

# tRNA Computing Units and Programming Languages

Natalya Odincova  
Department of Mathematics and  
Mechanics  
Ural State University  
Ekaterinburg, Russia, 620083  
Email: odincova.antalya@gmail.com

Vladimir Popov  
Department of Mathematics and  
Mechanics  
Ural State University  
Ekaterinburg, Russia, 620083  
Email: Vladimir.Popov@usu.ru

**Abstract**—In this paper we consider some new computing units for DNA-based computers. Construction of such unit essentially based on properties of tRNA. Therefore, we call them as tRNA computing units.

## I. INTRODUCTION

In the recent years several new ideas have been developed to use non electronic natural phenomena for real, efficient computation. In classical electronic-based computations the information is stored and modified bitwise by electric and electromagnetic means. It is typical for this kind of computations that the number of steps performed per time unit is huge but the number of processors running in parallel is small. The main objective for the new approaches mentioned above is not to speed up the number of steps per time unit but to increase the degree of parallelism considerably. In 1985 D. Deutsch [1] proposed computers using quantum-physical effects to store and modify information. The quasi-probabilistic physical effect of quantum parallelism and mutual dependences of between all bits (coherence effects) allow to construct quantum algorithms that solve certain problems faster than any known probabilistic algorithm. In [1] Quantum Turing Machines are introduced as a theoretical model of such a kind of computation. In [2] quantum machine algorithms for the discrete logarithm and for integer factoring are given which run in polynomial time. In 1994 different approaches came up that used biological properties of DNA strings to store and modify information. The general idea is to use a large number of DNA strings as processors which compute in parallel. In [3] P. Pudlák introduced Genetic Turing Machines that are probabilistic machines which can simulate the evolution of a population of strings using two special operators controlling the inheritance and the survival of strings. In this model on each of the randomly chosen paths one string is processed. Also in 1994, L. Adleman [4] used biological experiments with DNA strings to solve some particular instances of the directed hamiltonian path problem which is considered to be intractable because of its **NP**-completeness. In [5] – [7] R. Lipton showed how to extend this idea to solve any problem and discussed the practical relevance of this approach. He defined a model of biological computing that has, besides the

classical means, the ability of manipulating large collections of DNA strings. Performing one of the special operations on a test tube means some simple manipulation of each of the strings in the test tube. In that way each DNA string corresponds to a piece of information, and all these pieces can be modified in parallel. At current DNA manipulation technology levels, DNA computing provides no advantage over electronic computers, for example, when encoding the computing task with DNA molecule in Adlemans directed hamiltonian path problem, if the  $n$  is equal to 100, the amount of DNA required would be larger than the weight of the earth. There is not enough room for improvement on algorithm to make the number of DNA molecules practically small. At this stage, some people began to worry about the directions of DNA computing study. However, in other sub-fields of DNA computing, great progress has been made. There are currently several research disciplines driving towards the creation and use of DNA nanostructures for both biological and non-biological applications. These converging areas are:

- the miniaturization of biosensors and biochips into the nanometer scale regime;
- the fabrication of nano-scale objects that can be placed in intracellular locations for monitoring and modifying cell function;
- the replacement of silicon devices with nano-scale molecular-based computational systems, and the application of biopolymers in the formation of novel nano-structured materials with unique optical and selective transport properties.

DNA computing employs DNA molecule as a main resource to fulfill computing tasks. However, the concept of primary DNA computing unit keeps obscure. It is recently realized that there are multiple forms of basic DNA computing units. Adleman uses short oligonucleotides to encode mathematical problems. The computing process is mainly performed in the form of hybridization. Ligation and other molecular manipulation steps are used for output abstraction. The correct answer is hidden in a vast amount of different hybridization results. Rothmund proposed a Turing machinelike DNA computing unit [8]. In [9] – [11] published another study in which

an autonomous programmable DNA automaton is created. In particular, in [9] – [11] for DNA automaton used a double-stranded DNA as input, endonuclease and DNA ligase as main hardware, transition molecules as software, thus creating a two-state molecular finite automaton with a two-symbol input, eight transition rules and 765 syntactically distinct programs. DNA self-assembly has become one of the most important directions for DNA computing [12] – [19]. Because of its universal computing capability, DNA assembly provides another avenue for universal DNA computer development. DNA computing by self-assembly is basically a tiling process, and the tile types can vary a lot. The tiles can be formed with several singlestranded oligos, and each tile can have different sticky DNA ends for a number of combinations with other same or different tiles. The tiling can be designed in a twodimensional or three-dimensional way, and the scale for tiling should also be able to control. DNA assembly can be completely programmed, though molecular biology experiments are still a bottle-neck for large scale assembly. In [19] authors brought a new landscape for this avenue. Combinatorial cellular automata also used in designing any tiling shapes. Besides, the natural affinity of DNA to bind with proteins, some types of small molecules, even metal atoms, makes it possible that assembled DNA can work as an inherent or transient matrix for novel computing devices. In [20] – [23] published a study in ribozyme unit research area. Ribozyme is a piece of nucleic acid fragment with unique three-dimensional structure that has an enzymatic ability to cut specific complementary oligos as substrate. If another oligo binds with the ribozyme and prevents it from forming enzymatic conformation, the ribozyme stays in an inactive form. In [20] – [23] founded ribozymes that can be easily manipulated as logical gates. Thus such ribozyme can mimic conventional electronic computing devices and theoretically develop universal DNA computing system. Ribozymes can work as automaton, though for the time being ribozyme or deoxyribozyme automaton is still in its infancy. Ribozymebased DNA computing unit may be extremely useful in designing logical computing devices in the future, for example, single-molecule logical gate. In [24], [25] also trying to employ ribozyme-based DNA computing as a potential vehicle for *in vivo* DNA computing. Instead of making ribozymes into logical gates or automata, in [24], [25] ribozymes used to build simple automata that may be easier for *in vivo* usage. Membrane computing [26], [27] can be regarded as a unique biological computing system. A cell is the basic unit for membrane computing system. This unit is not a DNA computing unit. However, membrane system provides another sort of self-assembly tile, and each such unit can hold DNA in it and may be able to translocate DNA molecules between each unit in the future, so we would like to treat such unit as a special DNA computing unit. It might be also called cell computing, a natural distributed architecture of a computing unit where any other DNA computing unit processes might be embedded. Since no kind of artificial membrane computing systems has been tested in the form of biochemical or physical biochemical experiments, it is likely that the natural cells may

be firstly tried by cell molecular biology manipulations. So some *in vivo* DNA computing technology may be needed to develop beforehand.

In this paper we consider some new DNA computing units. Construction of such unit essentially based on properties of tRNA. Therefore, we call them as tRNA computing units.

## II. tRNA COMPUTING UNITS

Transfer RNA (tRNA) is RNA that transfers a specific active amino acid to a growing polypeptide chain at the ribosomal site of protein synthesis during translation. tRNA has a 3' terminal site for amino acid attachment. This covalent linkage is catalyzed by an aminoacyl tRNA synthetase. It also contains a three base region called the anticodon that can base pair to the corresponding three base codon region on mRNA. Each type of tRNA molecule can be attached to only one type of amino acid, but because the genetic code contains multiple codons that specify the same amino acid, tRNA molecules bearing different anticodons may also carry the same amino acid. An anticodon [28] is a unit made up of three nucleotides that correspond to the three bases of the codon on the mRNA. Each tRNA contains a specific anticodon triplet sequence that can base-pair to one or more codons for an amino acid. To provide a one-to-one correspondence between tRNA molecules and codons that specify amino acids, 61 types of tRNA molecules would be required per cell. However, many cells contain fewer than 61 types of tRNAs because the wobble base is capable of binding to several, though not necessarily all, of the codons that specify a particular amino acid. A minimum of 31 tRNA are required to translate, unambiguously, all 61 sense codons of the standard genetic code [29].

The main function of tRNA is to recognize a fragment of single-stranded DNA molecule which consists of three nucleotides. As a result of such action is established a correspondence between the triplet of nucleotides of DNA nucleotides and a triple contact element of the tRNA molecule. *In vivo* tRNA molecule used for the amino acids synthesis. However, at least *in vitro* using special enzymes, we can stop the natural process of protein synthesis at the stage of reading nucleotide triplets of the DNA molecule and start the process of reading information from the tRNA molecules. As a result, we obtain a new DNA molecule. In the classical model of tRNA function we do not get anything interesting. In view of one-to-one correspondence between triples of DNA and tRNA we simply obtain a copy or some subsequence of the original DNA molecule. More precisely, *in vitro* we can produce any of following operations. Let

$$F[P](x) = y$$

where  $F[P](x)$  is a function with a parameter  $P$  of the variable  $x$  defined as follows:

$$x = x[1]z[1]x[2]z[2] \dots x[n]z[n]x[n+1],$$

$$y = z[1]z[2] \dots z[n],$$

$$\begin{aligned}
z[i] \in P \subseteq \mathcal{P} = \{ & UUU, UUC, UUA, \\
& UUG, UCU, UCC, UCA, UCG, \\
& UAU, UAC, UGU, UGC, UGG, \\
& CUU, CUC, CUA, CUG, CCU, \\
& CCC, CCA, CCG, CAU, CAC, \\
& CAA, CAG, CGU, CGC, CGA, \\
& CCG, AUU, AUC, AUA, AUG, \\
& ACU, ACC, ACA, ACG, AAU, \\
& AAC, AAA, AAG, AGU, AGC, \\
& AGA, AGG, GUU, GUC, GUA, \\
& GUG, GCU, GCC, GCA, GCG, \\
& GAU, GAC, GAA, GAG, GGU, \\
& GGC, GGA, GGG\}^+, \\
1 \leq i \leq n, \\
x[j] \in \{A, U, C, G\}^*, \\
1 \leq j \leq n + 1.
\end{aligned}$$

Note that *in vitro* the length of each word  $x[j]$  depends on the specific experimental conditions and the presence in this words subwords from

$$\{UAA, UGA, UAG\}.$$

In general case we can suppose that the length of  $x[j]$  is an arbitrary number. *In vivo*

$$x[j] \in \{UAA, UGA, UAG\}^*.$$

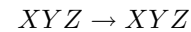
So, we obtain some set of tRNA computing units each of which is given by some operation  $F[P](x)$ . We call them as classical tRNA computing units.

Complete sets of tRNAs from one organism, including at least one isoacceptor species for each of the twenty amino acids, are known for several eubacteria (*Mycoplasma capricolum*, *Bacillus subtilis*, *Escherichia coli*), yeast (*Saccharomyces cerevisiae*) and chloroplasts (*Euglena gracilis*, *Marchantia polymorpha*, *Nicotiana tabacum*) or mitochondria (*Torulopsis glabra*, *ratus ratus*). The number of genes for a particular isoaccepting tRNA varies depending on the organism. Although these genes might have the same primary structure, it is more common that isoacceptor tRNAs feature the same anticodon but slightly differing sequences. In yeast, for example, the two tRNA<sub>GAA</sub><sup>phe</sup> [31] and the two tRNA<sub>IGU</sub><sup>thr</sup> [32] are identical except two nucleotides. Compensatory mutations frequently occur in the case when the difference between two isoacceptors is located in a stem. Again in yeast tRNA<sup>phe</sup>, an A-U base pair in the amino acid acceptor stem is exchanged for a G-C pair. The same replacement is found in yeast tRNA.

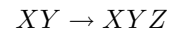
Transfer RNA is the most extensively modified nucleic acid in the cell. Modified nucleotides are contained in tRNAs from

all three phylogenetic domains (*archaea*, *bacteria*, *eucarya* [33], [34]). The modifications are not introduced during transcription, but are formed after the synthesis of the polynucleotide chain, serving for an improvement of the specificity and efficiency of tRNA biological functions. To date, more than eighty modified residues have been discovered and their chemical structures revealed [35]. Modified nucleotides are located at 61 different positions in tRNAs, mainly in loop regions. A large variety is present in the anticodon area, especially in the first position of the anticodon (position 34), and one base 3' to the anticodon (position 37). Apart from one exception (*archaeosine* at position 15 in archael tRNAs [36], all hypermodified residues are found in this region. Minor modifications like methylated or thiolated derivatives are usually situated outside the anticodon, with only one or two kinds of modified nucleotides present at each position. Some are common to almost all species, such as Dihydrouridine in D loops and Ribothymine in T loops, whereas others are characteristic of specific tRNAs. Examples are found in the hypermodified wybutosine residue (a guanosine derivative) at position 37 in almost all eukaryotic tRNA<sup>phe</sup> (except that from *Bombyx mori* and *Drosophila melanogaster*) and queuosine (another complicated post-transcriptional modification of guanosine) at the first anticodon position of certain tRNAs specific for tyr, his, asn and asp from eubacteria and eukaryotes. In both domains, modification takes place at different stages during the processing of precursor tRNA, depending strongly on the concentration of the substrate as well as on both the amount and the activity of tRNA-modifying enzymes. Several studies have been carried out on precursor tRNA<sup>tyr</sup>. The biosynthesis in *Xenopus laevis oocytes* initiated by injection of the yeast tRNA<sup>tyr</sup> gene into either the nucleus or the cytoplasm revealed that most base modifications occur in a sequential fashion in the nucleus before splicing [37], [38].

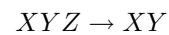
In many cases, the third nucleotide of the contact element of mutant tRNA is not functional. Non-functionality of third nucleotide of the contact element is connected with various mutations that lead to changes in the secondary and the tertiary structures of tRNA. It should be noted that these changes are stable. Note also that these mutations are quite common. In particular, the synthesis of some vital proteins of *Homo Sapiens* is only possible with the assistance of some mutant tRNA. The classical transformation



for some such mutant tRNAs can be represented in form



in case when a third nucleotide of tRNA contact element does not functional for the original DNA and for some other tRNAs can be represented in form



in case when a third nucleotide of tRNA contact element does not functional for the new DNA [30]. So, we obtain the

following set of relations:

$$\begin{aligned}
\mathcal{S} = \{ &UUU = UU, \\
&UUC = UU, UUA = UU, UUG = UU, \\
&UCU = UC, UCC = UC, UCA = UC, \\
&UCG = UC, UAU = UA, UAC = UA, \\
&UGU = UG, UGC = UG, UGG = UG, \\
&CUU = CU, CUC = CU, CUA = CU, \\
&CUG = CU, CCU = CC, CCC = CC, \\
&CCA = CC, CCG = CC, CAU = CA, \\
&CAC = CA, CAA = CA, CAG = CA, \\
&CGU = CG, CGC = CG, CGA = CG, \\
&CGG = CG, AUU = AU, AUC = AU, \\
&AUA = AU, AUG = AU, ACU = AC, \\
&ACC = AC, ACA = AC, ACG = AC, \\
&AAU = AA, AAC = AA, AAA = AA, \\
&AAG = AA, AGU = AG, AGC = AG, \\
&AGA = AG, AGG = AG, GUU = GU, \\
&GUC = GU, GUA = GU, GUG = GU, \\
&GCU = GC, GCC = GC, GCA = GC, \\
&GCG = GC, GAU = GA, GAC = GA, \\
&GAA = GA, GAG = GA, GGU = GG, \\
&GGC = GG, GGA = GG, GGG = GG\}
\end{aligned}$$

where

$$XYZ = XY$$

denotes the pair

$$XYZ \rightarrow XY,$$

$$XY \rightarrow XYZ.$$

We can produce any of following operations. Let

$$G[S, P](x) = y$$

where  $G[S, P](x)$  is a function with parameters  $S$  and  $P$  of the variable  $x$  defined as follows:

$$\begin{aligned}
x &= x[1]x[2] \dots x[n], \\
y &= y[1]y[2] \dots y[n], \\
y[i] &= \begin{cases} z, & x[i] \rightarrow z \in S \subseteq \mathcal{S} \\ x[i], & y[i] \in P \subseteq \mathcal{P} \\ \text{empty word} & \end{cases}
\end{aligned}$$

So, we obtain some set of tRNA computing units each of which is given by some operation  $G[S, P](x)$ . We call them as mutational tRNA computing units.

The frequent occurrence of non-canonical G-U base pairs [39] is a noticeable feature of stem regions. Since their first discovery in [40], other possible non-canonical pairs (for example A-A, C-C, C-U, G-A, U-U, U-Y) have been detected in the stems of various tRNAs [41]. G-U pairs, however, occur with the highest frequency. As to stems, a frequently occurring length can be attributed to loops as well. Anticodon and T loops contain seven nucleotides, whereas D loops and variable regions are areas of various lengths. An important discovery regarding the primary structure was made in the early 1970s. Certain positions in tRNAs are occupied by invariant or semi-invariant nucleotides.

Insights concerning characteristic behaviour of natural tRNA molecules were subsequently applied to the design of artificial tRNA molecules [30]. Using the complete sets of identity elements of some E. coli tRNAs as sequence constraints in inverse folding, a large amount of thermodynamically very stable sequences was obtained and subsequently sorted out due to inefficient folding behaviour.

Genes of interest can be selectively metallized via the incorporation of modified triphosphates [42]. These triphosphates bear functions that can be further derivatized with aldehyde groups via the use of click chemistry. Treatment of the aldehyde-labeled gene mixture with the Tollens reagent, followed by a development process, results in the selective metallization of the gene of interest in the presence of natural DNA strands.

In [43] reported a simple solution based method for the gold (Au) metallization of DNA resulting in a Au nanowire network. Advantage of solution based approach is that it allows the removal of excess gold ( $\text{Au}^{+3}$ ) ions by extraction with tetraoctylammonium bromide (TOAB) in order to avoid non specific metallization. Further it has been shown that Au metallized DNA obtained in aqueous phase can be transferred to organic phase using hexadecyl aniline (HDA). Au metallized DNA has potential application in nanoscale devices.

Also a number of small organic ribonucleases have been synthesized with rigid polycationic structures containing an aromatic framework with two residues of bis-quaternary salts of 1,4-diazabicyclo[2.2.2]octane (DABCO) bearing various substituents [44]. The compounds carrying positively charged groups connected via rigid linker are expected to bend the sugar-phosphate backbone and can stimulate the intramolecular phosphoester transfer reaction.

Since we can use artificial nucleotides and artificial tRNA molecules, we can consider artificial tRNA computing units. In this case we consider some alphabet  $\Sigma$  and the set of relations

$$\begin{aligned}
\mathcal{Q} = \{ &X_1Y_1Z_1 \rightarrow X_1Y_1Z_1, \\
&X_2Y_2 \rightarrow X_2Y_2Z_2, \\
&X_3Y_3Z_3 \rightarrow X_3Y_3 \mid \\
&X_i, Y_i, Z_i \in \Sigma, 1 \leq i \leq 3\}.
\end{aligned}$$

We can define following operations. Let

$$H[Q](x) = y$$

where  $H[Q](x)$  is a function with a parameters  $Q$  of the variable  $x$  defined as follows:

$$\begin{aligned} x &= x[1]x[2] \dots x[n], \\ y &= y[1]y[2] \dots y[n], \\ y[i] &= \begin{cases} z, x[i] \rightarrow z \in Q \subseteq \mathcal{Q} \\ \text{empty word} \end{cases} \end{aligned}$$

We obtain the set of tRNA computing units each of which is given by some operation  $H[Q](x)$ . We call them as artificial tRNA computing units.

### III. tRNA PROGRAMMING LANGUAGES

Let  $k \geq 0$  and  $m \geq 1$  be variables for natural numbers, let  $a, b \in \{0, 1\}$ , let  $x$  be a word variable and let  $T, T_1$  and  $T_2$  be set variables. Let  $I(x) \in \{0, 1\}^*$  be the contents of the word variable  $x$ , and let  $I(T) \subseteq \{0, 1\}^*$  be the contents of the set variable  $T$  in a given moment. We define the cut operation  $\setminus$  by  $\setminus av = v$  and  $\setminus \Delta = \Delta$  where  $\Delta$  is the empty word. Different types of DNA-computers use the following instructions with set operations and conditions with set tests [45].

$$\begin{aligned} T &= T_1 \cup T_2, \\ I(T_1) \cup I(T_2); \end{aligned}$$

$$\begin{aligned} T &= In(k), \\ \{0, 1\}^k; \end{aligned}$$

$$\begin{aligned} T &= T_1 \cdot T_2, \\ I(T_1) \cdot I(T_2); \end{aligned}$$

$$\begin{aligned} T &= \setminus T_1, \\ \{\setminus z \mid z \in I(T_1)\}; \end{aligned}$$

$$\begin{aligned} T &= Sw(T_1), \\ \{y \mid \exists v \exists w (vyw \in I(T_1))\}; \end{aligned}$$

$$\begin{aligned} T &= a \cdot T_1, \\ \{a\} \cdot I(T_1); \end{aligned}$$

$$\begin{aligned} T &= Eq(T_1 \cdot m \cdot a), \\ \{vaw \mid (v0w \in I(T_1) \vee v1w \in I(T_1)) \wedge |v| = m - 1\}; \end{aligned}$$

$$T = Bs(T_1 \cdot m \cdot a \cdot b),$$

$$\begin{aligned} \{vaw \mid v0w \in I(T_1) \wedge |v| = m - 1\} \cup \\ \{vbw \mid v1w \in I(T_1) \wedge |v| = m - 1\}; \end{aligned}$$

$$\begin{aligned} T &= Bx(T_1 \cdot m \cdot a), \\ I(T_1) \cap (\{0, 1\}^{m-1} a \{0, 1\}^*); \end{aligned}$$

$$\begin{aligned} T &= Br(T_1 \cdot m \cdot a \cdot x), \\ \{vI(x)w \mid vaw \in I(T_1) \wedge |v| = m - 1\}; \end{aligned}$$

$$\begin{aligned} T &= Bl(T_1 \cdot m \cdot a \cdot b), \\ \{vbw \mid vaw \in I(T_1) \wedge |v| = m - 1\}; \end{aligned}$$

$$\begin{aligned} x &\in T, \\ I(x) &\in I(T); \end{aligned}$$

$$\begin{aligned} T &= \emptyset, \\ I(T) &= \emptyset, \end{aligned}$$

$$\begin{aligned} T_1 &\subseteq T_2, \\ I(T_1) &\subseteq I(T_2). \end{aligned}$$

We can use our computing units independently or add them to this computing units. Depending on experimental conditions using the same computing units we can obtain essentially different programming languages. For example, if we allow unrestricted appliance of operations  $F[P](x)$  and  $G[S, P](x)$ , then we can consider the following semigroup as a model of computations:

$$\langle A, U, C, G \mid S \rangle.$$

Note that

$$\begin{aligned} UAA &= UACA = UAC = UA, \\ UGA &= UGGA = UGG = UG, \\ UAG &= UACG = UAC = UA. \end{aligned}$$

Therefore,

$$\begin{aligned} \langle A, U, C, G \mid S \rangle = \\ \{U, A, C, G, \\ UU, UA, UC, UG, \\ AU, AA, AC, AG, \\ CU, CA, CC, CG, \\ GU, GA, GC, GG\}. \end{aligned}$$

From other hand, using restricted appliance of operations  $F[P](x)$  and  $G[S, P](x)$ , we can easily obtain a semigroup with undecidable word problem.

Note that for tRNA programming languages we have only set and string variables and constants. This is a characteristic feature of all programming languages for DNA computing. In the case of DNA computing, we have significant difficulties with numerical operations and numbers themselves. But “difficult” does not mean “impossible”. For example, suppose that we have a binary register

$$a_1 a_2 a_3 a_4$$

where  $a_1, a_2, a_3, a_4 \in \{0, 1\}$ . Assume that we want to define some bit operations. We can emulate this binary register the following word:

$$GGA_1 GGA_2 GGA_3 GGA_4$$

where

$$A_i = \begin{cases} G, & a_i = 0 \\ C, & a_i = 1 \end{cases}$$

Let

$$G_{0,1,1}[S_1, P] : GGCx[1]x[2]x[3] \rightarrow GGx[1]x[2]x[3],$$

$$G_{0,1,2}[S_2, P] : GGx[1]x[2]x[3] \rightarrow GGGx[1]x[2]x[3],$$

$$G_{0,2,1}[S_1, P] : x[1]GGCx[2]x[3] \rightarrow x[1]GGx[2]x[3],$$

$$G_{0,2,2}[S_2, P] : x[1]GGx[2]x[3] \rightarrow x[1]GGGx[2]x[3],$$

$$G_{0,3,1}[S_1, P] : x[1]x[2]GGCx[3] \rightarrow x[1]x[2]GGx[3],$$

$$G_{0,3,2}[S_2, P] : x[1]x[2]GGx[3] \rightarrow x[1]x[2]GGGx[3],$$

$$G_{0,4,1}[S_1, P] : x[1]x[2]x[3]GGC \rightarrow x[1]x[2]x[3]GG,$$

$$G_{0,4,2}[S_2, P] : x[1]x[2]x[3]GG \rightarrow x[1]x[2]x[3]GGG,$$

$$G_{1,1,1}[S_3, P] : GGGx[1]x[2]x[3] \rightarrow GGx[1]x[2]x[3],$$

$$G_{1,1,2}[S_4, P] : GGx[1]x[2]x[3] \rightarrow GGCx[1]x[2]x[3],$$

$$G_{1,2,1}[S_3, P] : x[1]GGGx[2]x[3] \rightarrow x[1]GGx[2]x[3],$$

$$G_{1,2,2}[S_4, P] : x[1]GGx[2]x[3] \rightarrow x[1]GGCx[2]x[3],$$

$$G_{1,3,1}[S_3, P] : x[1]x[2]GGGx[3] \rightarrow x[1]x[2]GGx[3],$$

$$G_{1,3,2}[S_4, P] : x[1]x[2]GGx[3] \rightarrow x[1]x[2]GGCx[3],$$

$$G_{1,4,1}[S_3, P] : x[1]x[2]x[3]GGG \rightarrow x[1]x[2]x[3]GG,$$

$$G_{1,4,2}[S_4, P] : x[1]x[2]x[3]GG \rightarrow x[1]x[2]x[3]GGC,$$

$$x[i] \in P = \{GGG, GGC\},$$

$$S_1 = \{GGC \rightarrow GG\}, S_2 = \{GG \rightarrow GGG\},$$

$$S_3 = \{GGG \rightarrow GG\}, S_4 = \{GG \rightarrow GGC\}.$$

Let

$$B_{i,j}(x) \rightleftharpoons G_{i,j,2}(G_{i,j,1}(x)).$$

It is easy to check that using  $B_{i,j}$  we can obtain arbitrary bit operations.

## IV. CONCLUSION

In this paper we consider some new computing units which can be used in different programming languages for DNA-based computers.

As the main direction of further research we can mention the rigorous formalization and classification of programming languages based on tRNA computing units and the study of computational power of such programming languages.

## ACKNOWLEDGMENT

The work was partially supported by Grant of President of the Russian Federation MD-1687.2008.9 and Analytical Departmental Program “Developing the scientific potential of high school” 2.1.1/1775.

## REFERENCES

- [1] D. Deutsch. *Quantum theory, the Church-Turing principle and the universal quantum computer*, Proceedings of the Royal Society London, 1985, Vol. A400, 97–117.
- [2] P. W. Shor. *Algorithms for quantum computation: Discrete logarithms and factoring*, Proceedings of the 35th IEEE Symposium on Foundations of Computer Science, 1994. pp.124–134.
- [3] P. Pudlák. *Complexity theory and genetics*, Proceedings of 9th Conference on Structure in Complexity Theory, 1994. pp.183–195.
- [4] L. M. Adleman. *Molecular computation of solutions to combinatorial problems*, Science, 1994, Vol. 266, 1021–1024.
- [5] R. J. Lipton. *Speeding up computations via molecular biology*, Technical report, Princeton University, 1994.
- [6] R. J. Lipton. *Using DNA to solve NP-complete problems*, Technical report, Princeton University, 1995.
- [7] R. J. Lipton. *DNA solution of hard computational problem*, Science, 1995, Vol. 268, 542–545.
- [8] P. W. K. Rothmund. *A DNA and restriction enzyme implementation of Turing machines*, R.J.Lipton and E.B.Baum, editors. DNA Based Computers: Proceedings of the DIMACS Workshop, Princeton University, Providence, Rhode Island, 1996. pp.75–119.
- [9] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, E. Shapiro. *Programmable and autonomous computing machine made of biomolecules*, Nature, 2001, Vol. 414, 430–434.
- [10] Y. Benenson, R. Adar, T. Paz-Elizur, E. Keinan, Z. Livneh, E. Shapiro. *DNA molecule provides a computing machine with both data and fuel*, Proceedings of the National Academy of Sciences of the United States of America, 2003, Vol. 100, 2191–2196.
- [11] Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, E. Shapiro. *An autonomous molecular computer for logical control of gene expression*, Nature, 2004, Vol. 429, 423–442.
- [12] T. H. LaBen, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J. H. Reif, N. C. Seeman. *The Construction of DNA Triple Crossover Molecules*, Journal of the American Chemical Society, 2000, Vol. 122, 1848–1860.
- [13] H. Li, S. H. Park, J. H. Reif, T. H. LaBean, H. Yan. *DNA-Templated Self-Assembly of Protein and Nanoparticle Linear Arrays*, Journal of American Chemistry Society, 2004, Vol. 126, 418–419.
- [14] S. H. Park, H. Yan, J. H. Reif, T. H. LaBean, G. Finkelstein. *Electronic nanostructures templated on self-assembled DNA scaffolds*, Nanotechnology, 2004, Vol. 15, 525–527.
- [15] N. C. Seeman. *Nucleic Acid Junctions and Lattices*, Journal of Theoretical Biology, 1982, Vol. 99, 237–247.
- [16] A. T. Winfree. *The Geometry of Biological Time*, Springer-Verlag, Berlin, 2000.
- [17] H. Yan, T. H. LaBean, L. Feng, J. H. Reif. *Directed Nucleation Assembly of Barcode Patterned DNA Lattices*, Proceedings of the National Academy of Science of the United States of America, 2003, Vol. 100, 8103–8108.
- [18] H. Yan, S. H. Park, G. Finkelstein, J. H. Reif, T. H. LaBean. *DNA-Templated Self-Assembly of Protein Arrays and Highly Conductive Nanowires*, Science, 2003, Vol. 301, 1882–1884.
- [19] P. Yin, A. J. Turberfield, J. H. Reif. *Design of an Autonomous DNA Nanomechanical Device Capable of Universal Computation and Universal Translational Motion*, Tenth International Meeting on DNA Based Computers. LNCS 3384, Springer-Verlag, New York, 2005. pp.426–444.

- [20] M. N. Stojanovic, T. H. E. Mitchel, D. Stefanovic. *Deoxyribozyme-based Logic Gates*, Journal of American Chemistry Society, 2002, Vol. 124, 3555–3561.
- [21] M. N. Stojanovic, P. de Prada, D. W. Landry. *Homogeneous assays based on deoxyribozyme catalysis*, Nucleic Acids Reserch, 2000, Vol. 28, 2915.
- [22] M. N. Stojanovic, D. Stefanovic. *A deoxyribozyme-based Molecular Automaton*, Nature Biotechnology, 2003, Vol. 21, 1069.
- [23] M. N. Stojanovic, D. Stefanovic. *Deoxyribozyme-based Half-Adder*, Journal of American Chemistry Society, 2003, Vol. 125, 6673.
- [24] <http://bdcc.kmpn.net/htmls/dnacomputer/index.php>
- [25] <http://bi.snu.ac.kr/biocomputers2004>
- [26] G. Păun. *Membrane Computing: An Introduction*, Springer-Verlag, Berlin, 2002.
- [27] G. Păun. *Membrane computing: Main ideas, basic results, applications*, Molecular Computational Models: Unconventional Approaches (M. Gheorghe, ed.), Idea Group Publ., London, 2004. pp.1–31.
- [28] G. Felsenfeld, G. Cantoni. *Use of thermal denaturation studies to investigate the base sequence of yeast serine sRNA*, Proceedings of the National Academy of Science of the United States of America, 1964, Vol. 51, 818–826.
- [29] H. Lodish, A. Berk, P. Matsudaira, C. A. Kaiser, M. Krieger, M. P. Scott, S. L. Zipursky, J. Darnell. *Molecular Biology of the Cell*, W.H. Freeman, New York, 2004.
- [30] M. D. Friede. *Design of artificial tRNAs*, Dissertation zur Erlangung des akademischen Grades Doctor rerum naturalium, Vorgelegt der Form- und Naturwissenschaftlichen Fakultät der Universität Wien, Wien, 2001.
- [31] G. Keith and G. Dirheimer. *Evidence for the existence of an expressed minor variant tRNA<sup>phe</sup> in yeast*, Biochemical and Biophysical Research Communications, 142:183–187, 1987.
- [32] J. Weissenbach, I. Kiraly, and G. Dirheimer. *Primary structure of tRNA<sup>thr</sup> 1a and b from brewer's yeast*, Biochimie, 59:381–391, 1977.
- [33] M. Sprinzl, C. Horn, M. Brown, A. Ioudovitch, and S. Steinberg. *Compilation of tRNA sequences and sequences of tRNA genes*, Nucleic Acids Research, 26:148–153, 1998.
- [34] C. R. Woese, O. Kandler, and M. L. Wheelis. *Towards a natural system of organisms: proposal for the domains archaea, bacteria and eucarya*, Proceedings of the National Academy of Sciences of the United States of America, 87:4576–4579, 1990.
- [35] G. R. Bjork, J. M. Durand, T. G. Hagervall, R. Leipuviene, H. K. Lundgren, K. Nilsson, P. Chen, Q. Qian, and J. Urbonavicius. *Transfer RNA modification: in on translational frameshifting and metabolism*, FEBS Letters, 452:47–51, 1999.
- [36] C. G. Edmonds, P. F. Crain, R. Gupta, T. Hashizume, C. H. Hocart, J. A. Kowalak, S. C. Pomerantz, K. O. Stetter, and J. A. McCloskey. *Post-transcriptional modification of tRNA in thermophilic archae (archaeobacteria)*, Journal of Bacteriology, 173:3138–3148, 1991.
- [37] D. A. Melton, E. M. de Robertis, and R. Cortese. *Order and intracellular location of the events involved in the maturation of a spliced tRNA*, Nature, 284:143–148, 1980.
- [38] K. Nishikura and E. M. De Robertis. *RNA processing in microinjected xenopus oocytes. Sequential addition of base modifications in the spliced transfer RNA*, Journal of Molecular Biology, 145:405–420, 1981.
- [39] B. Masquida and E. Westhof. *On the wobble G-U and related pairs*, RNA, 6:9–15, 2000.
- [40] R. W. Holley. *Structure of an alanine transfer ribonucleic acid*, JAMA, 194:868–871, 1965.
- [41] N. B. Leontis and E. Westhof. *Conserved geometrical base-pairing patterns in RNA*, Quarterly Review of Biophysics, 31:399–455, 1998.
- [42] G. A. Burley, J. Gierlich, M. R. Mofid, H. Nir, S. Tal, Y. Eichen, and T. Carell. *Directed DNA Metallization*, Journal of the American Chemical Society, 128(5):1398–1399, 2006.
- [43] A. S. Swami, N. Brun, and D. Langevin. *Phase Transfer of Gold Metallized DNA*, Journal of Cluster Science, 20(2):281–290, 2009.
- [44] E. A. Burakova and V. N. Silnikov. *Molecular Design of Artificial Ribonucleases Using Electrostatic Interaction*, Nucleosides, Nucleotides and Nucleic Acids, 23(6-7):915–920, 2004.
- [45] D. Rooß and K. W. Wagner. *On the Power of DNA-Computing*, Information and Computation, 131(2)95–109, 1996.