

Gritsak–Groener V. V.

BIOCOMPUTERS 1

Laboratory of HRIT Corporation, USA, UK
Royal Holloway, University of London

This article contains the main results.

1. There exist a DNA-RNA mechanism X such that for all NP-full problems \mathbf{p} ,
 - (i) \mathbf{p} are solved;
 - (ii) $\forall \mathbf{p} \in \mathbf{p}$ is solved in polynomial computational time.
2. There exist a 2-Band Linear Cellular Automata (2LCA) such that for all NP#-full problems \mathbf{q} ,
 - (i) \mathbf{q} is solved in polynomial computational time.

Keywords: computer, genome, cellular automata, matroid, category, NP-complexis.

1. Introduction

Computer's mathematical constructions make possible to estimate theoretically the complexity of solving on computer problems. Assume that the entering into computer information volume is expressed by some unified measure, for example, in bits. Then the algorithm of problem decision is acceptable if the decision time, expressed, for example, in steps number of Turing machine, polynomial depends from input information volume. But for the overwhelming majority of problems have been not succeeded in to construct the algorithms with an acceptable computational complexity. Moreover, properly from [1, 2] the choice is highly improbably within the framework of classical recursive schemes.

2. Matroids

We begin with definitions.

A *matroid* [2] is a pair $\mathbf{M} = (\mathbf{A}, \mathbf{B})$, where a finite set \mathbf{A} and a set \mathbf{B} of independence subsets $\mathbf{B} \subseteq 2^{\mathbf{A}}$; here a set \mathbf{B} such that

- (1) $\emptyset \in \mathbf{B}$;
- (2) $(\mathbf{b}_1 \subseteq \mathbf{b}_2) \wedge (\mathbf{b}_2 \in \mathbf{B}) \Rightarrow \mathbf{b}_1 \in \mathbf{B}$;
- (3) $(|\mathbf{b}_2| - |\mathbf{b}_1| = 1) \wedge (\mathbf{b}_1, \mathbf{b}_2 \in \mathbf{B}) \Rightarrow (\exists \alpha \in \mathbf{b}_2 / \mathbf{b}_1) \wedge (\mathbf{b}_1 \cup \{\alpha\} \in \mathbf{B})$.

Suppose $\mathbf{M} = (\mathbf{A}, \mathbf{B})$ is a matroid and $|\mathbf{A}| = \mathbf{n}$. A matroid $\mathbf{M} = (\mathbf{A}, \mathbf{B})$ is called *representable* over a field \mathbf{K} if there exists a matrix $\mathbf{M}_{\mathbf{K}}$ such that $|\mathbf{M}_{\mathbf{K}}| = \mathbf{m} \times \mathbf{n}$ and $\mathbf{M}_{\mathbf{K}}$ -coefficients belong to \mathbf{K} such that exists a bijection $\varphi : \mathbf{A} \rightarrow \mathbf{M}_{\mathbf{K}}^{col}$, where $\mathbf{M}_{\mathbf{K}}^{col}$ is the columns set of a matrix $\mathbf{M}_{\mathbf{K}}$ and a subset $\mathbf{A}_0 \subseteq \mathbf{A}$ is independent in \mathbf{M} iff $\varphi(\mathbf{A}_0)$ is linearly independent in $\mathbf{M}_{\mathbf{K}}$. We shall say that a representable matroid \mathbf{M} is called *ternary* if $\mathbf{K} = \text{GF}(3)$.

Correctly are the following propositions.

Theorem 1. [See 3]. Suppose $\mathbf{M} = (\mathbf{A}, \mathbf{B})$ is a ternary matroid and $|\mathbf{A}| = \mathbf{n}$. \mathbf{M} is unique representable over $\text{GF}(3)$.

3. Genetic Information and Matroids

The genetic information of living organism's majority is contained in DAN-molecules. At the beginning the information "is rewritten" on RNA-molecule and in what follows is realized in protein

receiving [4]. The realization process in factor RNA-informational — protein is written in the code representably by 4-valued matrix size 22×3 , see table 2. The columns "end" and "end 2" of information are excepted. The map

$$f: \{\text{Ala, Cys, } \dots, \text{Trp, Tyr}\} \longrightarrow \text{Codons} \quad (1)$$

is ambiguous. For example, $f: \text{Arg} \mapsto ((3,4,3), (3,4,4), (2,4,4), (2,4,3), (2,4,2), (2,4,1))$, see table 1. In table 2 the amino acids takes a codons to the equivalent codons and by replace the rules for the change of positions in genetical codons, see table 2.

Table 1.

1 st position	2 nd position	2 nd position	2 nd position	2 nd position	3 rd position
	U	C	A	G	
U	Phe	Ser	Tyr	Cys	U
U	Phe	Ser	Tyr	Cys	C
U	Leu	Ser	End	End 2	A
U	Leu	Ser	End	Trp	G
C	Leu	Pro	His	Arg	U
C	Leu	Pro	His	Arg	C
C	Leu	Pro	Gln	Arg	A
C	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
A	Ile	Thr	Asn	Ser	C
A	Ile	Thr	Lys	Arg	A
A	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
G	Val	Ala	Asp	Gly	C
G	Val	Ala	Glu	Gly	A
G	Val	Ala	Glu	Gly	G

Table 2 (U ↔ 1, C ↔ 2, A ↔ 3, G ↔ 4).

Position	Ala	Cys	Asp	Glu	Phe	Gly	His	Ile	Lys
1	4	1	4	4	1	4	2	3	3
2	2	4	3	3	1	4	3	1	3
3	3	2	2	3	1	3	2	3	3

Pos	Leu	Met	Asn	Pro	Gin	Arg	Ser	Thr	Val	Trp	Tyr
1	1	3	3	2	2	3	3	3	4	1	1
2	1	1	3	2	3	4	4	2	1	4	3
3	3	4	2	3	3	3	2	3	3	4	2

These results can be summarized as following.

Theorem 2. A matrix M_0 size 18×3 described above is the matroid representability matrix over $GF(3)$.

The matrix M_0 and proofs for theorem 3 are found in [5] and [6].

Continuing in the same way, we prove

Theorem 3. Let f be given by (1). We have linear cell automata $C = (D, f, \text{end}, \text{end2})$, where D is DNA-molecule, f is stand function of C , end is beginning information and end2 is finishing of information.

Detailed see in [7].

4. Complexity

The notion NP-complexity is a great invention of computers. The NP-full is supercomplexity of computational problems. Note that a calculation C has NP-full complexity, then $t(C)$ is large, where $t(C)$ is a computer computational time (CCT).

Theorem 4. [See 8]. The ternary matroid recognition problem is NP-full.

5. Topoi

The topos theory development was led to creation of "non-classical" recursive theories, which have more broad computational possibilities. In particular, the following proposition is correctly:

Theorem 5. Exists a topos \mathbf{T} for recursive theory over which is valid

$$\mathbf{P}^{\mathbf{T}} = \mathbf{NP}^{\mathbf{T}}, \tag{2}$$

where $\mathbf{NP}^{\mathbf{T}}$ and $\mathbf{P}^{\mathbf{T}}$ — respectively, are analogs of NP- completeness and polynomial computational complexity over \mathbf{T} [9].

6. NP*- full

The notion NP-full can be generalized and on the sufficiently big finite totalities. The corresponding reduced notions we'll denote by P^* and NP*-full ($P^{\#}$ and $NP^{\#}$ -full) recursive classes (automata cell computed). Is correctly the following proposition?

Theorem 6. A ternary matroid recognition problem is NP*-full.

7. Superpower Computational Mediums 1.

In a similar way to classical recursive theory, the theory of computational complexity over toposes is reduced on the sufficiently big finite totalities.

One of perspective way of the new computers creation is based on theoretical principles of cell automates (CA) direction. At the last time here is a big progress, see [10, 1] including a creation of real acting systems. From the pure theory of complexity computations point of view, the CA are the systems in which a number of processors can change with each iteration including unlimitedly grow. But in the real problems decision a number of active cells (corresponding to the working processors) is constantly or sufficiently restrictedly. That make possible to hope on creation of completely satisfactory algorithmic software CA.

CA is a sufficient universal model of recursive theories. Really, if by \mathbf{T} is denoted a topos for which is fulfilled (1), then is correctly the following proposition.

Theorem 7. [9] , [1]. Flat CA having a finite number of cells states and forming a finite configuration of active cells can be realized a recursive theory over the topos \mathbf{T} .

Theorem 7 points to theoretical possibility of computer creation having more broad computational possibilities compared with "classical" computers, that is computers in the basis of which is used a "classical" recursive theory. Theorems 1–5 point to possibility in principle of similar computer creation.

8. Superpower Computational Mediums 2.

Linear Cell Automata (LCA), when it goes beyond the very elementary level of the General Cell Automata, makes considerable use of the results of DNA-Automata, as we remarked in the p.42.

Let $A = (a_1, \dots, a_i, \dots) \neq \emptyset$ be a set, whose elements will be called *codons*. Let $M(A)$ be the

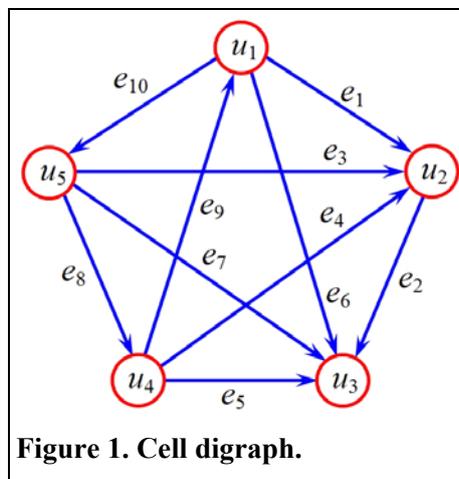


Figure 1. Cell digraph.

free monoid generated by A . The elements of $M(A)$ are called *g-words* (genetic words) and are identified with finite sequences

$$S = a_1 a_2 \dots a_s \#, \quad (3)$$

where $a_i \in A$, $i = [1, s]$, $\# = \text{end or end2}$. We recall that the *length* $l(S)$ of *g-words* (3) is s . The composition in $M(A)$ will be written multiplicatively, so that

$$S_1 \circ S_2 = a_1 a_2 \dots a_s \# b_1 b_2 \dots b_r \#, \quad (4)$$

is the sequence obtained by juxtaposition of $S_1 = a_1 a_2 \dots a_s \#$ and $S_2 = b_1 b_2 \dots b_r \#$.

The 0-words $\varepsilon = \emptyset$ is the identity element of the monoid $M(A)$. We are given a map

$$\begin{aligned} S_1 \\ w: A \longrightarrow N, \end{aligned}$$

where the set N is positive integers. $w(a_i)$ is called *arity* of the codon a_i . For each non-null word $S = a_1 a_2 \dots a_s \#$, we put

$$w(S) = \sum_{i=1}^s w(a_i),$$

and $w(\varepsilon) = 0$, $w(\#) = 2$. $w(S)$ is called the *significant* of the word S . If $S_1 = S \circ S_2 \circ S'$, the word S_2 is said to be a *segment* of S_1 . If $S_1 = S_2 \circ S_3 \circ S_4 \circ S_5 \circ S_6$, the segments S_3 and S_5 of S are said to be *disjoint*. We say that triple $J = (A, \zeta, U)$ has *cell universe*, if there is given an injective map

$$\zeta: \{\#, A\} \longrightarrow U \cup U^2 \cup \dots \cup U^m,$$

where U is non-empty set, $A = (a_1, \dots, a_i, \dots) \neq \emptyset$, and m is maximal arity of codons for A . The set U is called a *vertices cell* (*v-cell*). *Cell digraph* Γ is an ordered pair

$$\Gamma = (V, E), \quad V = A \setminus \{\#\}, \quad E \subset V \amalg V \quad (5)$$

where V is a nonempty set called the set of *vertices* of Γ ; E is a subset disjoint union from V , called the set of arrows of Γ if the following conditions hold:

- (1) $\forall x \in V, (x, x) \notin E$;
- (2) $\forall x, y \in V, (x, y) \in E \Rightarrow (y, x) \notin E$.

Before, if $e = \langle v_1, v_2 \rangle \in E \Rightarrow e^* = \langle v_2, v_1 \rangle \notin E$, $v_1, v_2 \in V$. If $e = \langle v_1, v_2 \rangle$ arrow of Γ , $v_1 \stackrel{\text{def}}{=} \partial^+ e$ is called the *initial* of e , and $v_2 \stackrel{\text{def}}{=} \partial^- e$ are called the *terminal* of e .

Let $\Gamma = (V, E)$ is cell digraph (5), $V^h \subseteq V$, $\# \in V^y \subseteq V$, and $D(\Gamma)$ is the set of all directed walk $\Gamma^{\rightarrow}(v_1, v_{n+1}) = v_1 e_1 v_2 e_2 v_3 \dots e_n v_{n+1}$, where $v_1 \in V^h$, $v_{n+1} \in V^y$. There is given a map $\psi: E \longrightarrow (\Xi, \Xi \times \Xi, \dots, \prod_{i=1}^d \Xi)$. The set Ξ is called an *arrows cell* (*a-cell*). V^h and V^y are called an *input* and an *output*.
Sekstant

$$CL = (\Gamma, \Xi, \psi, V^h, V^y, d) \quad (6)$$

is called *2-Band Cellular Automata* (2CA). Further, the word $S = a_1 a_2 \dots a_s \#$ induces 2LCA (6), if $a_1 \in V^h$, $a_s \in V^y$, $\zeta\{\#\} \longrightarrow U \cup U^2$, $\zeta\{a_i\} \longrightarrow U \cup U^2$, $\forall (a_i, a_{i+1}) \in E$, $\psi(a_i, a_{i+1}) \longrightarrow (\Xi, \Xi \times \Xi)$. Here, CL is called *2-Band Linear Cellular Automata* (2LCA). Suppose W_{CL}^2 is the set of all induces 2LCA words. W_{CL}^2 is called a *Induces 2LCA Languages*.

Detailed definition will be the objects of books [2, 3].

Examples of W_{CL}^2 :

- 1) English.
- 2) [See 2, 3] Ukrainian.
- 3) [See 8] Transcription and realization genome-information.

Theorem 8. [2]. 2LCA is computational full.

Moreover

Theorem 9. [10]. 2LCA is logical full.

Proposition. [Bixby, R.E. 1979]. A matroid M is ternary if and only if it has no minor isomorphic to any of the matroids $U_{2,5}$, $U_{3,5}$, F_7 , F_7^* , where $U_{2,5}$ is the uniform matroid of rank 2 on 5 elements, $U_{3,5}$ is the uniform matroid of rank 3 on 5 elements, F_7 is the matroid Fano on 7 elements, F_7^* is the dual matroid to matroid Fano on 7 elements.

Our main result is the following.

Theorem 10. Suppose M is a matroid; then exist the 2 -Band Linear Cellular Automata Ω of $P^\#$ -solution the following problems:

- 1) M has minor $U_{2,5}$;
- 2) M has minor $U_{3,5}$;
- 3) M has minor F_7 ;
- 4) M has minor F_7^* .

Finally

Consequence 1. Suppose π is problem of recognition submatrix M_0 in Theorem 2; then exist the 2 -Band Linear Cellular Automata Ω of $P^\#$ -solution of this problem.

Consequence 2. Exists the 2-Band Linear Cellular Automata Ω for recursive theory over which is valid

$$P^\# = NP^\#, \tag{7}$$

where $NP^\#$ and $P^\#$ — respectively, are analogs of NP — completeness and polynomial computational complexity over cell automata [10].

References :

1. Gritsak V. V. (Gritsak–Groener V. V.). // Proc. Nat. Acad. Sci. Ukraine SSR, ser. A, N6(1990)68.
2. Groener W. (Gritsak–Groener V. V.). Fundamental of Mathematical Cybernetics. MIT PRESS, 2004.
3. Gritsak–Groener V.V., Gritsak–Groener J. Arts Combinatoria. — Charkiv, NTU “ChPI”, 2003.
4. Alberts B., Bray D., Lewis J., Raff M., Roberts K., Watson J.D. Molecular biology of the cell. — New York, Garland Publishing, Inc., 1989.
5. Gritsak V. V. (Gritsak–Groener V. V.). Algorithms of Mathematical Chemistry. Preprint IK87-21. — Kyjiv, 1987.
6. Gritsak V. V. (Gritsak–Groener V. V.). Mathematical Chemistry. — SVITIZIR, Kyjiv, 1993.
7. Gritsak V. V. (Gritsak–Groener V. V.). // Proc. Nat. Acad. Sci. Ukraine SSR, ser. A, N7(1990)72.
8. Gritsak V. V. (Gritsak–Groener V. V.). Pattern Recognition Algorithms of Information in Hereditarily Alphabets Languages. Algorithms an software. 1984. — Kyjiv, 1985.
9. Gritsak V. V. (Gritsak–Groener V. V.). Combinatorial-Geometry Optimization Problems for Elementary Toposes, // Proceedings Republican Seminary “Discrete Optimisatin”.1985, IK AN Ukraine, Kyjiv, (1985) 31.
10. Gritsak V. V. (Gritsak–Groener V. V.). // Visnyk Shevtschenko National University, ser. phys.-math. science, N1(1993)5.
11. Valery V.Gritsak–Groener, Julia Gritsak–Groener, Hamid R. Arabia, Superpower Computational Creation for Biological Computing, Proceedings of the 2005 International Conference on Scientific Computing, Las Vegas, Nevada, USA June 20-23, 2005, pp.184-190.

Статья поступила в редакцию 12.05.2010 г.

Гритсак-Грёнер В. В.

Биокомпьютеры 1

В статье предложены следующие результаты.

1. Существует ДНК-РНК вычислительный механизм X такой, что для произвольной NP -полной проблемы

p :

- (i) p разрешима;
- (ii) $\forall \pi \in p$ разрешима в полиномиальное время.

2. Существует 2-ленточный клеточный автомат (2LCA) такой, что для произвольной $NP^\#$ -полной проблемы q :

- (i) q разрешим в полиномиальное вычислительное время.

Ключевые слова: компьютер, геном, клеточный автомат, матроид, категория, NP -сложность.