ISSN: 2834-7706

Research Article

Journal of Mathematical Techniques and Computational Mathematics

Introduction to Section of Dynamic Mathematics: Dynamic Sets Theory: Parallel Sprt-Elements and their Applications

Oleksandr Danilishyn* and Illia Danilishyn

Sumy State University, Ukraine

***Corresponding Author**
Oleksandr Danilishyn, Sumy State University, Ukraine

Submitted: 2024, Apr 23; **Accepted:** 2024, May 07; **Published:** 2024, May 20

Citation: Danilishyn, O., Danilishyn, I. (2024). Introduction to Section of Dynamic Mathematics: Dynamic Sets Theory: Parallel Sprt-Elements and their Applications. *J Math Techniques Comput Math, 3*(5), 01-27.

Abstract

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 parallel 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, we proposed using a short-pulse laser to achieve the desired goals by a directly parallel neural network [1]. We then proposed the fundamental development of this directly parallel neural network. In the articles new mathematical structures and operators are constructed through one action - "containment" [2-9]. 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.*

Keywords: Parallel Hierarchical Structure (Parallel Dynamic Operator), Parallel Sprt-Elements (Designation - PrSprt-Elements), Parallel tSpr- Elements (Designation - PrtSpr-Elements), Parallel Self-Type Structures.

1. 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 $\uparrow \mathrm{I} \downarrow_h^q$, 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. We then proposed the fundamental development of this directly parallel neural network. In the articles new mathematical structures and operators are constructed through one action - “containment” [2-9]. 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.

1.1 Parallel Sprt – Elements

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. Significance of the article: 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 dealt with by Synergetics.

We consider expression

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\mathrm{PrSprt}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \; (*_1)$$

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. The result of this process will be described by the expression

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\mathrm{PrSrt}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \; (*_2)$$

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 $(*_1)$ a parallel dynamic set. The need $(*_1)$ 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

$$\mathrm{PrSprt}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \; (**_1),$$

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

$$\mathrm{PrSpr}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \; (**_2)$$

or

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\mathrm{PrSprt} \; (***_1),$$

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 simultaneouslyб the result of this process will be described by the expression

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\mathrm{PrSpr} \; (***_2)$$

We consider the measure: $\mu^{**}(\begin{smallmatrix} C_1 & C_1 & \dots & C_1 \\ D_1 & D_2 & \dots & D_m \end{smallmatrix} \text{PrSprt} \begin{smallmatrix} A_1 & A_2 & \dots & A_n \\ C_1 & C_1 & \dots & C_1 \end{smallmatrix}) = \frac{\mu(A_1)*\mu(A_2)*\dots*\mu(A_n)}{\mu(D_1)*\mu(D_2)*\dots*\mu(D_m)}$, where $\mu(A_i)$,$\mu(D_j)$,–usual measures of sets A_i, D_j (i = 1, 2, …, n; j = 1, 2, …, m).

Remark. One can consider some generalization for (*): $\begin{smallmatrix} q_1(C_1) & q_2(C_2) & \dots & q_m(C_m) \\ D_1 & D_2 & \dots & D_m \end{smallmatrix} \text{PrSprt} \begin{smallmatrix} A_1 & A_2 & \dots & A_n \\ w_1(B_1) & w_2(B_2) & \dots & w_n(B_n) \end{smallmatrix}$ ${}^{q_1(C)}_{D} PrSprt^{A}_{q(B)}$, 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 $\begin{smallmatrix} q_1(C_1) & q_2(C_2) & \dots & q_m(C_m) \\ D_1 & D_2 & \dots & D_m \end{smallmatrix} \text{PrSpr} \begin{smallmatrix} A_1 & A_2 & \dots & A_n \\ w_1(B_1) & w_2(B_2) & \dots & w_n(B_n) \end{smallmatrix}$.

Similarly, for (**$_1$): $\text{PrSprt} \begin{smallmatrix} A_1 & A_2 & \dots & A_n \\ w_1(B_1) & w_2(B_2) & \dots & w_n(B_n) \end{smallmatrix}$, for (***$_1$): $\begin{smallmatrix} q_1(C_1) & q_2(C_2) & \dots & q_m(C_m) \\ D_1 & D_2 & \dots & D_m \end{smallmatrix} \text{PrSprt}$. The result of this process will be described by the expression $\begin{smallmatrix} q_1(C_1) & q_2(C_2) & \dots & q_m(C_m) \\ D_1 & D_2 & \dots & D_m \end{smallmatrix} \text{PrSpr}$.

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. Here, the axiom of regularity (A8) 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 [10]. Instead of the axiom of regularity, we introduce the following axioms: Axiom R1. $\forall B(Sprt^{CoB}_{CoB} = B)$. Axiom R2. $\forall B(\exists B^{-1})$.

1.2 PrSprt - Elements

Definition 1.1. The set of elements $\{g\} = (g_1, g_2, \dots, g_n)$ at one point x = $(x_1, x_2, \dots, x_n)$ 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 $\text{PrSprt} \begin{smallmatrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix}$.

Definition 1.2. $\text{PrSprt} \begin{smallmatrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix}$ - a parallel dynamic set $\{g\}$ at x.

Definition 1.3. An ordered set of elements at one point in the space is called an ordered PrSprt–element. It's possible to $\text{PrSprt} \begin{smallmatrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix}$ 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: $\text{PrSprt} \begin{smallmatrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix} + \text{PrSprt} \begin{smallmatrix} b_1 & b_2 & \dots & b_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix} = \text{PrSprt} \begin{smallmatrix} g_1 \cup b_1 & g_2 \cup b_2 & \dots & g_n \cup b_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix}$.

It's allowed to multiply PrSprt – elements: $\text{PrSprt} \begin{smallmatrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix} * \text{PrSprt} \begin{smallmatrix} b_1 & b_2 & \dots & b_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix} = \text{PrSprt} \begin{smallmatrix} g_1 \cap b_1 & g_2 \cap b_2 & \dots & g_n \cap b_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix}$.

The operator $\text{PrSprt} \begin{smallmatrix} g_1 \cup b_1 & g_2 \cup b_2 & \dots & g_n \cup b_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix}$ 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 $\text{PrSprt} \begin{smallmatrix} g_1 \cap b_1 & g_2 \cap b_2 & \dots & g_n \cap b_n \\ x_1 & x_2 & \dots & x_n \end{smallmatrix}$. 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.

1.3 Parallel Capacity in Itself

Definition 1.4. The capacity $\text{PrSprt} \begin{smallmatrix} g_1 & g_2 & \dots & g_n \\ A_1 & A_2 & \dots & A_n \end{smallmatrix}$ is called the parallel capacity A = $(A_1, A_2, \dots, A_n)$ for $(g_1, g_2, \dots, g_n)$.

Definition 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 = PrSprt$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$.

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

You can consider a more “hard” option: A*B=$PPrSprt_B^A$, where $PPrSprt_B^A$ – 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: $e^A = 1 + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots$, $(A+B)^n = \sum_{k=0}^{n} \binom{n}{k} A^k B^{n-k}$, $(1+A)^n = 1 + \frac{Ax}{1!} + \frac{n(n-1)A^2}{2!} + \cdots$, 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.

1.4 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$\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where $\{g\} = (g_1, g_2, \dots, g_n)$, i.e. n - elements at one point x = $(x_1, x_2, \dots, x_n)$, we can consider the capacity PrS_3f in itself with m elements from $\{g\}$, m<n, which is formed according to the form:

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

that is, the structure PrS_3f contains only m elements. Form (1.1) can be generalized into the following forms:

$w_{m,n,k}^1$ = (k, $\begin{pmatrix} (n_1, 1) \\ (\dots) \\ (n_m, 1) \end{pmatrix}$) (1.1.1)

or

$w_{m,n,k}^2$ = (k, (l, $\begin{pmatrix} (n_1) \\ (\dots) \\ (n_m) \end{pmatrix}$)) (1.1.2)

$w_{m,n,k,l}^3$ = Q(($\begin{matrix} d_1 \\ \dots \\ d_l \end{matrix}$), $\begin{pmatrix} (n_1, 1) \\ (\dots) \\ (n_m, 1) \end{pmatrix}$) (1.1.3),

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

$w_{m,m_1,n_1,m_2,n_2,m_3,n_3}^4$ = (m, ((m_1, n_1), ((m_2, n_2), (m_3, n_3)))) (1.1.4),

or

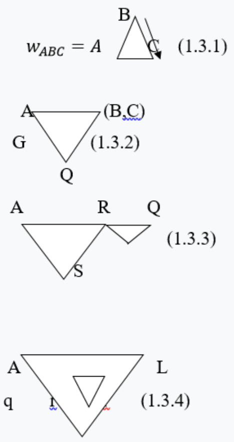
(Q, R) (1.1.5), where Q – any, R – any structure, R could be anything can be anything, not just structure. In this case, (1.1.5) can be used as another type of transformation from Q to R. Parallel capacities in themselves of the third type can be formed for any other structure, not necessarily Srt, only 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 PrS_3f can be introduced. For example, through the forms that generalizes (1.1):

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

where A is compressed (fits) in C in the compression fuzzy structure B in C ; or



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

$w_{A_1 A_2 \ldots A_n C} = (A_1, (A_2, (\ldots (A_n, C)\ldots)))$ (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.

1.5 Math Prself

Let's consider PrSprt arithmetic first :

1. Simultaneous parallel addition 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 is carried out using PrSprt$\begin{matrix} \{g_1\}+ & \{g_2\}+ & \cdots & \{g_n\}+ \\ x_1 & x_2 & \cdots & x_n \end{matrix}$.

2. Similarly, for simultaneous parallel multiplication: PrSprt$\begin{matrix} \{g_1\}* & \{g_2\}* & \cdots & \{g_n\}* \\ w_1 & w_2 & \cdots & w_n \end{matrix}$: the notation of the set B_l, l =1,2,…,n, with elements $b_{l i_1 i_2 \ldots i_{m_j}} = Sprt_{x_l}^{\left\{g_{l_{1_{i_1}}}*, g_{l_{2_{i_2}}}*, \ldots, g_{l_{m_{j_{i_{m_j}}}}}\right\}})_{R_l}$ 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):

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

Table 1: Index on the scale of discharges

Then $\mathrm{PrSprt}\begin{matrix} B_1+ & B_2+ & \dots & B_n+ \\ x_1 & x_2 & \dots & x_n \end{matrix}$ 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 PrSprt-multiplication:

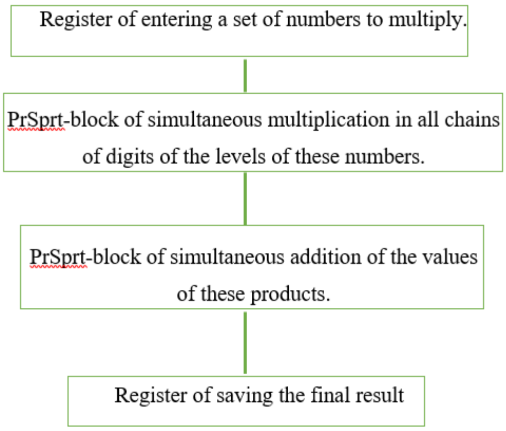


Figure 1: The straightforward functional scheme of the assumed arithmetic-logical device for PrSprt-multiplication.

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: $\mathrm{PrSprt}\begin{matrix} g_1q_1 & g_2q_2 & \dots & g_nq_n \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\{q\} = (q_1, \mathrm{q}_2, .., \mathrm{q}_\mathrm{n})$. q_i -an operation, i = 1, …, n.

4. Similarly, for the simultaneous execution of various operators: $\mathrm{PrSprt}\begin{matrix} F_1g_1 & F_2g_2 & \dots & F_ng_n \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\{F\} = (F_1, F_2, \dots, F_n)$. F_i is an operator, i = 1, …, n.

5. The arithmetic itself for 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: $PrS_1f\{\mathrm{a}q\}$, ($PrS_3f\{\mathrm{a}q\}$), and with different operators: $PrS_1f\{F\mathrm{a}\}$, ($PrS_3f\{F\mathrm{a}\}$).

7. $\mathrm{PrSrt}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$ – the result of the containment operator. For sets A_i, B_i, (i = 1, 2, …, n), we have

$\text{PrSrt}\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} = \{\sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \ , \sum_{i=1}^{n} D_i\} = \left\{\begin{matrix} \sum_{i=1}^{n} D_i \\ \sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \end{matrix}\right\}$, 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, $\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\text{PrSrt} = \left\{\begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\} & \{\} & \dots & \{\} \\ D_1 - D_1 \cap C_1 & D_2 - D_2 \cap C_2 & \dots & D_m - D_m \cap C_m \end{matrix}\text{PrSrt} \\ \sum_{i=1}^{m} (C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix}\right\}$, where Q_i is Proself-set for $(D_i \cap C_i)$ [16].

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 $\text{PrSprt} \begin{matrix} \frac{\partial}{\partial x_{1_i}} & \frac{\partial}{\partial x_{2_i}} & \dots & \frac{\partial}{\partial x_{k_i}} \\ f_1(x_1, x_2, \dots, x_n) & f_2(x_1, x_2, \dots, x_n) & \dots & f_k(x_1, x_2, \dots, x_n) \end{matrix}$, where $x=(x_{1_i}, x_{2_i}, \dots x_{k_i})$- 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 $\text{PrSprt} \begin{matrix} \int () dx_{1_i} & \int () dx_{2_i} & \dots & \int () dx_{k_i} \\ f_1(x_1, x_2, \dots, x_n) & f_2(x_1, x_2, \dots, x_n) & \dots & f_k(x_1, x_2, \dots, x_n) \end{matrix}$, where $(x_{1_i}, x_{2_i}, \dots x_{k_i})$- 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 $\text{PrSprt} \begin{matrix} \lim_{x_{1_i} \to a_{1_i}} & \lim_{x_{2_i} \to a_{2_i}} & \dots & \lim_{x_{k_i} \to a_{k_i}} \\ f_1(x_1, x_2, \dots, x_n) & f_2(x_1, x_2, \dots, x_n) & \dots & f_k(x_1, x_2, \dots, x_n) \end{matrix}$. The same is done for PrSprt-$\lim_{\substack{x_{1_i} \to a_{1_i} \\ \vdots \\ x_{k_i} \to a_{k_i}}} f(x_1, x_2, \dots, x_n)$.

$$PrS_3 f\{\lim_{x \to a}\} = \text{PrSprt} \begin{matrix} \lim_{x_{1_i} \to a_{1_i}} & \lim_{x_{2_i} \to a_{2_i}} & \dots & \lim_{x_{k_i} \to a_{k_i}} \\ \lim_{x_{1_i} \to a_{1_i}} & \lim_{x_{2_i} \to a_{2_i}} & \dots & \lim_{x_{k_i} \to a_{k_i}} \end{matrix}.$$

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.

1.6 Operator Pritself

Definition 7. An operator that transforms $\text{PrSprt}\begin{matrix} g_1 & g_2 & \dots & g_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, into any $\text{PrS}_i f\{b\}, i = 2,3$; where $\{b\} \subset \{g\}, \{g\} = (g_1, g_2, \dots, g_n)$; is the operator Pritself.

Example. The operator contains the set in Pritself.

1.7 Lim-Pritself

1. Lim PrSprt

For example, the double limit:$\lim_{\substack{x \to a1 \\ y \to a2}} G(x, y)$ corresponds to $PrSprt \begin{matrix} G(x, y) & G(x, y) \\ x \to a1 & y \to a2 \end{matrix}$.

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 $PrS_3 f$ – 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)$\frac{\partial u}{\partial x} \to$ 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 $PrS_1 f$. The differential operator $PrSprt_x^{\{\frac{\partial}{\partial x}, \frac{\partial}{\partial x}(\frac{\partial}{\partial x})\}}$ - is interesting too.

1.8 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 $\{p\} = (p_1, p_2, \ldots, p_n)$ to the variables $\{g\} = (g_1, g_2, \ldots, g_n)$. This is implemented via PrSprt$\begin{matrix} g_1 := & g_2 := & \ldots & g_n := \\ p_1 & p_2 & \ldots & p_n \end{matrix}$.

2. Simultaneous checking the set of conditions $\{f\} = (f_1, f_{2,} \ldots, f_n)$ for the set of expressions$\{B\} = (B_1, B_{2,} \ldots, B_n)$. 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.

$PrS_3 f$ – 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) and for more complex operators in the form (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 PrSrt we introduce the concept of PrSrt – the change in physical quantity B: PrSrt$\begin{matrix} \Delta_1 B & \Delta_2 B & \ldots & \Delta_n B \\ x_1 & x_2 & \ldots & x_n \end{matrix}$, . Then the mean PrSrt - velocity will be $v_{cpPrsrt}(t, \Delta t)$ = PrSrt $\begin{matrix} \frac{\Delta_1 B}{\Delta t} & \frac{\Delta_n B}{\Delta t} & \ldots & \frac{\Delta_n B}{\Delta t} \\ x_1 & x_2 & \ldots & x_n \end{matrix}$ and PrSrt-velocity at time t:$v_{Prsrt} = \lim_{\Delta t \to 0} v_{cpPrst}(t, \Delta t)$. PrSrt – acceleration $a_{st} = \frac{dv_{st}}{dt}$.

In normal use, simply $Sprt_x$ reduces to a sum at point x of space. When using $Sprt_x$ with "target weights", we get, depending on the "target weights", one or another modification, namely, for example, the velocity v_{st}^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 $v_{st}^f = Srt_x^{\{v_1 f, v_2\}}$, 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 PrSprt elements.

Similarly, the concepts of Sprt - force and Sprt - energy are introduced [3,4]. For example, $E_{st}^f = Sprt_x^{\{E_1 f, E_2\}}$ 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.

Remark 1.2. PrSprt – 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 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, $PrSrt_B^B$ will mean $PrS_1 f$ B. For example, Pr$Sprt_{DNA}^{DNA}$ 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 Cprt: 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 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.

Remark1.3
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) $SprtE_D^{\{a_1 q, a_2\}}$ – the energy of instantaneous substitution a_1 by a_2, where a_1, and a_2 are chemical elements, q is instant replacement.
*T*he ideology of PrSprt elements allows us to go to the border of the world familiar to us, which allows us to act more effectively.

2. Dynamic PrSprt – Elements
2.1 Dynamic PrSprt – Elements
We considered stationary PrSprt – elements earlier. Here we consider dynamic PrSprt – elements [5].
Definition 2.1
The process of fitting a set of elements $\{g(t)\} = (g_1(t), g_2(t), \dots, g_n(t))$ into one point x= $(x_1, x_2, \dots, x_n)$ 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 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)$\begin{matrix} b_1(t) & b_2(t) & \dots & b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$ = 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$\begin{matrix} g_1(t) \cap b_1(t) & g_2(t) \cap b_2(t) & \dots & g_n(t) \cap b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$

2.2 Parallel Dynamic Containment of Oneself

Definition 2.3 Parallel Self -Dynamic Capacity Q(t) = $(Q_1(t), Q_2(t), \dots, Q_n(t))$ is parallel fitting into Q(t) = $(Q_1(t), Q_2(t), \dots, Q_n(t))$: 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 2.4

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 2.5

Parallel dynamic capacity A(t) = $(A_1(t), A_2(t), \dots, A_n(t))$ 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 $PrS_1 f(t)A(t)$.

Definition 2.6

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 2.7

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 $PrS_3 f(t)B(t)$.

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.

2.3 Parallel Self Dynamic Capacity

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 $\{g(t)\} = (g_1(t), g_2(t), \dots, g_n(t))$, i.e. n – elements at one point x = $(x_1, x_2, \dots, x_n)$, 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)$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

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 $PrS_3 f(t)$.

2.4 Parallel Dynamic Math itself

1. The process of simultaneous parallel addition of sets elements $\{g_i(t)\} = \left(g_{i_1}(t), g_{i_2}(t), \dots, g_{i_{m_j}}(t)\right)$, i = 1,2,…,n, j =1,2,…,k are realized by PrSprt(t)$\begin{matrix} \{g_1(t)\}+ & \{g_2(t)\}+ & \dots & \{g_n(t)\}+ \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

2. By analogy, for simultaneous multiplication: PrSprt(t)$\begin{matrix} \{g_1(t)\}* & \{g_2(t)\}* & \dots & \{g_n(t)\}* \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

3. Similarly for simultaneous execution of various operations:PrSprt(t)$\begin{matrix} g_1(t)q_1(t) & g_2(t)q_2(t) & \dots & g_n(t)q_n(t) \\ w_1 & w_2 & \dots & w_n \end{matrix}$, 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)$\begin{matrix} F_1(t)g_1(t) & F_2(t)g_2(t) & \dots & F_n(t)g_n(t) \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\{F(t)\} = (F_1(t), F_2(t), \dots, F_n(t))$. $F_i(t)$ is an operator, i = 1, …, n.

5. Parallel dynamic arithmetic itself for containments of oneself will be similar: Parallel dynamic addition - $PrS_1 f(t)\{g(t) +\}$, (or $PrS_3 f(t)\{g(t) +\}$ for the third type), Parallel dynamic multiplication $PrS_1 f(t)\{g(t) *\}$, $(PrS_3 f(t)\{g(t) *\})$.

6. Similarly with different operations: $PrS_1 f(t)\{g(t)q(t)\}$, $(PrS_3 f(t)\{g(t)q(t)\})$ and with different operators: $PrS_1 f(t)\{F(t)g(t)\}$, $(PrS_3 f(t)\{F(t)g(t)\})$.

7. PrSrt(t)$\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}$ − gives the result

PrSrt(t)$\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix} = \{\sum_{i=1}^{n} A_i(t) \cup B_i(t) - A_i(t) \cap B_i(t)\ , \sum_{i=1}^{n} D_i(t)\}$, 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,

$$\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}\text{PrSrt(t)} =$$

$$\left\{ \begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\,\} & \{\,\}\ \dots & \{\,\} \\ D_1(t) - D_1(t) \cap C_1(t) & D_2(t) - D_2(t) \cap C_2(t) & \dots & D_2(t) - D_2(t) \cap C_2(t) \end{matrix}\text{PrSrt} \\ \sum_{i=1}^{m} (C_i(t) - D_i(t) \cap C_i(t)) - (D_i(t) - D_i(t) \cap C_i(t)) \end{matrix} \right\},$$

where $Q_i(t)$ is oself-set for $(D_i(t) \cap C_i(t))$ [16].

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 2.1. The parallel dynamic PrSprt-displacement will be denote by $\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}$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 $\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}$PrSrt(t). Then the notation $\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}$PrSprt(t)$\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}$ 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

$$\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}\text{PrSrt(t)}\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}$$

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 $PrS_1 f(t)B(t)$. $\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ C_1(t) & C_2(t) & \dots & C_m(t) \end{matrix}$PrSprt(t) denotes the parallel dynamic expelling C(t) = $(C_1(t), C_2(t),.., C_m(t))$ oneself out of oneself, $\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ A_1(t) & A_2(t) & \dots & A_n(t) \end{matrix}$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 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. 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.

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 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.

2.5 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 $\{p(t)\} = (p_1(t), p_2(t), \dots, p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), \dots, g_n(t))$. 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 $\{f(t)\} = \big(f(t)_1, f_{2,}(t) \dots, f(t)_n\big)$ for a set of expressions$\{B(t)\} = (B_1(t), B_2(t), \dots, B_n(t))$ 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) = $(x_1(t), x_2(t), \dots, x_n(t))$. can be any.

3. Similarly for loop operators and others.

$PrS_3f(t)$– software operators will differ only in that the aggregates $\{g(t)\}, \{p(t)\}, \{B(t)\}, \{f(t)\}$ will be formed from corresponding processes PrSprt(t) for the above-mentioned programming operators through form (1.1) or forms (1.1.1) – (1.4) for more complex operators.

Remark 2.2

With the help of dynamic Sprt-elements, the concepts of dynamic Sprt - force, dynamic Sprt – energy are introduced. For example, $E(t)_{sprt}^{f} = Sprt(t)_{x(t)}^{\{E_1(t)f, E_2(t)\}}$ 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 2.3

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.

Remark2.4PrSprt(t)$\begin{matrix} \text{PrSprt(t)}\begin{matrix} B_1(t) & B_2(t) & \dots & B_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix} & \text{PrSprt(t)}\begin{matrix} B_1(t) & B_2(t) & \dots & B_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix} & \dots & \text{PrSprt(t)}\begin{matrix} B_1(t) & B_2(t) & \dots & B_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix} \\ \text{PrSprt(t)}\begin{matrix} B_1(t) & B_2(t) & \dots & B_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix} & \text{PrSprt(t)}\begin{matrix} B_1(t) & B_2(t) & \dots & B_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix} & \dots & \text{PrSprt(t)}\begin{matrix} B_1(t) & B_2(t) & \dots & B_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix} \end{matrix}$

can increase parallel self- level of B(t) = $(B_1(t), B_2(t), \dots, B_n(t))$.

Remark 2.5. For example, the operator Pritself is $PrS_1f(t)$.

Remark 2.6

May be considered the following derivatives: $\frac{d\text{PrSrt(t)}\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}}{dt}$, $\frac{d\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}\text{PrSprt(t)}}{dt}$, $\frac{d\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}\text{PrSprt(t)}\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}}{dt}$, $\frac{dPrS_if(t)}{dt}$, i=1,2,3.

Remark 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 2.8

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

3. PrSprt – Elements for Continual Sets

3.1 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 $\{g\} = (g_1, g_2, \ldots, g_n)$ at one point x = $(x_1, x_2, \ldots, x_n)$ 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$\begin{matrix} g_1 & g_2 & \cdots & g_n \\ x_1 & x_2 & \cdots & x_n \end{matrix}$.

Definition 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$\begin{matrix} g_1 & g_2 & \cdots & g_n \\ x_1 & x_2 & \cdots & x_n \end{matrix}$ + PrSprt$\begin{matrix} b_1 & b_2 & \cdots & b_n \\ x_1 & x_2 & \cdots & x_n \end{matrix}$ = PrSprt$\begin{matrix} g_1 \cup b_1 & g_2 \cup b_2 & \cdots & g_n \cup b_n \\ x_1 & x_2 & \cdots & x_n \end{matrix}$, where some or any elements may be ordered elements. It's allowed to multiply PrSprt – elements: PrSprt$\begin{matrix} g_1 & g_2 & \cdots & g_n \\ x_1 & x_2 & \cdots & x_n \end{matrix} *$ PrSprt$\begin{matrix} b_1 & b_2 & \cdots & b_n \\ x_1 & x_2 & \cdots & x_n \end{matrix}$ = PrSprt$\begin{matrix} g_1 \cap b_1 & g_2 \cap b_2 & \cdots & g_n \cap b_n \\ x_1 & x_2 & \cdots & x_n \end{matrix}$.

Definition 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$\begin{matrix} A_1 & A_2 & \cdots & A_n \\ A_1 & A_2 & \cdots & A_n \end{matrix}$.

Definition 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$\begin{matrix} \sin\infty & tg(-\infty) & \cdots & \sin(-\infty) \\ x_1 & x_2 & \cdots & x_n \end{matrix}$ = PrSprt$\begin{matrix} \uparrow I \downarrow_{-1}^{1}. & \downarrow I \uparrow_{-\infty}^{\infty} & \cdots & \downarrow I \uparrow_{-1}^{1} \\ x_1 & x_2 & \cdots & x_n \end{matrix}$, don't confuse with values of these functions.

Definition 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 PrS_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 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.

3.2 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$\begin{matrix} g_1 & g_2 & \cdots & g_n \\ x_1 & x_2 & \cdots & x_n \end{matrix}$, where

$\{g\} = (g_1, g_2, \ldots, g_n)$, i.e. n - continual elements at one point x = $(x_1, x_2, \ldots, x_n)$, 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 structurePrSprt$\begin{matrix} g_1 & g_2 & \cdots & g_n \\ x_1 & x_2 & \cdots & x_n \end{matrix}$. 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.

3.3 Mathematics itself for Continual Elements

1.Simultaneous parallel addition of the sets continual elements $\{g_i\} = \left(g_{i_1}, g_{i_2}, \ldots, g_{i_{m_j}}\right)$, $i = 1,2,\ldots,n$, $j = 1,2,\ldots,k$, is implemented using PrSprt$\begin{matrix} \{g_1\} \cup & \{g_2\} \cup & \cdots & \{g_n\} \cup \\ x_1 & x_2 & \cdots & x_n \end{matrix}$.

2. By analogy, for simultaneous multiplication: PrSprt$\begin{matrix} \{g_1\} \cap & \{g_2\} \cap & \cdots & \{g_n\} \cap \\ x_1 & x_2 & \cdots & x_n \end{matrix}$.

3. Similarly for simultaneous execution of various operations: PrSprt$\begin{matrix} g_1 q_1 & g_2 q_2 & \cdots & g_n q_n \\ w_1 & w_2 & \cdots & w_n \end{matrix}$, where $\{q\} = (q_1, q_2, \ldots, q_n)$. q_i -an operation, i = 1, …, n.

4. Similarly, for the simultaneous execution of various operators: PrSprt$\begin{matrix} F_1 g_1 & F_2 g_2 & \cdots & F_n g_n \\ w_1 & w_2 & \cdots & w_n \end{matrix}$, where $\{F\} = (F_1, F_2, \ldots, F_n)$. 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: $PrS_1f\{aq\}$, ($PrS_3f\{aq\}$), and with different operators: $PrS_1f\{Fa\}$, ($PrS_3f\{Fa\}$).

7. PrSrt$\begin{matrix} A_1 & A_2 & \cdots & A_n \\ B_1 & B_2 & \cdots & B_n \end{matrix}$ $-$ the result of the parallel containment operator. For continual sets A_i, B_i, (i = 1, 2, …, n), we have PrSrt$\begin{matrix} A_1 & A_2 & \cdots & A_n \\ B_1 & B_2 & \cdots & B_n \end{matrix} = \{\sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i\ , \sum_{i=1}^{n} D_i\}$, 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, $\begin{matrix} C_1 & C_2 & \cdots & C_m \\ D_1 & D_2 & \cdots & D_m \end{matrix}$PrSrt $=$ $\left\{\begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\ \} & \{\ \} & \cdots & \{\ \} \\ D_1 - D_1 \cap C_1 & D_2 - D_2 \cap C_2 & \cdots & D_m - D_m \cap C_m \end{matrix} \text{PrSrt} \\ \sum_{i=1}^{m} (C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix}\right\}$, where Q_i is oself-set for $(D_i \cap C_i)$ [16].

Remark 3.1. $\begin{matrix} C_1 & C_2 & \cdots & C_m \\ D_1 & D_2 & \cdots & D_m \end{matrix}$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. $\begin{matrix} C_1 & C_2 & \cdots & C_m \\ D_1 & D_2 & \cdots & D_m \end{matrix}$PrSprt$\begin{matrix} A_1 & A_2 & \cdots & A_n \\ B_1 & B_2 & \cdots & B_n \end{matrix}$

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 PrSprt$\begin{matrix} A_1 & A_2 & \cdots & A_n \\ B_1 & B_2 & \cdots & B_n \end{matrix}$, 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$\begin{matrix} B_1 & B_2 & \cdots & B_n \\ B_1 & B_2 & \cdots & B_n \end{matrix}$ will mean 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 [6]. You can also try to optimize and research some processes with these elements using the techniques from [7]. Let's introduce operators for transforming capacity to Prself-capacity in itself as an element: $PrQ_1S(A)$ transforms A to Prf_1SA, $PrQ_0S(C)$ transforms C to $\begin{matrix} C_1 & C_2 & \dots & C_m \\ C_1 & C_2 & \dots & C_m \end{matrix}$PrSprt .

Can be considered $Q({}^{A}_{A}Sprt^{A}_{A})$, Q-any operator.

4. Dynamic Continual PrSprt – Elements

4.1 Dynamic Continual PrSprt – Elements

Definition 4.1

The process of containing the set of continual elements $\{g(t)\} = (g_1(t), g_2(t), \dots, g_n(t))$ into one point x= $(x_1, x_2, \dots, x_n)$ 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 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 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$\begin{matrix} g_1(t) \cap b_1(t) & g_2(t) \cap b_2(t) & \dots & g_n(t) \cap b_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

4.2 Parallel Dynamic Continual Containment of oneself in oneself as an Element

Definition 4.3

Parallel self dynamic continual capacity Q(t) = $(Q_1(t), Q_2(t), \dots, Q_n(t))$ is parallel fitting into Q(t) = $(Q_1(t), Q_2(t), \dots, Q_n(t))$: 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 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 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 $PrS_1f(t)A(t)$.

Definition 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 $PrS_2f(t)C(t)$.

Definition 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 $PrS_3f(t)B(t)$.

4.3 The connection of Dynamic Continual PrSprt – Elements with Parallel Self 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)$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where $\{g(t)\} = (g_1(t), g_2(t), \dots, g_n(t))$, i.e. n - continual elements at one point x= $(x_1, x_2, \dots, x_n)$, one can consider the parallel dynamic containment $PrS_3f(t)$ 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)$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$. 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 $PrS_3f(t)$.

4.4 Parallel Dynamic Continual Mathematics self

1. The process of simultaneous parallel addition of sets continual elements $\{g_i(t)\} = \left(g_{i_1}(t), g_{i_2}(t), \dots, g_{i_{m_j}}(t)\right)$, i = 1,2,…,n, j =1,2,…,k are realized by PrSprt(t)$\begin{matrix} \{g_1(t)\}\cup & \{g_2(t)\}\cup & \dots & \{g_n(t)\}\cup \\ x_1 & x_2 & \dots & x_n \end{matrix}$.
2. By analogy, for simultaneous multiplication: PrSprt(t)$\begin{matrix} \{g_1(t)\}\cap & \{g_2(t)\}\cap & \dots & \{g_n(t)\}\cap \\ x_1 & x_2 & \dots & x_n \end{matrix}$.
3. Similarly for simultaneous execution of various operations:PrSprt(t)$\begin{matrix} g_1(t)q_1(t) & g_2(t)q_2(t) & \dots & g_n(t)q_n(t) \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\{q(t)\} = (q_1(t), q_2(t), \dots, q_n(t))$. $q_i(t)$-an operation, i = 1, …, n.
4. Similarly, for the simultaneous execution of various operators: PrSprt(t)$\begin{matrix} F_1(t)g_1(t) & F_2(t)g_2(t) & \dots & F_n(t)g_n(t) \\ w_1 & w_2 & \dots & w_n \end{matrix}$, where $\{F(t)\} = (F_1(t), F_2(t), \dots, F_n(t))$. $F_i(t)$ is an operator, i = 1, …, n.
5. Parallel dynamic arithmetic itself for containments of oneself will be similar: Parallel dynamic addition - $PrS_1f(t)\{g(t)\cup\}$, (or $PrS_3f(t)\{g(t)\cup\}$ for the third type), Parallel dynamic multiplication $PrS_1f(t)\{g(t)\cap\}$, $(PrS_3f(t)\{g(t)\cap\})$.
6. Similarly with different operations: $PrS_1f(t)\{g(t)q(t)\}$, $(PrS_3f(t)\{g(t)q(t)\})$ and with different operators: $PrS_1f(t)\{F(t)g(t)\}$, $(PrS_3f(t)\{F(t)g(t)\})$.
7.

. PrSrt(t)$\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}$ – gives the result

PrSrt(t)$\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix} = \{\sum_{i=1}^{n} A_i(t)\cup B_i(t) - A_i(t)\cap B_i(t)\ , \sum_{i=1}^{n} D_i(t)\}$, 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,

$$\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}\text{PrSrt(t)} =$$

$$\left\{\begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\,\} & \{\,\} & \dots & \{\,\} \\ D_1(t) - D_1(t)\cap C_1(t) & D_2(t) - D_2(t)\cap C_2(t) & \dots & D_2(t) - D_2(t)\cap C_2(t) \end{matrix}\text{PrSrt} \\ \sum_{i=1}^{m}(C_i(t) - D_i(t)\cap C_i(t)) - (D_i(t) - D_i(t)\cap C_i(t)) \end{matrix}\right\},$$

where $Q_i(t)$ is oself-set for $(D_i(t)\cap C_i(t))$ [16].

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 1.1

The parallel dynamic continual PrSprt-displacement will be denote by $\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}$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 $\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}$PrSrt(t). Then the notation

$\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}\text{PrSprt(t)}\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}$

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

$\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}\text{PrSrt(t)}\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ B_1(t) & B_2(t) & \dots & B_n(t) \end{matrix}$

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) = $(C_1(t), C_2(t), \ldots, C_m(t))$. oneself out of oneself, $\begin{matrix} A_1(t) & A_2(t) & \dots & A_m(t) \\ A_1(t) & A_2(t) & \dots & A_m(t) \end{matrix}\text{PrSprt(t)}\begin{matrix} A_1(t) & A_2(t) & \dots & A_n(t) \\ A_1(t) & A_2(t) & \dots & A_n(t) \end{matrix}$ — simultaneous parallel dynamic containment continual A(t) = $(A_1(t), A_2(t), \ldots, A_n(t))$. 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.

4.5 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 $\{g(t)\} = (g_1(t), g_2(t), \ldots, g_n(t))$ i.e. n - continual elements with target weights{w(t)} at one point x = $(x_1, x_2, \ldots, x_n)$, we can consider the dynamic continual containment $\text{Pr}S_3 f(t)\{g(t)\}$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 $\text{Pr}S_3 f(t)\{g(t)\}$w(t)
Parallel dynamic containments of oneself with target weights of the third type can be formed for any other structure, not necessarily Sit, 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 $\text{Pr}S_3 f(t)\{g(t)\}$w(t) can be introduced.
Definition 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 $PrS_1 f(t)A(t)$w(t).

Definition 30

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 $PrS_2 f(t)C(t)$w(t).

Definition 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 $PrS_3 f(t)B(t)$w(t).

5. The usage of PrSprt-Elements for Networks

5.1 The usage of PrSprt-Elements for Networks

A. Galushkin's comprehensive monograph covers all aspects of networks, but traditional approaches go through classical mathematics, mainly through the usual correspondence operators [11]. 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 Sprt-networks. PrSprt-networks(Smnsprt) 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 in PrSprt(t)$\begin{matrix} g_1(t) & g_2(t) & \dots & g_n(t) \\ x_1 & x_2 & \dots & x_n \end{matrix}$ are chosen for necessary tasks. We will not touch on the issues of applications, or network optimization. They are described in detail by Galushkin [8]. 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 – 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, $\{g(t)\} = (g_1(t), g_2(t), \dots, g_n(t))$ – PrSprt-synapses weights. The first level of mnPrSprt consists of simple mnPrSprt. The second level of mnPrSprt consists of PrSprt(t)$\begin{matrix} \text{mnPrSprt} & \text{mnPrSprt} & \dots & \text{mnPrSprt} \\ D_1 & D_2 & \dots & D_n \end{matrix}$– 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)$\begin{matrix} \text{PrSprt(t)}\begin{matrix} \text{mnPrSprt} & \text{mnPrSprt} & \dots & \text{mnPrSprt} \\ D_1 & D_2 & \dots & D_n \end{matrix} & \text{PrSprt(t)}\begin{matrix} \text{mnPrSprt} & \text{mnPrSprt} & \dots & \text{mnPrSprt} \\ D_1 & D_2 & \dots & D_n \end{matrix} & \dots & \text{mnPrSprt} \\ D_1 & D_2 \quad \dots & & D_n \end{matrix}$ - PrSprt^2- node of 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-, n≥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 5.

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. mnPrSprt contains PrSprt(t)$\begin{matrix} \text{eprogram}_1(t) & \text{eprogram}_2(t) & \dots & \text{eprogram}_n(t) \\ \text{mnPrSprt} & \text{mnPrSprt} & \dots & \text{mnPrSprt} \end{matrix}$., 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 $Sprt_{\mathrm{SmnPrSprt}}^{\mathrm{SmnPrSprt}}$ 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 Sit-coding, Sit-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. $Sprt_{activation}^{\mathrm{SmnPrSprt},f}$ 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."

6. 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,

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrSprt(t)} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} = \begin{cases} \begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrSprt}, q_2 \geq t \geq q_1 \\ \begin{matrix} B_1 & B_2 & \dots & B_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrSprt} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}, q_3 \geq t > q_2 \\ \begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrSprt} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}, q_4 \geq t > q_3 \\ \mathrm{PrSprt} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}, q_5 \geq t > q_4 \\ \begin{matrix} \{\} & \{\} & \dots & \{\} \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrSprt}, t > q_5 \\ \dots \end{cases} (*_{6.1}),$$

In particular, $\begin{matrix} B_1 & B_2 & \dots & B_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrSprt} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$ 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 $\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrSprt}_1(t) \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$ =

$$= \begin{cases} \left\{ \begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\} & \{\} & \dots & \{\} \\ D_1 - D_1 \cap C_1 & D_2 - D_2 \cap C_2 & \dots & D_m - D_m \cap C_m \end{matrix} \mathrm{PrSrt} \\ \sum_{i=1}^{m} (C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix} \right\}, q_2 \geq t \geq q_1 \\ \sum_{i=1}^{n} \left(\begin{matrix} S1_0^e f B_i * \\ {}_{Q_i - B_i}^{B_i} S^1 t_{B_i}^{A_i - B_i} \end{matrix} \right), q_3 \geq t > q_2 \\ \sum_{j=1}^{m} \sum_{i=1}^{n} \left(\begin{matrix} S_{01}^{et} f B_i \\ {}_{C_j - B_i}^{C_j - B_i} S_1 t_{B_i}^{A - B_i} \\ {}_{D_j - C_j - B_i}^{C_j - B_i} S_1 t_{B_i}^{A - B_i} \end{matrix} \right), q_4 \geq t > q_3 \\ \left\{ \begin{matrix} \sum_{i=1}^{n} D_i \\ \sum_{i=1}^{n} A_i \cup B_i - A_i \cap B_i \end{matrix} \right\},, q_5 \geq t > q_4 \\ {}_{D}^{\{\}} Sprt \begin{matrix} \{\} & \{\} & \dots & \{\} \\ D_1 & D_2 & \dots & D_m \end{matrix} \mathrm{PrSrt}, t > q_5 \\ \dots \end{cases} (*_{6.2}),$$

where Q_i is oself-set for $(D_i \cap C_i)$ D_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, ${}_{Q-B}^{B} S^1 t_B^{A-B}$ is considered in [12,13].

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

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \\ A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix} \begin{matrix} a \\ c \\ d \end{matrix} \begin{matrix} b \\ PrVS \\ q \end{matrix} \begin{matrix} g \\ w \\ r \end{matrix} \begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \\ C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}$$

Fig.1

c) continuous variable parallel structure

PrVS

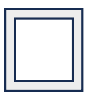


Figure 2

Where a continuous set represents the rim of the Fig.2.

We introduce the notation m_{VS_N} ,where m – 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_{PrVS_R} ,where q – any, R - connections in q in the variable parallel 2-hierarchical structure PrVS, in particular, q, 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)=$\sum_{k=1}^{n} c(B_k) * c(A/B_k)$, B_1, B_2,.., B_n-full group of hypotheses- containments: $\sum_{k=1}^{n} c(B_k)$=1("parallel structure"). PrSprt- structure of the first type for set of parallel structures 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$\begin{matrix} c(A_1) & c(A_2) & \dots & c(A_n) \\ x_1 & x_2 & \dots & x_n \end{matrix}$ PrSprt- structurability for these structures. It is possible to consider the parallel self-type structure PrS_3A with m parallel structures and 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$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. The same for parallel self-type structurability $PrS_3\{$ c(A_1), c(A_2), …, 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.

The example PrQHS=$HSprt_x^{\left[\!\left[PrSprt^{PrSprt^{\begin{matrix} \text{N−level of hierarchical structure}_1 & \text{N−level of hierarchical structure}_2 & \dots & \text{N−level of hierarchical structure}_n \\ x_1 & x_2 \; \dots & & x_n \end{matrix}}_{\begin{matrix} \text{i−level of hierarchical structure}_1 & \text{i−level of hierarchical structure}_2 & \dots & \text{i−level of hierarchical structure}_n \\ x_1 & x_2 \; \dots & & x_n \end{matrix}}}_{\begin{matrix} \text{1−level of hierarchical structure}_1 & \text{1−level of hierarchical structure}_2 & \dots & \text{1−level of hierarchical structure}_n \\ x_1 & x_2 \; \dots & & x_n \end{matrix}} \right]\!\right]}$ -
N-hierarchical structure compression into point x = $(x_1, x_2, \dots, x_n)$.

Let Prf(N, PrQHS)= $\text{PrQHS}^{\text{PRQHS}^{\text{PRQHS}^{\cdots^{\text{PrQHS}}}}} \Big\}$-N levels

It can be considered Prself- PrQHS, Prf(y, PrQHS) for any y, Prf(PrQHS, PrQHS).

Parallel Compression Hierarchy Examples:

1)

$$\text{PrSprt}^{\begin{matrix} \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} & \dots & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} \\ \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} + B & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} + B & \dots & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} + B \end{matrix}} =$$

$$\left(\text{PrSprt}^{\text{PrSprt}^{\begin{matrix} \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} & \dots & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} \\ \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} & \dots & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} \end{matrix}}} \begin{matrix} \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} & \dots & \text{PrSprt}\begin{matrix} () & () & \dots & () \\ () & () & \dots & () \end{matrix} \\ B & B \; \dots & & B \end{matrix} \right)$$

2) ${}^{C+{}^{()}_{()}St{}^{()}_{()}}_{D+{}^{()}_{()}St{}^{()}_{()}}S_1t^{A+{}^{()}_{()}St{}^{()}_{()}}_{B+{}^{()}_{()}St{}^{()}_{()}} = \begin{pmatrix} {}^{{}^{()}_{()}St{}^{()}_{()}}_{{}^{()}_{()}St{}^{()}_{()}}S_1t^{{}^{()}_{()}St{}^{()}_{()}}_{{}^{()}_{()}St{}^{()}_{()}} \\ {}^{C}_{D}S_1t^{A}_{B} \end{pmatrix}$

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 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 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 - $PrS_3\{\mathrm{ca}(A_1), \mathrm{ca}(A_2), \dots, \mathrm{ca}(A_n)$.

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 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 PrSprt(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. The same for Prself- hierarchization $PrS_3\{\mathrm{h}(A_1), \mathrm{h}(A_2), \dots, \mathrm{h}(A_n)$.Can be considered PrSprt(t)$\begin{matrix} \{\mathrm{ca}(A_1), \mathrm{c}(A_1), \mathrm{h}(A_1)\} & \{\mathrm{ca}(A_2), \mathrm{c}(A_2), \mathrm{h}(A_2)\} & \dots & \{\mathrm{ca}(A_n), \mathrm{c}(A_n), \mathrm{h}(A_n)\} \\ x_1 & x_2 & \dots & x_n \end{matrix}$.

Very interesting next parallel hierarchy type:

$\begin{matrix} \text{hierarchy } A_1 & \text{hierarchy } A_2 & \dots & \text{hierarchy } A_n \\ \text{hierarchy } A_1 & \text{hierarchy } A_2 & \dots & \text{hierarchy } A_n \end{matrix}$, PrSprt(t)$\begin{matrix} \text{hierarchy } A_1 & \text{hierarchy } A_2 & \dots & \text{hierarchy } A_n \\ \text{hierarchy } A_1 & \text{hierarchy } A_2 & \dots & \text{hierarchy } A_n \end{matrix}$. 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:

$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$, PrCrt(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n \end{matrix}$,

You can enter special parallel operator PrHprt to work with hierarches: $\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n \end{matrix}$PrHprt $\begin{matrix} R_1 & R_2 & \dots & R_m \\ Q_1 & Q_2 & \dots & Q_m \end{matrix}$

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).

7 Program Operators PrSprt, PrtSpr, S¹e, 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)$\begin{matrix} g_1(t)w_1(t) & g_2(t)w_n(t) & \dots & g_n(t)w_n(t) \\ activation & activation & \dots & activation \end{matrix}$, it will be possible to obtain the execution of a parallel action $(g_1(t), g_2(t), \dots, g_n(t))$ 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 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)$\begin{matrix} (\text{UHF AC})_1(t) \coloneqq Q_1(t) & (\text{UHF AC})_2(t) \coloneqq Q_2(t) & \dots & (\text{UHF AC})_n(t) \coloneqq Q_n(t) \\ activation & activation & \dots & activation \end{matrix}$, 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) 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:

4. Simultaneous assignment of the expressions $\{p\} = (p_1, p_2, \dots, p_n)$ to the variables $\{g\} = (g_1, g_2, \dots, g_n)$. This is implemented via PrSprt$\begin{matrix} g_1 \coloneqq & g_2 \coloneqq & \dots & g_n \coloneqq \\ p_1 & p_2 & \dots & p_n \end{matrix}$.
5. Simultaneous checking the set of conditions $\{f\} = (f_1, f_2, \dots, f_n)$ for the set of expressions$\{B\} = (B_1, B_2, \dots, B_n)$. Implemented via PrSprt$\begin{matrix} IF\{B_1 f_1\}\ then & IF\{B_2 f_2\}\ then & \dots & IF\{B_n f_n\}\ then \\ x_1 & x_2 & \dots & x_n \end{matrix}$, where x_i (i = 1, …, n) can be anything.
6. Similarly for loop operators and others.

PRSprt-algorithm Examples:

1) Simultaneous addition and simultaneous parallel multiplication of sets elements $\{g_i\} = \left(g_{i_1}, g_{i_2}, \dots, g_{i_{m_j}}\right)$, i = 1, 2, …, n, j =1, 2, …, k (See point 1, 2 in 1.5 **Math Prself**
2) parallel pattern recognition:

PrSprt$\begin{matrix} IF\{q_1 \epsilon\ image\ archive_1\}\ then & IF\{q_2 \epsilon\ image\ archive_2\}\ then & \dots & IF\{q_n \epsilon\ image\ archive_n\}\ then \\ Name\ of\ q_1 & Name\ of\ q_2 & \dots & Name\ of\ q_n \end{matrix}$

The example of PrSprt-program is

$$\text{PrSprt}^{\text{PrSprt}\begin{matrix} g_1 \coloneqq & g_2 \coloneqq & \dots & g_n \coloneqq \\ p_1 & p_2 & \dots & p_n \end{matrix}}_{x_1} \quad \text{PrSprt}^{\begin{matrix} IF\{B_1 f_1\}\ then & IF\{B_2 f_2\}\ then & \dots & IF\{B_n f_n\}\ then \\ w_1 & w_2 & \dots & w_n \end{matrix}}_{x_2 \ \dots} \quad \dots \quad \text{PrSprt}^{\begin{matrix} w_1 & w_2 & \dots & w_n \\ w_1 & w_2 & \dots & w_n \end{matrix}}_{x_n}$$

$PrS_3 f$ – 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, $Sprt_{g\{R\}}^{\{R\}}$ 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 $St^{\{St_x^{\{\{a(x)\}:\{p\}\}},St_x^{IF\{\{B\}\{f\}\}\ then\ Q},St_Q^Q\}}_{\{St_x^{\{\{a(x)\}:\{p\}\}},St_x^{IF\{\{B\}\{f\}\}\ then\ Q},St_Q^Q\}}$.

$$\text{PrSprt}\begin{matrix}\text{PrSprt}\begin{matrix} g_1 := & g_2 := & \dots & g_n := \\ p_1 & p_2 & \dots & p_n\end{matrix} & \text{PrSprt}\begin{matrix} IF\{B_1f_1\}\ then & IF\{B_2f_2\}\ then & \dots & IF\{B_nf_n\}\ then \\ w_1 & w_2 & \dots & w_n\end{matrix} & \dots & \text{PrSprt}\begin{matrix} w_1 & w_2 & \dots & w_n \\ w_1 & w_2 & \dots & w_n\end{matrix} \\ \text{PrSprt}\begin{matrix} g_1 := & g_2 := & \dots & g_n := \\ p_1 & p_2 & \dots & p_n\end{matrix} & \text{PrSprt}\begin{matrix} IF\{B_1f_1\}\ then & IF\{B_2f_2\}\ then & \dots & IF\{B_nf_n\}\ then \\ w_1 & w_2 & \dots & w_n\end{matrix} & \dots & \text{PrSprt}\begin{matrix} w_1 & w_2 & \dots & w_n \\ w_1 & w_2 & \dots & w_n\end{matrix}\end{matrix}$$

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$\begin{matrix} A_1 & A_2 & \dots & A_n \\ B_1 & B_2 & \dots & B_n\end{matrix}$.

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$\begin{matrix} A_1 & A_2 & \dots & A_n \\ A_1 & A_2 & \dots & A_n\end{matrix}$.

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 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)$\begin{matrix} (\text{UHF AC})_1(t) := Q_1(t) & (\text{UHF AC})_2(t) := Q_2(t) & \dots & (\text{UHF AC})_n(t) := Q_n(t) \\ activation & activation\ \dots & & activation\end{matrix}$. UHF AC field activation is used.

Consider the dynamic PrSprt and $PrS_3f(t)$ programming:

1. The process of simultaneous assignment of the expressions $\{p(t)\} = (p_1(t), p_2(t), \dots, p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), \dots, g_n(t))$ 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 $\{f(t)\} = \left(f(t)_1, f_{2,}(t) \dots, f(t)_n\right)$ for a set of expressions $\{B(t)\} = (B_1(t), B_2\,(t), \dots, B_n(t))$ 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) = $(x_1(t), x_2(t), \dots, x_n(t))$. can be any.
3. Similarly for loop operators and others.

$PrS_3f(t)$– software operators will differ only in that the aggregates $\{g(t)\}, \{p(t)\}, \{B(t)\}, \{f(t)\}$ will be formed from corresponding processes 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 Prt_{S_4f} is parallel analogue of t_{S_4f} can be used for programming. Here are some of the PrtSpr -program operators [13].

1. Simultaneous expelling assignment of the expressions $\{p\} = (p_1, p_2, \dots, p_n)$ from the variables $\{g\} = (g_1, g_2, \dots, g_n)$. It's implemented through $\begin{matrix} g_1 & g_2 & \dots & g_n \\ =: p_1 & =: p_2 & \dots & =: p_n\end{matrix}$ PrSprt.
2. Simultaneous expelling checks the set of conditions $\{f\} = \left(f_1, f_{2,} \dots, f_n\right)$ for a set of expressions $\{B\} = \left(B_1, B_{2,} \dots, B_n\right)$. It's implemented through PrSprt$\begin{matrix} x_1 & x_2 & \dots & x_n \\ IF\{B_1f_1\}\ then & IF\{B_2f_2\}\ then & \dots & IF\{B_nf_n\}\ then\end{matrix}$, where x_i (i = 1, …, n) can be anything.
3. Similarly for loop operators and others.

Prt_{S_4f} – 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

$$\begin{matrix} C_1 & C_2 & \dots & C_m \\ D_1 & D_2 & \dots & D_m \end{matrix}\text{PrSrt} = \left\{ \begin{matrix} \sum_{i=1}^{m} Q_i + \begin{matrix} \{\,\} & \{\,\} & \dots & \{\,\} \\ D_1 - D_1 \cap C_1 & D_2 - D_2 \cap C_2 & \dots & D_m - D_m \cap C_m \end{matrix}\text{PrSrt} \\ \sum_{i=1}^{m} (C_i - D_i \cap C_i) - (D_i - D_i \cap C_i) \end{matrix} \right\}$$, where Q_i is oself-set for $(D_i \cap C_i)$ [13].

Consider the dynamic PrtSpr and $Prt(t)_{S_4f}$ programming at time t.

1. The process of simultaneous assignment of the expressions $\{p(t)\} = (p_1(t), p_2(t), \dots, p_n(t))$ to the variables $\{g(t)\} = (g_1(t), g_2(t), \dots, g_n(t))$. 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 $\{f(t)\} = \left(f(t)_1, f_{2,}(t) \dots, f(t)_n\right)$ for a set of expressions$\{B(t)\} = (B_1(t), B_2(t), \dots, B_n(t))$ 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) = $(x_1(t), x_2(t), \dots, x_n(t))$. can be any.

3.Similarly for loop operators and others.

$Prt(t)_{S_4f}$ – 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:

$$\begin{matrix} C_1(t) & C_2(t) & \dots & C_m(t) \\ D_1(t) & D_2(t) & \dots & D_m(t) \end{matrix}\text{PrSrt(t)} =$$

$$\left\{ \begin{matrix} \sum_{i=1}^{m} Q_i(t) + \begin{matrix} \{\,\} & \{\,\} & \dots & \{\,\} \\ D_1(t) - D_1(t) \cap C_1(t) & D_2(t) - D_2(t) \cap C_2(t) & \dots & D_2(t) - D_2(t) \cap C_2(t) \end{matrix}\text{PrSrt} \\ \sum_{i=1}^{m} (C_i(t) - D_i(t) \cap C_i(t)) - (D_i(t) - D_i(t) \cap C_i(t)) \end{matrix} \right\}$$, where $Q_i(t)$ is oself-set for $(D_i(t) \cap C_i(t))$ [16].

Consider S[1]e -program operators (form ${}_{D}^{B}S^1t_B^A$). For example, ${}_{\{=:\{p(t)\}\}}^{\{a(t)\}}S^1t_{\{a(t)\}}^{IF\{\{B(t)\}\{f(t)\}\}\, then\, Q(t)}$.

Consider hierarchical dynamic S[1]e-program operator: (form $\begin{matrix} {}_{A}^{B}S^1t_B^A * \\ {}_{D-A}^{B}S^1t_B^A \end{matrix}$).

Consider PrSrt(t)- program operators

$PrSprt_{t_0}^{\{{}^{q}\left(\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}PrSpr\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}\right)_{W_q} PrSprt_{q\left(\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}PrSpr\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}\right)}^{Eq}, PrSprt_{d_r}^{\{El^{d_r}\}}\}}$. —program structure example, where the assemblage point d_r is the cursor, it is quite complex self—program.

Remark. Energy of a living organism:

Prf(r, u(E_q)) =$PrSprt_{t_0}^{\{{}^{q}\left(\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}PrSpr\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}\right)_{W_q} PrSprt_{q\left(\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}PrSpr\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}\right)}^{Eq}, PrSprt_{d_r}^{\{El^{d_r}\}}\}}$.

$\begin{matrix} u & u \\ u & u \end{matrix}PrSpr\begin{matrix} u & u \\ u & u \end{matrix}$ -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, El^{d_r} - a bundle of fibers of external energy self-capacities, collected at the point of assembly of the cocoon of a living organism.

$PrSprt_{t_0}^{\{{}^{q}\left(\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}PrSpr\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}\right)_{W_q} PrSprt_{q\left(\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}PrSpr\begin{smallmatrix} u & u \\ u & u \end{smallmatrix}\right)}^{Eq}, PrSprt_{d_r}^{\{El^{d_r}\}}\}}$ can be interpreted as a program operator.

Consider structure examples hierarchical Set_1-program operator

3. $\left(\begin{matrix} S_{01}^{et} fB \\ {}_{Q-B}^{R-B}S_1t_B^{A-B} \end{matrix} \right)$,

4. $\begin{pmatrix} S_{21}^{et} f_A^B \\ {}_{Q-A}^{R-A} S_1 t_B^A \end{pmatrix}$.

Appendix

Supplement for string theory: May be to try represent elementary particles in the form of continual self-elements of the type: $PrSprt\begin{matrix} \uparrow I \downarrow_{-1}^{1}. & \downarrow I \uparrow_{-\infty}^{\infty} & \dots & \downarrow I \uparrow_{-1}^{1} \\ x_1 & x_2 & \dots & x_n \end{matrix}$ 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 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(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. PrSprt(t)$\begin{matrix} f(A_1) & f(A_2) & \dots & f(A_n) \\ x_1 & x_2 & \dots & x_n \end{matrix}$.[- 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(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$. 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(t)$\begin{matrix} A_1 & A_2 & \dots & A_n \\ x_1 & x_2 & \dots & x_n \end{matrix}$, A is usual group, PrSprt(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: $PrSf_iA$, 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(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 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 a solutions or structures the inversion of the Prself-equation concerning unknowns, i.e., the structure of the Prself-equation is complete.

Remark. Energy of a living organism:

$$\text{Prf(r, u}(E_q)) = PrSprt_{t_0}^{\{ {}^{q}\begin{pmatrix} u & u^{PrSpr}{}_{u}^{u} & u \\ u & u & u \end{pmatrix}_{W_q} PrSprt_{q\begin{pmatrix} u & u^{PrSpr}{}_{u}^{u} & u \\ u & u & u \end{pmatrix}}^{E_q}, PrSprt_{d_r}^{\{El^{d_r}\}} \}}.$$

$\begin{matrix} u & u \\ u & u \end{matrix} PrSpr \begin{matrix} u & u \\ u & u \end{matrix}$ - 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, El^{d_r} - a bundle of fibers of external energy self-capacities, collected at the point of assembly of the cocoon of a living organism [14-17].

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.

Author's Contributions
The contribution of the authors is the same, we will not separate.

Acknowledgements
We are grateful to our academic supervisor Scientific director: Volodymyr Pasynkov - PhD of physic-mathematical science, assistant professor of applied mathematics and calculated techniques department of National Metallurgical Academy (Ukraine) for guidance and assistance in writing this article.

References

1. Danilishyn, I., & Danilishyn, O. Advances in Neurology and Neuroscience.
2. Danilishyn, I., Danilishyn, O., & Pasynkov, V. (2023). THE USAGE OF SIT-ELEMENTS FOR NETWORKS. *Collection of scientific papers «ΛΌΓΟΣ»,* (March 31, 2023; Zurich, Switzerland), 129-134.
3. Danilishyn, I., Danilishyn, O., & Pasynkov, V. (2023). MATHEMATICS ST, PROGRAMMING OPERATORS ST AND SOME EMPLOYMENT. *Collection of scientific papers «SCIENTIA»,* (March 10, 2023; Valencia, Spain), 123-127.
4. Danilishin, O., & Danilishin, I. (2023). Dynamic Sets Theory: Sit-elements and Their Applications.
5. Blokhintsev, D. I. (1981). Quantum Mechanics: Lectures on Selected Topics. *Moscow: Atomizdat,* 87-8.
6. Oleksandr, D., & Illia, D. (2023). Dynamical Sets Theory: S2t-Elements and Their Applications. *J Math Techniques Comput Math, 2*(12), 479-498.
7. 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.
8. Danilishyn, O., Danilishyn, I. (2023). Dynamic Sets S1et and Some of their Applications in Physics. *Science Set Journal of Physics*, 25 September,1-11, 815- article1695707465.pdf (mkscienceset.com)
9. Thomas J. Jech. (1971). LECTURES IN SET THEORY WITH PARTICULAR EMPHASIS ON THE METHOD OF FORCING. *SPRINGER-VERLAG Berlin*. Heidelberg. New York.
10. Dontchev, A. L. (Ed.). (1983). *Perturbations, approximations and sensitivity analysis of optimal control systems.* Berlin, Heidelberg: Springer Berlin Heidelberg.
11. Danilishyn, I. V., Danilishyn, O. V. (2023). 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.
12. Danilishyn, I. V., Danilishyn, O. V. (2023). CONTINUAL SIT-ELEMENTS AND DYNAMICAL SIT-ELEMENTS. Theoretical and practical aspectsof modern scientific research:Collection of scientific papers «ΛΌΓΟΣ» with Proceedings of the IIInternational Scientific and Practical Conference, Seoul,April28, *Seoul-Vinnytsia: Case Co., Ltd.& European Scientific Platform,* pp.144-150.
13. Danilishyn, I. V., Danilishyn, O. V. (2023). DYNAMICAL SIT-ELEMENTS. IV International Scientific and Practical Conference"*GRUNDLAGEN DER MODERNEN WISSENCHAFTLICHEN FORSCHUNG*", Zurich, Switzerland.
14. Krain, S. G. (1967). Linear differential equations in Banach space. M.,Science, (in Russian).
15. Galushkin, A. (2010). Networks: principles of the theory. Hot line-Telecom. M., (in Russian).
16. Danilishyn, I. V., Danilishyn, O. V. (2023). tS – ELEMENTS. Collection of scientific papers «ΛΌΓΟΣ» with Proceedings of the V International Scientific and Practical Conference, Oxford, June 23, Oxford-Vinnytsia: P.C. Publishing House & European Scientific Platform, *Theoretical and empirical scientific research: concept and trends*. pp.156-161