TRANSLATIONS OF NETWORK LANGUAGES

DRAFT[*]

BORIS STILMAN
Department of Computer Science & Engineering, University of Colorado at Denver
Campus Box 109, Denver, CO 80217-3364. Email: bstilman@cse.cudenver.edu

Abstract—In this paper we describe new results of research on geometrical properties of complex control systems, the so-called Linguistic Geometry. This research includes the development of syntactic tools for *knowledgerepresentation* and *reasoning* about large-scale hierarchical complex systems. It relies on the formalization of *searchheuristics* of high-skilled human experts that have resulted in the development of successful applications in different areas. A hierarchy of subsystems of a complex system, the networks of paths, is represented as a hierarchy of formal languages. In this paper we investigate transformations of these networks while system moves from one state to another. The investigation consists of formal, constructive separation of changed and unchanged parts of system representation, the hierarchy of languages. Thus, we address the problem relative to the well-known Frame Problem for planning systems. A partial solution is presented in the form of the theorem about translations of network languages. Formal considerations are illustrated by example of Air Force robotic vehicles.

1. Introduction

Important real-world problems can be formally represented as problems of reasoning about complex control systems. The difficulties we meet trying to find the optimal operation for real-world complex systems are well known. While the formalization of the problem, as a rule, is not difficult, an algorithm that finds its solution usually results in the search of many variations. For small-dimensional "toy" problems a solution can be obtained; however, for most real-world problems the dimension increases and the number of variations increases significantly, usually exponentially, as a function of dimension [1]. Thus, most real-world search problems are not solvable with the help of exact algorithms in a reasonable amount of time.

There have been many attempts to design different approximate algorithms. One of the basic ideas is to decrease the dimension of the real-world problem following the approach of a *human expert in a certain field*, by breaking the problem into smaller subproblems. There are two most important issues in this decomposition.

The first issue is to find out how to break a complex system down into subsystems, to study these subsystems separately or in combinations, making appropriate searches, and eventually combine optimal solutions for the subsystems into an approximately optimal solution for the entire system [2–4]. It is easy if the system can be decomposed into completely independent subsystems. Usually, the subsystems are not independent, and the system can be considered as *nearly decomposable* [2]. For such problems we need the techniques that can handle each subsystem separately and then introduce the impact of potential interactions of these subsystems into the final solution.

The second issue is to avoid recomputation of the entire system state provided that system operates by moving from one state to another. Instead, we should consider only that part of the state that may have changed. For complex systems the Problem of Change or *FrameProblem* [5–8] consists of representation of knowledge in such a way that we can effectively determine which

[*] The final version was published as Stilman, B., Translations of Network Languages, An International Journal: Computers & Mathematics with Applications., Vol. 27, No. 2, pp. 65-98, 1994.

facts must change and which must not. It is especially important to make this determination without exhaustive search when the complexity of the system state increases. There is no universal recipe for this step since such a recipe must be based on the connections between the facts of the particular problem.

These ideas have been implemented for many problems with varying degrees of success. Implementations based on the formal theories of linear and nonlinear planning [6–13] meet hard efficiency problems. An efficient planner requires an intensive use of heuristic knowledge. On the other hand, a pure heuristic implementation is unique. There is no general constructive approach for such implementations. Each new problem should be carefully studied and previous experience usually can not be applied. Basically, we can not answer the question: what are the formal properties of human heuristics which drove us to a successful hierarchy for a given problem and how can we apply the same ideas in a different problem domain. Moreover, every attempt to evaluate the computational complexity and quality of a pilot solution requires implementing its program, which in itself is a unique task for each problem.

In the 1960's a formal syntactic approach to the investigation of properties of natural language resulted in the fast development of a theory of formal languages by Chomsky [14], Ginsburg [15], and others [16, 17]. This development provided an interesting opportunity for dissemination of this approach to different areas. In particular, there came an idea of analogous linguistic representation of images. This idea was successfully developed into syntactic methods of pattern recognition by Fu [18], Narasimhan [19], and Pavlidis [20], and picture description languages by Shaw [21], Feder [22], and Phaltz, Rosenfeld [23]. The power of a linguistic approach might be explained, in particular, by the recursive nature and expressiveness of language generating rules, i.e., formal grammars.

Searching for the adequate mathematical tools formalizing human heuristics of dynamic hierarchy, we have transformed the idea of linguistic representation of complex real-world and artificial images into the idea of similar representation of complex hierarchical systems [24]. However, the appropriate languages should possess more sophisticated attributes than languages usually used for pattern description. They should describe mathematically all of the essential syntactic and semantic features of the system and search, and be easily generated by certain controlled grammars. The origin of such languages can be traced back to the origin of SNOBOL-4 programming language and the research on programmed attribute grammars and languages by Knuth [16], Rozenkrantz [17], and Volchenkov [25].

A mathematical environment (a "glue") for the formal implementation of this approach was developed following the theories of formal problem solving and planning by Nilsson, Fikes [6], Sacerdoti [9], McCarthy, Hayes [5], and subsequent work [8-13], based on first order predicate calculus.

To show the power of the linguistic approach it is important that the chosen models of the heuristic hierarchical system be sufficiently complex, poorly formalized, and have successful applications in different areas. Such a model was developed by Botvinnik, Stilman, and others, and successfully applied to scheduling, planning, and computer chess. The hierarchical constructions were introduced in [4] in the form of ideas and plausible discussions.

An application of this hierarchy to a chess model was implemented in full as program PIONEER [4]. Similar heuristic hierarchy was implemented for power equipment maintenance in a number of computer programs being used for maintenance scheduling all over the USSR [24, 26, 27, 33, 34]. The results shown by these programs in solving complex chess and scheduling problems indicate that implementations of the dynamic hierarchy resulted in the extremely goal-driven algorithms generating search trees with a branching factor close to 1.

In order to discover the inner properties of human expert heuristics, which were successful in a certain class of complex systems, we develop a formal theory, the Linguistic Geometry [28–34]. In these papers we described the domain of applicability of the theory, introduced a class of formal grammars to be used, specified and investigated the Hierarchy of Formal Languages. Papers [28, 29] include the general survey of this approach. The works [30-33] describe in detail and investigate the lowest level of the Hierarchy, the Language of Trajectories. The next level, the Family of Network Languages, considered in [34]. Basically, papers [28-34] deal with the formalization of the first issue of decomposition of the complex system, the break into subsystems and their representation.

In this paper we address the second issue, the Problem of Change. In order to approach a formal solution of this Problem, we employ relations of reachability [33] and trajectory connection. Based on that, we create a hierarchy of languages for each system state. Then we investigate the translations of one hierarchy into another while system moves to another state. Our goal is to separate changed and unchanged parts of this hierarchy during the translation. This separation should be done constructively, in the form of an algorithm. Theorem 8.1 (Section 8) gives a partial solution of this problem. In Sections 9, 10 we illustrate formal discussion by example of Air Force robotic vehicles.

2. Class of Problems

The *class of problems* to be studied includes problems of optimal operation of a complex system, with a twin-set of *elements* and *points* where elements are units moving from one point to another.

More precisely, a **Complex System** is the following eight-tuple:

$$< \mathbf{X}, \mathbf{P}, \mathbf{R_p}, \{\mathbf{ON}\}, \mathbf{v}, \mathbf{S_i}, \mathbf{S_t}, \mathbf{TR}>,$$

where

X$=\{x_i\}$ is a finite set of *points*; **P**$=\{p_i\}$ is a finite set of *elements*;

P is a union of two non-intersecting subsets P_1 and P_2;

$\mathbf{R_p(x, y)}$ is a set of binary relations of *reachability* in X (x and y are from X, p from P);

ON(p)=x, where ON is a partial function of *placement* from P into X;

v is a function on P with positive integer values; it describes the *values* of elements. The Complex System searches the state space, which should have initial and target states;

$\mathbf{S_i}$ and $\mathbf{S_t}$ are the descriptions of the *initial* and *target* states in the language of the first order predicate calculus, which matches with each relation a certain Well-Formed Formula (WFF). Thus, each state from S_i or S_t is described by a certain set of WFF of the form $\{ON(p_j) = x_k\}$;

TR is a set of operators, TRANSITION(p, x, y), of transition of the System from one state to another one. These operators describe the transition in terms of two lists of WFF (to be removed and added to the description of the state), and of WFF of applicability of the transition. Here,

Remove list: ON(p)=x, ON(q)=y;
Add list: ON(p)=y;
Applicability list: $(ON(p)=x)$^$R_p(x, y)$,

where p belongs to P_1 and q belongs to P_2 or vice versa. The transitions are carried out in turn with participation of elements p from P_1 and P_2 respectively; omission of a turn is permitted.

According to definition of the set P, the elements of the System are divided into two subsets P_1 and P_2. They might be considered as units moving along the reachable points. Element p can move from point x to point y if these points are reachable, i.e., $R_p(x, y)$ holds. The current location of each element is described by the equation ON(p)=x. Thus, the description of each state of the System $\{ON(p_j)=x_k\}$ is the set of descriptions of the locations of the elements. The operator TRANSITION(p, x, y) describes the change of the state of the System caused by the move of the element p from point x to point y. The element q from point y must be withdrawn (eliminated) if p and q belong to the different subsets P_1 and P_2.

The problem of the optimal operation of the System is considered as a search for the optimal sequence of transitions leading from one of the initial states of S_i to a target state S of S_t. It is a very general representation, e.g., in robot control problems *elements* are autonomous robots moving along the *points* of a complex hazardous environment on the surface or in space. The *elements* are divided into two or more sides; the goal of each side is to attack and destroy opposite side *elements* and to protect its own. Each side aims to maximize a gain, the total value of opposite *elements* destroyed and withdrawn from the system. Such a withdrawal happens if an attacking *element* moves to a point where there is already an *element* of the opposite side.

A robot control model can be represented as a Complex System naturally (Fig. 1). A set of **X** represents the operational district that could be the area of combat operation broken into squares, e.g., in the form of the table 8 x 8, n=64. It could be a space operation, where X represents the set of different orbits, or a navy battlefield, etc. **P** is the set of robots or autonomous vehicles. It is broken into two subsets P_1 and P_2 with opposing interests; $\mathbf{R_p(x,\ \ y)}$ represent moving capabilities of different robots: robot p can move from point x to point y if $R_p(x, y)$ holds. Some of the robots can crawl, the other can jump or ride, or even sail and fly. Some of them move fast and can reach point y (from x) in "one step", i.e., $R_p(x, y)$ holds, others can do that in *k* steps only, and many of them can not reach this point at all. **ON(p)=x**, if robot p is at the point x; **v(p)** is the value of robot p. This value might be determined by the technical parameters of the robot. It might include the immediate value of this robot for the given combat operation; $\mathbf{S_i}$ is an arbitrary initial state of operation for analysis, or the starting state; $\mathbf{S_t}$ is the set of target states. These might be the states where robots of each side reached specified points. On the other hand S_t can specify states where opposing robots of the highest value are destroyed. The set of WFF $\{ON(p_j) = x_k\}$ corresponds to the list of robots with their coordinates in each state. **TRANSITION(p,x,y)** represents the move of the robot p from square x to square y; if a robot of the opposing side stands on y, a removal occurs, i.e., robot on y is destroyed and removed.

Four robots with different moving capabilities are shown in Fig. 1. The operational district X is the table 8 x 8. Squares d5, e6, f7, g3, g4 representing a restricted area are excluded. Robot FIGHTER standing on f6, can move to any next square (shown by arrows). The other robot BOMBER from h5 can move only straight ahead, e.g., from h5 to h4, from h4 to h3, etc. Robot MISSILE standing on d7 can move only along diagonals but it can pass squares located on this diagonal without stops. Thus, robot FIGHTER on f6 can reach any of the points y {e5, e7, g7, g6, g5, f4} in one step, i.e., $R_{FIGHTER}(f6, y)$ holds, while MISSILE can reach a4, b5, c6, e8, c8 in one step. Robot BOMBER standing on h5 can reach only h4. Obviously, moving capabilities of these robots are similar to the well-known chess pieces King, Bishop, and Pawn, respectively.

Assume that robots FIGHTER and MISSILE belong to one side, while BOMBER belong to the opposite side: FIGHTER P_1, MISSILE P_1, BOMBER P_2. Also assume that the fourth robot, TARGET, (or unmoving device) stands on h1. TARGET belongs to P_1 which means that character function χ(BOMBER, TARGET)=0. (Function χ(p, q) is defined on P x P and equals 1 if p and q both belong to P_1 or P_2; χ(p, q) = 0 in the remaining cases.) Thus, robot BOMBER should reach point h1 to destroy the TARGET while FIGHTER and MISSILE will try to intercept this motion.

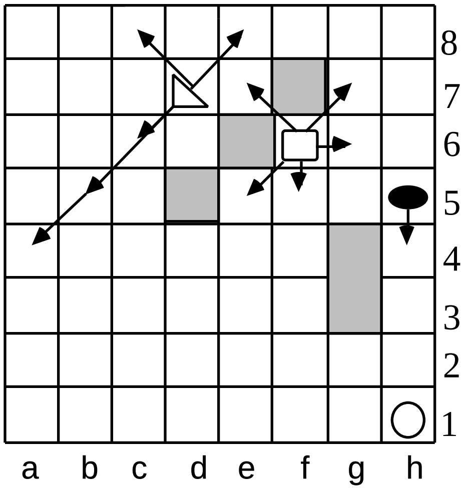

Fig. 1. *Interpretation of the Complex System for the robot control model.*

It was easy to show that robot control model can be considered as Complex System. Many different technical and human society systems (including military battlefield systems, economic competition, positional games) which can be represented as twin-sets of movable units (of two or more opposite sides) and their locations, thus can be considered as Complex Systems. But it is interesting that a wide class of operation research problems such as power maintenance scheduling, long-range planning, operations planning, without obvious movable units and opposed sides can be represented as Complex Systems [24, 33, 34]. The idea is that optimal variant of operation of these real-world systems may be artificially reduced to a two-sides positional game where one side strives to achieve a goal and the other is responsible for provision of resources.

With such a problem statement for the search of the optimal sequence of transitions leading to the target state, we could use formal methods like those in the problem-solving system STRIPS [6], nonlinear planner NOAH [10], or in subsequent planning systems [8-13]. However, the search would have to be made in a space of a huge dimension (for nontrivial examples, that, of course, do not include our “toy-dimensional” robot control model). Thus, in practice no solution would be obtained. We devote ourselves to search for an approximate solution of a reformulated problem. A one-goal, one-level system could be substituted for a multi-goal, multi-level system by introducing intermediate goals and breaking the system down into subsystems striving to attain these goals. The goals of the subsystems are individual but coordinated within the main mutual goal. For example, each second-level subsystem includes elements of both sides: the goal of one side is to attack and gain some element (a target), while the other side tries to protect it. In a robot control, this means the selection of a couple of robots of opposing sides: one – as an attacking element, and the other – as a local target, generation of the paths for approaching the target as well as the paths of other robots supporting the attack or protecting the target.

3. Language of Trajectories

Following a linguistic approach each subsystem could be represented as a string of symbols with parameters: $\boldsymbol{a}(x_1)\boldsymbol{a}(x_2)\ldots\boldsymbol{a}(x_n)$, where the values of the parameters incorporate the semantics of the problem domain or lower-level subsystems.

Here, we define the lowest-level language of the hierarchy of languages, the Language of Trajectories. It serves as a building block to create the upper-level languages. The Language of Trajectories actually formalizes the description of the set of lowest-level subsystems, the set of different paths between different points of the Complex System. An element might follow a path to achieve the goal “connected with the ending point”.

A ***trajectory*** for an element p of P with the beginning at x of X and the end at y of X ($x \neq y$) with a length l is the following string of symbols with parameters, points of X: $t_o=\boldsymbol{a}(x)\boldsymbol{a}(x_1)\ldots\boldsymbol{a}(x_l)$. Here each successive point x_{i+1} is reachable from the previous point x_i: $R_p(x_i, x_{i+1})$ holds for $i=0,1,\ldots, l–1$; element p stands at the point x: ON(p)=x. The empty string ***e*** is called a trajectory of the length 0. We denote $t_p(x, y, l)$ the set of trajectories in which p, x, y, and l are the same. $P(t_o)=\{x, x_1, \ldots, x_l\}$ is the set of parametric values of the trajectory t_o.

A ***shortest trajectory*** t of $t_p(x, y, l)$ is the trajectory of minimum length for the given beginning x, end y and element p.

A ***Language of Trajectories*** $L_t^H(S)$ for the Complex System in a state S is the set of all the trajectories of the length less or equal H.

A deeper study of the Language of Trajectories and generating grammars $\mathbf{G_t^{(1)}}$, $\mathbf{G_t^{(2)}}$ is presented in [30-33].

4. Languages of Trajectory Networks

After defining the Language of Trajectories, we have the tools for the breakdown of our System into subsystems. According to the ideas presented in [4], these subsystems should be various types of trajectory networks, i.e., some sets of interconnected trajectories with one singled out trajectory called the main trajectory. An example of such a network is shown in Fig. 2.

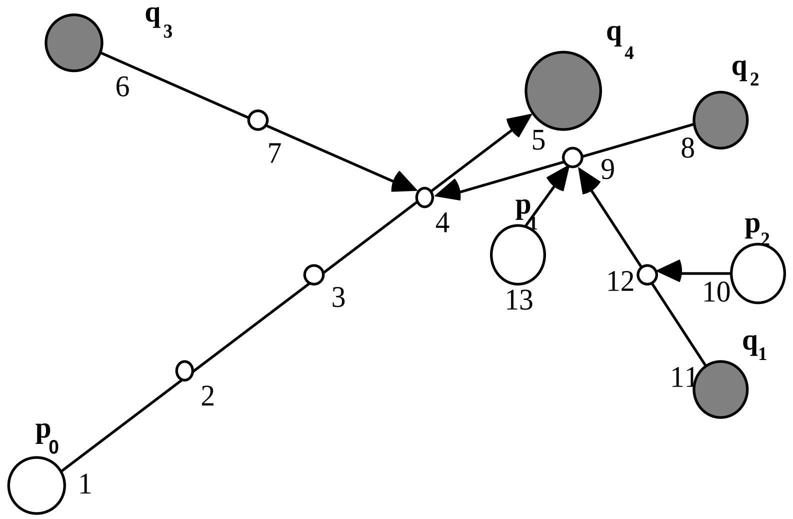

Fig. 2. *A network language interpretation.*

The basic idea behind these networks is as follows. Element p_o should move along the main trajectory $\boldsymbol{a}(1)\boldsymbol{a}(2)\boldsymbol{a}(3)\boldsymbol{a}(4)\boldsymbol{a}(5)$ to reach the ending point 5 and remove the target q_4 (an opposite element). Naturally, the opposite elements should try to disturb those motions by controlling the intermediate points of the main trajectory. They should come closer to these points (to the point 4 in Fig. 2) and remove element p_O after its arrival (at point 4). For this purpose, elements q_3 or q_2 should move along the trajectories $\boldsymbol{a}(6)\boldsymbol{a}(7)\boldsymbol{a}(4)$ and $\boldsymbol{a}(8)\boldsymbol{a}(9)\boldsymbol{a}(4)$, respectively, and wait (if necessary) on the next to last point (7 or 9) for the arrival of element p_o at point 4. Similarly, element p_1 of the same side as p_o might try to disturb the motion of q_2 by controlling point 9 along the trajectory $\boldsymbol{a}(13)\boldsymbol{a}(9)$. It makes sense for the opposite side to include the trajectory $\boldsymbol{a}(11)\boldsymbol{a}(12)\boldsymbol{a}(9)$ of element q_1 to prevent this control.

Similar networks are used for the breakdown of complex systems in different areas. Let us consider a formal linguistic formalization of such networks. The Language of Trajectories describes "one-dimensional" objects by joining symbols into a string employing reachability relation $R_p(x, y)$. To describe networks, i.e., “two-dimensional" objects made up of trajectories, we use the relation of *trajectory connection.*

Definition 4.1. A ***trajectory connection*** of the trajectories t_1 and t_2 is the relation $C(t_1,t_2)$. It holds, if the ending link of the trajectory t_1 coincides with an intermediate link of the trajectory t_2; more precisely t_1 is connected with t_2, if among the parameter values $P(t_2)=\{y,y_1,\ldots,y_l\}$ of trajectory t_2 there is a value $y_i = x_k$, where $t_1=\boldsymbol{a}(x_o)\boldsymbol{a}(x_1)\ldots\boldsymbol{a}(x_k)$. If t_1 belongs to some set of trajectories with the common end-point, than the entire set is said to be connected with the trajectory t_2.

For example, in Fig. 2 the trajectories $\boldsymbol{a}(6)\boldsymbol{a}(7)\boldsymbol{a}(4)$ and $\boldsymbol{a}(8)\boldsymbol{a}(9)\boldsymbol{a}(4)$ are connected with the main trajectory $\boldsymbol{a}(1)\boldsymbol{a}(2)\boldsymbol{a}(3)\boldsymbol{a}(4)\boldsymbol{a}(5)$ through point 4. Trajectories $\boldsymbol{a}(13)\boldsymbol{a}(9)$ and $\boldsymbol{a}(11)\boldsymbol{a}(12)\boldsymbol{a}(9)$ are connected with $\boldsymbol{a}(8)\boldsymbol{a}(9)\boldsymbol{a}(4)$.

Definition 4.2. A set of trajectories $CA_B(t)$ from B, with which trajectory t is connected, is called the ***bundle of trajectories*** for trajectory t relative to the set B of trajectories.

To formalize the trajectory networks we should define some routine operations on the set of trajectories: a k-th degree of connection and a transitive closure.

Definition 4.3. A ***k-th degree of the relation C*** on the set of trajectories A (denoted by $C_A{}^k$) is defined as usual by induction.

For k = 1 $C_A{}^k(t_1,t_2)$ coincides with $C(t_1,t_2)$ for t_1,t_2 from A.

For k > 1 $C_A{}^k(t_1,t_2)$ holds if and only if there exists a trajectory t_3 from A, such that $C(t_1,t_3)$ and $C_A{}^{k-1}(t_3,t_2)$ both hold.

Trajectory ***a***(11)***a***(12)***a***(9) in Fig. 2 is connected (degree 2) with trajectory ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5), i.e., C^2(***a***(11)***a***(12)***a***(9), ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5)) holds.

Definition 4.4. A ***transitive closure of the relation*** **C** on the set of trajectories A (denoted by $C_A{}^+$) is a relation, such that $C_A{}^+(t_1,t_2)$ holds for t_1 and t_2 from A, if and only if there exists i > 0 that $C_A{}^i(t_1,t_2)$ holds.

The trajectory ***a***(10)***a***(12) in Fig. 2 is in transitive closure to the trajectory ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5) because C^3(***a***(10)***a***(12), ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5)) holds by means of the chain of trajectories ***a***(11)***a***(12)***a***(9) and ***a***(8)***a***(9)***a***(4).

Definition 4.5. A ***trajectory network*** **W** relative to trajectory t_o is a finite set of trajectories $t_o,t_1,\ldots,t_k$ from the language $L_t{}^H(S)$ that possesses the following property: for every trajectory t_i from W (i = 1, 2,…,k) the relation $C_W{}^+(t_i,t_o)$ holds, i.e., each trajectory of the network W is connected with the trajectory t_o that was singled out by a subset of interconnected trajectories of this network. If the relation $C_W{}^{\mathbf{m}}(t_i, t_o)$ holds, trajectory t_i is called the ***m negation trajectory***.

Obviously, the trajectories in Fig. 2 form a trajectory network relative to the main trajectory ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5). We are now ready to define network languages.

Definition 4.6. A ***family of trajectory network languages*** $\mathbf{L_C(S)}$ in a state S of the Complex System is the family of languages that contains strings of the form

$$t(t_1, param)t(t_p, param)\ldots t(t_m, param),$$

where *param* in parentheses substitute for the other parameters of a particular language. All the symbols of the string t_1, t_2,…, t_m correspond to trajectories that form a trajectory network W relative to t_1.

Different members of this family correspond to different types of trajectory network languages, which describe particular subsystems for solving search problems. One of such languages is a language that describes specific networks called Zones. They play a main role in the model considered here [4, 34]. The formal definition of this language is essentially constructive and requires showing explicitly a method for generating this language, i.e., a certain formal grammar. This grammar will be discussed later. In order to make our points transparent, first, we define the Language of Zones informally.

A *Language of Zones* is a trajectory network language with strings of the form

$$Z=t(p_o,t_o,\tau_o)\ t(p_1,t_1,\tau_1)\ldots t(p_k,t_k,\tau_k),$$

where $t_o,t_1,\ldots,t_k$ are the trajectories of elements $p_o,p_2,\ldots,p_k$ respectively; $\tau_o,\tau_1,\ldots,\tau_k$ are positive integer numbers (or 0) which “denote the time allocated for the motion along the trajectories in a correspondence to the mutual goal of this Zone: to remove the target element – for one side, and to protect it – for the opposite side. Trajectory $t(p_o,t_o,\tau_o)$ is called the *main trajectory* of the Zone. The element q standing on the ending point of the main trajectory is called the *target.* The elements p_o and q belong to the opposite sides.

To make it clearer let us show the Zone corresponding to the trajectory network in Fig. 2.

Z =***t***(p_o, ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5), 4)***t***(q_3, ***a***(6)***a***(7)***a***(4), 3)***t***(q_2, ***a***(8)***a***(9)***a***(4), 3)***t***(p_1, ***a***(13)***a***(9), 1)
t(q_1, ***a***(11)***a***(12)***a***(9), 2) ***t***(p_2, ***a***(10)***a***(12), 1)

Assume that the goal of the white side is to remove target q_4, while the goal of the black side is to protect it. According to these goals element p_o starts the motion to the target, while blacks start in its turn to move their elements q_2 or q_3 to intercept element p_o. Actually, only those black trajectories are to be included into the Zone where the motion of the element makes sense, i. e., the *length of the trajectory is less than the amount of time (third parameter* τ) *allocated to it.* For example, the motion along the trajectories ***a***(6)***a***(7)***a***(4) and ***a***(8)***a***(9)***a***(4) makes sense, because they are of length 2 and time allocated equals 3: each of the elements has 3 time intervals to reach point 4 to intercept element p_o assuming one would go along the main trajectory without move omission. According to definition of Zone the trajectories of white elements (except p_o) could only be of the length 1, e.g., ***a***(13)***a***(9) or ***a***(10)***a***(12). As far as element p_1 can intercept motion of the element q_2 at the point 9, blacks include into the Zone the trajectory ***a***(11)***a***(12)***a***(9) of the element q_1, which has enough time for motion to prevent this interception. The total amount of time allocated to the whole bunch of black trajectories connected (directly or indirectly) with the given point of main trajectory is determined by the number of that point. For example, for the point 4 it equals 3 time intervals.

5. Language of Zones

Here we consider a formal definition of the Language of Zones employing class of controlled grammars. This class of grammars was formally introduced and considered in details in [33, 34]. An example of actual generation in such a grammar applied to the Network Language is presented in Section 9. Definition 5.1 has some differences with definition considered in [34]. It applies to the definition of function ***ALPHA*** (Table 2). Both definitions are valid but the definition in [34] should be considered as the definition of a *minimal Zone* while next we define a *maximum Zone*. The difference is that maximal Zone includes *all* the trajectories of *all* the opposite elements which can participate in the interception of the main element. The minimal Zone includes the reduced amount of trajectories in order to make the network smaller (for actual implementations).

Definition 5.1. A language $L_Z(S)$ generated by the grammar G_Z (**Tables 1, 2**) in a state S of a Complex System is called the ***Language of Zones***.

TABLE 1
A Grammar of Zones G_Z

L	*Q*	**Kernel,** π_k	π_n for all z from X	F_T	F_F
1	Q_1	$\boldsymbol{S}(u, v, w) \to \boldsymbol{A}(u, v, w)$		*two*	ø
2_i	Q_2	$\boldsymbol{A}(u, v, w) \to \boldsymbol{t}(\boldsymbol{h_i^o}(\mathrm{u}), l_o+1)$ $\boldsymbol{A}((0, 0, 0), \boldsymbol{g}(\boldsymbol{h_i^o}(u),w), zero)$	TIME(z)=$\boldsymbol{DIST}$(z,$\boldsymbol{h_i^o}(u)$)	3	ø
3	Q_3	$\boldsymbol{A}(u, v, w) \to \boldsymbol{A}(\boldsymbol{f}(u, v), v, w)$	NEXTTIME(z)= ***init***(u, NEXTTIME(z))	*four*	5
4_j	Q_4	$\boldsymbol{A}(u, v, w) \to \boldsymbol{t}(\boldsymbol{h_j}(u), \mathrm{TIME}(y)))$	NEXTTIME(z)=	3	3

		$\boldsymbol{A}(u, v, \boldsymbol{g}(\boldsymbol{h_j}(u), w))$	$\boldsymbol{ALPHA}$(z, $\boldsymbol{h_j}$(u), TIME(y)–$\boldsymbol{len}$($\boldsymbol{h_j}$(u))+1)		
5	$\boldsymbol{Q_5}$	$\boldsymbol{A}(u, v, w) \to \boldsymbol{A}((0, 0, 0), w, zero)$	TIME(z)=NEXTTIME(z)	3	6
6	$\boldsymbol{Q_6}$	$\boldsymbol{A}(u, v, w) \to \boldsymbol{e}$		ø	ø

$\boldsymbol{V_T}$ ={$\boldsymbol{t}$} is the alphabet of terminal symbols, $\boldsymbol{e}$ is an empty string,
$\boldsymbol{V_N}$ ={$\boldsymbol{S}$, $\boldsymbol{A}$} is the alphabet of nonterminal symbols,
$\boldsymbol{V_{PR}}$ =*Truth*U*Pred*U*Con*U*Var*U*Func*U{symbols of logical operations} is the alphabet of the first order predicate calculus ***PR***,

Truth ={*T*, *F*}

Pred ={$\boldsymbol{Q_1}$,$\boldsymbol{Q_2}$,$\boldsymbol{Q_3}$,$\boldsymbol{Q_4}$,$\boldsymbol{Q_5}$,$\boldsymbol{Q_6}$} are the following WFF of the predicate calculus ***PR***, the conditions of applicability of productions.

Assume that χ(p, q) is a character function of the set (P_1 x P_1)   (P_2 x P_2), where P_1 P_2 = P. It means that χ(p, q) = 1, if both p and q belong P_1 or P_2, otherwise χ(p, q) = 0.

$\boldsymbol{Q_1}(u)$ = (ON(p_o)=x) ^ (MAP_{x,p_o}(y) ≤ *l* ≤ l_o) ^ (q((ON(q)=y) ^ (χ(p_o, q)=0)))
$\boldsymbol{Q_2}(u)$ = *T*
$\boldsymbol{Q_3}(u)$ = (x≠n) ^ (y≠n)
$\boldsymbol{Q_4}(u)$ = (p_1((ON(p_1)=x) ^ (*l* > 0) ^ (((χ(p_o, p_1)=1) ^ (MAP_{x,p_1}(y)=1))
((χ(p_o, p_1)=0) ^ (MAP_{x,p_1}(y) ≤ *l*))))
$\boldsymbol{Q_5}(w)$ = (*w*≠*zero*)
$\boldsymbol{Q_6}$=*T*

Var = {x, y, *l*, τ, θ, v_1, v_2, ..., v_n, w_1, w_2, ..., w_n} are variables;
for the sake of brevity:
u=(x, y, *l*), *v* =(v_1, v_2, ..., v_n),
w=(w_1, w_2, ..., w_n), *zero* = (0, 0, ..., 0)

Con = {x_o,y_o,l_o,p_o} are constants;

Func=*Fcon* U*Fvar* are functional symbols;

Fcon={$\boldsymbol{f_x}$,$\boldsymbol{f_y}$,$\boldsymbol{f_l}$, $\boldsymbol{g_1}$,$\boldsymbol{g_2}$,...,$\boldsymbol{g_n}$, $\boldsymbol{h_1}$,$\boldsymbol{h_2}$,...,$\boldsymbol{h_M}$,$\boldsymbol{h_1}^{\mathbf{o}}$,$\boldsymbol{h_2}^{\mathbf{o}}$,...,$\boldsymbol{h_M}^{\mathbf{o}}$,***DIST***, ***init***, ***ALPHA***, ***len***}, $\boldsymbol{f}$ = ($\boldsymbol{f_x}$,$\boldsymbol{f_y}$,$\boldsymbol{f_l}$), $\boldsymbol{g}$ = ($\boldsymbol{g_{x1}}$,$\boldsymbol{g_{x2}}$,...,$\boldsymbol{g_{xn}}$),
M=|$\mathbf{L_t}^{\boldsymbol{l}\mathbf{o}}$(S)| is the number of trajectories $\mathbf{L_t}^{\boldsymbol{l}\mathbf{o}}$(S);
functions are defined in Table 2.

Fvar={x_o,y_o,l_o,p_o, TIME, NEXTTIME (are defined in Table 2)}

E =$\mathbf{Z_+}$U X U P U $\mathbf{L_t}^{\boldsymbol{l}\mathbf{o}}$(S) is the subject domain;

Parm: ***S*** –>*Var*, ***A*** –>{*u*, *v*, *w*}, ***t*** –>{p,τ, θ}, is such a mapping that matches each symbol of the alphabet V_TUV_N a set of formal parameters;

L= {1, 3, 5, 6} U *two* U *four*, *two*={2_1,2_2,...,2_M}, *four*={4_1,4_2,...,4_M}. *L* is a finite set called the set of labels; labels of different productions are different;

$\boldsymbol{F_T}$ is a subset of labels of the productions permitted on the next step of derivation if *Q*=*T*; it is called a permissible set;

$\boldsymbol{F_F}$ is analogous to $\boldsymbol{F_T}$ but permitted in case of *Q*=*F*.

At the beginning of derivation:

u=(x_o, y_o, l_o), *w* = *zero*, *v* = *zero*, x_o X, y_o X, l_o $\mathbf{Z_+}$,p_o P,
TIME(z) = 2n, NEXTTIME(z) = 2n for all z from X.

TABLE 2

Definition of functions of the Grammar of Zones G_Z

D(***init***) = X x X x $\mathbf{Z}_+$ x $\mathbf{Z}_+$

$$\boldsymbol{init}(u,\mathrm{r}) = \begin{cases} 2\mathrm{n}, & \text{if } u = (0,0,0), \\ \mathrm{r}, & \text{if } u \neq (0,0,0). \end{cases}$$

D(***len***)= P x $\mathbf{L_t}^{\boldsymbol{l}\mathbf{o}}$(S)

len(p, t) = m, t =$\boldsymbol{a}(z_o)\boldsymbol{a}(z_1)\ldots\boldsymbol{a}(z_m)$

D(***f***) = (X x X x $\mathbf{Z}_+$ U{0, 0, 0}) x $\mathbf{Z}_+^{\,n}$

$$\boldsymbol{f}(u, v) = \begin{cases} (\mathrm{x}+1,\mathrm{y},l), & \text{if} \quad (\mathrm{x} \neq \mathrm{n}) \quad (l > 0), \\ (1, \mathrm{y}+1, \mathrm{TIME}(\mathrm{y}+1) * v_{\mathrm{y}+1}), & \text{if} \quad (\mathrm{x} = \mathrm{n}) \quad ((l \leq 0) \quad (\mathrm{y} \neq \mathrm{n})), \\ (0,0,0), & \text{if} \quad (\mathrm{x} = \mathrm{n}) \quad (\mathrm{y} = \mathrm{n}). \end{cases}$$

D(***DIST***) = X x P x $\mathbf{L_t}^{\boldsymbol{l}\mathbf{o}}$(S). Let $t_o \quad \mathbf{L_t}^{\boldsymbol{l}\mathbf{o}}$(S), t_o =$\boldsymbol{a}(z_o)\boldsymbol{a}(z_1)\ldots\boldsymbol{a}(z_m)$, $t_o \quad t_{p_o}(z_o, z_m, m)$;

```
If
   for some k (1 ≤ k ≤ m)   x = z_k,
   then DIST(x, p_o, t_o) = k+1
   else DIST(x, p_o, t_o) = 2n
```

D(***ALPHA***) = X x P x $\mathbf{L_t}^{\boldsymbol{l}\mathbf{o}}$(S) x $\mathbf{Z}_+$

$$\boldsymbol{ALPHA}(\mathrm{x}, \mathrm{p_o}, \mathrm{t_o}, \mathrm{k}) = \begin{cases} \boldsymbol{max}(NEXTTIME(\mathrm{x}), \mathrm{k}), & \text{if } (\boldsymbol{DIST}(\mathrm{x},\mathrm{p_o},\mathrm{t_o}) \neq 2\mathrm{n}) \quad (NEXTTIME(\mathrm{x}) \neq 2\mathrm{n}) \\ NEXTTIME(\mathrm{x}), & \text{if } \boldsymbol{DIST}(\mathrm{x},\mathrm{p_o},\mathrm{t_o}) = 2\mathrm{n}), \\ \mathrm{k}, & \text{if } \boldsymbol{DIST}(\mathrm{x},\mathrm{p_o},\mathrm{t_o}) \neq 2\mathrm{n}). \end{cases}$$

D($\boldsymbol{g}_\mathbf{r}$) = P x $\mathbf{L_t}^{\boldsymbol{l}\mathbf{o}}$(S) x $\mathbf{Z}_+^{\,n}$, r   X.

$$\boldsymbol{g}_\mathbf{r}(\mathrm{p_o},\mathrm{t_o}, w) = \begin{cases} 1, & \text{if } \boldsymbol{DIST}(\mathrm{r},\mathrm{p_o},\mathrm{t_o}) < 2n, \\ w_\mathrm{r}, & \text{if } \boldsymbol{DIST}(\mathrm{r}, \mathrm{p_o},\mathrm{t_o}) = 2\mathrm{n}. \end{cases}$$

D ($\boldsymbol{h}_i^{\,o}$) = X x X x $\mathbf{Z}_+$; Denote $\mathrm{TRACKS}_{p_o} = \{p_o\} \times (\underset{1 \leq k \leq l}{\quad} \mathrm{L}[\, \mathbf{G_t}^{(1)}(x, y, k, p_o)]$

If

TRACKS$_{po}$ = e

then $\boldsymbol{h}_{\boldsymbol{i}}^{\boldsymbol{o}}(u) = e$

else $\mathrm{TRACKS}_{p_o} = \{(p_o,t_1),(p_o,t_2),\ldots,(p_o,t_b)\}, (b \leq M)$, $\quad \boldsymbol{h}_{\boldsymbol{i}}^{\boldsymbol{o}}(u) = \begin{cases} (p_o, t_i), & \text{if } i \leq b, \\ (p_o, t_b), & \text{if } i > b. \end{cases}$

D($\boldsymbol{h}_i$) = X x X x $\mathbf{Z}_+$; Denote TRACKS = $\underset{\mathrm{ON(p)=x}}{\quad}$ TRACKS$_p$, where TRACKS$_p$ is the same as for $\boldsymbol{h}_i^{\,o}$

If

TRACKS = *e*

then $\boldsymbol{h}_i(u) = e$

else TRACKS = $\{(p_1,t_1),(p_1,t_2),\ldots,(p_m,t_m)\}$, (m ≤ M), $\quad \boldsymbol{h}_i(u) = \begin{cases} (p_i, t_i), & \text{if } i \leq m, \\ (p_m, t_m), & \text{if } i > m. \end{cases}$

One of the Zones to be generated for the robot control model shown in Fig. 1 is as follows:

$\boldsymbol{t}(\text{BOMBER},t_B,4)\boldsymbol{t}(\text{FIGHTER}, t_F,5)\boldsymbol{t}(\text{MISSILE}, t_M,5)\boldsymbol{t}(\text{MISSILE}, t_M^1, 3)\boldsymbol{t}(\text{FIGHTER}, t_F^1, 2)$, where

$$t_B=\boldsymbol{a}(h5)\boldsymbol{a}(h4)\boldsymbol{a}(h3)\boldsymbol{a}(h2)\boldsymbol{a}(h1),\quad t_F=\boldsymbol{a}(f6)\boldsymbol{a}(e5)\boldsymbol{a}(e4)\boldsymbol{a}(f3)\boldsymbol{a}(g2)\boldsymbol{a}(h1),$$
$$t_M=\boldsymbol{a}(d7)\boldsymbol{a}(b5)\boldsymbol{a}(f1)\boldsymbol{a}(g2)\boldsymbol{a}(h1),\ t_M^1=\boldsymbol{a}(d7)\boldsymbol{a}(b5)\boldsymbol{a}(f1)\boldsymbol{a}(h3),\ t_F^1=\boldsymbol{a}(f6)\boldsymbol{a}(g5)\boldsymbol{a}(h4)$$

The generation of this Zone (Fig. 3) is considered in detail in Section 9.

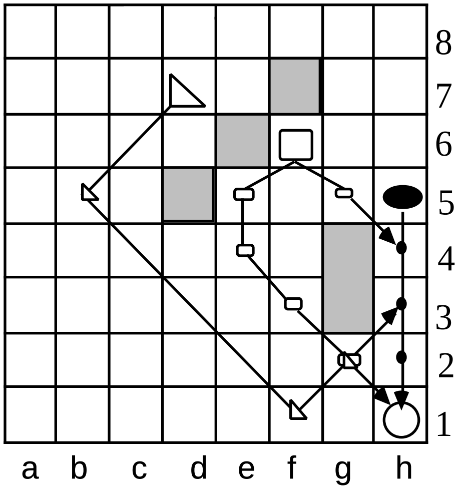

Fig. 3. *An interpretation of the trajectory network language for the robot control model.*

6. Translations of Languages

Network languages allow us to describe the "statics", i.e., the states of the System. We proceed with the description of the "dynamics" of the System, i.e., the transitions from one state to another. The transitions describe the change of the descriptions of states as the change of sets of WFF. After each transition a new hierarchy of languages should be generated. Of course, it is an inefficient procedure. To improve an efficiency of applications in a process of the search it is important to describe the change of the hierarchy of languages. A study of this change should help us in modifying the hierarchy instead of regenerating it in each state. The change may be described as a hierarchy of mappings – translations of languages. Each hierarchy's language should be transformed by the specific mapping called a translation.

Definition 6.1. A ***translation relation Tr*** from a language L_1 to a language L_2 is the binary relation *Tr* from L_1 into L_2 for which L_1 is the domain and L_2 is the range. If $Tr(\boldsymbol{a}, \boldsymbol{b})$ holds, then the string $\boldsymbol{b}$ is called the output for the input string $\boldsymbol{a}$.

In general, for the translation relation for each input string there may be several output strings. However, in our case we can consider the translation relation as a mapping, i.e., "for each input – no more than one output."

Definition 6.2. Let the Complex System move from the state S_1 to the state S_2 by applying the operator T_o= TRANSITION(p,x_o,y_o). A ***Translation of Languages of Trajectories*** is a mapping

$$\prod\nolimits_{T_o}: L_t^H(S_1) \text{–>} L_t^H(S_2),$$

of such a sort that trajectories of the form $\boldsymbol{a}(x)\boldsymbol{a}(y)\dots\boldsymbol{a}(z)$ are transformed as follows:

- are **"shortened"** by the exclusion of the first symbol $\boldsymbol{a}(x)$, if the transition T_o carries out along such a trajectory: $x = x_o$ & $y = y_o$. (If y = z, i.e., y is the ending point, the trajectory is transformed into the empty trajectory *e*.)
- are transformed into the **empty trajectory *e***, if **element p moves away** from such a trajectory: $x = x_o$ & $y \neq x_1$,

or this element is withdrawn: $x = x_1$ and WFF ON(q) = x_1 comes into the Remove list of the transition T_o.

– are transformed **into itself** in all the other cases.

Obviously, mapping $\prod_{M_o}$ is not a mapping "onto" and has a non-empty kernel, i.e., a nonempty co-image of the empty trajectory ***e***.

To proceed with the description of the hierarchy change we should define a translation of the hierarchy's next level, the Trajectory Network Languages. Let us consider the definition of the translation for the Language of Zones.

Definition 6.3. A ***Translation of Languages of Zones*** is a mapping of the following form:

$$\pi_{T_o}: L_Z(S_1) \text{ –> } L_Z(S_2),$$

where Zone Z_1 is translated into Zone Z_2, i.e., $\pi_{T_o}(Z_1) = Z_2$ if and only if the main trajectory $t_o{}^1$ of Zone Z_1 is translated into the main trajectory $t_o{}^2$ of the Zone Z_2 by the corresponding trajectory translation, $\prod_{T_o}(t_o{}^1) = t_o{}^2$.

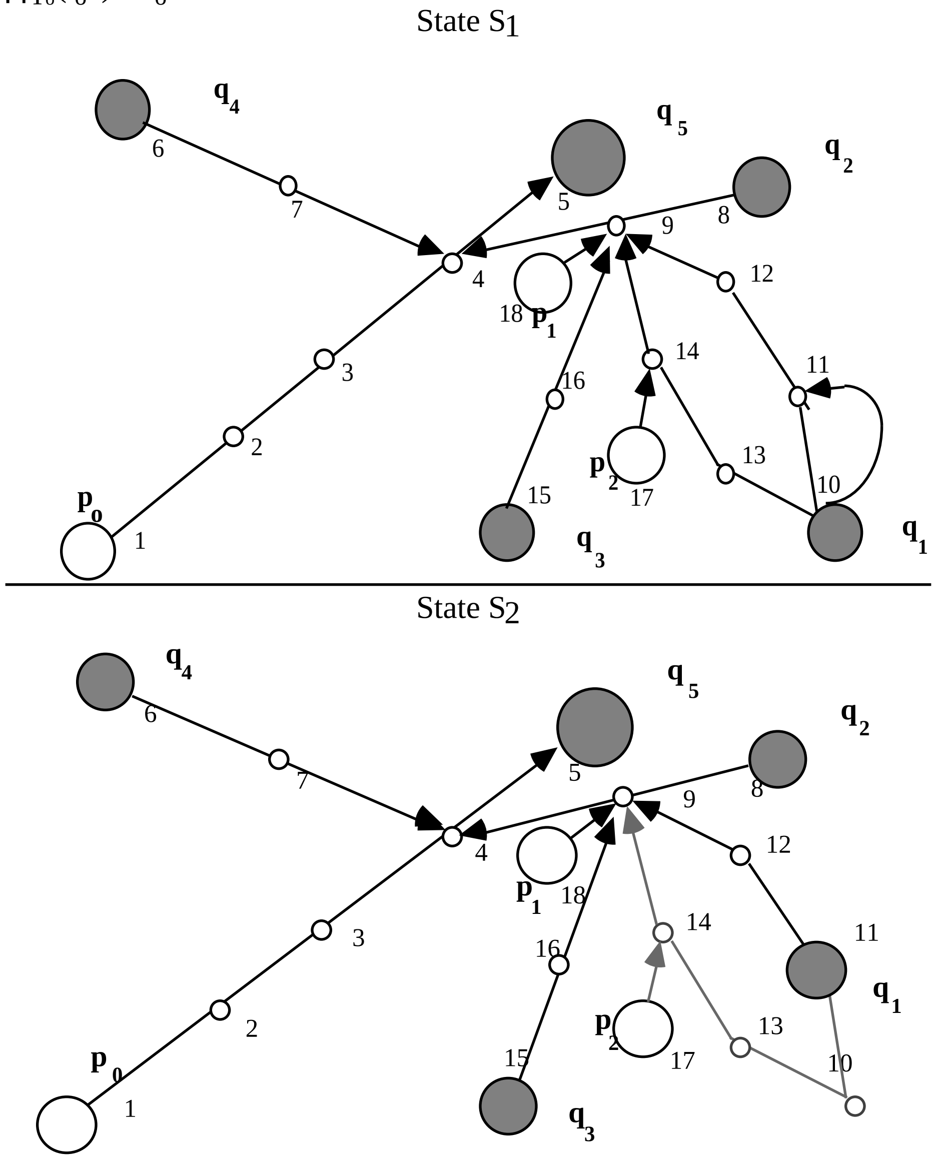

Fig. 4. *A translation of Languages of Zones.*

For example, in Fig. 2 after transition TRANSITION(p_2, 10, 12) the trajectory ***a***(10)***a***(12) is translated into the trajectory *e* and all the remaining trajectories are translated into itself. After transition TRANSITION(p_0, 1, 2) the Zone depicted in Fig. 2 is translated into a new Zone with the main trajectory ***a***(2)***a***(3)***a***(4)***a***(5), because this transition causes such a translation of trajectories that trajectory ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5) is translated into the trajectory ***a***(2)***a***(3)***a***(4)***a***(5).

Let us take a look at the different example (Fig. 4). The Language of Zones in State 1 consists of two Zones with the same main trajectory ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5). The difference between these Zones is in the trajectories of element q_1. Trajectory ***a***(10)***a***(11)***a***(12)***a***(9) is included into Zone 1 while ***a***(10)***a***(13)***a***(14)***a***(9) together with ***a***(17)***a***(14) are included into Zone 2. After TRANSITION(q_1,10, 11) the Language of Zones in State S_1 is translated into the new Language of Zones in State S_2. Trajectory ***a***(10)***a***(11)***a***(12)***a***(9) is shortened; it is translated into ***a***(11)***a***(12)***a***(9). This is the only difference between the Zone 1 and its translation. The change for Zone 2 is more essential. It “looses” trajectories ***a***(10)***a***(13)***a***(14)***a***(9) and ***a***(17)***a***(14) completely. (The trajectories and their links that are not included in the Language of Zones in a State S_2 are shown by dotted lines in Fig. 4.)

It is very important to show the difference between the Zone and its translation in *general case*, i.e., to describe which trajectories of the old Zone remain unchanged in the new one, which trajectories are shortened, as ***a***(1)***a***(2)***a***(3)***a***(4)***a***(5) in Fig. 2 or ***a***(10)***a***(11)***a***(12)***a***(9) in Fig. 4, which are not included, i.e., are translated into the empty trajectory *e*, and finally, what are the new trajectories of the new Zone. This knowledge for every transition would give us a description of the change of the Language of Zones.

7. Approaching a Solution of the Problem of Change

A description of the change for the Language of Trajectories is trivial and explicitly follows from the definition of translations of these languages. For the Translation of Languages of Zones it is a problem. The study of properties of translations should allow us to give a formal, constructive solution of the problem relative to the well-known Frame Problem [5-8] for this specific system. This is the problem of effective description of boundaries between the actual and outdated information about the system. This information is updated in the process of search for an optimal operation. An efficient and constructive description of the hierarchy adjustment is very important for the design of efficient applications in different fields.

To study this language formally we need some preliminary definitions.

Definition 7.1. An ***alphabet*** A(Z) ***of the string*** Z of the parameter language L is the set symbols of this language with given parameter values, where each of these symbols with parameters is included at least once in a string Z, and *e* (the empty symbol).

Definition 7.2. A ***trajectory alphabet*** TA(Z) of the Zone Z is the set of trajectories from $L_t^H(S)$ that correspond to the actual parameter values of the alphabet A(Z).

When Complex System moves from one state to another, the corresponding Hierarchy of Languages is changed by translation. Each language of the hierarchy for one state is translated into the similar language for another state. A translation of the given language causes a mapping of the alphabets of strings.

Definition 7.3. Let $\prod_{Mo}$ be a translation of languages of Zones, with $\pi_{Mo}(Z_1) = Z_2$. ***Mapping of alphabets*** $\pi_{\mathbf{o}}$ of Zones Z_1 and Z_2 is the mapping $\pi_{\mathbf{o}}$: A(Z_1) –> A(Z_2), which is constructed as follows. For all the symbols $\boldsymbol{t}(p, t_j, \tau_1)$ from A(Z_1)

$$\pi_{\mathbf{o}}(\boldsymbol{t}(p, t_j, \tau_1)) = \boldsymbol{t}(p, \textstyle\prod_{Mo}(t_j), \tau_2),$$

if there exists τ_2 $\mathbf{Z}_+$, $\tau_2 > 0$, such that $\boldsymbol{t}(p, \prod_{Mo}(t_j), \tau_2)$ A(Z_2) – {*e*};

$$\pi_{\mathbf{o}}(\boldsymbol{t}(p, t_{j,} \tau_1)) = e$$

in the remaining cases.

Generally speaking, π_o is a not a function because for each symbol from the domain π_o can yield several different values. For example, it can yield empty and non-empty values. We will introduce constraints for the domain of π_o, which allow to consider π_o as a function. In particular, we are going to constrain the domain of π_o within the so-called *invariant subnet* of Zone (see next).

Definition 7.4. In conditions of Definition 7.3 we denote by

$$Con_{\Pi}(Z_1)=\{\boldsymbol{t}(p_i, t_i, \tau_i) \quad A(Z_1)-\{e\} \mid C^{+}_{\Pi_{M_o}(TA(Z_1))}(\Pi_{Mo}(t_i), \Pi_{M_o}(t_o))=T\}$$

an ***invariant subnet*** of Zone Z_1 with respect to the translation π_{Mo}.

Consider the example of the translation of Zones shown in Fig. 6. After transition M_o = TRANSITION(q_1, 10, 11) trajectories $t_1 = \boldsymbol{t}(10)\boldsymbol{t}(13)\boldsymbol{t}(14)\boldsymbol{t}(9)$ and $t_2 = \boldsymbol{t}(17)\boldsymbol{t}(14)$ “loose” the C^+ connection with the main trajectory t_o=$\boldsymbol{t}(1)\boldsymbol{t}(2)\boldsymbol{t}(3)\boldsymbol{t}(4)\boldsymbol{t}(5)$. Indeed, $\Pi_{Mo}(t_1) = e$ and thus can not be connected with anything while $\Pi_{Mo}(t_2)$ can be connected with $\Pi_{Mo}(t_o)$ only through $\Pi_{Mo}(t_1)$ which is empty. Consequently, the symbols $\boldsymbol{t}(q_1, t_1, \tau_1)$ and $\boldsymbol{t}(p_2, t_2, \tau_2)$ from $A(Z_1)$ are not included into the invariant subnet of Zone Z_1 shown in Fig. 6. Thus $Con_{\Pi}(Z_1)$ should be considered as a collection of symbols of the alphabet of Zone which trajectories being translated do not loose the connection with the translation of the main trajectory of this Zone.

Now we are going to introduce a function called ***timer***$_\pi$. For every trajectory from the invariant subnet of Zone Z_1 this function should yield a correct value of the “time” (parameter τ) allocated to the image of this trajectory in the translation of Z_1. By comparing this value with the length of this image we should be able to conclude whether image of this trajectory is included into the translation of the Zone or not. Negative answer to this question means that the length of the trajectory image exceeds the time allocated to the motion along it.

Definition 7.5. Let $\pi_{Mo}(Z_1) = Z_2$ be a translation, with $Z_1 = \boldsymbol{t}(p_o, t_o, \tau_o)\boldsymbol{t}(p_1, t_1, \tau_1)\ldots\boldsymbol{t}(p_r, t_r, \tau_r)$, $Z_1 \quad L_Z(S_1)$, $Z_2 \quad L_Z(S_2)$. A mapping

$$\boldsymbol{timer}_\pi\colon Con_{\Pi}(Z_1) \to \mathbf{Z},$$

where $\mathbf{Z}$ is the set of all integer numbers, is constructed as follows. We consider three cases:

(1) If $\Pi_{M_o}(t_o) = t_o'$, i.e., the main trajectory of Zone Z_1 is shortened, that is transformed into a substring with an excluded first symbol (according to Definition 10.2),
then for all symbols $\boldsymbol{t}(p_c, t_c, \tau) \quad Con_{\Pi}(Z_1)$

$$\boldsymbol{timer}_\pi(\boldsymbol{t}(p_c, t_c, \tau)) = \tau - 1.$$

(2) If $\Pi_{M_o}(t_k) = t_k'$, i.e., some other trajectory t_k of Zone Z_1 is shortened ($k \neq 0$),
then we define ***timer***$_\pi$ recursively.

(a) $\boldsymbol{timer}_\pi(\boldsymbol{t}(p_o, t_o, \tau_o)) = \tau_o$,
$\boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i)) = \tau_i$ (if $C_{TA(Z_1)}(t_i, t_o)$ =T)

(b) Let $\boldsymbol{t}(p_c, t_c, \tau) \quad Con_{\Pi}(Z_1)$,
denote $CA(t_c) = \{t_i \quad Con_{\Pi}(Z_1) \mid C(t_c, t_i) = T\}$,
then $\boldsymbol{timer}_\pi(\boldsymbol{t}(p_c, t_c, \tau)) = \max\limits_{t_i \quad CA(t_c)} \{TNEW(t_i)\}$, where

$$\mathrm{TNEW}(t_i) = \begin{cases} \boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i)) - \boldsymbol{len}(p_i, t_i) + 1, & \text{if } t_i \neq t_k, \\ (\boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i)) + 1) - \boldsymbol{len}(p_i, t_i) + 1, & \text{if } t_i = t_k, \end{cases}$$

(***len***(p_i, t_i) is the length of t_i).

(3) If $\prod_{M_o}(t_m) = t_m$ for all t_m TA(Z_1), then ***timer***$_\pi$(***t***(p_c, t_c, τ)) = τ.

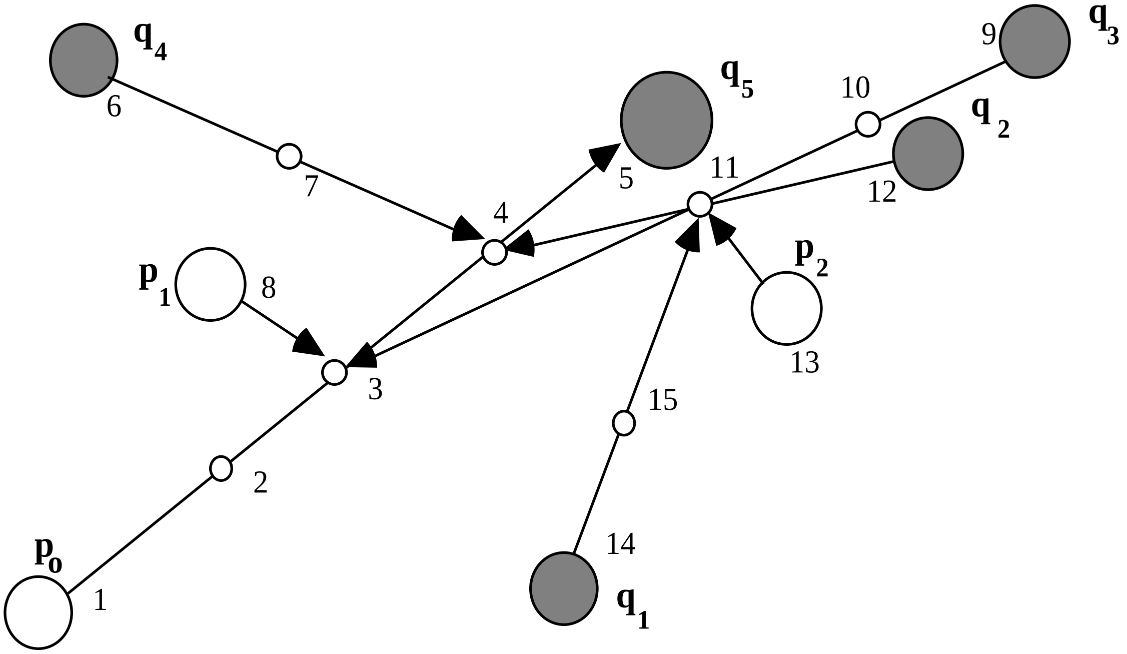

Fig. 5. *Interpretation of function* ***timer***$_\pi$.

Consider example of Zone shown in Fig. 5. In case of M_o = TRANSITION(p_o, 1, 2) we have case (1) of Definition 7.5. It means that function ***timer***$_\pi$ for all the symbols of A(Z_1) yields the value of $\tau - 1$, where τ is the value the third parameter of each symbol. For example, ***timer***$_\pi$(***t***(q_3, t_{q_3}, τ)) = $\tau - 1$, where t_{q_3} = ***t***(9)***t***(10)***t***(11)***t***(3), $\tau = 3$. It means that after TRANSITION(p_o, 1, 2) time allocated to the motion along trajectory t_{q_3} is less than the length of this trajectory (2 < 3) and, thus, trajectory t_{q_3} should not be included into the translation $\pi_{M_o}(Z_1)$ of Zone Z_1 (see Theorem 8.1). At the same time for the 2nd negation trajectory t_{q_1} = ***t***(14)***t***(15)***t***(11) connected with t_{q_3} ***timer***$_\pi$(***t***(q_1, t_{q_1}, 3)) = 2. It means that its length does not exceed the time allocated for the motion and, consequently, t_{q_1} should be included into Z_2. In spite of loosing the C^+ connection with t_o through t_{q_3} (which is not included), trajectory t_{q_1} keeps the C^+ connection with t_o through t_{q_2}.

For the same Zone (Fig. 5) consider different transition M_o = TRANSITION(q_3, 9, 10) assuming that it is an opposing side turn. According to Definition 7.5 (2, a) for the main trajectory ***timer***$_\pi$(***t***(p_o, t_o, 4)) = 4, for the 1st negation trajectories ***timer***$_\pi$(***t***(p_1, t_{p_1}, 3)) = 3, ***timer***$_\pi$(***t***(q_2, t_{q_2}, 4)) = 4, ***timer***$_\pi$(***t***(q_3, t_{q_3}, 3)) = 3, ***timer***$_\pi$(***t***(q_4, t_{q_4}, 4)) = 4. Obviously, the lengths of all the 1st negation trajectories here do not exceed values of ***timer***$_\pi$ for them, i.e., after transition M_o elements q_2, q_3, q_4 still have enough time for interception of element p_o. This means that these trajectories should be included into the translation of the Zone. Now we have to compute the value of ***timer***$_\pi$ for the 2nd negation trajectories t_{q_1} and t_{p_2} which corresponds to case (2,b) of Definition 7.5. Obviously, for both trajectories CA(t_{q_1}) = CA(t_{p_2}) = {t_{q_2}, t_{q_3}}. Then

$$\boldsymbol{timer}_\pi(\boldsymbol{t}(q_1, t_{q_1}, 3)) = \max\{\mathrm{TNEW}(t_{q_2}), \mathrm{TNEW}(t_{q_3})\},$$

$$\text{where} \quad \mathrm{TNEW}(t_{q_2}) = \boldsymbol{timer}_\pi(\boldsymbol{t}(q_2, t_{q_2}, 4)) - 2 + 1 = 4 - 2 + 1 = 3,$$

$$\mathrm{TNEW}(t_{q_3}) = (\boldsymbol{timer}_\pi(\boldsymbol{t}(q_3, t_{q_3}, 3)) + 1) - 3 + 1 = (3 + 1) - 3 + 1 = 2.$$

Consequently, $\boldsymbol{timer}_\pi(\boldsymbol{t}(q_1, t_{q_1}, 3)) = 3$. Thus while the length of t_{q_1} does not exceed the value of $\boldsymbol{timer}_\pi$ ($2 < 3$) it should be included into the translation.

Case (3) of Definition 7.5 takes place when transition is executed along the trajectory of some Zone Z' different from Z_1. It means that time allocation in Zone Z_1 should not be changed.

The following constructive definition actually gives us an algorithm for the computation of function π_o (see Theorem 8.1).

Definition 7.6. On conditions of Definition 7.3 the following set

$$\prod(\mathrm{Con}_\prod(Z_1)) =\{\boldsymbol{t}(p_i, \prod\nolimits_{Mo}(t_i), \boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i))) \mid \boldsymbol{t}(p_i, t_i, \tau_i) \quad \mathrm{Con}_\prod(Z_1)\}$$

is called a ***net image*** of Zone Z_1 with respect to translation π.

8. Theorem about Translations

THEOREM 8.1. *Let, for a translation* $\pi_{Mo}(Z_1) = Z_2$, *where for all symbols* $\boldsymbol{t}(p_1, t_1, \tau_1) \quad A(Z_2) - \prod(\mathrm{Con}_\prod(Z_1))$ *and* $\boldsymbol{t}(p_2, t_2, \tau_2) \quad \prod(\mathrm{Con}_\prod(Z_1))$, *therelation* $C_{TA(Z_2)}(t_2, t_1)$ *does not hold, i.e.,* $C_{TA(Z_2)}(t_2, t_1)=F$. *Then for every symbol* $\boldsymbol{t}(p, t_i, \tau) \quad \mathrm{Con}_\prod(Z_1)$, (*where* $t_i \quad t_p(x, y, l)$, $l > 1$)

$$\pi_o(\boldsymbol{t}(p, t_i, \tau)) = \boldsymbol{t}(p, \prod\nolimits_{Mo}(t_i), \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_i, \tau)))$$

is a mapping onto $A(Z_2)\cap\prod(\mathrm{Con}_\prod(Z_1))$, *if and only if* $l \le \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_i, \tau))$.

PROOF. Let $Z_1 = \boldsymbol{t}(p_o, t_o, \tau_o)\boldsymbol{t}(p_1, t_1, \tau_1)\ldots\boldsymbol{t}(p_{r_1}, t_{r_1}, \tau_{r_1})$. We consider three cases according to Definition 7.5.

1. Let $\prod_{Mo}(t_o) = t_o'$, i.e., the main trajectory of Zone Z_1 is shortened by exclusion of the first symbol, then we shall prove that for any symbol $\boldsymbol{t}(p_i, t_i, \tau_i) \quad \mathrm{Con}_\prod(Z_1)$

$$\pi_o(\boldsymbol{t}(p_i, t_i, \tau_i)) = \boldsymbol{t}(p_i, \prod\nolimits_{Mo}(t_i), \tau_i - 1)$$

if and only if the length of trajectory $t_i \quad l \le \tau_i - 1$.

We denote $t_i' = \prod_{Mo}(t_i)$. Obviously, $t_i' = t_i$ for all $t_i \neq t_o$. Consider the Grammar of Zones $\mathbf{G_Z}$ (Tables 1, 2). Next we are going to generate both Zones Z_1 and Z_2 simultaneously and independently of each other. In order to distinguish each of these derivations and compare the results, we will use Z_1 and Z_2 as indices of expressions derived in these grammars, where necessary.

We have to prove two statements. The direct statement is as follows: for any symbol $\boldsymbol{t}(p_i, t_i, \tau_i) \quad \mathrm{Con}_\prod(Z_1)$ such that $l \le \tau_i - 1$ symbol $\boldsymbol{t}(p_i, \prod_{Mo}(t_i), \tau_i - 1)$ belongs to $A(Z_2)$. The reverse statement requires that for every symbol $\boldsymbol{t}(p_i, t_i', \tau_i')$ from $A(Z_2)\cap\prod(\mathrm{Con}_\prod(Z_1))$ there exist symbol $\boldsymbol{t}(p_i, t_i, \tau_i'+ 1) \quad \mathrm{Con}_\prod(Z_1)$ such that $l \le \tau_i'$ and $\prod_{Mo}(t_i) = t_i'$.

Let us prove the direct statement. We are going to conduct this proof *by induction.*

The *basis* of the induction is as follows. Consider symbols $\boldsymbol{t}(p_i, t_i, \tau_i) \quad \mathrm{Con}_\prod(Z_1)$ for which $C_{TA(Z_1)}(t_i, t_o) = T$, i.e., the 1-st negation trajectories. Obviously, in this case $C_{TA(Z_2)}(t_i', t_o') = T$, where $t_i'= \prod_{Mo}(t_i) = t_i$, except for the trajectories with the end coincided with the beginning of the main trajectory. The last ones, obviously, are not included into $\mathrm{Con}_{\prod}(Z_1)$. Assume that

$$\tau_i - 1 = \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i)) \geq \boldsymbol{len}(p_i, t_i). \qquad (8.1)$$

Let us prove that symbol $\boldsymbol{t}(p_i, t_i, \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i)))$ will be generated by the grammar G_Z in a state S_2 and attached to Zone Z_2.

Indeed, the maximum length of trajectories t_i to be the parameter value of the attaching symbol is determined by the value of function $\boldsymbol{f}(u, v)$ in production 3 (Table 1). This length is determined by the value of the third parameter of function $\boldsymbol{f}(u, v)$ which in this case is as follows (Table 2):

$$\boldsymbol{f}(u, v) = (1, y + 1, \mathrm{TIME}(y + 1) * v_{y+1}).$$

Points y + 1 are the ending points of prospective 1-st negation trajectories and, thus, belong to $P(t_o) - \{y_o\}$. The values of TIME were computed by application of production 2_i (section $\pi_{\boldsymbol{n}}$).

From the expression for the kernel of the production 2_i, it follows that $\tau_o = l_o + 1$ for the terminal symbol $\boldsymbol{t}(p_o, t_o, \tau_o) = \boldsymbol{t}(\boldsymbol{h}_i{}^o(x_o, y_o, l_o), \tau_o)$. In such case $\pi_o(\boldsymbol{t}(p_o, t_o, \tau_o)) = \boldsymbol{t}(\boldsymbol{h}_i{}^o(x_1, y_o, l_o - 1), \tau_o')$, with $\tau_o' = l_o + 1$. Consider the following main trajectory of the Zone $t_o = \boldsymbol{a}(y_o)\boldsymbol{a}(y_1)\ldots\boldsymbol{a}(y_l)$ and its image $t_o' = \boldsymbol{a}(x_o)\boldsymbol{a}(x_1)\ldots\boldsymbol{a}(x_{l-1})$. Let us take into account that $\boldsymbol{P}(t_o') = \{x \quad X \mid \boldsymbol{DIST}(x, p_o, t_o') < 2n\}$ and mapping $\prod_{M_o}$ causes the following one-to-one correspondence between $P(t_o) - \{y_o\}$ and $P(t_o')$:

$$x_i = y_{i+1}$$

for i = 1, 2,..., $l - 1$. Then from the section $\pi_{\boldsymbol{n}}$ of production 2_i it follows that

$$\mathrm{TIME}_{Z_2}(x_i) = \boldsymbol{DIST}(x_i, p_o, t_o') = i + 1 = [(i + 1) + 1] - 1 = \boldsymbol{DIST}(y_{i+1}, p_o, t_o) = \mathrm{TIME}_{Z_1}(y_{i+1}) - 1$$

Consequently, for each point x $P(t_o) - \{y_o\}$

$$\mathrm{TIME}_{Z_2}(x) = \mathrm{TIME}_{Z_1}(x) - 1. \qquad (8.2)$$

At the same time TIME(x) determines the value of parameter τ_i of each symbol $\boldsymbol{t}(p_i, t_i, \tau_i)$; this follows from production 4_j. Consequently, $\mathrm{TIME}_{Z_1}(x) = \tau_i$, and taking (8.2) into account, we obtain

$$\mathrm{TIME}_{Z_2}(x) = \mathrm{TIME}_{Z_1}(x) - 1 = \tau_i - 1 = \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i)) \geq \boldsymbol{len}(p_i, t_i). \qquad (8.3)$$

Hence, trajectories t_i of the length $\boldsymbol{len}(p_i, t_i)$ will be generated by G_Z and corresponding symbols $\boldsymbol{t}(p_i, t_i, \tau')$ will be attached to Zone Z_2. The only question to be answered is the question of the value of parameter τ'. It was shown above that τ' is determined by the value of $\mathrm{TIME}_{Z_2}(x)$ in production 4_j. According to (8.3),

$$\tau_i' = \mathrm{TIME}_{Z_2}(x) = \mathrm{TIME}_{Z_1}(x) - 1 = \tau_i - 1 = \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i)),$$

and our statement about 1-st negation trajectories is proved:

$$\boldsymbol{t}(p_i, t_i, \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i))) \quad A(Z_2).$$

The *basis of induction is proved* by the preceding.

Assume that for all the m-negation trajectories t_m $m < m_o$ and $\boldsymbol{t}(p_m, t_m, \tau_m)$, the statement of Theorem 8.1 (1) is true.

Let t_{m_o} is an arbitrary m_o-negation trajectory, $\boldsymbol{t}(p, t_{m_o}, \tau)$ $\mathrm{Con}_{\prod}(Z_1)$. According to condition of Theorem 8.1 $\pi_o(\boldsymbol{t}(p, t_{m_o}, \tau)) = \boldsymbol{t}(p, \prod_{M_o}(t_{m_o}), \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_{m_o}, \tau)))$. Assume also that

$$\boldsymbol{len}(p, t_{m_o}) \leq \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_{m_o}, \tau)). \qquad (8.4)$$

We are going to prove that symbol $\boldsymbol{t}(p, \prod_{M_o}(t_{m_o}), \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_{m_o}, \tau)))$ will be generated by the grammar G_Z in a state S_2 and attached to Zone Z_2.

From Definition 7.2 and 7.5 it follows that $\prod_{M_o}(t_{m_o}) = t_{m_o}$, $\boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_{m_o}, \tau)) = \tau - 1$. Let us show that $\boldsymbol{t}(p, t_{m_o}, \tau - 1) \quad A(Z2)$.

The maximum length of trajectory t_{m_o} to be included into Z_2 as a parameter value of the attaching symbol, is determined by the value of function $\boldsymbol{f}(u, v)$ in production 3 (Table 1). This length is determined by the value of the third parameter of function $\boldsymbol{f}(u, v)$ which in this case is as follows (Table 2):

$$\boldsymbol{f}(u, v) = (1, y + 1, TIME(y + 1) * v_{y+1}).$$

Points y+1 are the parameter values of the (m_o–1) negation trajectories. Values of TIME(y) are assigned by applying production 5 (section $\pi_{\boldsymbol{n}}$, Table 1). This application happens each time when generation of current negation is completed. Last application of production 5 took place when generation of (m_o–1)st negation trajectories was completed. Thus, values of NEXTTIME were assigned to TIME. Values of NEXTTIME(z) were computed during earlier applications of productions 4_j for attaching symbols with (m_o–1) negation trajectories.

Let $m = m_o - 1$. Consider the generation of symbol $\boldsymbol{t}(p_m, t_m, \tau_2')$ with trajectory $t_m \quad t_{p_m}(x_o, x_e, l_m)$. Thus, applying formula $\pi_{\boldsymbol{n}}$ of production 4_j for $u = (x_o, x_e, l_m)$, we obtain

$$NEXTTIME\ (x_i) = \boldsymbol{ALPHA}(x_i, p_m, t_m, TIME(x_e) - l_m + 1).$$

Consequently, for $y \quad \boldsymbol{P}(t_m)$

$$\begin{aligned} NEXTTIME_{Z_2}(y) = & \qquad (8.5)\\ & \boldsymbol{ALPHA}_{Z_2}(y, \boldsymbol{h_j}(x_o, x_e, l_m), TIME_{Z_2}(x_e) - l_m + 1) = \\ & \max(NEXTTIME^o{}_{Z_2}(y), TIME_{Z_2}(x_e) - l_m + 1) = \\ & \max(NEXTTIME^o{}_{Z_2}(y), \tau_2' - l_m + 1), \end{aligned}$$

where $NEXTTIME^o{}_{Z_i}(y)$ are the values of function NEXTTIME(y) before current application of production 4_j in the derivation of Z_2.

Trajectory t_m is m-negation trajectory with $m < m_o$. On condition of the theorem, for all symbols $\boldsymbol{t}(p_1, t_1, \tau_1) \quad A(Z2) - \prod(Con_{\prod}(Z_1))$ and $\boldsymbol{t}(p_2, t_2, \tau_2) \quad \prod(Con_{\prod}(Z_1)) \quad C_{TA(Z_2)}(t_2, t_1) =$ F. This means that for any trajectory $t \quad TA(Z_2)$ such that $C^k{}_{TA(Z_2)}(t_{m_o}, t), \quad \boldsymbol{t}(p, t, \tau) \quad \prod(Con_{\prod}(Z_1))$. According to the assumption of induction and Definition 7.5 (1)

$$\tau_2' - l_m + 1 = \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_m, t_m, \tau_2)) - l_m + 1 = (\tau_2 - 1) - l_m + 1. \qquad (8.6)$$

The value of $NEXTTIME^o{}_{Z_2}(y)$ was computed by successive application of production 4_j for attaching symbols with trajectories t_i containing y among parameter values, i.e., $y \quad P(t_i)$.

An example of such situation is shown in Fig. 5. Trajectory t_{q_1} can be considered as trajectory t_{m_o}, i.e., 2nd negation trajectory. Trajectory t_{q_2} (as t_m) is the 1st negation trajectory such that $C(t_{q_1}, t_{q_2})$ through point y = 11. The other trajectory "crossing" the same point is t_{q_3}. Assume that t_{q_3} was computed and attached to the Zone earlier than t_{q_2}. Consequently, t_{q_3} is one of the trajectories t_i, which we are going to consider.

Thus, taking into account that such trajectories are of r negation, $r < m_o$, we conclude that assumption of induction is true for them. Considering contribution to computation of $NEXTTIME^o{}_{Z_2}(y)$ from each trajectory t_i (during application of production 4_j), we obtain:

$$\begin{aligned} NEXTTIME^o{}_{Z_2}(y) = & \max_{t_i \quad CT(m-1)}\{\tau_i' - l_i + 1\} = \qquad (8.7)\\ & \max_{t_i \quad CT(m-1)}\{\boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i)) - l_i + 1\} = \\ & \max_{t_i \quad CT(m-1)}\{(\tau_i - 1) - l_i + 1\} \end{aligned}$$

where $CT(r) = \{t_i \quad CA_{Con_{\prod}(Z_1)}(t_{m_o}), i \le r\}$. Then, provided (8.5), (8.6) and (8.7), we obtain

$\mathrm{NEXTTIME}_{Z_2}(y) =$

$$\max(\max_{t_i \quad CT(m-1)}\{(\tau_i - 1) - l_i + 1\}, (\tau_2 - 1) - l_m + 1) =$$

$$\max([\max_{t_i \quad CT(m-1)}\{\tau_i - l_i + 1\}] - 1, [\tau_2 - l_m + 1] - 1) =$$

$$\max(\mathrm{NEXTTIME}^{o}{}_{Z_1}(y) - 1, ((\mathrm{TIME}_{Z_1}(x_e) - 1) - l_m + 1) =$$

$$\boldsymbol{ALPHA}_{Z_1}(y, \boldsymbol{h_j}(x_o, x_k, l_m), \mathrm{TIME}_{Z_1}(x_e) - l_m + 1) - 1 =$$

$\mathrm{NEXTTIME}_{Z_1}(y) - 1$,
Hence,

$$\mathrm{NEXTTIME}_{Z_2}(y) = \mathrm{NEXTTIME}_{Z_1}(y) - 1, \qquad (8.8)$$

As we know from production 4_j TIME(x) determines the value of parameter τ of each symbol, in particular, $\boldsymbol{t}(p, t_{m_o}, \tau)$. Consequently, $\mathrm{TIME}_{Z_1}(x) = \tau$, and, taking (8.8) into account, we obtain

$$\mathrm{TIME}_{Z_2}(x) = \mathrm{NEXTTIME}_{Z_2}(x) = \mathrm{NEXTTIME}_{Z_1}(x) - 1 = \mathrm{TIME}_{Z_1}(x) - 1 =$$

$$\tau - 1 = \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_{m_o}, \tau)) \geq \boldsymbol{len}(p, t_{m_o}). \qquad (8.9)$$

Hence, trajectory t_{m_o} of the length $\boldsymbol{len}(p, t_{m_o})$ will be generated by G_Z and corresponding symbol $\boldsymbol{t}(p_i, t_i, \tau')$ will be attached to Zone Z_2. Now we have to determine the value of parameter τ'. Obviously, τ' is determined by the value of $\mathrm{TIME}_{Z_2}(x)$ in production 4_j. According to (8.9),

$$\tau' = \mathrm{TIME}_{Z_2}(x) = \mathrm{TIME}_{Z_1}(x) - 1 = \tau - 1 = \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_{m_o}, \tau)),$$

and our statement about m_o-negation trajectories is proved: $\boldsymbol{t}(p, t_{m_o}, \tau - 1) \quad A(Z_2) \cap \prod(\mathrm{Con}_{\prod}(Z_1))$. Thus, by induction the general direct statement is *proved.*

Let us prove the *reverse* statement. Analogously, we conduct this proof by *induction.* For brevity we show only a general outline of this proof.

The *basis* of the induction is as follows. Consider an arbitrary 1-st negation trajectory t_i' and a corresponding symbol $\boldsymbol{t}(p_i, t_i', \tau_i') \quad A(Z_2) \cap \prod(\mathrm{Con}_{\prod}(Z_1))$. Let us show that there exists a 1-st negation trajectory t_i such that $\boldsymbol{t}(p_i, t_i, \tau_i) \quad \mathrm{Con}_{\prod}(Z_1)$, where $t_i = t_i'$, $\tau_i = \tau_i' + 1$, and

$$\boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i) = \tau_i - 1, \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i)) \geq \boldsymbol{len}(p_i, t_i). \qquad (8.10)$$

Consider trajectory $t_i = t_i$'. Let us prove that symbol $\boldsymbol{t}(p_i, t_i, \tau_i' + 1)$ will be generated by the grammar G_Z in a state S_1 and attached to Zone Z_1.

Analogously with the proof of direct statement, the maximum length of trajectories t_i to be the parameter value of the attaching symbol is determined by the value of function $\boldsymbol{f(u, v)}$ in production 3 (Table 1). This length is determined by the value of the third parameter of function $\boldsymbol{f(u, v)}$ which in this case is as follows (Table 2):

$$\boldsymbol{f}(u, v) = (1, y + 1, \mathrm{TIME}(y + 1) * v_{y+1}).$$

Points y + 1 are the ending points of prospective 1-st negation trajectories and, thus, belong to $P(t_o) - \{y_o\}$. According to (8.2) for each point x $\quad P(t_o) - \{y_o\}$ $\mathrm{TIME}_{Z_1}(x) = \mathrm{TIME}_{Z_2}(x) + 1$. At the same time TIME(x) determines the value of parameter τ_i' for each symbol $\boldsymbol{t}(p_i, t_i, \tau_i')$; it follows from production 4_j. Consequently, $\mathrm{TIME}_{Z_2}(x) = \tau_i'$, and taking into account, that t_i is included into Z_2 we obtain

$$\mathrm{TIME}_{Z_2}(x) \geq \boldsymbol{len}(p_i, t_i).$$

Thus,

$$\mathrm{TIME}_{Z_1}(x) = \mathrm{TIME}_{Z_2}(x) + 1 = \tau_i' + 1 \geq \boldsymbol{len}(p_i, t_i) \qquad (8.11)$$

Hence, the trajectories t_i of the length $\boldsymbol{len}(p_i, t_i)$ will be generated by G_Z and corresponding symbols $\boldsymbol{t}(p_i, t_i, \tau)$ will be attached to Zone Z_1. The only question to be answered is the question of the value of parameter τ. As we know τ is determined by the value of $TIME_{Z_1}(x)$ in production 4_j. According to (8.11),

$$\tau_i = TIME_{Z_1}(x) = TIME_{Z_2}(x) + 1 = \tau_i' + 1,$$

consequently,

$$\boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i) = \tau_i - 1, \boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i)) \geq \boldsymbol{len}(p_i, t_i),$$

and our statement about 1-st negation trajectories is proved: $\boldsymbol{t}(p_i, t_i, \tau_i)))\quad Con_\Pi(Z_1)$.

The *basis of induction is proved* by the preceding.

Assume that for all the m-negation trajectories t_m $m < m_o$ and $\boldsymbol{t}(p_m, t_m', \tau_m')$ from $A(Z_2)$, the statement of Theorem 8.1 (1) is true.

Let t_{m_o} is an arbitrary m_o negation trajectory, $\boldsymbol{t}(p, t_{m_o}', \tau')\quad A(Z_2)$. Let us show that there exists m_o- negation trajectory t_{m_o} such that $\boldsymbol{t}(p_i, t_{m_o}, \tau)\quad Con_\Pi(Z_1)$, where $t_{m_o} = t_{m_o}'$, $\tau = \tau' + 1$, and

$$\boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau) = \tau - 1, \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau)) \geq \boldsymbol{len}(p, t_{m_o}) \qquad (8.12)$$

Consider trajectory $t_{m_o} = t_{m_o}'$. Let us prove that symbol $\boldsymbol{t}(p, t_{m_o}, \tau' + 1)$ will be generated by the grammar G_Z in a state S_1 and attached to Zone Z_1. Then we shall prove (8.12).

The maximum length of trajectories t_{m_o} to be included into Z_1, i.e., to be the parameter value of the attaching symbol, is determined by the value of function $\boldsymbol{f}(u, v)$ in production 3 (Table 1). This length is determined by the value of the third parameter of function $\boldsymbol{f}(u, v)$, which in this case is as follows (Table 2):

$$\boldsymbol{f}(u, v) = (1, y + 1, TIME(y + 1) * v_{y+1}).$$

Points y+1 are the parameter values of the (m_o– 1)st negation trajectories. Values of TIME(y) are assigned by applying production 5 (section $\pi_{\boldsymbol{n}}$, Table 1). This application happens each time when generation of current negation is completed. Last application of production 5 took place when generation of (m_o–1)st negation trajectories was completed. Thus, values of NEXTTIME were assigned to TIME. The values of NEXTTIME were computed in the course of earlier applications of productions 4_j for attaching symbols with (m_o–1)st negation trajectories.

Let $m = m_o - 1$. Consider the generation of symbol $\boldsymbol{t}(p_m, t_m, \tau_2)\quad Con_\Pi(Z_1)$ with trajectory $t_m\quad t_{p_m}(x_o, x_k, l_m)$. Thus, applying formula $\pi_{\boldsymbol{n}}$ of production 4_j for $u = (x_o, x_k, l_m)$, $y\quad \boldsymbol{P}(t_m)$, we obtain

$$NEXTTIME\ (x_i) = \boldsymbol{ALPHA}(x_i, p_m, t_m, TIME(x_k) - l + 1).$$

Analogously to the proof of direct statement it is easy to show (basing on the assumption of induction) that on condition of reverse statement, (8.8) holds as well. As we know, TIME(x) determines the value of parameter τ' for each symbol $\boldsymbol{t}(p, t_{m_o}, \tau')$; it follows from production 4_j. Consequently, $TIME_{Z_2}(x) = \tau'$, and taking into account, that t_{m_o} is included into Z_2 we obtain

$$TIME_{Z_2}(x) \geq \boldsymbol{len}(p, t_{m_o}).$$

Thus, according to (8.8)

$$TIME_{Z_1}(x) = TIME_{Z_2}(x) + 1 = \tau' + 1 \geq \boldsymbol{len}(p, t_{m_o}) \qquad (8.13)$$

Hence, trajectory t_{m_o} of the length $\boldsymbol{len}(p, t_{m_o})$ will be generated by G_Z and corresponding symbol $\boldsymbol{t}(p, t_{m_o}, \tau)$ will be attached to Zone Z_1. The only question to be answered is the question of the value of parameter τ. As we know the value of τ is determined by the value of $TIME_{Z_1}(x)$ as well as τ' is determined by the value of $TIME_{Z_2}(x)$ (in production 4_j). According to (8.13),

$$\tau = TIME_{Z_1}(x) = TIME_{Z_2}(x) + 1 = \tau' + 1,$$

consequently,

$$\boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_{m_o}, \tau) = \tau - 1,\ \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_{m_o}, \tau)) \geq \boldsymbol{len}(p, t_{m_o}),$$

and our statement about m_o negation trajectories is proved: $\boldsymbol{t}(p_i, t_i, \tau_i)))$ $Con_{\Pi}(Z_1)$. Thus, by induction the general reverse statement is proved.

Theorem 8.1(1) is proved.

2. Let $\Pi_{Mo}(t_k) = t_k$', i.e., the non-main trajectory t_k of Zone Z_1 is shortened by exclusion of the first symbol, then we shall prove that for any symbol $\boldsymbol{t}(p_i, t_i, \tau_i)$ $Con_{\Pi}(Z_1)$

$$\pi_o(\boldsymbol{t}(p_i, t_i, \tau_i)) = \boldsymbol{t}(p, \Pi_{Mo}(t_i), \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_i, \tau_i)))$$

if and only if the length of trajectory t_i $l \leq \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_i, \tau_i))$.

We denote $t_i' = \Pi_{Mo}(t_i)$. Obviously, $t_i' = t_i$ for all $t_i \neq t_k$. As in case 1 we have to prove two statements. The direct statement is as follows: for any symbol $\boldsymbol{t}(p_i, t_i, \tau_i)$ $Con_{\Pi}(Z_1)$ such that $l \leq \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_i, \tau_i))$ symbol $\boldsymbol{t}(p_i, \Pi_{Mo}(t_i), \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_i, \tau_i)))$ belongs to $A(Z_2)$. The reverse statement requires that for every symbol $\boldsymbol{t}(p_i, t_i', \tau_i')$ from $A(Z_2) \cap \Pi(Con_{\Pi}(Z_1))$ there exists symbol $\boldsymbol{t}(p_i, t_i, \tau_i)$ $Con_{\Pi}(Z_1)$ such that $\Pi_{Mo}(t_i) = t_i'$, $\boldsymbol{timer}_{\pi}(\boldsymbol{t}(p, t_i, \tau_i)) = \tau_i'$ and $l \leq \tau_i$.

Let us prove the direct statement. We are going to conduct this proof *by induction.*

The *basis* of the induction is as follows. Consider symbols $\boldsymbol{t}(p_i, t_i, \tau_i)$ $Con_{\Pi}(Z_1)$ for which $C_{TA(Z_1)}(t_i, t_o) = T$, i.e., the 1-st negation trajectories. Obviously, in this case $C_{TA(Z_2)}(t_i', t_o') = T$, where $t_i' = \Pi_{Mo}(t_i) = t_i$, except for the case when t_k is one of the 1-st negation trajectories. According to Definition 7.5 (2, a) $\tau_i = \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i))$. Assume that

$$\tau_i = \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i)) \geq \boldsymbol{len}(p_i, t_i). \qquad (8.14)$$

Let us prove that symbol $\boldsymbol{t}(p_i, t_i', \boldsymbol{timer}_{\pi}(\boldsymbol{t}(p_i, t_i, \tau_i)))$ will be generated by the grammar G_Z in a state S_2 and attached to Zone Z_2.

Indeed, the maximum length of trajectories t_i' to be the parameter value of the attaching symbol is determined by the value of function $\boldsymbol{f}(u, v)$ in production 3 (Table 1). This length is determined by the value of the third parameter of function $\boldsymbol{f}(u, v)$ which in this case is as follows (Table 2):

$$\boldsymbol{f}(u, v) = (1, y + 1, TIME(y + 1) * v_{y+1}).$$

Points y + 1 are the ending points of prospective 1-st negation trajectories and, thus, belong to $P(t_o)$. The values of TIME were computed by application of production 2_i (section $\pi_{\boldsymbol{n}}$).

From the expression for the kernel of the production 2_i, it follows that $\tau_o = l_o + 1$ for the terminal symbol $\boldsymbol{t}(p_o, t_o, \tau_o) = \boldsymbol{t}(\boldsymbol{h_i}^{\mathbf{o}}(x_o, y_o, l_o), \tau_o)$. In such case $\pi_o(\boldsymbol{t}(p_o, t_o, \tau_o)) = \boldsymbol{t}(\boldsymbol{h_i}^{\mathbf{o}}(x_1, y_o, l_o - 1), \tau_o')$, with $\tau_o' = l_o + 1$. Consider the following main trajectory of the Zone $t_o = \boldsymbol{a}(y_o)\boldsymbol{a}(y_1)\ldots\boldsymbol{a}(y_l)$ and its image $t_o' = \boldsymbol{a}(x_o)\boldsymbol{a}(x_1)\ldots\boldsymbol{a}(x_l)$. Obviously, $t_o = t_o$'. Let us take into account that $\boldsymbol{P}(t_o') = \{x \quad X \mid \boldsymbol{DIST}(x, p_o, t_o') < 2n\}$ and mapping Π_{M_o} causes the following one-to-one correspondence between $P(t_o)$ and $P(t_o')$:

$$x_i = y_i$$

for i = 0, 1,..., l . Then from the section $\pi_{\boldsymbol{n}}$ of production 2_i it follows that

$$TIME_{Z_2}(x_i) = \boldsymbol{DIST}(x_i, p_o, t_o') = \boldsymbol{DIST}(y_i, p_o, t_o) = TIME_{Z_1}(y_i)$$

Consequently, for each point x $P(t_o)$

$$TIME_{Z_2}(x) = TIME_{Z_1}(x). \qquad (8.15)$$

At the same time TIME(x) determines the value of parameter τ_i of each symbol $\boldsymbol{t}(p_i, t_i, \tau_i)$; this follows from production 4_j. Consequently, $TIME_{Z_1}(x) = \tau_i$, and taking (8.15), (8.14) into account, we obtain

$$\mathrm{TIME}_{Z_2}(x) = \mathrm{TIME}_{Z_1}(x) = \tau_i = \boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i)) \geq \boldsymbol{len}(p_i, t_i). \qquad (8.16)$$

Consider trajectories t_i' of the length $\boldsymbol{len}(p_i, t_i')$. Obviously, if $t_i = t_i$' these trajectories will be generated by G_Z in a state S_2 and corresponding symbols $\boldsymbol{t}(p_i, t_i, \tau')$ will be attached to Zone Z_2. In case of the shortening trajectory t_k we come to the same conclusion because t_k and t_k' have the same end, and $\boldsymbol{len}(p_i, t_k) > \boldsymbol{len}(p_i, t_k')$. The only question to be answered is the question of the value of parameter τ'. It was shown above that τ' is determined by the value of $\mathrm{TIME}_{Z_2}(x)$ in production 4_j. According to (8.16),

$$\tau_i' = \mathrm{TIME}_{Z_2}(x) = \mathrm{TIME}_{Z_1}(x) = \tau_i = \boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i)),$$

and our statement about 1-st negation trajectories is proved:

$$\boldsymbol{t}(p_i, t_i, \boldsymbol{timer}_\pi(\boldsymbol{t}(p_i, t_i, \tau_i))) \quad A(Z_2) \cap \prod(\mathrm{Con}_\prod(Z_1)).$$

The *basis of induction is proved* by the preceding.

Assume that for all the m negation trajectories t_m $m < m_o$ and $\boldsymbol{t}(p_m, t_m, \tau_m)$, the statement of Theorem 8.1 (2) is true.

Let t_{m_o} is a m_o negation trajectory, $\boldsymbol{t}(p, t_{m_o}, \tau)$ $\quad \mathrm{Con}_\prod(Z_1)$. According to condition of Theorem 8.1 $\pi_o(\boldsymbol{t}(p, t_{m_o}, \tau)) = \boldsymbol{t}(p, \prod_{M_o}(t_{m_o}), \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau)))$. Assume also that

$$\boldsymbol{len}(p, t_{m_o}) \leq \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau)). \qquad (8.17)$$

We are going to prove that symbol $\boldsymbol{t}(p, \prod_{M_o}(t_{m_o}), \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau)))$ will be generated by the grammar G_Z in a state S_2 and attached to Zone Z_2.

Obviously, $\prod_{M_o}(t_{m_o}) = t_{m_o}$ if $t_{m_o} = t_k$, otherwise $\prod_{M_o}(t_k) = t_k$'(shortened). From Definitions 7.2 and 7.5 (2,b) it follows that the value of $\boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau))$ is computed recursively. Let us show that

$$\boldsymbol{t}(p, \prod_{M_o}(t_{m_o}), \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau))) \quad A(Z_2) \cap \prod(\mathrm{Con}_\prod(Z_1)).$$

As we know the maximum length of a trajectory to be included into Z_2 is determined by the value of function $\boldsymbol{f}(u, v)$ in production 3 (Table 1). This length is determined by the value of the third parameter of function $\boldsymbol{f}(u, v)$ which in this case is as follows (Table 2):

$$\boldsymbol{f}(u, v) = (1, y + 1, \mathrm{TIME}(y + 1) * v_{y+1}).$$

Points y+1 are the parameter values of the (m_o-1)st negation trajectories. Values of TIME are assigned by applying production 5 (section $\pi_{\boldsymbol{n}}$, Table 1). This application happens each time when generation of current negation is completed. Last application of production 5 took place when generation of (m_o-1)st-negation trajectories was completed. Thus, values of NEXTTIME were assigned to TIME. Values of NEXTTIME(z) were computed in the course of earlier applications of productions 4_j for attaching symbols with (m_o-1) negation trajectories.

Let $m = m_o - 1$. Consider the generation of symbol $\boldsymbol{t}(p_m, \prod_{Mo}(t_m), \tau_2')$ with trajectory $\prod_{Mo}(t_m)$ $\quad t_{p_m}(x_o, x_e, l_m')$. Obviously, $\prod_{Mo}(t_m) = t_m$, if $t_m \neq t_k$. Let us apply formula $\pi_{\boldsymbol{n}}$ of production 4_j for $u = (x_o, x_e, l_m)$:

$$\mathrm{NEXTTIME}\,(x_i) = \boldsymbol{ALPHA}(x_i, p_m, t_m, \mathrm{TIME}(x_e) - l_m + 1).$$

Consequently, for y $\quad P(t_m)$

$$\begin{aligned}\mathrm{NEXTTIME}_{Z_2}(y) = {} & \qquad (8.18)\\ & \boldsymbol{ALPHA}_{Z_2}(y, \boldsymbol{h_j}(x_o, x_e, l_m), \mathrm{TIME}_{Z_2}(x_e) - l_m' + 1) = \\ & \max(\mathrm{NEXTTIME}^o{}_{Z_2}(y), \mathrm{TIME}_{Z_2}(x_e) - l_m' + 1) = \\ & \max(\mathrm{NEXTTIME}^o{}_{Z_2}(y), \tau_2' - l_m' + 1),\end{aligned}$$

where $\mathrm{NEXTTIME}^o{}_{Z_2}(y)$ are the values of function NEXTTIME(y) before current application of production 4_j in the derivation of Z_2.

Trajectory t_m is m negation trajectory with $m < m_o$. According to Definition 7.5 (2) and assumption of the induction if $t_m \neq t_k$ $(l_m' = l_m)$

$$\tau_2' - l_m' + 1 = \boldsymbol{timer}_\pi(\boldsymbol{t}(p_m, t_m, \tau_2)) - \boldsymbol{len}(p_m, t_m) + 1 = TNEW(t_m).$$

If $t_m = t_k$, i.e., $l_k' = \boldsymbol{len}(p_m, \prod_{Mo}(t_m)) = l_k - 1$,

$$\tau_2' - l_k' + 1 = (\tau_2'+1) - l_k + 1 = \boldsymbol{timer}_\pi(\boldsymbol{t}(p_k, t_k, \tau_2)) - \boldsymbol{len}(p_k, t_k) + 1 = TNEW(t_k).$$

Thus, in all cases

$$\tau_2' - l_m' + 1 = TNEW(t_m). \qquad (8.19)$$

Analogously, it is easy to show that

$$NEXTTIME^o{}_{Z_2}(y) = \max_{t_i \quad CT(m-1)}\{TNEW(t_i)\}, \text{ where } CT(r) = \{t_i \quad CA_{Con_{\prod}(Z_1)}(t_{m_o}),\ i \leq r\}. \qquad (8.20)$$

Indeed, trajectories t_i from $Con_{\prod}(Z_1)$ as well as $\prod_{Mo}(t_i)$ from $\prod(Con_{\prod}(Z_1))$ are r negation trajectories with $r < m_o$, so the assumption of induction is true for them. Then, provided (8.18), (8.19) and (8.20) we obtain

$$\max(NEXTTIME^o{}_{Z_2}(y), \tau_2' - l_m' + 1) = \qquad (8.21)$$

$$\max(\max_{t_i \quad CT(m-1)}\{TNEW(t_i)\}, TNEW(t_m)) =$$

$$\max_{t_j \quad CT(m)}\{TNEW(t_j)\}.$$

Finally, combining (8.18) and (8.21), we have

$$NEXTTIME_{Z_2}(y) = \max_{t_i \quad CA(t_{m_o})}\{TNEW(t_i)\} = \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau)), \qquad (8.22)$$

For computation and attaching t_{m_o} we go to the next, m_o negation. Consequently, taking (8.22) into account, we obtain

$$TIME_{Z_2}(x) = NEXTTIME_{Z_2}(x) = \qquad (8.23)$$

$$\boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau)) \geq \boldsymbol{len}(p, t_{m_o}) \geq \boldsymbol{len}(p, \prod_{M_0}(t_{m_o})).$$

Hence, trajectory $\prod_{M_0}(t_{m_o})$ of the length $\boldsymbol{len}(p, \prod_{M_0}(t_{m_o}))$ will be generated by G_Z and corresponding symbol $\boldsymbol{t}(p, \prod_{M_0}(t_{m_o}), \tau')$ will be attached to Zone Z_2. Now we have to determine the value of parameter τ'. Obviously, τ' is determined by the value of $TIME_{Z_2}(x)$ in production 4_j. According to (8.23),

$$\tau' = TIME_{Z_2}(x) = \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau)),$$

and our statement about m_o negation trajectories is proved:

$$\boldsymbol{t}(p, \prod_{M_0}(t_{m_o}), \boldsymbol{timer}_\pi(\boldsymbol{t}(p, t_{m_o}, \tau))) \quad A(Z_2) \cap \prod(Con_{\prod}(Z_1)).$$

Thus, by induction the general direct statement is *proved.*

The reverse statement can be proved by induction analogously with the proof of reverse statement in case 1.

3. $\prod_{Mo}(t_i) = t_i$ for all $t_i \quad TA(Z_1)$. The proof is obvious.
Theorem 8.1 *is proved.*

Theorem 8.1 gives a partial solution of the Frame Problem for the Hierarchy of Languages.

In the next Sections an example of the robot control model for the Air Force vehicles will be considered. Employing this example we are going to show in detail the generating of the Language of Zones and Translations.

9. An Example of Generating Techniques

Consider the Grammar of Zones (Tables 1, 2). This is a controlled grammar [33, 34]. Such grammars operate as follows. The initial permissible set of productions consists of the production with label 1. It should be applied first. Let us describe the application of a production in such grammar. Suppose that we attempt to apply production with label $\boldsymbol{l}$ to rewrite a symbol $\boldsymbol{A}$. We choose the leftmost entry of symbol $\boldsymbol{A}$ in the current string and compute the value of predicate $\boldsymbol{Q}$, the condition of applicability of the production. If the current string *does not* contain $\boldsymbol{A}$ or $\boldsymbol{Q} = F$, then the application of the production is ended, and the next production is chosen from the failure section $\boldsymbol{F_F}$; $\boldsymbol{F_F}$ becomes the current permissible set. If the current string *does* contain the symbol $\boldsymbol{A}$, it is replaced by the string in the right side of the production; we carry out the computation of the values of all formulas either standing separately (from section $\pi_{\mathbf{n}}$) or corresponding to the parameters of the symbols ($\pi_{\mathbf{k}}$), and the parameters assume new values thus computed. Then, application of the production is ended, and the next production is chosen from the success section $\boldsymbol{F_T}$, which is now the current permissible set. If the applicable section is empty, the derivation halts.

Let us return to the robot control model shown in Fig.1, 3. We are going to apply grammar of Zones (shown in Table 1, 2) for generating trajectory network language for this model.

Let us generate the Language of Zones. Here we identify points of X with their ordinal numbers; thus, a1 corresponds to 1, a2 to 2, etc., h8 corresponds to 59 (g3, g4, d5, e6, f7 are excluded). We shall use both notations, algebraic and numerical, where it is convenient.

Let us apply grammar $\mathbf{G_Z}$ (Tables 1, 2) for different values of *u*. Production 1 is applicable for u = (h5, h1, 4) = (39, 8, 4), $l = l_0 = 4$ because

$$\boldsymbol{Q_1}(u) = (\mathrm{ON(BOMBER)=h5}) \wedge (\mathrm{MAP_{h5,BOMBER}(h1)} \le 4 \le 4) \wedge ((\mathrm{ON(TARGET)=h1}) \wedge$$
$$(\chi(\mathrm{BOMBER,\ TARGET})=0)) = \boldsymbol{T}.$$

Thus,

$$\boldsymbol{S}(u, zero, zero) \overset{1}{=>} \boldsymbol{A}(u, zero, zero)$$

and $\boldsymbol{F_T}$ = *two* is a permissible set. Therefor, next we have to apply one of the productions

2_i *two*. $\boldsymbol{Q_2}(u)$ is always true, so

$$\boldsymbol{A}(u, zero, zero) \overset{2_i}{=>} \boldsymbol{t}(\boldsymbol{h_i}^{\mathbf{o}}(\mathrm{u}), 5)\ \boldsymbol{A}((0, 0, 0), \boldsymbol{g}(\boldsymbol{h_i}^{\mathbf{o}}(u), zero), zero)$$

In order to compute $\boldsymbol{h_i}^{\mathbf{o}}(u)$ we have to generate all the shortest trajectories from h5 to h1 for the robot BOMBER. The length of these trajectories should be less or equal l = 4.

$$\mathrm{TRACKS_{BOMBER}} = \{\mathrm{BOMBER}\}\ \mathrm{x}\ (\ \ \underset{1\le k\le 4}{}\ \mathrm{L}[\ \mathbf{G_t}^{(1)}(\mathrm{h5,\ h1,\ k,\ BOMBER})].$$

According to the grammar of trajectories $\mathbf{G_t}^{(1)}$ [31-33] only one such trajectory t_1 exists, and it is generated by this grammar :

$$t_B = \boldsymbol{a}(\mathrm{h5})\boldsymbol{a}(\mathrm{h4})\boldsymbol{a}(\mathrm{h3})\boldsymbol{a}(\mathrm{h2})\boldsymbol{a}(\mathrm{h1}).$$

Thus, TRACKS ={(BOMBER, t_B)}, the number of trajectories b =1 and $\boldsymbol{h_1}^{\mathbf{o}}(u)$ =(BOMBER,t_B). In that way we generated the main trajectory of the Zone:

$$\boldsymbol{t}(\mathrm{BOMBER},\ t_B,\ 5).$$

Next we have to compute $\boldsymbol{g}(\boldsymbol{h_1}^{\mathbf{o}}(u), zero) = \boldsymbol{g}$ (BOMBER, t_B, *zero*). According to Table 2, for all r   X the r-th component of function $\boldsymbol{g}$ is as follows:

$$\boldsymbol{g}_{\mathbf{r}}(\mathrm{BOMBER}, t_B, zero) = \begin{cases} 1, & \text{if } \boldsymbol{DIST}(\mathrm{r,BOMBER},t_B) < 118, \\ 0, & \text{if } \boldsymbol{DIST}(\mathrm{r,BOMBER},t_B) = 118, \end{cases}$$

The value of the function $\boldsymbol{DIST}$ (x, BOMBER, t_B) = k+1, where k is the number of symbol of the trajectory t_B, whose parameter value equals x. Consequently

$$\boldsymbol{DIST}\ (\mathrm{h4,\ BOMBER},\ t_B) = 2,\ \boldsymbol{DIST}\ (\mathrm{h3,\ BOMBER},\ t_B) = 3$$
$$\boldsymbol{DIST}\ (\mathrm{h2,\ BOMBER},\ t_B) = 4,\ \boldsymbol{DIST}\ (\mathrm{h1,\ BOMBER},\ t_B) = 5$$

For the rest of x from X $\boldsymbol{DIST}$ (x, BOMBER, t_B) = 2 x 62 =124. Thus for r   {h1, h2, h3, h4} = {8, 16, 23, 30} $\boldsymbol{g}_{\mathbf{r}}$(BOMBER, t_B, *zero*) = 1, for the rest of r $\boldsymbol{g}_{\mathbf{r}}$= 0.

Now we can complete application of production 2_1:

$$\boldsymbol{A}(u, zero, zero) \Rightarrow \boldsymbol{t}(\text{BOMBER}, t_B, 5)\boldsymbol{A}((0, 0, 0), \boldsymbol{g}\ (\text{BOMBER},t_B,zero), zero).$$

Non-kernel functional formula from $\pi_{\mathbf{n}}$ remains for computation:

$$\text{TIME}(z) = \boldsymbol{DIST}(z, \text{BOMBER}, t_B).$$

Symbol “=“ in these formulas should be considered as an assignment, i.e., the current value of the right side expression should be assigned to the left side. The computation of $\boldsymbol{DIST}$(z, BOMBER, t_B) for all z from X has been performed above, so TIME(z) equals 124 for all z X except {h1, h2, h3, h4}, where TIME(z) equals 5, 4, 3, 2, respectively.

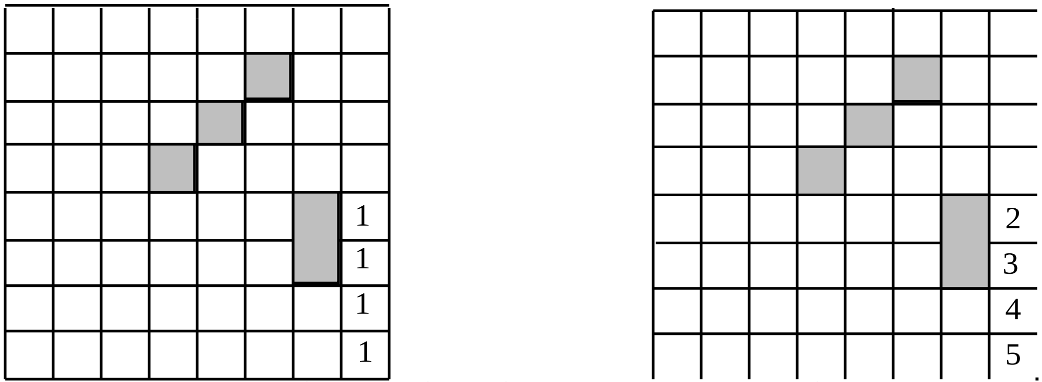

Fig. 6. *A representation of values of v* (left) *and* TIME(z) (right) *after generating trajectory* $\boldsymbol{a}$(h5)$\boldsymbol{a}$(h4)$\boldsymbol{a}$(h3)$\boldsymbol{a}$(h2)$\boldsymbol{a}$(h1).

Values of function $\boldsymbol{g}$ and, consequently, values of the components of vector v (Fig. 6, left), different from zero, mark ending points of prospective trajectories of robots from P_1 that could intercept motion of BOMBER along the main trajectory: points h1, h2, h3, h4. Values of TIME (Fig. 6, right) for the same points designate maximum lengths of those prospective trajectories. According to Definition 4.5 these trajectories are the *1-st negation* trajectories. Points h1, h2, h3, h4 are considered as targets by the other side, P_2, as well. It means that the grammar should generate trajectories of robots (if they exist) which could support motion of BOMBER by preventing its interception, the so-called *own* trajectories. By definition of the Grammar of Zones (Table 1, predicate $\boldsymbol{Q_4}$) the length of such trajectories is restricted by 1. Obviously, there are no own trajectories in the problem shown in Fig. 3.

Let us continue derivation of Zone. Production 2_1 was applied successfully, so we have to go to the production with label 3 and try to apply it to the left-most entry of nonterminal $\boldsymbol{A}$. This production is applicable because $\boldsymbol{Q_3}$ ((0, 0, 0)) = (0 ≠ 59) ^ (0 ≠ 59). Thus,

$$\boldsymbol{t}(\text{BOMBER}, t_1, 5)\boldsymbol{A}((0, 0, 0), v, zero) \ {}^{3}\!\!\Rightarrow \boldsymbol{t}(\text{BOMBER}, t_1, 5)\ \boldsymbol{A}(\boldsymbol{f}\ ((0, 0, 0), v), v, zero).$$

Next we have to compute value of the function $\boldsymbol{f}$. According to Table 2 for u = (x, y, l) = (0, 0, 0) and $v_{y+1} = v_1 = 0$:

$$\boldsymbol{f}(u, v) = (1, y+1, \text{TIME}(y+1) * v_{y+1}) = (1, 1, 0).$$

Therefor,

$${}^{3}\!\!\Rightarrow \boldsymbol{t}(\text{BOMBER}, t_B, 5)\ \boldsymbol{A}((1, 1, 0), v, zero)$$

It remains to compute values of the functional formula from $\pi_{\mathbf{n}}$.

$$\text{NEXTTIME}(z) = \boldsymbol{init}((0, 0, 0), \text{NEXTTIME}(z)) = 2n = 118 \text{ for all z from X.}$$

Application of the production 3 was successful so next we have to apply one of the productions 4_j to the left-most entry of the nonterminal $\boldsymbol{A}(u, v, w)$. Here u = (x, y, l) =(1, 1, 0), i.e., l = 0 and consequently $\boldsymbol{Q_4} = \boldsymbol{F}$. Thus, productions 4_j cannot be applied, so $\boldsymbol{F_F}$ is a permissible set here and we have to go back to the production 3.

We try to apply production to the nonterminal $\boldsymbol{A}(u, v, w)$ with u = (x, y, l) =(1, 1, 0), v shown in Fig. 6(left), and w = *zero*. Obviously, $\boldsymbol{Q_3}$(1, 1, 0) = $\boldsymbol{T}$, and this production is applicable:

$${}^{3}\!\!\Rightarrow \boldsymbol{t}(\text{BOMBER}, t_B, 5)\ \boldsymbol{A}(\boldsymbol{f}\ ((1, 1, 0), v), v, zero).$$

As far as (l =0) ^ (y =1) and $v_{y+1} = v_2 = 0$,

$$\boldsymbol{f}(u, v) = (1, \text{y+1}, \text{TIME(y+1)} * v_{y+1}) = (1, 2, 0).$$

Therefor,

$$\overset{3}{=>}\boldsymbol{t}(\text{BOMBER}, t_B, 5)\, \boldsymbol{A}((1, 2, 0), v, zero)$$

A computation of function NEXTTIME takes place as follows:

$$\text{NEXTTIME(z)} = \boldsymbol{init}((1, 1, 0), \text{NEXTTIME(z)}).$$

To prevent misunderstanding we have to remind that symbol "=" here means that value of the right side should be assigned to the left side, i.e., the new values of NEXTTIME are computed basing on the current values. Thus,

$$\text{NEXTTIME(z)} = 118 \text{ for all z from X.}$$

Application of the production 3 was successful so next again we will try to apply one of the productions 4_j. But $\boldsymbol{Q_4}(1, 2, 0) = \boldsymbol{F}$, and again we have to go back to production 3. $\boldsymbol{Q_3}(1, 2, 0) = \boldsymbol{T}$, this production is applicable, and this loop continues until u changes either way:

$$l = \text{TIME(y+1)} * v_{y+1} \neq 0 \quad \text{or} \quad \text{y} = 118.$$

In our case $v_{7+1} = 1$ ($\neq$ 0). Thus, the 8-th application of production 3 will result in the following string:

$$\overset{3}{=>}\boldsymbol{t}(\text{BOMBER}, t_B, 5)\, \boldsymbol{A}((1, 8, 5), v, zero)$$

because for u = (1, 7, 0) y+1 corresponds to h1, TIME(y+1) * v_{y+1} = TIME(h8) * 1 = 5.

This means that point h1 is determined as the ending point for generating trajectories of robots which intercept motion of the BOMBER. The following derivation steps would allow us to find possible starting points of such trajectories.

The next attempt of applying production 4_j will result in failure because there no robots at point x = 1, i.e., at point a1, and $\boldsymbol{Q_4}(1, 8, 5) = \boldsymbol{F}$. Again we return to production 3 but with $l > 0$ and x ≠ 59. This means the beginning of a new loop, which consists of multiple applications of production 3 after failures of attempts to apply one of productions 4_j .

$$\overset{3}{=>}\boldsymbol{t}(\text{BOMBER}, t_B, 5)\, \boldsymbol{A}((2, 8, 5), v, zero)$$
$$\overset{3}{=>}\boldsymbol{t}(\text{BOMBER}, t_B, 5)\, \boldsymbol{A}((3, 8, 5), v, zero)$$
$$\dots\dots\dots\dots$$
$$\overset{3}{=>}\boldsymbol{t}(\text{BOMBER}, t_B, 5)\, \boldsymbol{A}((42, 8, 5), v, zero)$$

With u = (42, 8, 5) this loop will be terminated because

$$\boldsymbol{Q_4}(42, 8, 5) = (\text{ON(FIGHTER)} = 44) \wedge (5 > 0) \wedge (\chi(\text{BOMBER, FIGHTER}) = 0) \wedge$$
$$(\text{MAP}_{\text{f6,FIGHTER}}(\text{h1}) = 5) = \boldsymbol{T}$$

which means that productions 4_j are applicable. These productions will generate intercepting trajectories from f6 to h1.

$$\overset{4_j}{=>}\boldsymbol{t}(\text{BOMBER}, t_B, 5)\boldsymbol{t}(\boldsymbol{h_j}(42, 8, 5), \text{TIME}(8))\boldsymbol{A}((42, 8, 5), v, \boldsymbol{g}(\boldsymbol{h_j}(42, 8, 5), zero))$$

In order to compute $\boldsymbol{h_j}(42, 8, 5)$ we have to generate all the shortest trajectories from point f6 to h1 for robot FIGHTER (Table 2). The length of these trajectories should be less or equal l = 5.

$$\text{TRACKS}_{\text{FIGHTER}} = \{\text{FIGHTER}\} \text{ x } (\underset{1\le k\le 5}{\quad} L[\, \mathbf{G_t^{(1)}}(\text{f6, h1, k, FIGHTER})].$$

$$\text{TRACKS} = \{(\text{FIGHTER}, t_1), (\text{FIGHTER}, t_2), (\text{FIGHTER}, t_3)\}, \text{ m} = 3 \text{ and}$$
$$\boldsymbol{h_1}(42, 8, 5) = (\text{ FIGHTER}, t_1),\ \ t_1=\boldsymbol{a}(\text{f6})\boldsymbol{a}(\text{e5})\boldsymbol{a}(\text{e4})\boldsymbol{a}(\text{f3})\boldsymbol{a}(\text{g2})\boldsymbol{a}(\text{h1}),$$
$$\boldsymbol{h_2}(42, 8, 5) = (\text{ FIGHTER}, t_2),\ t_2=\boldsymbol{a}(\text{f6})\boldsymbol{a}(\text{e5})\boldsymbol{a}(\text{f4})\boldsymbol{a}(\text{f3})\boldsymbol{a}(\text{g2})\boldsymbol{a}(\text{h1}),$$
$$\boldsymbol{h_3}(42, 8, 5) = (\text{ FIGHTER}, t_3),\ t_3=\boldsymbol{a}(\text{f6})\boldsymbol{a}(\text{f5})\boldsymbol{a}(\text{f4})\boldsymbol{a}(\text{f3})\boldsymbol{a}(\text{g2})\boldsymbol{a}(\text{h1}).$$

According to [33] there are three such trajectories, and they are generated by the certain grammar $\mathbf{G_t^{(1)}}$. (Of course, there is one more trajectory, $\boldsymbol{a}$(f6)$\boldsymbol{a}$(g5)$\boldsymbol{a}$(h4)$\boldsymbol{a}$(h3)$\boldsymbol{a}$(h2)$\boldsymbol{a}$(h1), which partially coincides with the main trajectory of the Zone and thus should be rejected.) Beginning with this step the derivation can be continued with three strings depending on the production applied on this step: 4_1, 4_2 or 4_3. It means we can derive *three* Zones with the same main trajectory and different intercepting trajectories from f6 to h1. Let us apply production 4_1 and continue derivation of Zone with the following trajectory

$$t_F = t_1 = \boldsymbol{a}(f6)\boldsymbol{a}(e5)\boldsymbol{a}(e4)\boldsymbol{a}(f3)\boldsymbol{a}(g2)\boldsymbol{a}(h1).$$

Thus, taking into account that TIME(8) = 5, we have

$$4_1 \Rightarrow \boldsymbol{t}(\text{BOMBER}, t_B, 5)\boldsymbol{t}((\text{FIGHTER}, t_F), 5)\boldsymbol{A}((42, 8, 5), v, \boldsymbol{g}(\text{FIGHTER}, t_F, \mathit{zero})).$$

Next we have to compute ***g***(FIGHTER, t_F,*zero*). According to Table 2, for all r X the r-th component of function ***g*** is as follows:

$$\boldsymbol{g}_{\mathbf{r}}(\text{FIGHTER}, t_F, \mathit{zero}) = \begin{cases} 1, & \text{if } \boldsymbol{DIST}(\text{r}, \text{FIGHTER}, t_F) < 118, \\ 0, & \text{if } \boldsymbol{DIST}(\text{r}, \text{FIGHTER}, t_F) = 118, \end{cases}$$

The value of function ***DIST*** (x, FIGHTER, t_F) = k+1, where k is the number of symbol of the trajectory t_F, whose parameter value equals x. Consequently

DIST (e5, FIGHTER, t_F) = 2, ***DIST*** (e4, FIGHTER, t_F) = 3, ***DIST*** (f3, FIGHTER, t_F) = 4
DIST (g2, FIGHTER, t_F) = 5, ***DIST*** (h1, FIGHTER, t_F) = 6

For the rest of x from X ***DIST*** (x, FIGHTER, t_F) = 2 x 59 =118. Thus for r {e5, e4, f3, g2, h1}={35, 28, 21, 15, 8} $\boldsymbol{g}_{\mathbf{r}}$(FIGHTER, t_F, *zero*) = 1, for the rest of r $\boldsymbol{g}_{\mathbf{r}}$= 0.

Now we can complete application of production 4_1. It remains to compute values of functional formula:

NEXTTIME(z) = ALPHA(z, (FIGHTER, t_F), 5–5+1).

As we know from previous steps NEXTTIME(x) = 118 for all x from X. Therefor, according to Table 2

$$\boldsymbol{ALPHA}(\text{x}, \text{FIGHTER}, t_F, 1) = \begin{cases} \boldsymbol{max}(\mathit{NEXTTIME}(\text{x}), 1), & \text{if } (\boldsymbol{DIST}(\text{x}, \text{FIGHTER}, t_F) \neq 118) \\ & (\mathit{NEXTTIME}(\text{x}) \neq 118) \\ \mathit{NEXTTIME}(\text{x}), & \text{if } \boldsymbol{DIST}(\text{x}, \text{FIGHTER}, t_F) = 118), \\ 1, & \text{if } \boldsymbol{DIST}(\text{x}, \text{FIGHTER}, t_F) \neq 118). \end{cases}$$

Thus, for x {e5, e4, f3, g2, h1} ***ALPHA***(x, FIGHTER, t_F, 1) = 1, while for other x ***ALPHA***(x, FIGHTER, t_F, 1) = 118. The same values should be assigned to NEXTTIME(z).

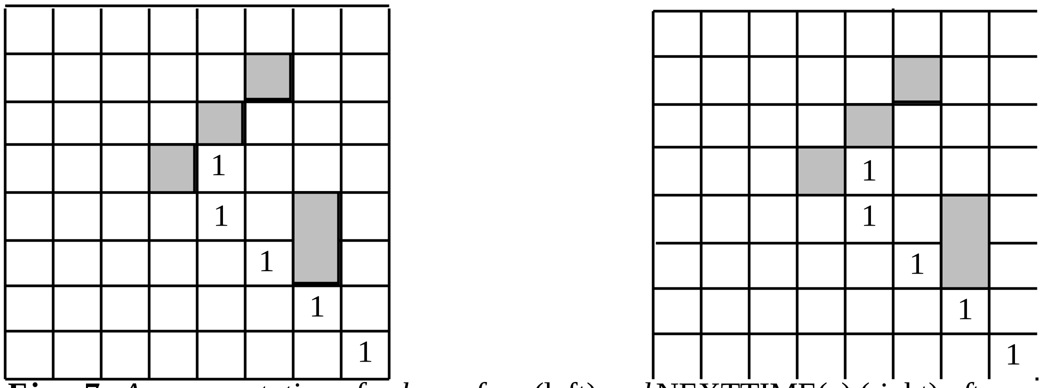

Fig. 7. *A representation of values of w* (left) *and* NEXTTIME(z) (right) *after generating trajectory* ***a***(f6)***a***(e5)***a***(e4)***a***(f3)***a***(g2)***a***(h1).

Values of function ***g*** and, consequently, values of components of vector *w*, different from zero, mark ending points of prospective trajectories of robots from P_1 that could support interception of BOMBER by protecting points the 1-st negation trajectories, points e5, e4, f3, g2, h1 in Fig. 7. According to Definition 4.5 these trajectories are the *2-nd negation* trajectories. Values of NEXTTIME for the same points (Fig. 7, right) designate maximum lengths of those prospective trajectories. These values are equal 1 because trajectory t_F is an intercepting trajectory of maximum length (5). It means that no one robot has enough time to intercept BOMBER at point h1 while moving along the trajectory of a greater length. Thus there is no extra time for robots from P_1 to approach points of trajectory t_F (for possible protection) while robot FIGHTER is moving along t_F. Values of *w* and NEXTTIME are computed employing productions 3 and 4_j, while 1-st negation trajectories are generated. After completion of this generation these values will be assigned to *v* and TIME, respectively, (production 5) to be used for generation of the 2-nd negation trajectories.

Points e5, e4, f3, g2, h1 are considered as *targets* by the other side P_2 as well. It means that the grammar should generate trajectories of robots (if they exist) which could intercept motion of FIGHTER, and thus prevent interception of BOMBER, the *own* trajectories. By definition of the Grammar of Zones (Table 1, predicate $\boldsymbol{Q_4}$) the length of such trajectories is restricted by 1. (Obviously, there are no own trajectories in the problem shown in Fig. 1, 3.)

Let us continue derivation of Zone. Production 4_1 was applied successfully, so we have to go to the production with label 3 and proceed with searching possible starting points of the trajectories with h1 as the ending point. We return to production 3 but with u =(42, 8, 5), i.e., with l =5 > 0 and x ≠ 59. This means the beginning of a new loop which consists of multiple applications of production 3 after failures of attempts to apply one of productions 4_j

$$^{3}\!\Rightarrow \boldsymbol{t}(\text{BOMBER}, t_B, 5)\boldsymbol{t}(\text{FIGHTER}, t_F, 5)\, \boldsymbol{A}(43, 8, 5), v, w)$$

$$^{3}\!\Rightarrow \boldsymbol{t}(\text{BOMBER}, t_B, 5)\boldsymbol{t}((\text{FIGHTER}, t_F, 5)\, \boldsymbol{A}(44, 8, 5), v, w)$$

$$\ldots\ldots\ldots\ldots$$

$$^{3}\!\Rightarrow \boldsymbol{t}(\text{BOMBER}, t_B, 5)\boldsymbol{t}(\text{FIGHTER}, t_F, 5)\, \boldsymbol{A}(48, 8, 5), v, w).$$

The intercepting trajectory to be found is as follows: $t_M{}^1 = t_1 =\boldsymbol{a}(\text{d7})\boldsymbol{a}(\text{b5})\boldsymbol{a}(\text{f1})\boldsymbol{a}(\text{g2})\boldsymbol{a}(\text{h1})$. We have

$$^{4_1}\!\Rightarrow \boldsymbol{t}(\text{BOMBER}, t_B, 5)\boldsymbol{t}(\text{FIGHTER}, t_F, 5)\boldsymbol{t}(\text{MISSILE}, t_M, 5)$$
$$\boldsymbol{A}((48, 8, 5), v, \boldsymbol{g}(\text{MISSILE}, t_M, w)),$$

Now we have to compute values of the following formula:

$$\text{NEXTTIME}(z) = \boldsymbol{ALPHA}(z, \text{MISSILE}, t_M, 5\text{–}4+1).$$

According to Table 2:

$$\boldsymbol{ALPHA}(x,\text{MISSILE},t_M, 2) = \begin{cases} \boldsymbol{max}(\mathit{NEXTTIME}(x),2), & \text{if } (\boldsymbol{DIST}(x,\text{MISSILE},t_M) \neq 118) \\ & (\mathit{NEXTTIME}(x) \neq 118) \\ \mathit{NEXTTIME}(x), & \text{if } \boldsymbol{DIST}(x,\text{MISSILE},t_M) = 118), \\ 2, & \text{if } \boldsymbol{DIST}(x, \text{MISSILE},t_M) \neq 118). \end{cases}$$

Thus, for x {b5, f1, g2, h1} $\boldsymbol{ALPHA}$(x, MISSILE, t_M, 2) = 2, while for other x $\boldsymbol{ALPHA}$(x, MISSILE, t_M, 2) = 118. These values should be assigned to NEXTTIME(z).

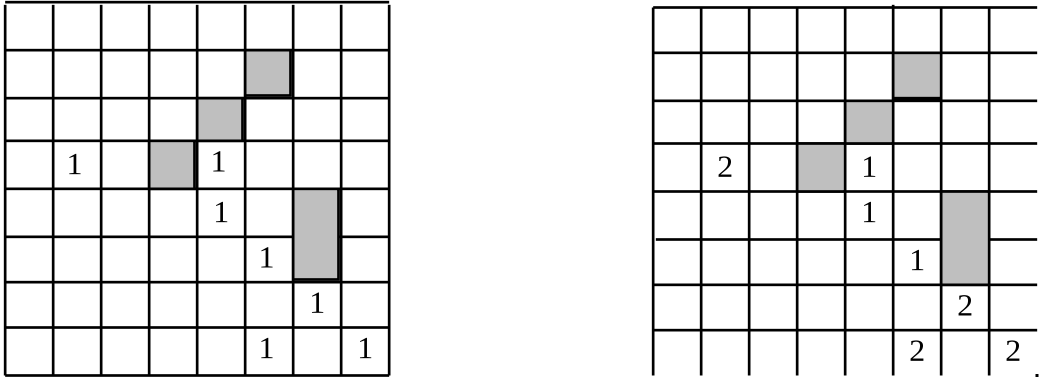

Fig. 8. *A representation of values of w* (left) *and* NEXTTIME(z) (right) *after generating trajectory* $\boldsymbol{a}$(d7)$\boldsymbol{a}$(b5)$\boldsymbol{a}$(f1)$\boldsymbol{a}$(g2)$\boldsymbol{a}$(h1).

Application of production 4_1 will result in the change of the values of *w* and NEXTTIME shown in Fig. 8. As we know values of NEXTTIME for the points b5, f1, g2, h1 designate maximum lengths of prospective 2-nd negation trajectories ending at those points. These values are equal 2 because trajectory t_M is an intercepting trajectory of non-maximum length (4) while the length of 5 is allowed. It means there is an extra time (2 time intervals) for robots from P_1 to approach points of trajectory t_M (for possible protection) while robot MISSILE is moving along t_M.

Then we continue searching for possible starting points of the trajectories with h1 as the ending point. We return to production 3 but with u =(48, 8, 5), i.e., with l =5 > 0 and x≠59. This means the beginning of a new loop, which consists of multiple applications of production 3 after failures of attempts to apply one of productions 4_j . This loop will be terminated when ***Q*₃**(*u*) = ***F***, i.e., (x = 59) ^ (y = 8):

^{3}=>***t***(BOMBER, t_B, 5)***t*(**FIGHTER, t_F, 5)***t*(**MISSILE, t_M, 5) ***A***(49, 8, 5), *v*, *w*)
^{3}=>***t***(BOMBER, t_B, 5)***t*(**(FIGHTER, t_F, 5)***t*(**MISSILE, t_M, 5) ***A***(50, 8, 5), *v*, *w*)
....
^{3}=>***t***(BOMBER, t_B, 5)***t*(**FIGHTER, t_F, 5)***t*(**MISSILE, t_M, 5) ***A***(59, 8, 5), *v*, *w*).

Computations of NEXTTIME(z) in production 3 will not change its values. With *u* = (59, 8, 5) this loop is terminated which means that no other starting points are found. Then a new loop begins. The grammar changes ending point of prospective trajectories:

^{3}=>***t***(BOMBER, t_B, 5)***t*(**FIGHTER, t_F, 5)***t*(**MISSILE, t_M, 5) ***A***(1, 9, 0), *v*, *w*)
^{3}=>***t***(BOMBER, t_B, 5)***t*(**(FIGHTER, t_F, 5)***t*(**MISSILE, t_M, 5) ***A***(1, 10, 0), *v*, *w*)
....

and eventually

^{3}=>***t***(BOMBER, t_B, 5)***t*(**FIGHTER, t_F, 5)***t*(**MISSILE, t_M, 5) ***A***(1, 16, 4), *v*, *w*),

because for *u* = (1, 15, 0) y+1 corresponds to h2, TIME(y+1) * v_{y+1} = TIME(h2) * 1 = 4.

This means that point h2 is determined as the next ending point for generating trajectories of robots that can intercept motion of the BOMBER. The following derivation steps would allow us to search for possible starting points of such trajectories. Obviously, nothing will be found. But the next ending point h3 will be successful. The following trajectory will be found: t_M^1 =***a***(d7)***a***(b5)***a***(f1)***a***(h3). We have

4_1=>***t***(BOMBER, t_B, 5)***t*(**FIGHTER, t_F, 5)***t*(**MISSILE, t_M, 5)***t*(**MISSILE, t_M^1, 5)
A((48, 23, 3),*v*, ***g***(MISSILE, t_M^1,*w*)),

Now we have to compute values of the following formula:

NEXTTIME(z) = ***ALPHA***(z, MISSILE, t_M^1, 3–3+1).

Values of NEXTTIME are shown in Fig. 9.

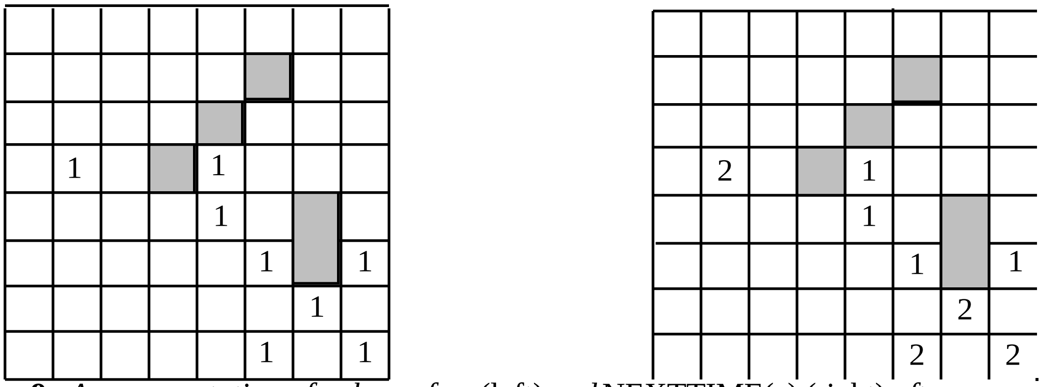

Fig. 9. *A representation of values of w* (left) *and* NEXTTIME(z) (right) *after generating trajectory* ***a***(d7)***a***(b5)***a***(f1)***a***(h3).

The same positive result will be achieved with the next ending point, h4. The intercepting trajectory to be found is as follows: t_F^1 =***a***(f6)***a***(g5)***a***(h4). We have

4_1=>***t***(BOMBER, t_B, 5)***t*(**FIGHTER, t_F, 5)***t*(**MISSILE, t_M, 5)***t*(**MISSILE, t_M^1, 5)
t*(**FIGHTER, t_F^1, 2)A***((42, 30, 2),*v*, ***g***(FIGHTER, t_F^1,*w*)),

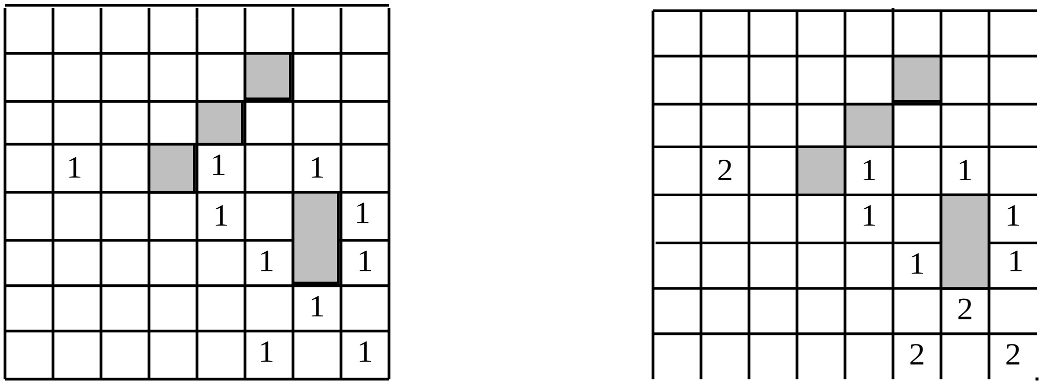

Fig. 10. *A representation of values of w* (left) and NEXTTIME(z) (right) *after generating trajectory* ***a***(f6)***a***(g5)***a***(h4).

Application of production 4_1 will result in the change of the values of *w* and NEXTTIME shown in Fig. 10. Then we continue applying production 3 returning to it each time after unsuccessful attempt of applying production 4_j. This loop will be terminated when $\boldsymbol{Q_3}(u) = \boldsymbol{F}$ for x=59.

Next we have to go to production 5. This production is applicable because $\boldsymbol{Q_5}(w) = (w \neq 0) = \boldsymbol{T}$ (current values of *w* are shown in Fig. 10). Thus,

^{5}=>***t***(BOMBER, t_B, 5)***t***(FIGHTER, t_F, 5)***t***(MISSILE, t_M, 5)***t***(MISSILE, t_M^1, 3)
t(FIGHTER, t_F^1, 2)***A***((0, 0, 0), *w*, *zero*)
TIME(z) = NEXTTIME(z)

This is the completion of generation of the 1-st negation trajectories, so production 5 performs the assignment we promised above. Values of *w* are assigned to *v* while NEXTTIME(z) are assigned to TIME(z). All the steps, 3 and 4_j, which have been executed (or tried) for generating 1-st negation trajectories, will be repeated for generating 2-nd negation trajectories. No one such trajectory should be found. The next return to production 5 will happen with *w = zero* (nothing is found). It means this production is not applicable, and we complete derivation applying production 6:

^{6}=>***t***(BOMBER, t_B, 5)***t***(FIGHTER, t_F, 5)***t***(MISSILE, t_M, 5)***t***(MISSILE, t_M^1, 3)
t(FIGHTER, t_F^1, 2).

10. An Example of Translations

Let us consider the model shown in Fig. 3 in dynamics. Assume that MISSILE has been lunched in advance, i.e., TRANSITION(MISSILE, d7, b5) took place. On its turn, the BOMBER from h5 took off, i.e., TRANSITION(BOMBER, d5, d4) happened. Consider values of π_o after these transitions. Let,

M_1 = TRANSITION(MISSILE, d7, b5)
M_2 = TRANSITION(BOMBER, d5, d4).

Let us apply M_1. Thus, according to Definitions 7.3, 7.5 (2, a), for the main trajectory we obtain:

π_o(***t***(BOMBER, t_B,5)) = (***t***(BOMBER,$\Pi_{M_1}(t_B)$, ***timer***$_\pi$(***t***(BOMBER,t_B, 5)) = ***t***(BOMBER, t_B, 5)

Similarly, according to Definition 7.5 (2, a) for all the 1-st negation trajectories we obtain

π_o(***t***(FIGHTER, t_F, 5)) = ***t***(FIGHTER, t_F, 5),
π_o(***t***(FIGHTER, t_F^1, 2)) = ***t***(FIGHTER, t_F^1, 2),
π_o(***t***(MISSILE, t_M, 5) = ***t***(MISSILE, $\Pi_{M_1}(t_M)$, 5),
π_o(***t***(MISSILE, t_M^1, 3) = ***t***(MISSILE, $\Pi_{M_1}(t_M^1)$, 3),

where $\Pi_{M_1}(t_M)$ and $\Pi_{M_1}(t_M^1)$ are shortened trajectories with excluded first symbol, i.e.,

$\Pi_{M_1}(t_M) = t_{M,s}$ =***a***(b5)***a***(f1)***a***(g2)***a***(h1), $\Pi_{M_1}(t_M^1) = t_{M,s}^1$ = ***a***(b5)***a***(f1)***a***(h3).

Lengths of all the 1-st negation trajectories of this Zone after the translation do not exceed values of $\boldsymbol{timer}_{\pi}$, consequently, according to Theorem 8.1, all these trajectories should be included into the new Zone $Z_2 = \pi_{M_1}(Z_1)$. Let us continue,

M_2 = TRANSITION(BOMBER, h5, h4).

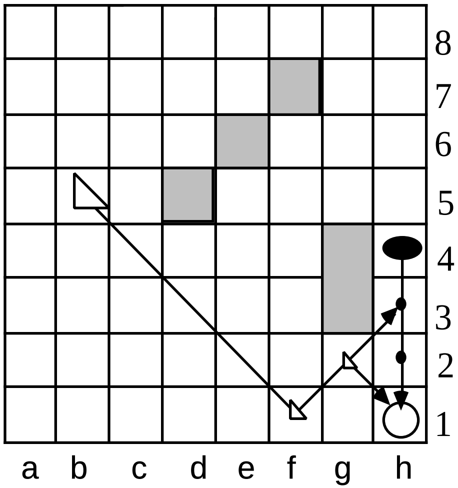

Fig. 11. *Network language in a state after transitions*

M_1 = TRANSITION(MISSILE, d7, b5), M_2 = TRANSITION(BOMBER, d5, d4).

Then, according to Definition 7.5 (1)

$\pi_o(\boldsymbol{t}(\text{BOMBER}, t_B,5)) = \boldsymbol{t}(\text{BOMBER}, \prod_{M_2}(t_B), 4)$

$\pi_o(\boldsymbol{t}(\text{MISSILE}, t_{M,s}, 5) = \boldsymbol{t}(\text{MISSILE}, t_{M,s}, 4),$

$\pi_o(\boldsymbol{t}(\text{MISSILE}, {t_{M,s}}^1 , 3) = \boldsymbol{t}(\text{MISSILE}, {t_{M,s}}^1 , 2),$

where $\prod_{M_2}(t_B) = t_{B,s} = \boldsymbol{a}(h4)\boldsymbol{a}(h3)\boldsymbol{a}(h2)\boldsymbol{a}(h1)$ is a shortened trajectory. For BOMBER and MISSILE the following inequalities hold

$\boldsymbol{len}(\text{BOMBER}, t_{B,s}) = 3 < 4,$

$\boldsymbol{len}(\text{MISSILE}, t_{M,s}) = 3 < 4,$

$\boldsymbol{len}(\text{MISSILE}, {t_{M,s}}^1) = 2 \leq 2.$

According to Theorem 8.1 it means that trajectories $t_{B,s}$, $t_{M,s}$, ${t_{M,s}}^1$ of BOMBER and MISSILE should be included into the new Zone $Z_3 = \pi_{M_2}(Z_2)$., i.e., MISSILE has enough time to intercept BOMBER at h3 or h1. But, considering trajectories of FIGHTER, we have

$\boldsymbol{t}(\text{FIGHTER}, t_F, \boldsymbol{timer}_{\pi}(\boldsymbol{t}(\text{FIGHTER}, t_F, 5)) = \boldsymbol{t}(\text{FIGHTER}, t_F, 4),$

$\boldsymbol{t}(\text{FIGHTER}, {t_F}^1, \boldsymbol{timer}_{\pi}(\boldsymbol{t}(\text{FIGHTER}, {t_F}^1, 2)) = \boldsymbol{t}(\text{FIGHTER}, {t_F}^1, 1),$

$\boldsymbol{len}(\text{FIGHTER}, {t_F}^1) = 2 > 1,$

$\boldsymbol{len}(\text{FIGHTER}, t_F) = 5 > 4,$

which means that trajectories ${t_F}^1$, ${t_F}^1$ of FIGHTER are not included into the new Zone Z_3. Indeed, after transition M_2 FIGHTER does not have enough time for interception of BOMBER at h4 or at h1.

Consider different variant of transitions leading from the initial state. Let

M_1 = TRANSITION(FIGHTER, f6, e5)
M_2 = TRANSITION(BOMBER, d5, d4).

Let us apply M_1. Thus, according to Definitions 7.3, 7.5 (2, a), for the main trajectory we obtain:

$\pi_o(\boldsymbol{t}(\text{BOMBER}, t_B,5)) = \boldsymbol{t}(\text{BOMBER}, t_B, 5)$

Similarly, according to Definition 7.5 (2, a) for all the 1-st negation trajectories we obtain

$\pi_o(\boldsymbol{t}(\text{FIGHTER}, t_F, 5)) = \boldsymbol{t}(\text{FIGHTER}, \prod_{M_1}(t_F), 5),$

$\pi_o(\boldsymbol{t}(\text{MISSILE}, t_M, 5) = \boldsymbol{t}(\text{MISSILE}, t_M, 5),$

$\pi_o(\boldsymbol{t}(\text{MISSILE}, {t_M}^1, 3) = \boldsymbol{t}(\text{MISSILE}, {t_M}^1, 3),$

where $\prod_{M_1}(t_F)$ is a shortened trajectory with excluded first symbol, i.e.,

$$\prod_{M_1}(t_F) = t_{F,s} = \boldsymbol{a}(e5)\boldsymbol{a}(e4)\boldsymbol{a}(f3)\boldsymbol{a}(g2)\boldsymbol{a}(h1).$$

Concerning $\prod_{M_1}(t_F{}^1)$, we conclude that after transition M_1 $t_F{}^1$ loose the connection with the main trajectory t_B, $\prod_{M_1}(t_F{}^1) = e$, hence $t_F{}^1$ $\quad Con_{\prod}(Z_1)$. Lengths of the 1-st negation trajectories of this Zone, accept for $t_F{}^1$, after translation $\prod_{M_1}$ do not exceed values of $\boldsymbol{timer}_\pi$, consequently, according to Theorem 8.1, all these trajectories should be included into the new Zone $Z_2 = \pi_{M_1}(Z_1)$. It means that both FIGHTER and MISSILE have enough time for interception.

Let us continue,

$$M_2 = \text{TRANSITION(BOMBER, h5, h4)}.$$

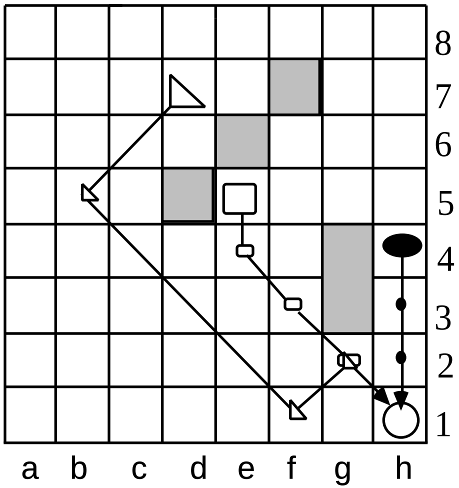

Fig. 12. *Network language in a state after transitions*

$$M_1 = \text{TRANSITION(FIGHTER, f6, e5)}, M_2 = \text{TRANSITION(BOMBER, d5, d4)}.$$

Then, according to Definition 7.5 (1)

$$\pi_o(\boldsymbol{t}(\text{BOMBER}, t_B, 5)) = \boldsymbol{t}(\text{BOMBER}, \textstyle\prod_{M_2}(t_B), 4)$$
$$\pi_o(\boldsymbol{t}(\text{FIGHTER}, t_{F,s}, 5)) = \boldsymbol{t}(\text{FIGHTER}, t_{F,s}, 4),$$
$$\pi_o(\boldsymbol{t}(\text{MISSILE}, t_M, 5) = \boldsymbol{t}(\text{MISSILE}, t_M, 4),$$

where $\prod_{M_2}(t_B) = t_{B,s} = \boldsymbol{a}(h4)\boldsymbol{a}(h3)\boldsymbol{a}(h2)\boldsymbol{a}(h1)$ is a shortened trajectory. For BOMBER, FIGHTER and MISSILE the following inequalities hold

$$\boldsymbol{len}(\text{BOMBER}, \ t_{B,s}) = 3 < 4,$$
$$\boldsymbol{len}(\text{FIGHTER}, \ t_{F,s}) = 4 \le 4,$$
$$\boldsymbol{len}(\text{MISSILE}, \ t_M) = 4 \le 4.$$

According to Theorem 8.1 it means that trajectories $t_{B,s}$, $t_{F,s}$, t_M of BOMBER, FIGHTER and MISSILE should be included into the new Zone $Z_3 = \pi_{M_2}(Z_2)$., i.e., FIGHTER and MISSILE have enough time to intercept BOMBER at h1. But, considering trajectory $t_M{}^1$ of MISSILE, we have

$$\boldsymbol{t}(\text{MISSILE}, t_M{}^1, \boldsymbol{timer}_\pi(\boldsymbol{t}(\text{MISSILE}, t_M{}^1, \ 3)) = \boldsymbol{t}(\text{FIGHTER}, t_F, 2),$$
$$\boldsymbol{len}(\text{MISSILE}, \ t_M{}^1) = 2 > 1,$$

which means that this trajectory is not included into the new Zone Z_3. Indeed, after transition M_2 MISSILE does not have enough time for interception of BOMBER at h3.

11. Discussion

In this paper we made a step towards a complete solution of the problem relative to the well known Frame Problem in Linguistic Geometry. This is the problem of efficient and constructive description of the change of the system representation while the system moves from one state to another one looking for an optimal operation. This problem is ever present in many existing artificial intelligence planning systems. A complete solution of the problem for the given model would permit us to avoid recomputation of the entire hierarchy of languages in each system state. Instead, we would be able to accomplish the differential recomputation of the changed part of the hierarchy, as well as computation of the completely new part. For a complete solution we have to investigate the trajectories that *loose the connection* with the main trajectory of the Zone in the new state, as well as new trajectories, which *did not exist* in the previous state. This investigation is in progress. A complete solution will ensure high effectiveness of the implementations of the hierarchy of languages.

References

1. M.R. Garey and D.S.Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co.: San Francisco, (1991).
2. H.A.Simon, *The Sciences of the Artificial, 2-nd ed.*, The MIT Press: Cambridge, MA, (1980).
3. M.D. Mesarovich, D. Macko, Y. Takahara Y., Theory of Hierarchical Multilevel Systems, Academic Press, New York, (1970).
4. M.M. Botvinnik, *Computers in Chess: Solving Inexact Search Problems.* Springer Series in Symbolic Computation, Springer-Verlag: New York , (1984).
5. J. McCarthy and P.J. Hayes, Some Philosophical Problems from the Standpoint of Artificial Intelligence, *Machine Intelligence*, vol. 4, 463–502, (1969).
6. R.E. Fikes and N.J. Nilsson, STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving, *Artificial Intelligence*, 2, 189–208, (1971).
7. J. McCarthy, Circumscription – A Form of Non-Monotonic Reasoning, *Artificial Intelligence*, 13, 27-39, (1980).
8. N.J. Nilsson, *Principles of Artificial Intelligence*, Tioga Publishing Co., Palo Alto, CA, (1980).
9. E.D. Sacerdoti, Planning in a Hierarchy of Abstraction Spaces, *ArtificialIntelligence*, 5-1, 115-135, (1974).
10. E.D. Sacerdoti, The Nonlinear Nature of Plans, *Proceedings of the International Joint Conference on Artificial Intelligence*, (1975).
11. M. Stefik, Planning and meta-planning (MOLGEN: Part 2), *ArtificialIntelligence*, 16-2, 141-169, (1981).
12. D. Chapman, Planning for conjunctive goals, *Artificial Intelligence*, 32-3, (1987).
13. D. McAllester and D. Rosenblitt, Systematic Non-Linear Planning, *Proc. of AAAI-91*, 634-639, 1991.
14. N. Chomsky, Formal Properties of Grammars, in *Handbook of MathematicalPsychology*, ed. R.Luce, R.Bush, E. Galanter., vol. 2, 323-418, John Wiley & Sons, New York, (1963).
15. S. Ginsburg, The Mathematical Theory of Context-Free Languages, McGraw Hill, New York, (1966).
16. D.E. Knuth, Semantics of Context-Free Languages, *Mathematical Systems Theory*, 2-2, 127–146, (1968).
17. D.J. Rozenkrantz, Programmed Grammars and Classes of Formal Languages, *Journal of the ACM*, 16-1, 107–131, (1969).
18. K.S. Fu, *Syntactic Pattern Recognition and Applications,* Prentice Hall, Englewood Cliffs, NJ, (1982).
19. R.N. Narasimhan, Syntax–Directed Interpretation of Classes of Pictures, *Communications of the ACM*, 9, 166–173, (1966).
20. T. Pavlidis, Linear and Context-Free Graph Grammars, *Journal of the ACM*, 19, 11–22, (1972).

21. A.C. Shaw, A Formal Picture Description Scheme as a Basis for Picture Processing System, *Information and Control*, 19, 9–52, (1969).
22. J. Feder, Plex languages, *Information Sciences*, 3, 225–241, (1971).
23. J.L. Pfaltz and A. Rosenfeld, WEB Grammars, *Proceedings of the 1-st International Joint Conference on Artificial Intelligence*, Washington, D.C., 609–619, (May 1969).
24. B.M. Stilman, Hierarchy of Formal Grammars for Solving Search Problems, in *Artificial Intelligence. Results and Prospects*, *Proceedings of the International Workshop*, Moscow, 63–72, (1985), [in Russian].
25. N.G. Volchenkov, The Interpreter of Context-Free Controlled Parameter Programmed Grammars, in *Cybernetics Problems. Intellectual Data Banks*, ed. by L.T. Kuzin, The USSR Academy of Sciences: Moscow, 147–157, (1979) [in Russian].
26. A.I. Reznitskiy and B.M. Stilman, Use of Method PIONEER in Automating the Planning of Maintenance of Power-Generating Equipment, *Automatics and Remote Control*, 11, 147-153, (1983) [in Russian].
27. M. Botvinnik, E. Petriyev, A. Reznitskiy, et al., Application of New Method for Solving Search Problems For Power Equipment Maintenance Scheduling", *Economics and Mathematical Methods*, 19-6, 1030-1041, (1983) [in Russian].
28. B. Stilman, A Linguistic Geometry of Complex Systems, *Abstr. of the Second Int. Symposium on Artificial Intelligence and Mathematics*, Ft. Lauderdale, FL, (Jan. 1992).
29. B. Stilman, A Syntactic Structure for Complex Systems, *Proc. of the Second Golden West International Conference on Intelligent Systems*, Reno, NE, 269-274, (June 1992).
30. B. Stilman, A Geometry of Hierarchical Systems: Generating Techniques, *Proc. of the Ninth Israeli Conference on Artificial Intelligence and Computer Vision*, Tel Aviv, Israel, 95-109, (Dec. 1992).
31. B. Stilman, A Syntactic Approach to Geometric Reasoning about Complex Systems, *Proc. of the Fifth International Symposium on ArtificialIntelligence*, Cancun, Mexico, 115-124, (Dec. 1992).
32. B. Stilman, A Linguistic Geometry of Complex Systems, *Annals of Mathematics and Artificial Intelligence*, (1992), (submitted).
33. B. Stilman, A Linguistic Approach to Geometric Reasoning, *Int. J. Computers and Mathematics with Applications*, (1992), (to appear).
34. B. Stilman, Network Languages for Complex Systems, *Int. J. Computers and Mathematics with Applications*, (1992), (to appear).