Bull. Nov. Comp. Center, Comp. Science, 36 (2014), 1–10

Modeling an artificial biological cell in a fine-grain structure

S.M. Achasova

Abstract. Two models of an artificial biological cell, built in a fine-grain structure, are presented. They can be elements of computing structures that mimic the properties of living organisms - growth, self-reproduction, self-healing. The models are built on the basis of the Parallel Substitution Algorithm, which is a spatial system for representing fine-grain parallel algorithms and architectures. An artificial biological cell is constructed from an artificial genome that is fed on the input tape. The model of an artificial biological cell contains the phenotype as a set of fixed data and the genotype as a set of mobile data. The phenotype is involved in performing a task that may be given an artificial biological organism; the genotype is able to produce daughter cells.

1. Introduction

John von Neumann used the concept of a cellular automaton for presenting and studying a logical form of self-reproduction [1]. His goal was to describe the fundamental principles and algorithms of information processing involved in the process of self-reproduction, in other words, to separate the logical form from the natural process of self-reproduction. It is interesting to note that a few years before J. Watson and F. Crick discovered the DNA double helix [2, 3] von Neumann had proposed an one-dimensional description (genome) of the self-replicating structure, which is fed on the input tape, and then generates the structure in the cellular automata space. In addition, von Neumann has formulated the principle of the dual use of the genome: it serves as a program for construction of the mother structure (translation of the genome), and, also, is copied to the mother structure (transcription of the genome) so that a daughter structure could then be produced.

Further study of self-reproduction was associated with the Langton loop [4–7]. This is a cellular automaton incapable of universal construction, but capable exclusively of self-replication. Originally the Langton loop was a rectangular loop in a two-dimensional cellular automata space. It is created on the basis of the periodic emitter, which is a fragment of the Codd cellular automaton [8], obtained, in turn, by simplifying the von Neumann machine. The self-description (genome) of the mother loop circulates in the Langton loop in the form of a sequence of cell states. Simultaneously with construction of a daughter loop the genome is rewritten into it, and then this loop creates its daughter. The Langton loop was used as a model for verification of hypotheses relating to the emergence of biological life [7,9,10]. The Langton loop was endowed with the ability to interact with the external observer [11]. Attempts were made to create "a useful replicator" on the basis of the Langton loop [12, 13], this is a cellular structure which executes a computational program together with constructing a copy. As the successful development of such a direction, self-replicating structures can be considered to be a new paradigm for designing fine-grain parallel algorithms and architectures.

The topic of this paper was inspired by papers [14–16]. In these papers as well as in [17–19] a new type of a self-replicating loop (the authors call it an artificial biological cell) is proposed, which can be a component of an artificial multicellular biological organism and provide simulation of the properties of living organisms: growth, self-replication, self-repair.

In this paper, two fine-grain models of an artificial biological cell are presented. These are a "star" and a "hedgehog". The models are built on the basis of the Parallel Substitution Algorithm (PSA), which is a spatial system for representing fine-grained parallel algorithms and architectures [20]. The PSA is an expanded paradigm of the classical cellular automaton (CA) and has some new properties as compared to the CA, which enhances its functional and expressive abilities. These properties are as follows. An arbitrary substitution template is admitted. At each clock cycle, one substitution can change the states of several cells. A new type of substitution is introduced. This is the functional substitution, in which the new states of the cells are functions of the states of the adjacent cells. These properties of the PSA enable the creation of a compact, easily foreseeable and structured description of the process of building fine-grain models of artificial biological cells. The experience gained in constructing the Langton loop on the basis of the PSA is presented in [21]. WinALT simulating system for algorithms with a fine-grain parallelism is presented in [22].

2. A parallel substitution algorithm

In this paper, the PSA operates in the 2D cellular automata space and in discrete time. Each cell can change its current state according to a local rule or a substitution, where a new state of a cell is determined by its current state and the states of the neighbor cells are included into the substitution template. Substitutions can have an arbitrary template, which is a geometrical figure in a cellular automata space. The left-hand side of a substitution defines the condition of its applicability and consists of the base and the context. The right-hand side of a substitution defines new states of the base cells. The substitution does not change the states of its context cells. The new states of the base cells can be either states from a set of possible states (then, the substitution is referred to as a symbolic one), or functions of the states of the cells from the left-hand side of the substitution (such a substitution is referred to as functional one). All the substitutions that are applicable at a clock cycle are implemented simultaneously. Since the PSA allows arbitrary substitution templates, it is possible that the same cell can appear in the applicability zones of two substitutions. If this cell is a context cell for both substitutions or it is a context cell for one substitution and a base cell for the other one, then no collisions with a change in its state occur. There is no conflict in the case when a cell is a base cell for both substitutions and its new state is the same in these substitutions. A contradiction in substitution applicability occurs when the state of a common cell is changed by two substitutions in different ways. A parallel substitution algorithm must contain a consistent set of substitutions. The consistency criteria and the ways to test a set of parallel substitutions for consistency are given in [20].

3. The artificial biological cell "star"

The fact is there is a conflict of terminology: the same term "cell" is applied to a biological unity and a mathematical element of a cellular automata space. It is proposed to overcome this conflict in such a way. An element of the cellular automata space or a fine-grain structure is generally called "cell" and the adjective "cellular" refers only to a mathematical cell. When mentioning an artificial biological cell we will use two names given to an artificial biological cell: these are "star" and "hedgehog", or fully "artificial biological cell" or a slightly truncated one "artificial cell". So, in this section we present a parallel substitution algorithm STAR for constructing an artificial cell "star".

The "star" is constructed from four fields. Each field is made up of four cells. Two adjacent fields are perpendicular to each other (Figure 1). The artificial genome for the "star" is a string of eight characters [N 1 E 2 S 3 W 4]. The letters correspond to the control flags that are needed to build the skeleton of an artificial cell, i.e. to establish connections between its fields to the north, east, south and west. The numbers are function codes of an artificial biological cell. Two copies of the genome are fed on the "star" in order to construct the phenotype as a set of fixed symbols and the genotype

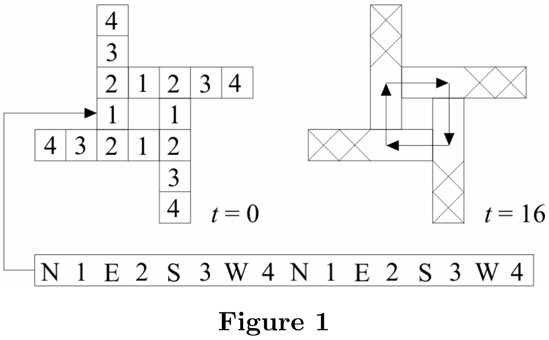


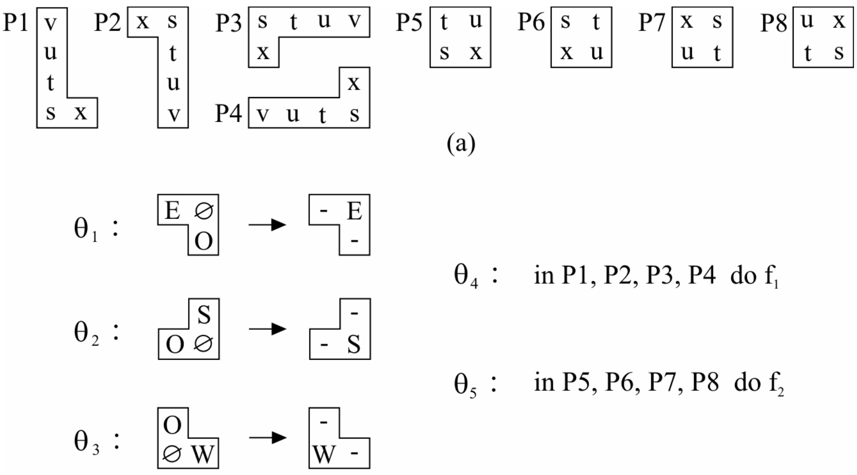
Figure 1

as a set of mobile symbols. The phenotype is involved in performing a task that may be given an artificial biological organism; the genotype is able to produce daughter cells.

Figure 1 shows the "star" at a zero time step ($t = 0$), where the cells of the fields are numbered, as well as the "star" in the form of an image at the final time step ($t = 16$), when constructing is completed. In the third and fourth cells of the fields of the "star" (the cells with crosses) a phenotype is recorded. A genotype circulates in the first and in the second cells of the fields.

The cells of the fine-grain structure, in which the "star" is built, can be in one of the ten states [N, E, S, W, 1, 2, 3, 4, $\emptyset$, O]. The eight states are the elements of the genome. The symbol $\emptyset$ is the quiescent state, in the figures it is represented by empty cells. The symbol O is an additional control flag. The PSA STAR contains three symbolic substitutions and two functional ones. Figure 2(a) shows the templates for the functional substitutions. Figure 2(b) shows the symbolic substitutions (the left column) and the functional ones (the right column). The functions used in the functional substitutions are given in Figure 2(c).

The dashes on the right-hand sides of the symbolic substitutions correspond to context cells. For the sake of brevity, we thought it correct to unite different templates that operate with the same function into one functional



(a)

(b)

f_1 : $t = s,\ u = t,\ v = u,\ \text{if}\ (v = \emptyset \wedge x = O)$

f_2 : $t = s,\ u = t,\ \text{if}\ (s \neq \emptyset \wedge t \neq \emptyset \wedge u \neq \emptyset \wedge x = O)$

(c)

Figure 2

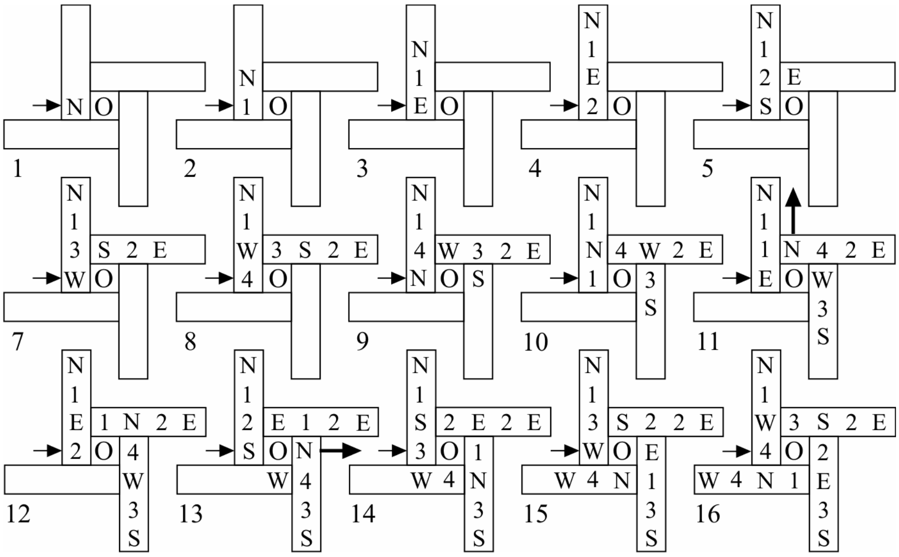


Figure 3

substitution, since these templates differ only in the orientation in space. The symbolic substitutions θ_1, θ_2, θ_3, and the functional one θ_4 build the "star", the functional substitution θ_5 controls the circulation of the genome in the "star". The functional substitution θ_4 is responsible for constructing the fields of the "star" and for forming the phenotype. First the north field (viewing to the north) is built; the genome comes to the first cell of the north field. Then the east, south and west fields are built. The substitutions θ_1, θ_2, θ_3 provide the transition of information from one field of the "star" to another. The field has been constructed, when elements of the phenotype are fixed in its third and fourth cells, and motion continues in its first and second cells. In Figure 3, the step-by-step execution of the PSA STAR is presented. At each step, an arrow approaches the first cell of the north field. This means that the next character of the genome arrives at this cell.

The bold arrows to the north at the 11th time step and to the east at the 13th time step mean that daughter artificial cells in the vertical and horizontal directions, respectively, can be constructed.

4. The artificial biological cell "hedgehog"

In this section, we describe the PSA HEDGEHOG for constructing the artificial biological cell "hedgehog". The artificial genome for this artificial cell contains sixteen characters [N 1 E 2 E 3 E 4 S 5 W 6 W 7 W 8]. The letters correspond to control flags and the numbers are function codes of an artificial biological cell. Two copies of the genome are fed on the "hedgehog" in

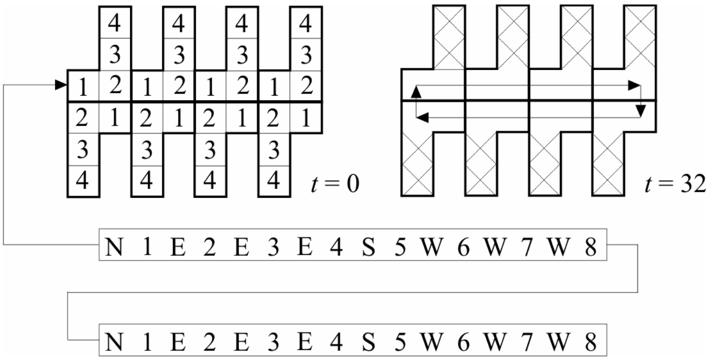


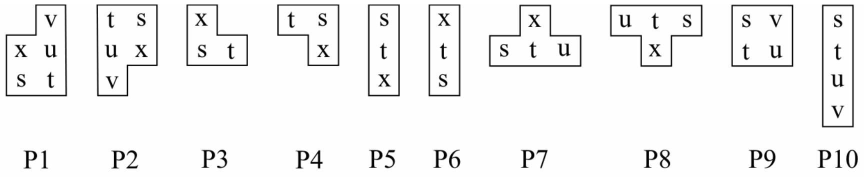
Figure 4

order to construct the phenotype as a set of fixed symbols and the genotype as a set of mobile symbols. The “hedgehog” is constructed from the eight fields. Four fields are located above and four — below. Each field contains four cells that form a right angle.

Figure 4 shows the “hedgehog” at the zero time step ($t = 0$), where the cells of the fields are numbered, and at the final time step ($t = 32$). In the third and fourth cells of the fields of the “hedgehog” (the cells with crosses) a phenotype is recorded. A genotype circulates in the first and second cells of the fields. The cells of the fine-grain structure, in which the “hedgehog” is built, can be in one of the fifteen states [N, E, S, W, 1, 2, 3, 4, 5, 6, 7, 8, $\emptyset$, O, I]. The twelve states are the elements of the genome. The symbols O and I are additional control flags.

The PSA HEDGEHOG contains eight functional substitutions θ_1, θ_2,…, θ_8 and two symbolic ones θ_9, θ_{10}. Figure 5(a) shows templates for the functional substitutions. Figure 5(b) shows the substitutions. The functions used in the functional substitutions are given in Figure 5(c).

All 10 substitutions can be divided into three groups according to their action types: the flag group, the motion group and the construction group. The substitutions θ_4, θ_5, θ_6 form the flag group. They set the control states O and I into the “hedgehog”. The motion group contains the substitutions θ_2, θ_3 that control the circulation of the genome into the “hedgehog”. The substitutions θ_1, θ_7, θ_8, θ_9, θ_{10} form the construction group. They are responsible for constructing the fields of the “hedgehog”, including forming a phenotype. The fields of the “hedgehog” are constructed one from another starting from the upper leftmost field, from which the connection passes to the east, and then three upper fields are built. From the upper rightmost field the connection passes to the south to the rightmost lower field and this field is built. From the rightmost lower field the connection passes to the west and three lower fields are built. In Figure 6, execution of the PSA HEDGEHOG is presented. At the 17th step of the construction of the



(a)

θ_1 : in P1, P2 do f_1

θ_2 : in P3, P4, P5, P6 do f_2

θ_3 : in P7, P8 do f_3

θ_4 : in P9 do f_4

θ_5 : in P9 do f_5

θ_6 : in P9, P10 do f_6

θ_7 : in P9 do f_7

θ_8 : in P9 do f_8

$$\theta_9 : \boxed{\begin{array}{cc} \varnothing & 6 \\ \varnothing & W \end{array}} \longrightarrow \boxed{\begin{array}{cc} - & - \\ O & - \end{array}}$$

$$\theta_{10} : \boxed{\begin{array}{c} 4 \\ S \\ \varnothing \\ \varnothing \end{array}} \longrightarrow \boxed{\begin{array}{c} - \\ - \\ S \\ - \end{array}}$$

(b)

f_1 : $t = s,\ u = t,\ v = u,$ if $[v = \varnothing \wedge (x = O \vee x = I)]$

f_2 : $t = s,$ if $[t \neq \varnothing \wedge (x = O \vee x = I)]$

f_3 : $t = s,\ u = t,$ if $[t \neq \varnothing \wedge u \neq \varnothing \wedge (x = O \vee x = I)]$

f_4 : $u = I,$ if $\{(u = \varnothing \wedge v = \varnothing) \wedge$
$[(s = N \wedge t = 1) \vee (s = E \wedge t = 3)]$

f_5 : $t = I,$ if $\{(s = \varnothing \wedge t = \varnothing) \wedge$
$[(u = 5 \wedge v = S) \vee (u = 7 \wedge v = W)]$

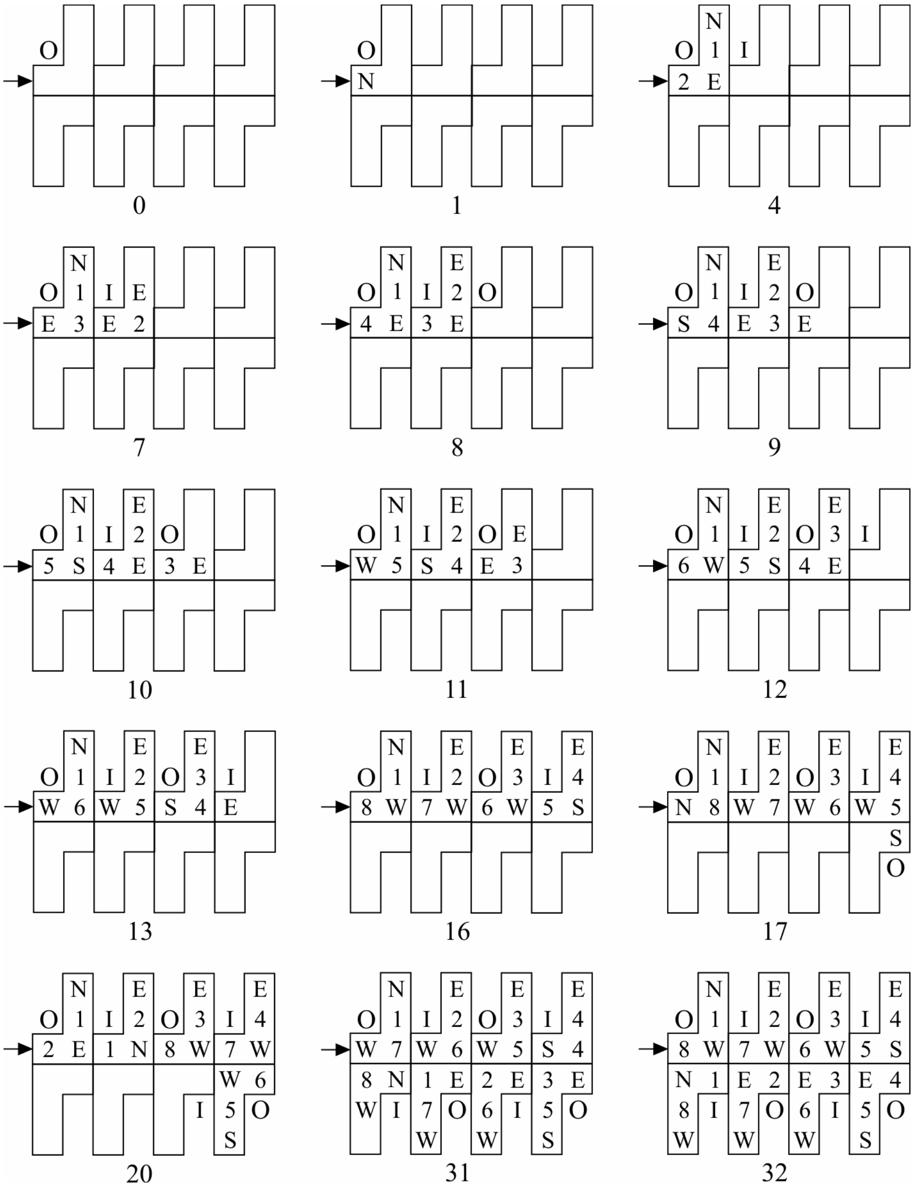
f_6 : $v = O,$ if $[(s = E \wedge u = \varnothing \wedge v = \varnothing) \wedge (t = 2 \vee t = 4)]$

f_7 : $v = t,$ if $\{(v = \varnothing \wedge t = E) \wedge [(u = I \wedge s = 1) \vee$
$(u = O \wedge s = 2) \vee (u = I \wedge s = 3)]\}$

f_8 : $s = u,$ if $\{(s = \varnothing \wedge u = W) \wedge [(t = I \wedge v = 5) \vee$
$(t = I \wedge v = 7) \vee (t = O \wedge v = 6)]\}$

(c)

Figure 5



0 1 4

7 8 9

10 11 12

13 16 17

20 31 32

Figure 6

"hedgehog", building a daughter artificial cell in the vertical direction can be initiated. Building a daughter artificial cell in the horizontal direction can be initiated at the 24th step.

There is a unicellular alga micrasterias radiate in Figure 7. It is slightly similar to the "hedgehog", is not it?

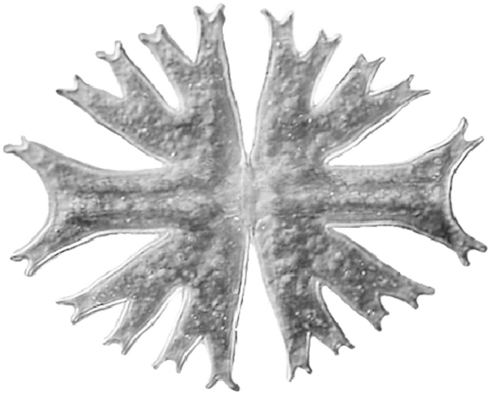


Figure 7

5. Conclusion

The programs for constructing the two models of an artificial biological cell in the form of a self-replicating structure in a fine-grain space are presented. The models are built on the basis of the Parallel Substitution Algorithm, which is a spatial system for representing fine-grain parallel algorithms and architectures. An artificial biological cell is constructed from a genome that is fed on the input tape. An artificial biological cell contains the phenotype as a set of fixed data and the genotype as a set of mobile data. The artificial cells can be elements of computing structures that mimic the properties of living organisms — growth, self-reproduction, self-healing. The devices possessing such properties can be used in the space research, in radioactive environments, avionics, etc.

References

[1] von Neumann J. Theory of Self-Replication Automata / Burks A.W., ed. — University of Illinois Press, 1966.

[2] Watson J., Crick F. A structure for deoxyribose nucleic acid // Nature. — 1953. — Vol. 171. – P. 737–738.

[3] Watson J.D. The Double Helix. — New York: Atheneum, 1968.

[4] Langton C.G. Self-replication in cellular automata // Physica D. — 1984. — Vol. 10. — P. 135–144.

[5] Byl J. Self-reproduction in small cellular automata // Physica D. — 1989. — Vol. 34. — P. 295–300.

[6] Reggia J.A., Armentrout S.I., Chou H.-H., Peng Y. Simple systems that exhibit self-directed replication // Science. — 1993. — Vol. 259. — P. 1282–1287.

[7] Langton C.G. Studying artificial life with cellular automata // Physica D. — 1986. — Vol. 22. — P. 120–149.

[8] Codd E.F. Cellular Automata. — New York: Academic Press, 1968.

[9] Chou H.-H., Reggia J.A. Emergence of self-reproducing structures in a cellular automata space // Physica D. — 1997. — Vol. 110. — P. 252–276.

[10] Azpeitia I., Ibanez J. Spontaneous emergence of robust cellular replicators // Lect. Notes in Comput. Sci. — 2002. — Vol. 2493. — P. 132–143.

[11] Stauffer A., Sipper M. Externally controllable and destructible self-replicating loops // Lect. Notes in Artificial Intelligence. — 2001. — Vol. 2159. — P. 282–291.

[12] Chou H.-H., Reggia J.A. Problem solving during artificial selection of self-replicating loops // Physica D. — 1998. — Vol. 115. — P. 293–312.

[13] Petraglio E., Henry J.-M., Tempesti G. Arithmetic operations on self-replicating cellular automata // Lect. Notes in Artificial Intelligence. — 1999. — Vol. 1674. — P. 447–456.

[14] Mange D., Stauffer A., Petraglio E., Tempesti G. Embryonic machines that divide and differentiate. // Lect. Notes in Comput. Sci. — 2004. — Vol. 3141. — P. 201–216.

[15] Mange D., Stauffer A., Petraglio E., Tempesti G. Self-replicating loop with universal construction // Physica D. — 2004. — Vol. 191. — P. 178–192.

[16] Stauffer A., Mange D., Tempesti G. Bio-inspired computing machines with self-repair mechanisms // Lect. Notes in Comput. Sci. — 2006. — Vol. 3853. — P. 128–140.

[17] Stauffer A., Mange D., Rossier J. Self-organizing systems based on bio-inspired properties // Lect. Notes in Artificial Intelligence. — 2007. — Vol. 4648. — P. 1171–1181.

[18] Stauffer A., Mange D., Vannel F. Bio-inspired self-organizing cellular systems // Biosystems. — 2008. — Vol. 94, Iss. 1–2. — P. 164–169.

[19] Tempesti G., Mange D., Stauffer A. Self-replicating and cellular automata // Encyclopedia of Complexity and Systems Science / Robert A. Meyers, ed. — Springer, 2009. — P. 8066–8084.

[20] Achasova S.M., Bandman O.L., Markova V.P., Piskunov S.V. Parallel Substitution Algorithm. Theory and Application. — Singapore: World Scientific, 1994.

[21] Achasova S.M. Program constructor of cellular self-reproducing structures // Programming and Computer Software. — 2009. — Vol. 35, No. 4. — P. 190–197.

[22] Ostapkevich M.B., Piskunov S.V. WinALT simulating system for algorithms with fine-grain parallelism // Vestnik Novosibirsk State University. — 2012. — Vol. 10, Iss. 3. — P. 34–45 (In Russian).