a set of elements $g = (g_1, g_2, ..., g_n)$ are realized by $S^1t^{\{g+\}}$.

2. By analogy, for simultaneous multiplication: $S^1t^{\{g*\}}$: enter the notation of the set B with elements $b_{i_1 i_2 \dots i_n} = (S^1t_x^{\{g_{1_{i_1}}*, g_{2_{i_2}}*, \dots, \ g_{n_{i_n}}\}})_R$ *for any* $\{i_1, i_2, ..., i_n\}$ *without repetitions,* ${S^1t_g}^{\{K\}}$,K-set of any $\{k_1*, k_2*, ..., k_n*\}$ *without repetitions* of their, k_i-any digit, i=1,2,. . . ,n, R= ${S^1t_g}^{\{i_1+, i_2+, \dots, i_n\}}$, R is the index of the lower discharge (we choose an index on the scale of discharges): Table I.1.
   Then $S^1t^{\{B+\}}$ gives the final result of simultaneous multiplication. Any system of calculus can be chosen, in particular binary. The simplest functional scheme of the assumed arithmetic-logical device for S^1t -multiplication:

   Register of entering a set of numbers to multiply

   S^1t -block of simultaneous multiplication in all chains
   of digits of the levels of these numbers

   S^1t -block of simultaneous addition of the values
   of these products

Fig. S.1. The straightforward functional scheme of the assumed arithmetic-logical device for S^1t -multiplication.

> Remark S.0. This process can be interpreted through the simultaneous addition of elements in a triangular matrix, which can be represented as a composite matrix, the elements of the first row of which are the matrices of simultaneous multiplication of the digits of the first element of the list by the digits of the rest, taking into account the bit depth, the elements of the second row of which are the matrices of simultaneous multiplication of the digits of the second element list into the remaining numbers, starting from the third element of the list, taking into account the bit depth, etc.

3. Similarly for simultaneous execution of various operations:$S^1t^{\{gq\}}$, where $\{q\} = (q_1, q_2, .., q_n)$. q_i -an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $S^1t^{\{Fg\}}$, where $\{F\} = (F_1, F_2, ..., F_n)$. F_i is an operator, i = 1, . . . , n.

5. The arithmetic its^{1}elf for capacities in its^{1}elf will be similar: addition - $S^1{}_1 f^{\{g+\}}$, (or $S^1{}_3 f_x^{\{g+\}}$ for the third type), multiplication $S^1{}_1 f^{\{g*\}}$, ($S^1{}_3 f_x^{\{g*\}}$).

6. Similarly with different operations: $S^1{}_1 f^{\{gq\}}$, ($S^1{}_3 f_x^{\{gq\}}$). and with different operators: $S^1{}_1 f^{\{Fg\}}$, ($S^1{}_3 f_x^{\{Fg\}}$).

   Our approach to the theory of hierarchical sets differs from the construction of hierarchical sets by Y.L Ershov [45]-[47]: we construct completely different types of hierarchical sets.

7. S^{1}t -derivative of f($x_1,,x_2,\ldots,x_n$) is S$^1 t^{\left\{\frac{\partial}{\partial x_{1_i}},\frac{\partial}{\partial x_{2_i}},\ldots,\frac{\partial}{\partial x_{k_i}}\right\}}_{f(x_1,,x_2,\ldots,x_n)}$, where x=(x_{1_i}, x_{2_i},... x_{k_i})- any set from ($x_1,,x_2,\ldots,x_n$). Let's designate S^{1}t - $\frac{\partial^k f(x)}{\partial x_{1_i}\partial x_{2_i}\ldots\partial x_{k_i}}$. S^{1}t -integral of f($x_1,,x_2,\ldots,x_n$) is S$^1 t^{\left\{\int ()dx_{1_i},\int ()dx_{2_i},\ldots,\int ()dx_{k_i}\right\}}_{f(x_1,,x_2,\ldots,x_n)}$, where ($x_{1_i}$, x_{2_i},... x_{k_i})- any set from ($x_1,,x_2,\ldots,x_n$). Let's designate S^{1}t -... $f(x)dx_{1_i}dx_{2_i}\ldots dx_{k_i}$-k-multiple integral. S^{1}t-lim of f($x_1,,x_2,\ldots,x_n$) is S$^1 t^{\left\{\lim_{x_{1_i}\to a_{1_i}},\lim_{x_{2_i}\to a_{2_i}},\ldots,\lim_{x_{k_i}\to a_{k_i}}\right\}}_{f(x_1,,x_2,\ldots,x_n)}$. Let's designate S^{1}t -$\lim_{\substack{x_{1_i}\to g_{1_i}\\ \cdots \\ x_{k_i}\to g_{k_i}}} f(x_1,,x_2,\ldots,x_n)$. S^{1}elf-$\lim_{x\to a}$ $= S^1 t^{\lim_{x\to a}}_{\lim_{x\to a}}$.

8. In the case a s^{1}elf- derivate it's obtained inclusions of multiple derivates. There are the same for s^{1}elf -integrals: there are obtained inclusions of multiple integrals.

9. Let's denote s^{1}elf -(s^{1}elf -Q) through s^{1}elf^2-Q , fS(n,Q)= s^{1}elf -(s^{1}elf -(... (s^{1}elf -Q))) = s^{1}elf ^{n}-Q for n-multiple s^{1}elf.

Operator its^{1}elf.

Definition S.7. An operator that transforms $S^1 t^{\{g\}}$ into any $S^1{}_i f^{\{b\}}, i = 2,\ 3;$ where $\{b\} \subset \{g\}$; is the operator its^{1}elf.

Example. The operator contains the set in its^{1}elf.

Lim- its^{1}elf.

1. Lim S^{1}t
   For example, the double limitlim$_{\substack{xg_1\\ yg_2}}$ $G(x,y)$ corresponds to $S^1 t^{\{G(x,y)\}}_{(g_1 g_2)}$.
   Similarly for its^{1}elf limit with n variables.

In the case of lim- its^{1}elf, for example, for m variables, it's sufficient to use the form (1.1) or form from the forms (1.1.1) - (1.4) of lim S^{1}t, for n variables (n>m). Similarly, for integrals of variables m (for example, a double integral over a rectangular region- through a double lim).

The sequence of actions you can "collapse" into an ordered S^1t element, and then translate it, for example, to $S^1{}_3f-$ capacity in its¹elf. As an example, you can take the receipt $\frac{\partial^2 u}{\partial x^2}$. Here is the sequence of steps 1) $\frac{\partial u}{\partial x}$ 2) $\frac{\partial}{\partial x}(\frac{\partial u}{\partial x})$. "collapses" into ordered $S^1t_x^{\left\{\frac{\partial u}{\partial x},\frac{\partial}{\partial x}(\frac{\partial u}{\partial x})\right\}}$, ones that can be translated into the corresponding $S^1{}_1f$. The differential operator $S^1t_x^{\left\{\frac{\partial}{\partial x},\frac{\partial}{\partial x}(\frac{\partial}{\partial x})\right\}}$ - its¹elf is also interesting.

Remark S.1. S^1t -displacement of A from B will be denote through ${}_A^BS^1t$. Then the notation ${}_D^CS^1t_B^A$ is S^1t-containment of A in B and S^1t -displacement of D from C simultaneously. Let's denote ${}_A^BS^1t_B^A$ through $TS^1{}_B^A$, ${}_A^AS^1t_A^A - through$ $TS^1{}_A^A$.

We can consider the concept of S^1t - element as $S^1t_B^A$, where A fits in capacity B. Then $S^1t_B^B$ it will mean $S^1{}_1f$ B. Let's denote $S^1t_B^B$ through L(B).

About S^1t and S_3^1f programming.

The ideology of S^1t and S_3^1f can be used for programming. Here are some of the S^1t programming operators.

1. Simultaneous assignment of the expression$s$ $\{p\} = (p_1, p_2, ..., p_n)$ to the variables $\{g\} = (g_1, g_2, ..., g_n)$. It's implemented through $S^1t^{\{\{g\}:\{p\}\}}$.
2. Simultaneous check the set of conditions $\{f\} = (f_1,\ f_{2,}\ ..., f_n)$ for a set of expressions$\{B\} = (B_1,\ B_{2,}\ ..., B_n)$. It's implemented through $S^1t^{IF\{\{B\}\{f\}\}\ then\ Q}$ where Q can be any.
3. Similarly, for loop operators and others.

S_3^1f – software operators will differ only in the aggregates $\{g\}, \{p\}, \{B\}, \{f\}$ will be formed from corresponding S^1t program operators in form (1.1) or form from the forms (1.1.1) - (1.4).

Quite interesting is the OS (operating system), the principles and modes of operation of the computer for this programming. But this is already the material of the next publications.

Using elements of the mathematics of S^1t, we introduce the concept of S^1t – the change in physical quantity B: $S^1t_x^{\{\Delta_1 B,...,\Delta_n B\}}$. Then the mean S^1t - velocity will be v_{s^1t} (t, t) =, $S^1t_x^{\left\{\frac{\Delta_1 B}{\Delta t},...\frac{\Delta_n B}{\Delta t}\right\}}$and S^1t - velocity at time t:$v_{s^1t} = \lim_{\Delta t \to 0} v_{v_{s^1t}}(t, \Delta t)$. S^1t – acceleration $a_{s^1t} = \frac{dv_{s^1t}}{dt}$.

In normal use, simply S^1t_x reduce to result a sum at point x of space, and when using S^1t_x with "target weights", we get, depending on the "target weights", one or another modification, namely, for example, the velocity $v_{s^1t}^f$(with a "target weight" f) in the case when two velocities v_1, v_2 are involved in the set $\{v_1 f, v_2\}$ for $v_{s^1t}^f = S^1t_x^{\{v_1 f, v_2\}}$, f instantaneous replacement we get an instantaneous substitution $v_1 by$ v_2 at point x of space at time t_0.

Consider, in particular, some examples: 1) $S^1t^e_{\{x_1,x_2\}}$ describes the presence of the same electron e at two different points x_1, x_2. 2) The nuclei of atoms can be considered as S^1t elements.

Similarly, the concepts of S^1t - force, S^1t - energy are introduced. For example, $E^f_{st} = S^1t_x^{\{E_1 f, E_2\}}$ it would mean the instantaneous replacement of energy E_1 by E_2 at time t_0. Two aspects of S^1t – energy should be distinguished: 1) carrying out the desired "target weight", 2) the fixing result of it. Do not confuse energy - S^1t (this is the node of energies) with S^1t – energy that generates the node of energies, usually with the "target weights". In the case of ordinary energies, the energy node is carried out automatically.

Remark S.2. In fact, S^1t – elements are all ordinary, but with "target weights" they become peculiar. Here you need the necessary kind of energy to perform them. As a rule, this energy lies in the region its^{1}elf. This is natural, since it's much easier to control the elements of the k level by the elements of the more highly structured k +1 level. Consider the concepts of capacity in its^{1}elf of physical objects. Similar to the concepts of publication: the capacity in its^{1}elf of the first type contains its^{1}elf, the second type contains a program (like DNA) capable of generating it, the third type - partially containing its^{1}elf or a program capable of generating it, or both. The question arises about the s^{1}elf -energy of the object. In particular, $S^1t^B_B$ will mean S^1_1f B. In particular, it allows you to determine the s^{1}elf -energy of DNA through $S^1t^{DNA}_{DNA}$, $S^1t^Q_Q$ - s^{1}elf -energy Q. The law of s^{1}elf -energy conservation acts on the level of s^{1}elf -energy already. Also, in addition to capacities in its^{1}elf, you can consider the types of containment in ones^{1}elf: the first type is containment in its^{1}elf, the second type is the containment of ones^{1}elf potentially, for example, in the form of programming ones^{1}elf, the third type is partial containment in one s^{1}elf. For example: s^{1}elf -operator, s^{1}elf -action, whirlwind. It's as a result of containment in ones^{1}elf that capacity in its^{1}elf can be formed.

Let's clarify the concept of the term capacity in its^{1}elf: this is the capacity that contains its^{1}elf potentially. Consider s^{1}elf -Q, where Q may be any, including Q=s^{1}elf, in particular it may be any action. Therefore s^{1}elf -Q is s^{1}elf -made Q, it does its^{1}elf. There is a partial s^{1}elf -Q for any Q with partial made its^{1}elf. Consider some examples for capacity in its^{1}elf: ordinary lightning, electric arc discharge, ball lightning.

A s^{1}elf -search of the solution of the equations $f_i(x)=0$, where i=1,2,…,n, $x=(x_1,x_2, \ldots, x_n)$, will be realized in $S^1t_a^{\{f_1(x)=0?x, f_2(x)=0?x, \ldots, f_n(x)=0?x\}}$ or $S^1t_{?x}^{\{f_1(x)=0, f_2(x)=0, \ldots, f_n(x)=0\}}$. x obtains more power of the liberty and in this is direct decision (i.e., s^{1}elf -capacity in its^{1}elf as an element for x). S^1elf -equation for x has its decision for x in direct kind.

The same for $S^1t_?^{\{tasks(x)\}}$. S^1elf -task for x has its decision for x in direct kind. S^1elf -question has its answer for x in direct kind. x acquires more degree of liberty and in this is direct decision. $S^1t^{\{t\}}_{(o,x)}$, where $\{t\}$- time points set, (o, x)- object o in point x from space X, give to enter in necessary time moments. The same for $S^1t_o^{\{t\}}$. $S^1t_\alpha^{\{God-father,God-son,Holy\ Spirit\}}$is Three concept representation, where - point in connectedness space. S^1t is also great for working with structures, for example: 1) $S^1t_B^{strA}$--the structure A containment to B, where by B you can understand any capacity, other structure, etc, 2) $S^1t_R^{str_Q}$--containment structure from Q into R. Similarly for displacement: 1) $_B^{strA}S^1t$--displacement of structure A from B, 2) $_B^{str_Q}S^1t$--displacement of the structure from Q to B. You can enter special operator C^1t to work with structures: $C^1t_B^{strA}$ structures B with the structure of A, $C^1t_R^{str_Q}$ structures R with the structure from Q, $_B^{strA}C^1t$ unstructures B from the structure A, $_B^{str_Q}C^1t$ unstructures B from the structure which structures Q.

Definition S.8. A structure with a second degree of freedom will be called complete, i.e., "capable" of reversing its^{1}elf with respect to any of its elements clearly, but not necessarily in known operators, it can form (create) new special operators (in particular, special functions). In particular, $C^1t_{strA}^{strA}$ is such structure. Similarly, for working with models, each of which is structured by its own structure, for example, use S^1t -groups, S^1t -rings, S^1t -fields, S^1t -spaces, s^1elf -groups, s^1elf -rings, s^1elf -fields, s^1elf -spaces. Like any task, this is also a structure of the appropriate capacity. S^1elf -H (s^1elf - hydrogen), like other s^1elf -particles, does not exist in the ordinary, but in fact all s^1elf -molecules, s^1elf -atoms, s^1elf -particles are elements of the energy space.

Remark S.3. The concept of elements of physics S^1t is introduced for energy space. The corresponding concept of elements of chemistry S^1t is introduced accordingly. Examples: 1) $S^1tE_D^{\{a_1q,a_2\ \}}$ – the energy of instantaneous substitution and a_1 by a_2, where a_1, and a_2 are chemical elements, q is instant replacement. Similarly, one can consider for the node of chemical reactions $S^1t_{reaction}^{\{chemical\ elements\ with\ "target\ weights"\}}$. The periodic table its^{1}elf can also be thought of as the S^1t – element: $S^1t_{Mendeleev\ table}^{\{list\ of\ chemical\ elements\}}$ The ideology of S^1t elements allows us to go to the border of the world familiar to us, which allows us to act more effectively.

Dynamical S^1t – elements.

We considered stationary S^1t – elements earlier. Here we consider dynamical S^1t – elements.

Definition S.9. The process of the containment of the set of elements $\{g(t)\} = (g_1(t), g_2(t), ..., g(t))$ in one point x of the space X at time t we shall call dynamical S^1t – element. We shall denote $S^1t(t)_x^{\{g(t)\}}$.

Definition S.10. The process of the containment of the ordered set of elements in one point in space is called dynamical ordered S^1t – element.

It is allowed to multiply dynamical S^1t – elements:

$S^1t(t)_x^{\{g(t)\}} * S^1t(t)_x^{\{b(t)\}}$ $=S^1t(t)_x^{\{g(t)\}\cup\{b(t)\}}$ $(*_{\text{S.1.4}})$,

$S^1t(t)^x_{\{g(t)\}} * S^1t(t)^x_{\{b(t)\}} = S^1t(t)^x_{\{g(t)\}\cup\{b(t)\}}$ $(*_{\text{S.1.5}})$,

where some or any elements may be by ordered elements.

Dynamical S^1t – elements can be elements of a groups by multiplication for $(*_{\text{S.1.4}})$, $(*_{\text{S.1.5}})$.

Dynamical containment of ones^{1}elf.

Definition S.11. Dynamical capacity Q(t) is called the process of a containment in Q(t).

Definition S.12. Dynamical S^1t -capacity $S^1t(t)^{R(t)}_{Q(t)}$ is called the process of a containment R(t) in Q(t).

Definition S.13. The dynamical containment of ones^{1}elf A(t) of the first type is the process of putting A(t) into A(t). Denote $S^1{}_1f(t)A(t)$.

Definition S.14. The dynamical containment of ones^{1}elf C(t) of the second type is the process of a containment of the program that allows C(t) to be generated. Let's denote $S^1{}_2f(t)C(t)$.

Definition S.15. Dynamical partial containment of ones^{1}elf B(t) of the third type is the process of partial containment of B(t) into ones^{1}elf or a program that allows B(t) to be generated in part, or both simultaneously. Let us denote $S^1{}_3f(t)B(t)$.

All dynamical capacities in dynamical s^{1}elf -space are containments of ones^{1}elf by definition. The dynamical containments of ones^{1}elf may to appear as dynamical S^1t -capacities and usual dynamical capacities. In these cases are used usual measure and topology methods.

Connection of dynamical S^1t – elements with dynamical containment of ones^{1}elf. Consider a third type of dynamical partial containment of ones^{1}elf. For example, based on $S^1t(t)^{\{g(t)\}}$, where $\{g(t)\} = (g_1(t), g_2(t), ..., g(t))$, i.e. n - elements in one point x, it is possible to consider the dynamical containment of ones^{1}elf $S^1{}_3f(t)$ with m elements from $\{g(t)\}$, at m<n, which is process to be formed by the form from (1.1) – (1.4). Dynamical containments of ones^{1}elf of the third type can be formed for any other structure, not necessarily S^1t, only through the obligatory reduction in the number of elements in the structure. Structures more complex than $S^1{}_3f(t)$ can be introduced.

Dynamical mathematics its^{1}elf.

1. The process of simultaneous addition of a set of elements $\{g(t)\} = (g_1(t), g_2(t), ..., g(t))$ are realized by $S^1t(t)^{\{g(t)+\}}$.

2. By analogy, for simultaneous multiplication: $S^1t(t)^{\{g(t)*\}}$.

3. Similarly for simultaneous execution of various operations:$S^1t(t)^{\{g(t)q(t)\}}$, where $\{q(t)\} = (q_1(t), q_2(t), .., q_n(t))$. $q_i(t)$-an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $S^1t(t)^{\{F(t)g(t)\}}$where $\{F(t)\} = (F_1(t), F_2(t), ..., F_n(t))$. $F_i(t)$ is an operator, i = 1, . . . , n.

5.The dynamical arithmetic its^{1}elf for containments of ones^{1}elf will be similar: dynamical addition - $S^1{}_1f(t)^{\{g(t)+\}}$, (or $S^1{}_3f(t)_x^{\{g(t)+\}}$ for the third type), dynamical multiplication $S^1{}_1f(t)^{\{g(t)*\}}$, $(or\ S^1{}_3f(t)_x^{\{g(t)*\}}$).

6. Similarly with different operations: $S^1{}_1f(t)^{\{g(t)q(t)\}}$, $(S^1{}_3f(t)_x^{\{g(t)q(t)\}})$, and with different operators: $S^1{}_1f(t)^{\{F(t)g(t)\}}$, $(S^1{}_3f(t)_x^{\{F(t)g(t)\}})$.

7. Similarly for dynamical S^{1}t -derivatives, dynamical S^{1}t -integrals, dynamical S^{1}t -lim, dynamical s^{1}elf - derivatives, dynamical s^{1}elf -integrals

8. Let's denote dynamical s^{1}elf -(dynamical s^{1}elf -Q(t)) through dynamical s^{1}elf ^{2} -Q(t) , fS^1(t)(n,Q(t))= dynamical s^{1}elf -(dynamical s^{1}elf -(. . . (dynamical s^{1}elf -Q(t)))) = dynamical s^{1}elf ^{n}-Q(t) for n-multiple dynamical s^{1}elf.

Remark S.4. Dynamical S^{1}t -displacement of A(t) from B(t) will be denote through ${}^{B(t)}_{A(t)}S^1t(t)$. Then the notation ${}^{C(t)}_{D(t)}S^1t(t)^{A(t)}_{B(t)}$ is dynamical S^{1}t -containment of A(t) in B(t) and dynamical S^{1}t -displacement of D(t) from C(t) simultaneously. Let's denote ${}^{B(t)}_{A(t)}S^1t(t)^{A(t)}_{B(t)}$ through $TS^1(t)^{A(t)}_{B(t)}$, ${}^{A(t)}_{A(t)}S^1t(t)^{A(t)}_{A(t)} - through\ TS^1(t)^{A(t)}_{A(t)}$.

We can consider the concept of dynamical S^{1}t - element as $S^1t(t)^{A(t)}_{B(t)}$, where A(t) fits in dynamical capacity B(t). Then $S^1t(t)^{B(t)}_{B(t)}$ it will mean $S^1{}_1$f(t) B(t). Let's denote $S^1t(t)^{B(t)}_{B(t)}$ through L(t)(B(t)). ${}^{A(t)}_{A(t)}S^1t(t)$ denotes the dynamical expelling ones^{1}elf A(t) out of ones^{1}elf A(t), ${}^{A(t)}_{A(t)}S^1t(t)^{A(t)}_{A(t)}$—simultaneous dynamical containment of ones^{1}elf A(t) in ones^{1}elf A(t) and dynamical expelling ones^{1}elf A(t) out of ones^{1}elf A(t). ${}^A_AS^1t$ will be called anti-capacity from ones^{1}elf. For example, "white hole" in physics is such simple anti-capacity. The concepts of "white hole" and "black hole" were formulated by the physicists proceeding from the physics subjects –usual energies level. The mathematics allows to find deeply and to formulate the concepts singular points in the Universe proceeding from levels of more thin energies. The experiments of Nobel laureates in 2022 year Asle Ahlen, Clauser John, Zeilinger Anton correspond to the concept of the Universe as its s^{1}elf -containment in its^{1}elf. The connection between the elements of s^{1}elf -containment in its^{1}elf is a property of s^{1}elf -containment in its^{1}elf and therefore does not disappear when their location in it changes. The energy of s^{1}elf -containment in its^{1}elf is closed on its^{1}elf.

Hypothesis: the containment of the galaxy in ones^{1}elf as spiral curl and the expelling her out of ones^{1}elf defines its existence. A s^{1}elf -capacity in its^{1}elf as an element A is the god of A, the s^{1}elf -capacity in its^{1}elf as an element the globe—the god of the globe, the s^{1}elf -capacity in its^{1}elf as an element man-- the god of the man, the s^{1}elf -capacity in its^{1}elf as an element of the universe-- the god of the universe, the containment of A into ones^{1}elf is spirit of A, the containment of the globe into ones^{1}elf is is spirit of globe, the containment of the man into ones^{1}elf is spirit of the man (soul), the containment of the universe into ones^{1}elf is spirit of the universe. We may consider next axiom: any capacity is capacity of ones^{1}elf in its^{1}elf. This is for each energy capacity.

About dynamical S^{1}t and S_3^1f(t) programming.

The ideology of dynamical S^{1}t and S_3^1f(t) can be used for programming:

1. The process of simultaneous assignment of the expression$s$ $\{p\} = (p_1, p_2, ..., p_n)$ to the variables $\{g\} = (g_1, g_2, ..., g_n)$ is implemented through $S^1t(t)^{\{\{g\}:\{p\}\}}$.

2. The process of simultaneous check the set of conditions $\{f(t)\} = (f(t)_1,\ f_{2,}\ (t)\ ...,f(t)_n)$ for a set of expressions$\{B(t)\} = (B_1(t),\ B_2\ (t),\ ...,B_n(t))$ is implemented through $S^1t(t)^{IF\{\{B(t)\}\{f(t)\}\}\ then\ Q(t)}$ where Q(t) can be any.

3.Similarly, for loop operators and others.

$S^1{}_3f(t)$– software operators will differ only in these aggregates $\{g\}$, $\{p\}$, $\{B(t)\}$, $\{f(t)\}$ will be formed from corresponding processes S^{1}t(t) for above mentioned programming operators through form (1.1) or form from the forms (1.1.1) - (1.4).

Remark S.5. Using dynamical S^{1}t -elements, we introduce the concepts odynamical S^{1}t - force, dynamical S^{1}t – energy. For example, $E(t)^f_{s^1t} = S^1t(t)^{\{E_1(t)f,E_2(t)\}}_{x(t)}$ it would mean the process of instantaneous replacement f of energy E$_1$(t) by E$_2$(t) at time t. Similarly, using $S_i^1f(t)$ we introduce the concepts of $S_i^1f(t)$ -force, $S_i^1f(t) -$ energy, i=1,2,3, and etc.

Remark S.6. Namely the putting ones^{1}elf into ones^{1}elf may "gives birth" the capacities in its^{1}elf –that's what it is s^{1}elf -organization.

Remark S.7. $S^1t(t)^{S^1t(t)^{B(t)}_{B(t)}}_{S^1t(t)^{B(t)}_{B(t)}}$ may to increase B(t) s^{1}elf level.

Remark S.8. For example, operator its^{1}elf [1] is $S_1^1f(t)$.

Remark S.9. May be considered the next derivatives: $\frac{dS^1t(t)^{A(t)}_{B(t)}}{dt}$, $\frac{d^{B(t)}_{A(t)}S^1t(t).}{dt}$, $\frac{d^{C(t)}_{D(t)}S^1t(t)^{A(t)}_{\quad B(t)}}{dt}$, $\frac{dS_i^1f(t)}{dt}$, i=1,2,3.

Remark S.10. Namely a containment of ones^{1}elf in ones^{1}elf may be interpreted as dynamical capacitiesin its^{1}elf.

Remark S.11. By far not each capacity in its^{1}elf will appear as S^{1}t – capacities or capacities.

S^1t –elements for continual sets.

Earlier we considered finite-dimensional discrete S^1t -elements and s^1elf -capacities in its^1elf as an element. Here we consider some continual S^1t -elements and continual s^1elf -capacities in its^1elf as an element.

Definition S.16. The set of continual elements $\{g\} = (g_1, g_2, ..., g_n)$ at one point x of space X we shall call S^1t – element, and such a point in space is called capacity of the continual S^1t – element. We shall denote $S^1t_x^{\{g\}}$.

Definition S.17. An ordered set of continual elements at one point in space is called an ordered continual S^1t – element.

It's allowed to multiply continual S^1t – elements:

$S^1t_x^{\{g\}} * S^1t_x^{\{b\}} = S^1t_x^{\{g\}\cup\{b\}}$ $(*_5)$

$S^1t_{\{g\}}^x * S^1t_{\{b\}}^x = S^1t_{\{g\}\cup\{b\}}^x$ $(*_6)$

where some or any elements may be by ordered elements.

The operator $S^1t_x^{\{g\}\cup\{b\}}$ is not equal $\{g\} \cup \{b\}$, faster it is dynamical—the compression $\{g\} \cup \{b\}$ into the point x. The continual S^1t – elements can be elements of a groups by multiplication for $(*_6)$ and for $(*_5)$.

Definition S.18. The continual s^1elf -capacity in its^1elf as an element A of the first type is the capacity containing its^1elf as an element. Denote $S^1{}_1fA$.

Definition S.19. The ordered continual s^1elf -capacity in its^1elf as an element A of the first type is the ordered capacity containing its^1elf as an element. Denote $\overrightarrow{S^1{}_1fA}$.

For example, S_∞^+= sin has such type. It denotes continual ordered s^1elf -capacities in its^1elf as an element of next type—the range of simultaneous "activation" of numbers from [-1,1] in mutual directions: $\uparrow I \downarrow_{-1}^{1}$. Also we consider next elements: S_∞^-=sin(-)--$\downarrow I \uparrow_{-1}^{1}$, T_∞^+= tg--$\uparrow I \downarrow_{-\infty}^{\infty}$, T_∞^-=tg(-)--$\downarrow I \uparrow_{-\infty}^{\infty}$, don't confuse with values of these functions. Such elements can be summarized. For example: gS_∞^++bS_∞^-=(g-b)S_∞^+ = $(b-g)$ S_∞^-.

Definition S.20. The continual s^1elf -capacity in its^1elf as an element of the second type is the capacity that contains the program that allows it to be generated. Let's denote $S^1{}_2fA$. An example of continual s^1elf -capacity in its^1elf as an element of the second type is a living organism, since it contains a program: DNA, RNA.

Definition S.21. Partial continual s^1elf -capacity in its^1elf as an element of the third type is called continual s^1elf -capacity in its^1elf as an element, which contains its^1elf in part or contains a program that allows it to be generated in part, or both simultaneously. Let us denote $S^1{}_3fA$.

Also, may be considered operators for them. For example: fS^{+}_{∞}(t-t$_0$) $= \begin{cases} S^{+}_{\infty}\ ,\ t = t_0 \\ 0,\ t \neq\ t_0 \end{cases}$.

All continual capacities in s[1]elf -space are continual s[1]elf -capacities in its[1]elf as an element by definition. The continual s[1]elf -capacities in its[1]elf as an element may to appear as continual S[1]t - capacities and usual continual capacities. In these cases there is used usual measure and topology methods.

s[1]elf -vector $\underset{\leftarrow}{\overset{\rightarrow}{[(0,0),(g,b)]}}$ b

s[1]elf -discrete vector : $\uparrow \begin{pmatrix} q \\ d \\ c \end{pmatrix} \downarrow$ g

s[1]elf -ordered curve Fig. S.2

Fig. S.3.

Connection of continual S[1]t – elements with s[1]elf -capacities in its[1]elf as an element.
Consider a third type of continual s[1]elf -capacity in its[1]elf as an element. For example, based on $S^1t^{\{g\}}$, where $\{g\} = (g_1, g_2, ..., g_n)$, i.e. g_i - continual elements at one point, i=1,2, . . . , n. It's possible to consider the continual s[1]elf -capacity in its[1]elf as an element $S^1{}_3f$ with m continual elements from $\{g\}$, at m<n, which is formed by the form from (1.1) – (1.4). Continual s[1]elf -capacities in its[1]elf as an element of the third type can be formed for any other structure, not necessarily S[1]t, only through the obligatory reduction in the number of continual elements in the structure. Structures more complex than $S^1{}_3f$ can be introduced.

Mathematics its[1]elf for continual elements.

1.Simultaneous addition of a set of continual elements $\{g\} = (g_1, g_2, ..., g_n)$ are realized by $S^1t^{\{g\cup\}}$.

2. By analogy, for simultaneous multiplication: $S^1t^{\{g\cap\}}$.

3. Similarly for simultaneous execution of various operations: $S^1t^{\{gq\}}$where $\{q\} = (q_1, q_2, .., q_n)$. q$_i$ -an operation, i = 1, . . . , n.

4. Similarly, for the simultaneous execution of various operators: $S^1t^{\{Fg\}}$, where $\{F\} = (F_1, F_2, ..., F_n)$. F$_i$ is an operator, i = 1, . . . , n.

5. For continual s[1]elf -capacities in its[1]elf as an element will be similar: addition - $S^1{}_1f^{\{g+\}}$, (or $S^1{}_3f_x^{\{g+\}}$ for the third type), multiplication $S^1{}_1f^{\{g*\}}$, $(S^1{}_3f_x^{\{g*\}})$.

6.Similarly with different operations: $S^1{}_1f^{\{gq\}}$, $(S^1{}_3f_x^{\{gq\}})$. and with different operators: $S^1{}_1f^{\{Fg\}}$, $(S^1{}_3f_x^{\{Fg\}})$.

Remark S.12. S^1t -displacement of A from B will be denote through ${}_A^BS^1t$. Then the notation ${}_D^CS^1t_B^A$ is S^1t -containment of A in B and S^1t -displacement of D from C simultaneously. Let's denote ${}_A^BS^1t_B^A$ through $TS^1{}_B^A$, ${}_A^AS^1t_A^A - through$ $TS^1{}_A^A$. Three in one is $S^1t_\propto^{\{\infty\ in\ itself,\ no\ element,\ \ 0\ out\ oneself\}}$, - point space connectedness.

We can consider the concept of continual S^1t - element as $S^1t_B^A$, where A fits in continual capacity B. Then $S^1t_B^B$, it will mean$S^1{}_1f$B.

Namely such elements are used for S^1t -coding, S^1t translation, coding s^1elf, translation s^1elf for networks, what for electric current of ultrahigh frequency is suitable. May be considered more complex elements as continual sets of numbers with mutual directions "activation" them. For example, ranges of functions values, in particular, functions, which represent lightning form. Certainly, here may be applied differential geometry. Also, may be considered n-dimensional elements. The space of such elements is Banach space if we introduce usual norm for functions or vectors excluding their exceptions. We call this space-- Selb-space. Then we introduce scalar product for functions or vectors excluding their exceptions and get the hilbert space. We call this space-- Selh-space. In particular, may try to describe some processes with these elements by differential equations and to use methods from [28]. Also, may try to apply an optimization and others researches to some processes with these elements by methods from [31]. Let's introduce operators to transform capacity to s^1elf -capacity in its^{1}elf as an element:

Q_1S^1(A) transforms A to f_1S^1A, Q_0S^1(A) transforms A to ${}_A^AS^1t$, S^1O(A) transforms A to$\uparrow A\downarrow$, $\uparrow A\downarrow$ -- ordered s^1elf -capacity in its^{1}elf as an element of simultaneous "activation" of all elements of A in mutual directions. For example, S^1O([-1,1])= S_∞^+, S^1O([1,-1])= S_∞^- , S^1O ([-,])= T_∞^+, S^1O ([,-])= T_∞^-. Operator $(Q_1S^1(A))^2$ increases s^1elf level for A: it transforms S^1elf-A= f_1S^1A to s^1elf^2-A, $(Q_1S^1(A))^n \rightarrow s^1elf^n$-A, $e^{Q_1S^1(A)}\ \rightarrow\ e^{s^1elf} - A$.

Dynamical continual S^1t – elements.

Also, may be considered dynamical continual S^1t -elements, where may be transfer these definitions, operations using on them by analogy:

Definition S.22. The process of the containment of the set of continual elements $\{g(t)\} = (g_1(t), g_2(t), ..., g(t))$ in one point x of the space X at time t we shall call dynamical continual S^1t – element. We shall denote $S^1t(t)_x^{\{g(t)\}}$.

Definition S.23. The process of the containment of ordered set of continual elements in one point in space is called dynamical continual ordered S^1t – element.

It is allowed to multiply dynamical continual S^1t – elements:

$S^1t(t)_x^{\{g(t)\}} * S^1t(t)_x^{\{b(t)\}} = S^1t(t)_x^{\{g(t)\}\cup\{b(t)\}}$ $(*_{\mathrm{S.1.6}})$,

$S^1t(t)^x_{\{g(t)\}} * S^1t(t)^x_{\{b(t)\}} = S^1t(t)^x_{\{g(t)\}\cup\{b(t)\}}$ $(*_{\mathrm{S.1.7}})$,

where some or any elements may be by ordered elements.

Dynamical continual S^1t – elements can be elements of a groups by multiplication for $(*_{\mathrm{S.1.6}})$, $(*_{\mathrm{S.1.7}})$.

Dynamical continual containment of ones^{1}elf.

Definition S.24. Dynamical continual capacity Q(t) is called the process of a containment in Q(t).

Definition S.25. Dynamical continual S^1t -capacity $S^1t(t)_{Q(t)}^{R(t)}$ is called the process of a containment R(t) in Q(t).

Definition S.26. The dynamical containment of ones^{1}elf continual A(t) of the first type is the process of putting A(t) into A(t). Denote $S^1{}_1f(t)A(t)$.

Definition S.27. The dynamical containment of ones^{1}elf continual C(t) of the second type is the process of a containment of the continual program that allows C(t) to be generated. Let's denote $S^1{}_2f(t)C(t)$.

Definition S.28. Dynamical partial containment of ones^{1}elf B(t) of the third type is the process of partial containment of continual B(t) into ones^{1}elf or continual program that allows B(t) to be generated in part, or both simultaneously. Let us denote $S^1{}_3f(t)B(t)$.

Connection of dynamical continual S^1t – elements with dynamical containment of ones^{1}elf.
Consider a third type of dynamical continual partial containment of ones^{1}elf. For example, based on $S^1t(t)^{\{g(t)\}}$, where $\{g(t)\} = (g_1(t), g_2(t), ..., g(t))$, i.e. n - continual elements in one point x, it is possible to consider the dynamical containment of ones^{1}elf $S^1{}_3f(t)$ with m continual elements from $\{g(t)\}$, at m<n, which is process to be formed by the form from (1.1) – (1.4). Dynamical continual containments of ones^{1}elf of the third type can be formed for any other structure, not necessarily S^1t, only through the obligatory reduction in the number of continual elements in the structure. Structures more complex than $S^1{}_3f(t)$ can be introduced.

Dynamical continual mathematics its^{1}elf.

1. The process of simultaneous addition of a set of continual elements $\{g(t)\} = (g_1(t), g_2(t), ..., g(t))$ are realized by $S^1t(t)^{\{g(t)\cup\}}$.

2. By analogy, for simultaneous multiplication: $S^1t(t)^{\{g(t)\cap\}}$.

3. Similarly for simultaneous execution of various operations:$S^1t(t)^{\{g(t)q(t)\}}$, where $\{q(t)\} = (q_1(t), q_2(t), .., q_n(t))$. q_i(t)-an operation, i = 1,. . . ,n.

4. Similarly, for the simultaneous execution of various operators: $S^1 t(t)^{\{F(t)g(t)\}}$where $\{F(t)\} = (F_1(t), F_2(t), ..., F_n(t))$. $F_i(t)$ is an operator, i = 1,. . . ,n.
5. The dynamical arithmetic its¹elf for containments of ones¹elf will be similar: dynamical addition - $S^1{}_1 f(t)^{\{g(t)\cup\}}$, (or $S^1{}_3 f(t)_x^{\{g(t)\cup\}}$ for the third type), dynamical multiplication $S^1{}_1 f(t)^{\{g(t)\cap\}}$, $(or\ S^1{}_3 f(t)_x^{\{g(t)\cap\}}$).
6. Similarly with different operations: $S^1{}_1 f(t)^{\{g(t)q(t)\}}$, $(S^1{}_3 f(t)_x^{\{g(t)q(t)\}})$, and with different operators: $S^1{}_1 f(t)^{\{F(t)g(t)\}}$, $(S^1{}_3 f(t)_x^{\{F(t)g(t)\}})$.
7. Similarly for dynamical S^1t -derivatives, dynamical S^1t -integrals, dynamical S^1t -lim, dynamical s^1elf - derivatives, dynamical s^1elf-integrals
8. Let's denote dynamical s^1elf -(dynamical s^1elf -Q(t)) through dynamical s^1elf -Q(t) , fS^2(t)(n,Q(t))= dynamical s^1elf -(dynamical s^1elf -(. . . (dynamical s^1elf -Q(t)))) = dynamical s^1elf ^{n}-Q(t) for n-multiple dynamical s^1elf.

Remark S.13. Dynamical S^1t -displacement of A(t) from B(t) will be denote through ${}^{B(t)}_{A(t)}S^1 t(t)$. Then the notation ${}^{C(t)}_{D(t)}S^1 t(t)^{A(t)}_{B(t)}$ is dynamical S^1t -containment of A(t) in B(t) and dynamical S^1t -displacement of D(t) from C(t) simultaneously. Let's denote ${}^{B(t)}_{A(t)}S^1 t(t)^{A(t)}_{B(t)}$ through $TS^1(t)^{A(t)}_{B(t)}$, ${}^{A(t)}_{A(t)}S^1 t(t)^{A(t)}_{A(t)} -$ $through\ TS^1(t)^{A(t)}_{A(t)}$.

We can consider the concept of dynamical S^1t - element as $S^1 t(t)^{A(t)}_{B(t)}$, where A(t) fits in dynamical capacity B(t). Then $S^1 t(t)^{B(t)}_{B(t)}$ it will mean $S^1{}_1 f(t)^{B(t)}$. Let's denote $S^1 t(t)^{B(t)}_{B(t)}$ through L(t)(B(t)). ${}^{A(t)}_{A(t)}S^1 t(t)$ denotes the dynamical expelling ones¹elf A(t) out of ones¹elf A(t), ${}^{A(t)}_{A(t)}S^1 t(t)^{A(t)}_{A(t)}$—simultaneous dynamical containment of ones¹elf A(t) in ones¹elf A(t) and dynamical expelling ones¹elf A(t) out of ones¹elf A(t). ${}^{A(t)}_{A(t)}S^1 t(t)$will be called anti-capacity from ones¹elf. For example, "white hole" in physics is such simple anti-capacity. The concepts of "white hole" and "black hole" were formulated by the physicists proceeding from the physics subjects –usual energies level. The mathematics allows to find deeply and to formulate the concepts singular points in the Universe proceeding from levels of more thin energies. Hypothesis: the containment of the galaxy in ones¹elf as spiral curl and the expelling her out of ones¹elf defines its existence. A s^1elf -capacity in its¹elf as an element A is the god of A, the s^1elf -capacity in its¹elf as an element the globe—the god of the globe, the s^1elf -capacity in its¹elf as an element man-- the god of the man, the s^1elf -capacity in its¹elf as an element of the universe-- the god of the universe, the containment of A into ones¹elf is spirit of A, the containment of the globe into ones¹elf is is spirit of globe, the containment of the man into ones¹elf is spirit of the man (soul), the containment of the universe into ones¹elf is spirit of the universe. We may consider next axiom: any capacity is capacity of ones^{1}elf in its^{1}elf. This is for each energy capacity. The experiments of Nobel laureates in 2022 year Asle Ahlen, Clauser John, Zeilinger Anton correspond to the concept of the Universe as its s^{1}elf -containment in its^{1}elf. The connection between the elements of s^{1}elf -containment in its^{1}elf is a property of s^{1}elf -containment in its^{1}elf and therefore does not disappear when their location in it changes. The energy of s^{1}elf -containment in its^{1}elf is closed on its^{1}elf.

Connection of dynamical continual S^{1}t – elements with target weights with dynamical continual containment of ones^{1}elf with target weights.

It is allowed to multiply dynamical continual S^{1}t – elements with target weights w(t):

$S^1t(t)_x^{\{g(t)w(t)\}} * S^1t(t)_x^{\{b(t)w(t\}} = S^1t(t)_x^{\{(g(t)\}\cup\{b(t))w(t)\}}$ $(*_{\mathrm{S.1.8}})$,

$S^1t(t)^x_{\{g(t)w(t)\}} * S^1t(t)^x_{\{b(t)w(t)\}} = S^1t(t)^x_{\{g(t)\}\cup\{b(t)\}w(t)}$ $(*_{\mathrm{S.1.9}})$,

where some or any elements may be by ordered elements.

Dynamical continual S^{1}t – elements with target weights can be elements of a groups by multiplication for $(*_{\mathrm{S.1.8}})$, $(*_{\mathrm{S.1.9}})$.
Consider a third type of dynamical partial containment of ones^{1}elf with target weights w(t). For example, based on $S^1t(t)^{\{g(t)w(t)\}}$, where $\{g(t)\} = (g_1(t), g_2(t), ..., g(t))$, i.e. n - continual elements with target weights{w(t)} in one point x, it is possible to consider the dynamical containment of ones^{1}elf with target weights $S^1{}_3f(t)w(t)$ with m continual elements with target weights{w(t)} from $\{g(t)\}$, at m<n, which is process to be formed by the form from (1.1) – (1.4). Dynamical containments of ones^{1}elf with target weights of the third type can be formed for any other structure, not necessarily S^{1}t, only through the obligatory reduction in the number of continual elements with target weights in the structure. Structures more complex than $S^1{}_3f(t)w(t)$ can be introduced.

Definition S.29. The dynamical containment of ones^{1}elf continual A(t) with target weights {g(t)} of the first type is the process of putting A(t) into A(t) with target weights. Denote $S^1{}_1f(t)A(t)w(t)$

Definition S.30. The partial dynamical containment of ones^{1}elf continual C(t) with target weights{w(t)} of the second type is the process of a containment of the continual program that allows C(t) to be generated. Let's denote $S^1{}_2f(t)C(t)w(t)$.

Definition S.31. Dynamical partial containment of ones^{1}elf B(t) with target weights {w(t)} of the third type is the process of partial containment of continual B(t) into ones^{1}elf or continual program that allows B(t) to be generated partially. Let us denote $S^1{}_3f(t)B(t)w(t)$.

The usage of S^{1}t -elements for networks.

The all-encompassing monograph of Galushkin A. [40] embraces all aspects of networks but usual traditional approaches to networks are through classical mathematics, in particular through usual conformity operators. Here we consider another approach – through new mathematics partition with containment operators, which though may be interpreted as a result of some conformity operators, but themselves are no conformity operators. The containment operators are more convenient for networks. Also, main lay stress on the processors use, which work with triodes use, that does not use in S^1t -networks in mainly. S^1t -networks is represented by S^1t -structure, which may be constructed for necessary weights. S^1t -OS (S^1t-operating system) are used S^1t -coding and S^1t -translation. In the first the coding is realized through 2-measured matrix –row (a, b), where the number b – the code of the action, the number a- the object code of this action. S^1t -coding (or s^1elf -coding) is realized through the matrix, which has 2 columns (in continuous case- 2 numbers intervals). Here initial coding is used for all matrix rows simultaneously. S^1t -translation is realized by the inversion. In this case s^1elf -coding and s^1elf -translation will be more stable in particular. The target weights $\mathrm{f_i}$ in $S^1t_a^{\{fx\}}$ are chosen for necessary tasks. We will touch no questions of the applications, optimization of networks. They are detailed by A. Galushkin [40]. We touch difference of it for complex networks hierarchy only. The same simple executing programs are in the cores of simple artificial neurons of type S^1t (designation - mnS^1t) for simple information processing. More complex executing programs are used for mnS^1t nodes. Unfortunately, we change name S^1t -elements to S^1t -elements because we find that S^1t -elements were used by other authors early. S^1t -threshold element –sgn($S^1t_b^{\{gx\}}$), b- mnS^1t, x=($\mathrm{x_{1,}x_{2,\ldots,}x_n}$) – source signals values, $\{g\} = (g_1, g_2, ..., g_n)$ – S^1t -synapses weights. The first level of mnS^1t consists from simple mnS^1t . The second level of mnS^1t consists from $S^1t_D^{\{mnS^2t\}}$ – S^1t -node of mnS^1t in range D, D- capacity for mnS^1t node. The third level of mnS^1t consists from $S^1t_D^{\{S^1t_D^{\{mnS^1t\}}\}}$- S^1t ^{1}- node of mnS^1t in range D, thus D becomes capacity in its^{1}elf for mnS^1t. The usage of S^1t ^{1}- nodes of mnS^1t is enough for our networks, but s^1elf level is more higher in living organisms, in particular S^1t $^\mathrm{n}$-, n3. Target structure or corresponding eprogram by corresponding s^1elf -code enters to target block by means of alternating current. After that here takes place the activation of all networks or its part according to indicative target. May arise the opinion, that we go out from networks ideology, but in fact networks presents complex hierarchy with capacity in its^{1}elf of different levels in living organisms.

Remark S.14. Traditional scientific approaches through classical mathematics allows to describe only on usual energy level. Here is approach- on more thin energy level.

In mnS^1t are $S^1t_{mnS^1t}^{\{eprograms\}}$, $eprogram-$executing program in S^1t - OS . In this connection S^1t -OS (or S^1elf-OS) is based on S^1t -assembly language (or S^1elf - assembly language), which is based on assembly language through S^1t -approach in turn in the case of the sufficiency of the S^1t -networks elements base. The eprograms are in S^1t -programming environments (or S^1elf - programming environments), but this question and S^1t -networks base will be considered in next articles. In particular, eprograms may contain S^1t - programming operators. In mnS^1t cores the constant memory S^1t with correspondent eprograms depending on mnS^1t.

The ideology of S^1t and $S^1{}_3f$ can be used for programming. Here are some of the

Quite interesting is the OS (operating system), the principles and modes of operation of the S^1t -networks for this programming. But this is already the material of the next articles.

Here is based on the elements of S^1t – physics and special neural networks with artificial neurons operating in normal and S^1t – modes, a model of a helicopter without a main and tail rotors was developed. Let's denote this model through S^1mnS^1t. To do this, it's proposed to use mnS^1t of different levels, for example, for the usual mode, mnS^1t serves for the initial processing of signals and the transfer of information to the second level, etc. to the nodal center, then checked and in case of anomaly - local S^1t – mode with the desired "target weight" is realized in this section, etc. to the center. Here, in case of anomaly during the test, S^1mnS^1t is activated with the desired "target weight". Here are realized other tasks also. To reach the s^1elf -energy level, the mode $S^1t^{SmnS^1t}_{SmnS^1t}$ is used. In normal mode, it's planned to carry out the movement of $SmnS^1t$ on jet propulsion with the conversion of the energy of the emitted gases into a vortex, to obtain additional thrust upwards. For this purpose, a spiral-shaped chute (with "pockets") is arranged at the bottom of the S^1mnS^1t for the gases emitted by the jet engine, which first exit through a straight chute connected to the spiral one. There is a drainage of exhaust gases outside the S^1mnS^1t. Otherwise, $SmnS^1t$ is represented by a neural network that extends from the center of one of the main clusters of S^1t - artificial neurons to the shell, turning on into the shell its^{1}elf. Above the operator's cabin is the central core of the neural network and the target block, which is responsible for performing the "target weights" and auxiliary blocks, the functions and roles of which we will discuss further. Next is the space for the movement of the local vortex. The unit responsible for Smnst's actions is located below the operators' cab. In S^1t – mode the entire network or its sections are S^1t – activated to perform certain tasks, in particular, with "target weights". In target block are used S^1t -coding, S^1t -translation for activation all networks to "target weights" simultaneously, then –the reset of this S^1t -coding after activation. Unfortunately, triodes are not suitable for S^1t -neural network. In the most primitive case usual separaters with corresponding resistances and core for eprograms may be used instead triodes since there is not necessity in the unbending of the alternating current to direct. The belt of S^1t -memory operative is disposed around central core of $SmnS^1t$. There are S^1t -coding, S^1t -translation, S^1t -realize of eprograms and of the programs from the archives without extraction theirs. For S^1t -coding and S^1t -translation may be use high-intensity, ultra-short optical pulses laser of Nobel laureates 2018 year Gerard Mourou, Donna, Strickland.

S^1t – structure or a eprogram if one is present of needed «target weight» are taken in target block at S^1t – activation of the networks. $S^1t^{SmnS^1t,f}_{activation}$ derives S_{mnS^1t} to the s^1elf level boundary with target weight f.

It's used an alternating current of above high frequently and ultra-violet light, which are able to work with S^1t – structures in S^1t – modes by it‘s nature for an activation of the networks or some of its parts in S^1t – modes and at local using S^1t – mode. Above high frequently alternating current go through mercury bearers that overheating does not occur. The power of the alternating current of above high frequently increase considerably for target block. The activation of all network is realized to indicative "target weights".

Supplement

Connection S^1t – elements with usual functionals and operators

We consider functional g(x): X→g, x∈ X, g—numerical value of functional g(x). It is specific capacity for X. $S^1t^{\{g(x)\}}_{\{g(x)\}}$ made from her the s^1elf -capacity in its^{1}elf as an element $S^1{}_1f$ {g(x)}, {g(x)}—all functional for all X. In particular, probability p(X) –is such functional, X—an event. Here $S^1t^{p(X)}_{p(X)}$—$S^1{}_1f$ p(X), denote it through pS^1(X). Usual event is dynamical capacity.

Definition S.32. S^1t -probability of events A, B is p($S^1t^A_B$), denote $S^1p^A_B$. In particular,

$S^1p^A_B$ for joint A, B: $S^1p^A_B$= p($S^1t^A_B$)=p({AB-AB, D})= p(A)+p(B) – p(AB)+pS^1(D), D-- the s^1elf-capacity in its^{1}elf as an element from AB, pS^1(D)—probability s^1elf of D of next level—s^1elf level. The probability for stochastic value X is capacity. We represent their distribution in the kind of S^1t -element:

$S^1t(t)^{\{(x_1,p_1),(x_2,p_2),\ldots,(x_n,p_n)\}}_X$ $(*_{S.1.10})$

Here interest represent partial distribution s^1elf from $(*_{S.1.10})$ by the form from (1.1) – (1.4) with value s^1elf of stochastic value X for some subset {x$x_{1_1}, x_{2_1}, \ldots, x_{j_1}$} ∈{$x_1$, x_2, …, x_n} with probabilities s^1elf {$pS^1{}_1$, $pS^1{}_2$, …, $pS^1{}_j$}.

For operator $X_1 \overset{F}{\to} X_2$: $\overset{F}{\to}$ X_2 is capacity for X_1. $S^1t^{\overset{F}{\to}X2}_{\overset{F}{\to}X2}$ -- s^1elf -capacity in its^{1}elf as an element for X_1. More complex for implicit operator: F(X_1, X_2)=0. Then $S^1t^{F(X1,X2)=0}_{F(X1,X2)=0}$ forms s^1elf -capacity in its^{1}elf as an element for X_1 relatively of X_2 or for X_2 relatively of X_1.

x obtains more power of the liberty and in this is direct decision (i.e., s^1elf -capacity in its^{1}elf as an element for x). S^1elf -equation for x has its decision for x in direct kind. S^1elf -task for x has its decision for x in direct kind. S^1elf

-question has its answer for x in direct kind. x acquires more degree of liberty and in this is direct decision.

We consider $S^1t^D_D$, D-block over execution subject in ${S^1}_{mnS^1t}$ for networks. Then we have s^1elf -capacity in its^{1}elf as an element D, where full realization requires correspondent s^1elf -energy. $S^1t^{SmnS^1t}_{SmnS^1t}$ increase s^1elf level of ${S^1}_{mnS^1t}$ and may made no visual its.

Conclusions: New concepts and new processing methods of information based on them and new software operators were introduced. We have a fairly non-classical section of mathematics, so the presentation of the material is a bit non-classical to emphasize this. Classical science reaches the horizon and beyond the horizon through chance, while we immediately start from the horizon. We have found a way to construct some mathematical class of complex processes with which classical science can work through chance at best. Classical science proceeds from the objective components, while we immediately start from the wave ones, bypassing the objective ones. Further development is associated with changing the structure of the arithmetic-logical device, the corresponding software and application for new technologies, in the light of the new approach. The entire neural network as instantaneous simultaneous RAM in S1t elements and s^1elf-elements.$s^1elf^{s^1elf^{\cdots^{s^1elf}}}$, $f1{\downarrow}\, I \uparrow^1_{-1} f_2{}^{f1\downarrow I\uparrow^1_{-1}f_2{}^{\cdots^{f1\downarrow I\uparrow^1_{-1}f_2}}}$, $sin\infty^{sin\infty^{\cdots^{sin\infty}}}$.When activated in a neural network, the entire neural network becomes a working memory. Use of s^1elf -energy as activation or from outside. ${Q^1}_0= S^1t^{S^1t^{SmnS^1t}_{activation}}_{S^1t^{SmnS^1t}_{activation}} \rightarrow$ s^1elf -RAM, ${Q^1}_{00}={}^{S^1mnS^1t}_{activation}{}^{S^1t}_{S^1mnS^1t}{}_{activation}{}_{S^1t}{Q^1}_0$, ${Q^1}_{01}={}^{S^1t^{S^1mnS^1t}_{activation}}_{S^1t^{S^1mnS^1t}_{activation}}{Q^1}_0$.

${Q^1}_0$, ${Q^1}_{00}$, ${Q^1}_{01}$-coding, translation, realization eprograms, ${Q^1}_0$, ${Q^1}_{00}$, ${Q^1}_{01}$-${S^1}_{mnS^1t}$, Assembler

Supplement 2

Some aspects of dental restoration through the use of neural networks

Here proposes the principle of a new approach to dental restoration using a neural network of a fundamentally new type - directly parallel, similar to the human nervous system. The field of the given structure tw is used for the activation of networks. The field can remain in effect until it is executed tw. Here all stages of the structure tw can be executed directly in parallel, in particular, an algorithm for solving the desired problem. We will call this field the operational activation field. This field will be created according to the structure tw. We will try to interpret the energy of amorphous objects through a mathematical model $Sprt_q^q$. Then we will try to interpret the energy of the crystals through a mathematical model $Sprt_{Sprt_q^q}^{Sprt_q^q}$. Then we will try to interpret the energy of living crystals, in particular teeth, through a mathematical model $Sprt_{St_{St_q^q}^{St_q^q}}^{St_{St_q^q}^{St_q^q}}$ or

$Sprt_{Sprt_q^{Sprt_q^q}}^{Sprt_q^{Sprt_q^q}}$, teeth damaged by caries - $Sprt_{Sprt_{Sprt_{q-}^q}^{Sprt_{q-}^q}}^{Sprt_{Sprt_{q-}^q}^{Sprt_{q-}^q}}$ $(1_{S.2})$

or

$Sprt_{Sprt_{q-}^{Sprt_{q-}^q}}^{Sprt_{q-}^{Sprt_{q-}^q}}$ $(2_{\ S.2})$,

where may be small and corresponds to the missing area of the tooth that has been affected by caries. Then we will try to done by using the required activation of S_{mnSprt} the following transformation:

Then we will try to fulfill dental restoration through transformation: $Sprt_{Sprt_{Sprt_{q-}^q}^{Sprt_{q-}^q}}^{Sprt_{Sprt_{q-}^q}^{Sprt_{q-}^q}} \to Sprt_{Sprt_{Sprt_q^q}^{Sprt_q^q}}^{Sprt_{Sprt_q^q}^{Sprt_q^q}}$ or $Sprt_{Sprt_{q-}^{Sprt_{q-}^q}}^{Sprt_{q-}^{Sprt_{q-}^q}} \to Sprt_{Sprt_q^{Sprt_q^q}}^{Sprt_q^{Sprt_q^q}}$.

Let's consider two scenarios:

a) with complete restoration of the tooth structure:

   using ($***_H$) during activation of S_{mnSprt} with target weight tw = complete restoration of the tooth structure through $(1_{\ S.2})$ or $(2_{\ S.2})$,

b) with partial restoration of the tooth structure:

   using ($***_H$) during activation of S_{mnSprt} with target weight tw = the specified type partial restoration of the tooth structure through $(1_{\ S.2})$ or $(2_{\ S.2})$.

Different types of neural networks: based on living structure

1) tree
2) insect
3) virus
4) etc

The energy of the container itself {A}, abstracted from its contents $\{\}_A$:

$$Sprt^{\{\}_A}_{\{\}_A},\ {}^{\{\}_A}_{\{\}_A}Sprt^{\{\}_A}_{\{\}_A}$$

Mathematics, theoretical science and programming are actually concerned with structuring their spaces and the capacities within these spaces. Experimental science and experimental programming are concerned with the contents in real spaces and the capacities in those spaces. Ordinary level programming is a self- theoretical science, self-programming is programming oneself, in particular programming environments, self^2-programming is, in particular, programming the creation of programming environments etc. (oself-self) -science - this is work S_{mnSprt} in activation mode, for example, by a short-pulse laser or through microwave current with minimum amplitude and maximum frequency or ultraviolet; the same applies to practical (experimental) programming with real objects and energies [1].

Implicitness is a sign of a mathematical cocoon of a mathematical equation (problem). The "form" of implicitness is the mathematical cocoon (structure). **Conclusions** New concepts and new processing methods of information based on them and new software operators were introduced. We have a fairly non-classical section of mathematics, so the presentation of the material is a bit non-classical to emphasize this. Classical science reaches the horizon and beyond the horizon through chance, while we immediately start from the horizon. We have found a way to construct some mathematical class of complex processes with which classical science can work through chance at best. Classical science proceeds from the objective components, while we immediately start from the wave ones, bypassing the objective ones. Further development is associated with changing the structure of the arithmetic-logical device, the corresponding software and application for new technologies, in the light of the new approach.If you transform through self, i.e. through the middle level, using the reserve that is formed in the self-object by definition, the self-object can "catch" on to another object and as a result form a new self-object. For example, a simplest chemical reaction, when one simplest inorganic self-object is obtained from two inorganic simplest self-objects and, as a rule, energy is released. Next come cyclic organic molecules with increasing degree of self ("folding").The simplest organic self-object benzene with delocalized orbitals: 3 bonding orbitals↑ I ↓3 antibonding orbitals.Next come DNA and RNA with a complex multi-cyclic helical structure with a significantly increasing degree of self-organization ("folding"). If you transform through ||| (identification), i.e. through the upper level, then this can usually be done through replacement: "replacing" what is available

with what is needed. Also, through the upper level (|||), you can go to the virtual double of the self-object (in fact, the self-virtual object from this one) to perform unusual actions, as well as perform other manipulations, in particular, the creation of new singularities of the upper level. By gaining the necessary energy, the self-object can turn into self-action, generating a singularity of the upper level. We consider the usual (sequential) hierarchy for levels: the 0th level corresponds to usual energies, the next higher lower level corresponds to objects, the next higher middle level corresponds to self and finally the upper level corresponds to |||. This is the most simplified representation (simplified 2-interpretation). Of course, we can also consider more complex representations with greater possibilities, in particular, through more complex hierarchies: parallel hierarchies, self-hierarchies, |||-hierarchies, etc.In particular, one can use the hierarchy A = |||B (identification of the hierarchies set B). From it one can make all sorts of self-hierarchies by (1.1) – (1.4). It is possible to introduce and use ||| - science (science of synthesis) by Smnsprt, in particular, ||| - mathematics (mathematics of synthesis), - self-science, in particular, self-mathematics. This is of particular interest for biology, in particular, work with “virtual” components of living organisms, DNA, RNA, etc., and in general with “virtual” components of science and technology. We have simply outlined some aspects; it is too early to develop them without experimental study. All problems have a solution, the only question is: a) in complete identification with the problem, b) the “breadth of the field” of values that you can provide. Considering object A as the value of dynamic variable P, we can reassign P:= B at the top level and instead of A we will have object B, and also perform any other parallel dynamic operations with objects. That is, our world will be perceived as a Dynamic Programming environment. Dynamic programming is an extension of science for "working" with the top and middle levels. Here information is transformed into actions.
Energy Internet: a) transmission of necessary singularities (subtle energies) (connection to necessary singularities (subtle energies)), b) transmission of necessary self-structures (subtle energies) (connection to necessary self-structures) etc.
It is also necessary to construct a bioSmnsprt on living, grown elements of the central nervous system, in particular, neurons. It is also possible to construct another version of bioSmnsprt, using a viral model, on living, grown bacteriophages.

REFERENCES:

1) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3, pp.1-31, March 27, 2024 https://cskscientificpress.com/articles$_{f}ile/527-article$1712797848*.pdf*

2) Danilishyn I., Danilishyn O. PROGRAM OPERATORS SIT, tS, S1e, Set1. Preprint. Research Square. : 2023-08-04 08:00:27 https://www.researchsquare.com/article/rs-3228799/v1 https://doi.org /10.21203/rs.3.rs-3228799/v1

3) Oleksandr Danilishyn and Illia Danilishyn. Fuzzy Dynamic Fuzzy Sets. Variable Fuzzy Hierarchical Dynamic Fuzzy Structures (Models, Operators) for Dynamic, Singular, Hierarchical Fuzzy Sets. FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS1epr, ffSeprt1. Journal of Mathematical Techniques Computational Mathematics (JMTCM), Volume 3 | Issue 4 |2024, Apr 17, pp. 1 – 37 Journal DOI: 10.33140/JMTCM https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

4) Oleksandr Danilishyn and Illia Danilishyn. Introduction to Section of Dynamic Mathematics: Dynamic Sets Theory: Parallel Sprt-Elements and their Applications. J Math Techniques Comput Math (Journal DOI: 10.33140/JMTCM), Volume 3 | Issue 5 (2024), pp.1- 27. https://www.opastpublishers.com/open-access-articles/introduction-to-section-of-dynamic-mathematics-dynamic-sets-theory-parallel-sprtelements-and-their-applications.pdf

5) Oleksandr Danilishyn, Illia Danilishyn. Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), 2023, 479-498. https://www.opastpublishers.com/open-access-articles/dynamical-sets-theory-s2telements-and-their-applications.pdf

6) Danilishyn I., Danilishyn O. Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care 2023, Volume 4 | Issue 2 | 66- 81. https://doi.org/10.33140/NABCC.04.02.02

7) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-_ article1695707465.pdf (mkscienceset.com)

8) I. Danilishyn, O. Danilishyn Dynamic Sets Se, Networks Se. *Advances in Neurology and Neuroscience*, 2023, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

9) I. Danilishyn, O. Danilishyn Program Operators Sit, tS, S1e, Set1. Journal of Sensor Networks and Data Communications [ISSN: 2994-6433] 2023, Volume

3 | Issue 1 | 138-143, https://www.opastpublishers.com/table-contents/jsndc-volume-3-issue-1-year-2023

10) 10) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square. 2023-08-01 04:20:16 https://doi.org/10.21203/rs.3.rs-3217178/v1 Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

11) Oleksandr Danilishyn and Illia Danilishyn. Intrtoduction to section of Dynamic mathematics: DYNAMIC fuzzy SETS THEORY: Parallel fuzzy Sprt – elements AND THEIR APPLICATIONS. J Math Techniques Comput. Math, 3(7), 2023, 1 – 27.

12) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference "GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ", 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

13) Danilishyn .V. Danilishyn O.V. MATHEMATICS ST, PROGRAMMING OPERATORS ST AND SOME EMPLOYMENT. III International Scientific and Theoretical Conference «Theoretical and practical scientific achievements: research and results of their implementation» *07.04.2023 | Pisa, Italia.* Collection of scientific papers "SCIENTIA", 2023. **https://previous.scientia.report/index.php/archive/issue /view/07.04.2023**

14) Danilishyn .V. Danilishyn O.V. DYNAMICAL SIT-ELEMENTS . IV International Scientific and Practical Conference "GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ", 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/is sue/view/9

15) Danilishyn .V. Danilishyn O.V. tS - ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

16) Danilishyn .V. Danilishyn O.V. SET1 - ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June 2023 with the proceedings of the: Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine)

LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

17) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO SETS THEORY AND OTHERS. Scientific practice: modern and classical research methods: Collection of scientific papers «O» with Proceedings of the IV International Scientific and Practical Conference, Boston, May 26, 2023. Boston-Vinnytsia: Primedia eLaunch & European Scientific Platform, 2023, pp.166-171. **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/11**

18) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO CONTINUALVALUED LOGIC AND OTHERS. Features of the development of modern science in the pandemic's era: a collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, May 19, 2023. Berlin, Federal Republic of Germany: European Scientific Platform, pp.79-84. **https://previous.scientia.report/index.php/archive/issue/view /19.05.2023**

19) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO SETS THEORY.Development of modern science: topical issues of theory and practices: materials of the III All-Ukrainian Student Scientific Conference conference, Kharkiv, May 19, 2023 / NGO "Youth Scientific league". - Vinnytsia: NGO "European Scientific Platform", 2023,pp. 270-272. .

20) Danilishyn .V. Danilishyn O.V. CONTINUAL SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS. Theoretical and practical aspects of modern scientific research: Collection of scientific papers «O» with Proceedings of the II International Scientific and Practical Conference, Seoul, April 28, 2023. Seoul-Vinnytsia: Case Co., Ltd.& European Scientific Platform, **pp. 144-150. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/10**

21) Danilishyn .V. Danilishyn O.V. SIT-DYNAMICAL AND CONTINUAL ELEMENTS. MATERIALS OF THE VIII INTERNATIONAL PRIORITY SCIENTIFIC AND PRACTICAL CONFERENCE DIRECTIONS OF RESEARCH IN SCIENTIFIC AND EDUCATIONAL ACTIVITIES April 29 - 30 Lviv in 2023 , pp.24-31. http://lviv-forum.inf.ua/save/2023/29-30.04

22) Danilishyn .V. Danilishyn O.V. CONTINUAL SIT-ELEMENTS WITH TARGET WEIGHTS. Formation of modern science: methodology and practice: materials III All-Ukrainian Student Scientific Conference, Uzhgorod, 21 April 2023 / NGO "Youth Scientific League". - Vinnytsia: NGO "European Scientific Platform", 2023 **pp. 105-107. https://archive.liga.science/index.php/conference-proceedings/issue/view/ukr-21.04.2023**

23) Danilishyn .V. Danilishyn O.V. SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS FOR CONTINUAL SETS. Modernization of science and its influence on global processes: collection of scientific papers «SCIENTIA» with Proceedings of the III International Scientific and Theoretical Conference, April 14, 2023. Bern, Swiss Confederation: European Scientific Platform. Pp.61-66. **https://previous.scientia.report/index.php/archive/issue/view/14.04.2023**

24) Danilishyn .V. Danilishyn O.V. SIT-ELEMENTS AND THEIR APPLICATIONS. LVIV SCIENTIFIC FORUM MATERIALS OF THE VIII INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE THEORY AND PRACTICE OF MODERN SCIENCE AND EDUCATION March 19-20, 2023, Lviv, pp.33-38. http://lviv-forum.inf.ua/save/2023/29-30.04/

25) Danilishyn .V. Danilishyn O.V. SIT-NETWORKS. Lviv SCIENTIFIC FORUM MATERIALS OF THE VIII INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE THEORY AND THE PRACTICE OF MODERN SCIENCE AND EDUCATION March 19–20, 2023 Lviv, pp.39- 41. http://lviv-forum.inf.ua/save/2023/29-30.04/

26) Danilishyn .V. Danilishyn O.V. MATHEMATICS ST, PROGRAMMING OPERATORS STAND SOME EMPLOYMENT comprehensive approach to the modernization of science: methods, models, and multidisciplinarity: materials of the II International scientific conference, Lutsk, March 3, 2023. / International Center scientific research. - Vinnytsia: European Scientific Platform, 2023,**p.p. 114-119. https://archive.mcnd.org.ua/index.php/conference-proceeding/issue/view/03.03.2023**

27) Danilishyn .V. Danilishyn O.V. ST –ELEMENTS APPLICATIONS FOR SOME TASKS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 24 February 2023 with the proceedings of the: Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 17 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), pp.400-402. https://archive.journal-grail.science/index.php/2710-3056/issue/view/17.02.2023

28) Danilishyn .V. Danilishyn O.V. THE INTRODUCTORY CONCEPTS AND OPERATIONS OF ST MATHEMATICS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 24 February 2023 with the proceedings of the: Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 17 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), pp.403- 406. https://archive.journal-grail.scie

nce/index.php/2710-3056/issue/view/17.02.2023**https://archive.journal-grail.science/index.php/2710-3056/issue/view/17.02.2023**

29) Danilishyn .V. Danilishyn O.V. SOME St – ELEMENTS APPLICATIONS. Information society: technological, economic and technical aspects formation (issue 74): materials of the International Scientific Internet conferences (Ternopil, Ukraine – Perevorsk, Poland, February 6-7, 2023) / [editor. : O. Patryak and others] ; GO “Naukova community”; WSSG in Przeworsk. - Ternopil. http://www.konferenciaonline.org.ua/ua/article/id-947/

30) Danilishyn I.V. Danilishyn O.V. MATHEMATICS ST, PROGRAMMING OPERATORS STAND SOME EMPLOYMENT A comprehensive approach to the modernization of science: methods, models and multidisciplinarity: materials of the II International Scientific Conference, Lutsk, March 3, 2023. / International Center for Scientific Research. — Vinnytsia: European Scientific Platform, 2023, p.p. 114-119. https://archive.mcnd.org.ua/index.php/conference-proceeding/issue/view/03.03.2023

31) Danilishyn .V. St – ELEMENTS AND PROGRAMMING OPERATORS.Informational society: technological, economic and technical aspects of formation (issue 74): materials of the International Scientific Internet Conference, (m. Ternopil, Ukraine – Perevorsk, Poland, March 6-7, 2023) / [editor. : O. Patryak and others] ; NGO “Scientific community”; WSSG w Przeworska. – Ternopil, pp.9-12. The Internet address of the article on web-site: http://www.konferenciaonline.org.ua/ua/article/id-917

32) Danilishyn I.V. Danilishyn O.V. MATHEMATICS SIT, PROGRAMMING OPERATORS SIT AND SOME APPLICATIONS. Information society: technological, economic and technical aspects of development (issue 74): materials of the International Scientific Internet Conference, (Ternopil, Ukraine - Perevorsk, Poland, March 6-7, 2023) / [editor: O. Patryak and others] ; NGO "Scientific Community"; WSSG in Przeworsk. – Ternopil, pp. 4-7. http://www.konferenciaonline.org.ua/ua/article/id

33) Danilishyn I.V. Danilishyn O.V. ELEMENTS OF ST MATHEMATICS AND ST PROGRAMMING. LVIV SCIENTIFIC FORUM. MATERIALS OF THE VII INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE PROBLEMS AND PROSPECTS OF MODERN SCIENCE AND EDUCATION January 29-30, 2023, Lviv, 2023, pp. 40-43. Collection.pdf (lviv-forum.inf.ua)

34) Danilishyn I.V. Danilishyn O.V. THE NETWORKS SITE. Information society: technological, economic and technical aspects of development (issue 75): materials of the International Scientific Internet Conference, (Ternopil, Ukraine - Perevorsk, Poland, April 3-4, 2023) / [editor. : O. Patryak and others] ; NGO "Scientific Community"; WSSG in Przeworsk. – Ternopil, pp. 6-9. http://www.konferenciaonline.org.ua/ua/article/id-1060/

35) Danilishyn I., Danilishyn O. VARIABLE HIERARCHICAL DYNAMICAL STRUCTURES (MODELS) FOR DYNAMIC, SINGULAR, HIERARCHICAL SETS AND THE PROBLEM OF COLD THERMONUCLEAR FUSIONThe driving force of science and trends in its development: a collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, July 14, 2023. Coventry, United Kingdom: European Scientific. Pp.113-119. Platform.**https://previous.scientia.report/index.php/archive/issue /view/14.07.2023**

36) Krain S.G. Linear differential equations in Banach space. M., Science,1967(in Russian).

37) Kantor G. Fundamentals of the general doctrine of diversity. *New ideas in mathematics*, 6,1914 (in Russian).

38) N.Y. Belenkov. The principle of the integrity of brain activity. M., Meditsina,1980(in Russian).

39) A.L.Dontchev .Perturbations, approximations and sensitivity analysis of optimal control systems. Sptinger-Verllag, Berlin, New York,1983.

40) Galushkin A. Networks: principles of the theory. Hot line-Telecom. M.,2010 (in Russian).

41) Thomas J.Jech. LECTURES IN SET THEORY WITH PARTICULAR EMPHASIS ON THE METHOD OF FORCING. SPRINGER-VERLAG Berlin. Heidelberg. New York, 1971.

42) D.I.Blokhintsev. Quantum Mechanics Lectures on Selected Topics.Moscow,Atomizdat,1981.

43) George J. Klir, Bo Yuan. Fuzzy sets and fuzzy logic: theory and applications / Prentice Hall P T R Upper Saddle River, New Jersey 07458. 1995

44) Guanrong Chen, Trung Tat Pham. Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems / CRC Press LLC Boca Raton London New York Washington, D.C. 2001

45) Ershov Y.L. On a Hierarchy of Sets I, Albebra and Logic, 7 (1968), 47 - 73.

46) Ershov Y.L. On a Hierarchy of Sets II, Albebra and Logic, 7 (1968),15 - 47.

47) Yrshov Y.L.On a Hierarchy of Sets III, Algebra and Logic, 9(1970), 34 - 51.

48) Oleksandr Danilishyn, Illia Danilishyn. Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), 2023, 479-498. https://www.opastpublishers.com/open-access-articles/dynamical-sets-theory-s2telements-and-their-applications.pdf

49) A.Mostowski. Constructive sets.UNIVERSITY OF WARSAW-North-Holland Publishing Company-Amsterdam. PWN-Polish Scientific Publishers-Warszawa.1969.

50) Illia Danilishyn, Oleksandr Danilishyn. Hierarchical dynamic mathematical structures (models) theory: Scientific monograph DOI https://doi.org/10.36074/hdmsmt.monograph-2024 Published 2024-08-28 Shawnee, USA Primedia eLaunch LLC 2024 https://bookwire.bowker.com : 9798892178099

Declarations:

Availability of data and material

REFERENCES:

1) Danilishyn I., Danilishyn O. Dynamic Sets Set and Some of Their Applications to Neuroscience, Networks Set New Advances in Brain & Critical Care 2023, Volume 4 | Issue 2 | 66- 81. https://doi.org/10.33140/NABCC.04.02.02

2) Oleksandr Danilishyn and Illia Danilishyn. Dynamic Sets S1et and Some of their Applications in Physics. Science Set Journal of Physics, 25 September 2023,1-11, 815-_ article1695707465.pdf (mkscienceset.com)

3) I. Danilishyn, O. Danilishyn Dynamic Sets Se, Networks Se. *Advances in Neurology and Neuroscience*, 2023, Volume 6 | Issue 2 | 278-294. https://www.opastpublishers.com/open-access-articles/dynamic-sets-se-networks-se.pdf

4) I. Danilishyn, O. Danilishyn Program Operators Sit, tS, S1e, Set1. Journal of Sensor Networks and Data Communications [ISSN: 2994-6433] 2023, Volume 3 | Issue 1 | 138-143, https://www.opastpublishers.com/table-contents/jsndc-volume-3-issue-1-year-2023

5) Danilishyn .V. Danilishyn O.V. THE USAGE OF SIT-ELEMENTS FOR NETWORKS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

6) Danilishyn .V. Danilishyn O.V. MATHEMATICS ST, PROGRAMMING OPERATORS ST AND SOME EMPLOYMENT. **III International Scientific and Theoretical Conference «Theoretical and practical scientific achievements: research and results of their implementation»** *07.04.2023 | Pisa, Italia.* Collection of scientific papers “SCIENTIA”, 2023. **https://previous.scientia.report/index.php/archive/issue /view/07.04.2023**

7) Danilishyn .V. Danilishyn O.V. DYNAMICAL SIT-ELEMENTS. IV International Scientific and Practical Conference ”GRUNDLAGEN DER MOD-

ERNEN WISSENCHAFTLICHEN FORSCHUNG ”, 31.03.2023/Zurich, Switzerland. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/9

8) Danilishyn .V. Danilishyn O.V. tS - ELEMENTS. Collection of scientific papers «O» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, 2023. Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, 2023.Theoretical and empirical scientific research: concept and trends. **pp.156-161,** DOI 10.36074/logos-23.06.2023.42, **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/12**

9) Danilishyn .V. Danilishyn O.V. SET1 – ELEMENTS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 28 June 2023 with the proceedings of the: Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 9 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), **pp.** 239-254. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/09.06.2023**

10) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO SETS THEORY AND OTHERS. Scientific practice: modern and classical research methods: Collection of scientific papers «O» with Proceedings of the IV International Scientific and Practical Conference, Boston, May 26, 2023. Boston-Vinnytsia: Primedia eLaunch & European Scientific Platform, 2023, pp.166-171. **https://archive.logos-science.com/index.php/conference-proceedings/issue/view/11**

11) Danilishyn .V. Danilishyn O.V. SOME APPLICATIONS OF SIT- ELEMENTS TO CONTINUALVALUED LOGIC AND OTHERS. Features of the development of modern science in the pandemic’s era: a collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, May 19, 2023. Berlin, Federal Republic of Germany: European Scientific Platform, pp.79-84. **https://previous.scientia.report/index.php/archive/issue/view /19.05.2023**

12) Danilishyn .V. Danilishyn O.V. CONTINUAL SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS. Theoretical and practical aspects of modern scientific research: Collection of scientific papers «O» with Proceedings of the II International Scientific and Practical Conference, Seoul, April 28, 2023. Seoul-Vinnytsia: Case Co., Ltd.& European Scientific Platform, **pp. 144-150. https://archive.logos-science.com/ index.php/conference-proceedings/issue/view/10** 2023,

pp. 270-272. **https://archive.liga.science/index.php/conference-proceedings/issue/view/ukr-19.05.2023**.

13) Danilishyn .V. Danilishyn O.V. CONTINUAL SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS. Theoretical and practical aspects of modern scientific research: Collection of scientific papers «O» with Proceedings of the II International Scientific and Practical Conference, Seoul, April 28, 2023. Seoul-Vinnytsia: Case Co., Ltd.& European Scientific Platform, **pp. 144-150. https://archive.logos-science.com/index.php/conference-proceedings/issue/view/10**

14) Danilishyn .V. Danilishyn O.V. SIT-DYNAMICAL AND CONTINUAL ELEMENTS. MATERIALS OF THE VIII INTERNATIONAL PRIORITY SCIENTIFIC AND PRACTICAL CONFERENCE DIRECTIONS OF RESEARCH IN SCIENTIFIC AND EDUCATIONAL ACTIVITIES April 29 - 30 2023, Lviv. pp.24-31. http://lviv-forum.inf.ua/save/2023/29-30.04/

15) Danilishyn .V. Danilishyn O.V. CONTINUAL SIT-ELEMENTS WITH TARGET WEIGHTS.Formation of modern science: methodology and practice: materials III All-Ukrainian Student Scientific Conference, Uzhgorod, 21 April 2023 / NGO "Youth Scientific League".— Vinnytsia: NGO "European Scientific Platform", 2023 **pp. 105-107. https://archive.liga.science/index.php/conference-proceedings/issue/view/ukr-21.04.2023**.

16) Danilishyn .V. Danilishyn O.V. SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS FOR CONTINUAL SETS. Modernization of science and its influence on global processes: collection of scientific papers «SCIENTIA» with Proceedings of the III International Scientific and Theoretical Conference, April 14, 2023. Bern, Swiss Confederation: European Scientific Platform. Pp.61-66. **https://previous.scientia.report/index.php/archive/issue/view/14.04.2023**

17) Danilishyn .V. Danilishyn O.V. SIT-ELEMENTS AND THEIR APPLICATIONS. LVIV SCIENTIFIC FORUM MATERIALS OF THE VIII INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE THEORY AND PRACTICE OF MODERN SCIENCE AND EDUCATION March 19–20, 2023, Lviv 2023, pp.33-38. http://lviv-forum.inf.ua/save/2023/29-30.04/

18) Danilishyn .V. Danilishyn O.V. SIT-NETWORKS. LVIV SCIENTIFIC FORUM MATERIALS OF THE VIII INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE THEORY AND PRACTICE OF MODERN SCIENCE AND EDUCATION March 19–20, 2023, Lviv, pp. 39- 41. http://lviv-forum.inf.ua/save/2023/29-30.04

19) Danilishyn .V. Danilishyn O.V. MATHEMATICS ST, PROGRAMMING OPERATORS STAND SOME EMPLOYMENT A comprehensive approach

to the modernization of science: methods, models, and multidisciplinarity: materials of the II International scientific conference, Lutsk, March 3, 2023. / International Center scientific research. - Vinnytsia: European Scientific Platform, 2023, pp. 114-119. https://archive.mcnd.org.ua/index.php/conference-proceeding/issue/view/03.03.2023

20) Danilishyn .V. Danilishyn O.V. ST –ELEMENTS APPLICATIONS FOR SOME TASKS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 24 February 2023 with the proceedings of the: Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 17 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), pp.400-402. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/17.02.2023**

21) Danilishyn .V. Danilishyn O.V. THE INTRODUCTORY CONCEPTS AND OPERATIONS OF ST MATHEMATICS. INTERNATIONAL SCIENTIFIC JOURNAL GRAIL OF SCIENCE № 24 February 2023 with the proceedings of the: Correspondence International Scientific and Practical Conference SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS held on June 17 th, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) LLC International Centre Corporative Management (Vienna, Austria), pp.403- 406. **https://archive.journal-grail.science/index.php/2710-3056/issue/view/17.02.2023**

22) Danilishyn .V. Danilishyn O.V. SOME St – ELEMENTS APPLICATIONS. Information society: technological, economic and technical aspects formation (issue 74): materials of the International Scientific Internet conferences (Ternopil, Ukraine – Perevorsk, Poland, February 6-7, 2023) / [editor. : O. Patryak and others] ; GO "Naukova community"; WSSG in Przeworsk. – Ternopil. http://www.konferenciaonline.org.ua/ua/article/id-947/

23) Danilishyn .V. Danilishyn O.V. MATHEMATICS SIT, PROGRAMMING OPERATORS SIT AND SOME APPLICATIONS.Information society: technological, economic and technical aspects formation (issue 74): materials of the International Scientific Internet conferences (Ternopil, Ukraine – Perevorsk, Poland, February 6-7, 2023) / [editor. : O. Patryak and others] ; GO "Naukova community"; WSSG in Przeworsk. - Ternopil http://www.konferenciaonline.org.ua/ua/article/id-1040/

24) Danilishyn .V. Danilishyn O.V. THE NETWORKS SIT. Informational society: technological, economic and technical aspects of formation (issue 75): materials of the International Scientific Internet Conference, (m. Ternopil, Ukraine – Perevorsk, Poland, April 3-4, 2023) / [editor : O. Patryak and

others] ; NGO “Scientific community”; WSSG w Przeworska. - Ternopil, pp. 6-9. http://www.konferenciaonline.org.ua/ua/article/id-1060/

25) Danilishyn .V. Danilishyn O.V. MATHEMATICS ST, PROGRAMMING OPERATORS ST AND SOME EMPLOYMENT. Scientific forum: theory and practice of research: collection of scientific papers «SCIENTIA» with Proceedings of the III International Scientific and Theoretical Conference, March 10, 2023. Valencia, Kingdom of Spain: European Scientific Platform. Collection of scientific papers «SCIENTIA». pp.123-127 DOI:https://doi.org/10.36074/scientia-10.03.2023

26) Danilishyn .V. Danilishyn O.V. ELEMENTS OF ST MATHEMATICS AND ST PROGRAMMING.LVIV SCIENTIFIC FORUM. MATERIALS OF THE VII INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE PROBLEMS AND PROSPECTS OF MODERN SCIENCE AND EDUCATION January 29-30, 2023, Lviv, pp.40-43. http://lviv-forum.inf.ua/save/2023/29-30.01

27) Danilishyn .V. St – ELEMENTS AND PROGRAMMING OPERATORS. Informational society: technological, economic and technical aspects of formation (issue 74): materials of the International Scientific Internet Conference, (m. Ternopil, Ukraine – Perevorsk, Poland, March 6-7, 2023) / [editor. : O. Patryak and others] ; NGO “Scientific community”; WSSG w Przeworska. – Ternopil, pp.9-12. The Internet address of the article on web-site: http://www.konferenciaonline.org.ua/ua/article/id-917

28) Danilishyn I., Danilishyn O. VARIABLE HIERARCHICAL DYNAMICAL STRUCTURES (MODELS) FOR DYNAMIC, SINGULAR, HIERARCHICAL SETS AND THE PROBLEM OF COLD THERMONUCLEAR FUSIONThe driving force of science and trends in its development: a collection of scientific papers «SCIENTIA» with Proceedings of the IV International Scientific and Theoretical Conference, July 14, 2023. Coventry, United Kingdom: European Scientific. Pp.113-119. Platform.**https://previous.scientia.report/index.php/archive/issue/view/14.07.2023**

29) Danilishyn I., Danilishyn O. PROGRAM OPERATORS SIT, tS, S1e, Set1. Preprint. Research Square. **RESEARCH ARTICLE** *Published: 2023-08-04 08:00:27* https://www.researchsquare.com/article/rs-3228799/v1 https://doi.org/10.21203/rs.3.rs-3228799/v1

30) Danilishyn I., Danilishyn O. DYNAMIC SETS THEORY: SIT-ELEMENTS AND THEIR APPLICATIONS. Preprint. Research Square.**METHOD ARTICLE** *published 2023-08-01 04:20:16* https://doi.org/10.21203/rs.3.rs-3217178/v1 Dynamic Sets Theory: Sit-elements and Their Applications | Research Square www.researchsquare.com/article/rs-3217178/v1

31) Oleksandr Danilishyn, Illia Danilishyn. Dynamical Sets Theory: S2t-Elements and Their Applications. J Math Techniques Comput Math, 2(12), 2023, 479-498. https://www.opastpublishers.com/open-access-articles/dynamical-sets-theory-s2telements-and-their-applications.pdf

32) Oleksandr Danilishyn and Illia Danilishyn. Introduction to Section of Dynamic Mathematics: Dynamic Sets Theory: Parallel Sprt-Elements and their Applications. J Math Techniques Comput Math (Journal DOI: 10.33140/JMTCM), Volume 3 | Issue 5 (2024), pp.1- 27. https://www.opastpublishers.com/open-access-articles/introduction-to-section-of-dynamic-mathematics-dynamic-sets-theory-parallel-sprtelements-and-their-applications.pdf

33) Oleksandr Danilishyn and Illia Danilishyn. Fuzzy Dynamic Fuzzy Sets. Variable Fuzzy Hierarchical Dynamic Fuzzy Structures (Models, Operators) for Dynamic, Singular, Hierarchical Fuzzy Sets. FUZZY PROGRAM OPERATORS ffSprt, fftprS, ffS1epr, ffSeprt1. Journal of Mathematical Techniques Computational Mathematics (JMTCM), Volume 3 | Issue 4 |2024, Apr 17, pp. 1 – 37 Journal DOI: 10.33140/JMTCM https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

34) Oleksandr Danilishyn and Illia Danilishyn. Intrtoduction to section of Dynamic mathematics: DYNAMIC fuzzy SETS THEORY: Parallel fuzzy Sprt – elements AND THEIR APPLICATIONS. J Math Techniques Comput. Math, 3(7), 2023, 1 – 27. https://www.opastpublishers.com/journal/journal-of-mathematical-techniques-and-computational-mathematics/articles-in-press

35) Oleksandr Danilishyn, Illia Danilishyn. Introduction to Dynamic Sets Theory: Sprt-elements and Their Applications to the Fhysics and Chemistry. JOURNAL OF PHYSICS AND CHEMISTRY, vol. 2, issue 3, pp.1-31, March 27, 2024 https://cskscientificpress.com/articles$_{file}/527 - article$1712797848*.pdf*

36) Illia Danilishyn, Oleksandr Danilishyn. Hierarchical dynamic mathematical structures (models) theory: Scientific m onograph DOI https://doi.org/10.36074/hdmsmt.monograph-2024 Published 2024-08-28 Shawnee, USA Primedia eLaunch LLC 2024 https://bookwire.bowker.com : 9798892178099

SCIENTIFIC EDITION

Illia DANILISHYN
Oleksandr DANILISHYN

INTRODUCTION TO DYNAMIC MATHEMATICS: DYNAMIC SETS, DYNAMIC OPERATORS AND THEIR APPLICATIONS

MONOGRAPH

Edited by Volodymyr Pasynko

Date of publication: 24.12.2024.
The volume of data is 7.8 MB.
Conventionally printed sheets 25,69.

Publisher: Primedia E-launch LLC.
KS 5518, United States, Kansas, Shawnee, Flint St.
E-mail: info@primediaelaunch.com
URL: www.primediaelaunch.com