Sent to Cybernetics and Systems: An International Journal [1]
My draft
Preprint

THE INFERENCE BASED ON MOLECULAR COMPUTING

PIOTR WĄSIEWICZ, TOMASZ JANCZAK, JAN J. MULAWKA
Institute of Electronic Systems, Warsaw University of Technology, Warsaw, Poland

ANDRZEJ PŁUCIENNICZAK
Institute of Biotechnology and Antibiotics, Warsaw, Poland

Molecular computing is a new paradigm to perform calculations using nanotechnology. This paper presents the overall research direction from which molecular inference and expert systems are emerging. It introduces the subject matter and a general description of the problems involved. This includes selected methods of knowledge representation by DNA oligonucleotides, strategies of the inference mechanism, concept of the inference engine based on circular DNA molecules, particularly derived from plasmids, practical experience in DNA inference engine implementation, and discussion of the experimental results. The approach allows evaluating logical statements and drawing inferences for generating other statements via DNA computing. Series of experiments have been conducted to confirm practical utility of this approach. In these experiments, parameters of biochemical reactions were varied to determine truth/false recognition accuracy. In addition, we discuss the fundamental issues of inference engine and try to enhance physical insight into the dominating features of the approach proposed.

[1]This work was supported by the Polish Committee for Scientific Research, through the KBN grant 8T11F00816.
Address correspondence to Piotr Wąsiewicz, Institute of Electronic Systems, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland, E-mail: pwas@ieee.org

In recent years, progress in silicon technology of integrated networks has come to a standstill, as fundamental limits are being reached. This fact, however, provided opportunity for alternative devices and stimulated development of new techniques to perform computations in more effective way (Mulawka et al., 1998b). Seminal work of Adleman (1994) made a breakthrough in this area demonstrating that biochemical molecular experiments on DNA may be used in solving a number of complex problems (Mulawka, 1997). Subsequent proposals have continued to use this paradigm of computation (Baum, 1996; Kolata et al., 1995). Papers on evolutionary computation (Goldberg, 1989; Koza, 1992; Wąsiewicz et al., 1997a; Wąsiewicz & Mulawka, 1997b; Wąsiewicz & Klebus, 1997c) in connection with DNA computing (Dassen, 1999; Mulawka & Wasiewicz, 1998e; Mulawka et al., 1999c; Paun et al., 1998) have been written (Mulawka et al., 1999b; Stanczak et al., 1999), also a concept of molecular computer (Biswas, 1993; Hill et al., 1994; Pederson, 1989) and its new architectures has been envisioned (Gehani & Reif, 1998; Heath et al., 1999; Lipton, 1995; Malone, 1995; McCaskill et al., 1997b; Montemerlo et al., 1996, 1997; Murphy et al., 1997; Wąsiewicz et al., 1999c).

Simultaneously these advances heralded the rapid acceleration of a new field known as moletronics, or molecular electronics. Ionic and covalent bounded 'artificial molecules' have been discussed. Based on these facts a concept of quantum computer has appeared (Ashoori, 1996; Beth, 1997). Recently, significant progress has been achieved in calculations based on rotaxanes (Collier et al., 1999). Crude molecular computer components have been created. Molecular approach - if implemented - opens a new opportunity in computing and signal processing (Mulawka & Nowak, 1999d), but its potential has not been fully explored as yet (Jagielska et al., 1998).

In DNA computing, information is stored in molecules that are linear polymers composed of nucleotides. DNA molecules are composed of single or double DNA fragments and often called strings, primers, oligonucleotides or oligos, and rarely strands. A single-stranded primer has a phospho-sugar backbone and four bases denoted by the symbols A, T, C, and G. A double-stranded oligo may be formed of two single primers due to the hybridization reaction, because A is complementary with T and C is complementary with G (Węgleński, 1995). Since DNA strings are composed of four nucleotides, from informatic point of view, they represent chains of symbols over a four-letter alphabet. Therefore, DNA computing is adequate for processing symbols and logical structures that has been reported in a number of publications (Adleman et al., 1996; Amos, 1997b; Amos & Dunne, 1997a; Gupta et al., 1997; Leete et al., 1997; Liu et al., 1998; Mulawka et al., 1998c,d, 1999a; Ogihara & Ray, 1996, 1998; Roweis et al., 1998; Wąsiewicz et al., 1999a,b; Winfree et al., 1996).

Identification of problems that can be solved by molecular computing more efficiently than on classical electronic machines would contribute to the assessment of the efforts put into the development of this approach and obviously need further study. Question arises of which other areas of computer science may be stimulated by this paradigm. Artificial intelligence (Freund et al., 1997; Mihalache, 1997; Rooß et al., 1996; Sakamoto et al., 1998) and especially expert systems made by Mulawka et al. (1998a) seem to be promising for such research.

It has been observed that expert systems would mimic more precisely intellectual ability of people if they were based on associative memories. Human memory operates in an associative manner; a portion of recollection can produce a larger related memory. One thing reminds us of one, and that one of another. If we allow our thoughts to wander, they move from topic to topic based on a chain

of mental associations. Such chains of associations may be formed in biochemical DNA reactions (Ausubel & Struhl, 1995; Sambrook et al., 1989) by putting molecules together.

Pursuing "Adleman-experiment" Baum (1995) envisioned realization of associative DNA memory of the larger capacity than human memory. By contrast, classical expert systems are based on ordinary computer memories, which are location addressable. To retrieve a datum an address is applied and the information occupying that address is returned. Thus, to find adequate data, contents of particular memory cells must be tested if matching is achieved. Such process is cumbersome and takes long time for large sets of data.

In this paper it will be shown that a concept of molecular computing based on DNA may be applied in the area of expert systems, and it will be demonstrated that using DNA reactions an inference engine may be implemented in a straightforward manner.

SYMBOLIC DESCRIPTION OF THE DNA STRANDS

As has been mentioned, symbolic computation is well suited for implementation by means of DNA strings and genetic engineering methodology (Csuhaj et al., 1996; Head, 1987; Head et al., 1997; Li, 1998; Pisanti, 1997).

Examples given below show our *DNA* notation created in order to simplify descriptions of our experiments. One letter marks one oligo as is seen in Figure 1A. In our systems, the upper oligo usually presents data and lower (underlined) - logic devices, rules and so on. Therefore, they should not be exchanged. Symbols $\lceil$ and $\rfloor$ denote position of single strings in space. A corner of such symbol represents 5' end, while an open edge represents 3' end. An underlined primer is always in the same position (3' on the left, 5' on the right) even in other double-stranded oligos.

$$\begin{array}{l} 5'\ \boxed{\texttt{CTAGAGAGGATGAGAGGTCATGCATCTCTTCACAC}}\ 3' \\ 3'\ \boxed{\texttt{GATCTCTCCTACTCTCCAGTACGTAGAGAAGTGTG}}\ 5' \end{array} \quad = \left\lceil \begin{array}{c} a \\ \underline{b} \end{array} \right\rfloor$$

$$\begin{array}{l} 5'\ \boxed{\texttt{CTAGAGAGGATGAGAGGTCATGCA}}\ 3' \\ 3'\ \boxed{\texttt{GATCTCTCCTACTCTCCAGTACGTAGAGAAGTGTG}}\ 5' \end{array} \quad = \left\lceil \begin{array}{cc} a_1 & \\ \underline{b} & \underline{+} \end{array} \right\rfloor = \left\lceil \begin{array}{cc} a_1 & + \\ \underline{b} & \underline{-} \end{array} \right\rfloor = \left\lceil \begin{array}{cc} a_1 & + \\ \underline{-} & \underline{b} \end{array} \right\rfloor$$

$$\begin{array}{l} 5'\ \boxed{\texttt{CTAGAGAG}}\ 3' \\ 3'\ \boxed{\texttt{GATCTCTCCTACTCTCCAGTACGTAGAGAAGTGTG}}\ 5' \end{array} \quad = \left\lceil \begin{array}{cc} a_2 & \\ \underline{b} & \underline{-} \end{array} \right\rfloor = \left\lceil \begin{array}{cc} a_2 & \\ \underline{-} & \underline{b} \end{array} \right\rfloor$$

$$\begin{array}{l} 5'\ \boxed{\texttt{CTAGAGAGGATGAGAGGTCAT}}\ 3' \\ 3'\ \boxed{\texttt{GATCTCTCCTACTCTCCAGTA}}\ 5' \end{array} \quad = \left\lceil \begin{array}{c} a_3 \\ \underline{b_1} \end{array} \right\rfloor = \left\lceil \begin{array}{ccc} - & a_3 & - \\ \underline{-} & \underline{b_1} & \underline{-} \end{array} \right\rfloor = \left\lceil \begin{array}{c} a_3 \\ \underline{\tilde{a_3}} \end{array} \right\rfloor$$

Figure 1: Analytical representation of DNA oligos.

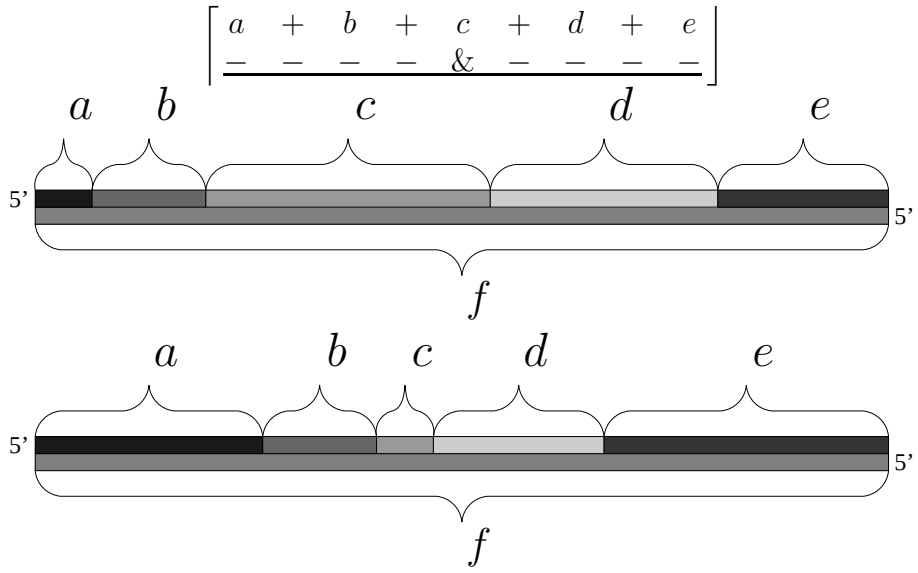

Figure 2: The exemplary double-stranded oligos described in the above equation.

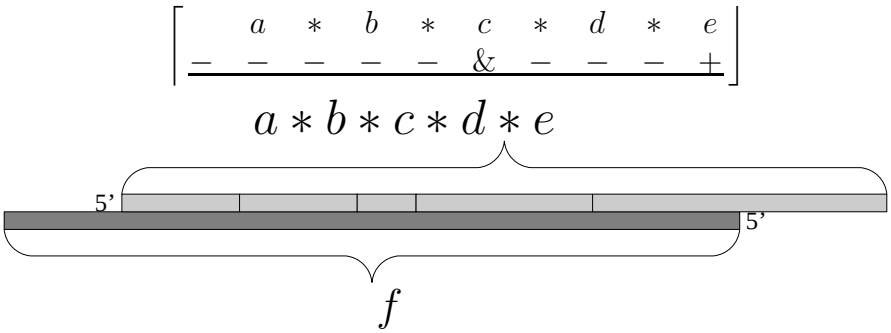

Figure 3: The exemplary double-stranded strings with sticky ends.

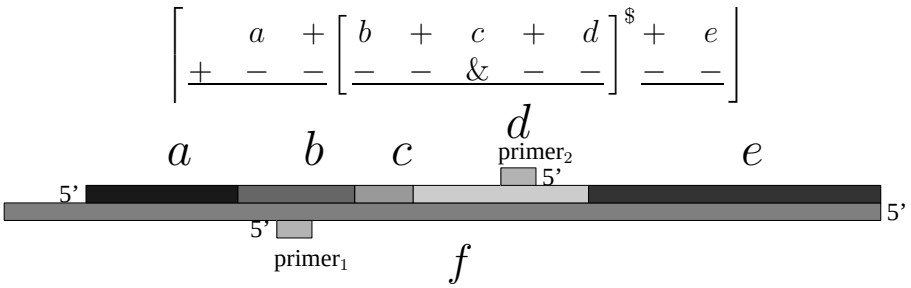

Figure 4: The exemplary double-stranded oligos with the blunt end.

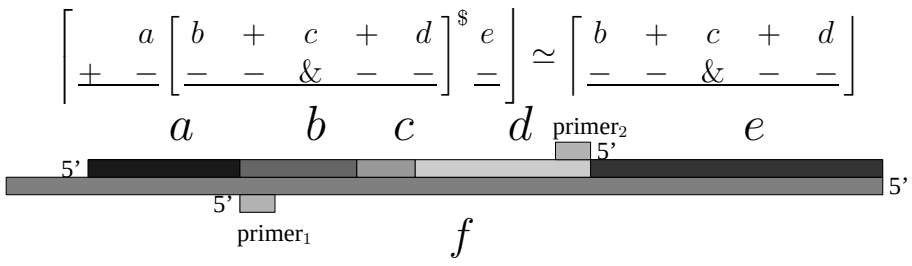

Figure 5: Another representation of the double-stranded strings before amplification.

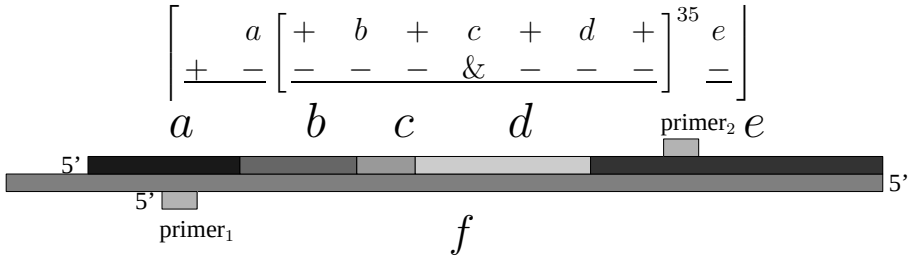

Figure 6: The exemplary double-stranded oligos before 35 cloning cycles of PCR.

In Figure 1B, a sign $+$ at the right side of b describes a sticky end of b shorter than the nearest complementary oligo a. In Figure 1C, a sign $-$ at the right side of b describes a sticky end of b longer than the nearest complementary string a. In Figure1D, the same signs $-$ at the same sides of complementary strings a and b mean that these strings form a double stranded oligo with blunt ends. As is seen, signs can be omitted or exchange by the pair of $+$. If complementary strings have a same letter, e.g., a, then in order to distinguish them, a sign tilde ($\tilde{}$) is added to the one. "Halves" of strings are denoted by letters r (a right string part) or l (a left string part) in upper, right indices of these strings, e.g., $a = a^l * a^r$.

In Figure 2 - 6, examples of notation in different cases are explained. Note that the sign $+$ may be additionally applied to mark a symbolic disjunction between two hybridized primers, and the sign $*$ to denote concatenation of strings (after hybridization and ligation) and the sign $-$ to lengthen a string, e.g., $\&$, of course, only in the equations, not in real experiments. Polymerase Chain Reaction (*PCR*) may be used to lengthen an oligo by a length of a sticky end of a primer. A *DNA* oligo has its length. In our experiments, its length will be written in an upper right index of an oligo letter, e.g., a^{345}, and ligated *DNA* strings with letters in brackets $|$ and $|$, e.g., $|bcd|^{123}$, $|ab|^{67}$.

In Figure 4-6, *PCR* process description is presented. The 5' ends of both primers are like $[$ and $]$ brackets and an unknown (or not so important to be written) number of *PCR* cycles is defined by a symbol $\$$. In Figure 5, both 5' ends of primers terminate exactly above 3' ends of oligos a and d. Thus, brackets $[$ and $]$ are just between oligo letters. Now the number of PCR cycles is equal to 35. After *PCR* reaction in a vessel, there are millions of amplified molecules and very small amounts of others. Therefore, a sign $\simeq$ is also utilized in Figure 5. As is seen, oligos are amplified from 5' end of the primer number 1 to 5' end of the primer number 2.

THE INFERENCE MECHANISMS

Knowledge representation plays an important role in inference systems (Mulawka, 1996), where the knowledge should be formalized and structured. One of the methods that can support knowledge structuring is known as production rules. Such rules are also referred to as IF-THEN rules. An alternative designation for IF-THEN rules is that of condition-action or premise-conclusion statements. There may be several premise statements within a single rule. Among rules we can distinguish those ones whose conclusions are not final for the inference system. Such rules are called indirect rules. In our approach the rules will be represented by DNA molecules, which are described in the next point.

Another part of the knowledge base constitutes facts. As a fact, we consider a statement to be valid. The fact exists in our inference system if adequate DNA oligo representing it is included in the knowledge base (Baum, 1995). If the premise of a rule is satisfied based on the facts and its conclusion part has been implemented, the rule is said to fire. In traditional expert systems, the knowledge base not only stores rules for processing knowledge and individual facts, it also includes complex objects, their attributes, relationships between objects, and rules for deriving new knowledge from existing knowledge, i.e., heuristics. However, in our approach the knowledge base is simplified to rules and facts only.

The inference mechanism, in general, is a part of an expert system that draws inferences from a knowledge base according to a fixed problem-solving method. The functions of the inference mechanism in classical systems include controlling the actions between the individual parts of the expert system, determining the time for and the type of rule processing, controlling the dialogue with the users. In simple case, the rules may be graphically represented by so-called inference networks. These networks comprise assertions and intermediate conclusions which may be combined by logical connectives, specifically AND or OR operators. The inference mechanism can often be made clearer by means of the inference network.

A method used by the inference mechanism in problem solving is called inference strategy. Three main strategies may be distinguished: forward chaining, backward chaining, and mixed chaining. For example, in backward chaining all rules in the knowledge base leading to the hypothesis are selected, followed by a check to determine whether applicable rules exist that can be used for satisfying the conclusions. The same procedure is used for the conditions of these rules. On the contrary, in forward chaining the knowledge base is searched for the rules associated with known facts and the action part of these rules is executed until the solution is reached or no more rules can be applied.

The ordinary premise may be denoted by the symbol S_k for $k \in \{1, K\}$ (K - a number of such premises), and their set by $\mathbf{S}$. The ordinary conclusion (always as the first premise of the next rule) is depicted by the symbol Z_l for $l \in \{1, L\}$ (L - a number of all such conclusions), and their set by $\mathbf{Z}$. The symbol $\sum$ signifies hybridization, and $\prod$ - concatenation for $S_k \in \mathbf{S}$ and $Z_l \in \mathbf{Z}$. The rule $S_k^1 \wedge S_k^2 \wedge \ldots \wedge S_k^M \Rightarrow Z_l$ is represented by connected oligonucleotides creating linear form:

$$R =< \sum_{m=1}^{m=M} S_k^m Z_l| =< S_k^1 \ldots S_k^M Z_l|.$$

Premises and conclusions are just DNA strings with orientation $3' \rightarrow 5'$ and are connected in complementary manner with their "halves" to DNA fragments s_h for $h \in \mathbf{N}$:

$$\hat{R} = \left[\begin{array}{cccccccccc} + & s_h & + & s_{h+1} & \ldots & s_{h-1+M} & + & s_{h+M} & + \\ \underline{S_k^1} & + & S_k^2 & + & \ldots & + & S_k^M & + & \underline{Z_l} \end{array} \right]$$

The premise S_k^1 is the first in the rule number k. The next premises are with indices m up to the conclusion Z_l at the end of the rule string. Indices of strings s are not connected with indices of premises and conclusions. Hybridized with the help of these strings premises and conclusions can be concatenated.

The given in the work of Mulawka et al. (1998a) method called "modus ponens" has been applied. Molecules representing rules create inference paths during the inference process consisting of the following operations: hybridization, concatenation, and resulting string detection. These paths are truth paths for hypotheses (last conclusions in the paths). Conclusions Z_l of first rules become first premises S_k^1 of second rules creating the complete reasoning path, which can be represented by the sequence of DNA strings:

$$R_1 \cdot R_W = \sum_{w=1}^{w=W} R_w = R_1 + R_2 + \ldots + R_W =< S_k^1 \sum_{w=1}^{w=W} \left\{ \sum_{m=2}^{m=M_w} S_{k,w}^m + Z_{l,w} \right\} |$$

$$\stackrel{*}{\Rightarrow} < S_k^1 \prod_{w=1}^{w=W} \left\{ \prod_{m=2}^{m=M_w} S_{k,w}^m * Z_{l,w} \right\} | = \prod_{w=1}^{w=W} R_w = R_1 * R_2 * \ldots * R_W$$

where W is a number of rules in the path, and M_w - a number of premises in the rule number w. The orientation $3' \rightarrow 5'$ is denoted by signs: $<$ and $|$. The sign $\stackrel{*}{\Rightarrow}$ means execution of concatenation process. Next, premises S_k (from the second one to the last one number M_w) are basic facts known at the beginning of inference process. If the fact has a value equal to *TRUTH*, than its representing string is present in the reaction vessel. To distinguish between premises of different rules, the ordinal number of rule R_w is written after a comma in the lower, right index of the premise: $S_{k,w}^m$, the conclusion: $Z_{l,w}$, and the complementary string: $s_{h,w}$.

The truth path in the inference process from the first premise $S_{k,1}$ of the first rule to the conclusion $Z_{l,W}$ of the last rule number W is described by the expression $< S_{k,1} \cdot Z_{l,W}|$. Such paths has their own orientation, because they consist of premises and conclusions or complementary to premises and conclusions strings s_h. In the last case this complementary path is described by an expression $|\widetilde{S_{k,1}} \cdot \widetilde{Z_{l,W}} >$. This is a need to detect the whole path $< S_{k,1} \cdot Z_{l,W}|$ or her complementary string $|\widetilde{S_{k,1}} \cdot \widetilde{Z_{l,W}} >$ in order to prove the final hypothesis $Z_{l,W}$. Paths built in the process of hybridization and concatenation from DNA molecules are detected by their length in the process of electrophoresis. If the DNA fragments called DBFs: $|\widetilde{Z_{l,w}^p} \widetilde{S_{k,1}^l} >$ are added to the reaction vessel, then circular molecules are created from the whole paths. DNA basic fragments (DBFs) connect starting premises with final hypotheses.

The decision tree, also called the tree of intermediate rules, has one node called the root $S_{k,1}^1$ and

the rest of its nodes $Z_{l,w} = S^1_{k,v}$ for $v > w$ has exactly one node-predecessor $S^1_{k,w}$ and may have several node-successors. Nodes without successors are final nodes called leaves. The path from the root (the starting fact) to the leaf (final fact - hypothesis) is just named the inference path.

It is the first step to synthesize DNA strings representing premises, conclusions, and DNA fragments called DBFs. The algorithm of the inference process looks like this:

1. Initiation. First all synthesized DNA strings are added to the reaction vessel;
2. $\mathrm{h}eat\uparrow$; - mixing
3. $\mathrm{h}eat\downarrow$; - hybridization
4. Concatenation of hybridized DNA strings;
5. Detection of complete inference paths and checking truth of ending conlusions - hypotheses.

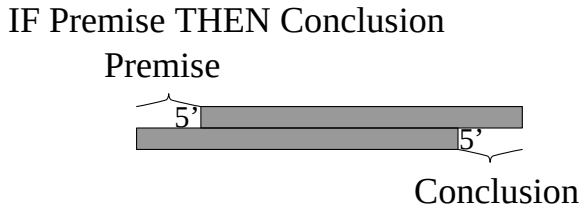

Figure 7: The indirect rule.

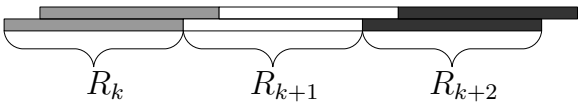

Figure 8: The indirect rules concatenation.

IMPLEMENTATION OF EXEMPLARY INFERENCE SYSTEM BASED ON CIRCULAR DNA MOLECULES

In our approach, the idea of molecular inference system has been developed. Consider rule base comprising a set of indirect rules. All these rules, with one premise and one conclusion, are converted to double-stranded oligos with sticky ends. One of them is depicted in Figure 7. Such a molecule is made of three sectors. The first single-stranded one encodes a given premise, the third single-stranded - a given conclusion and the second double-stranded sector connects both previous. Two rules can hybridize on that condition that the conclusion sector of one rule is complementary to the premise sector of the second rule. Sticky ends of DNA fragments anneal to each other in the way depicted in Figure 8.

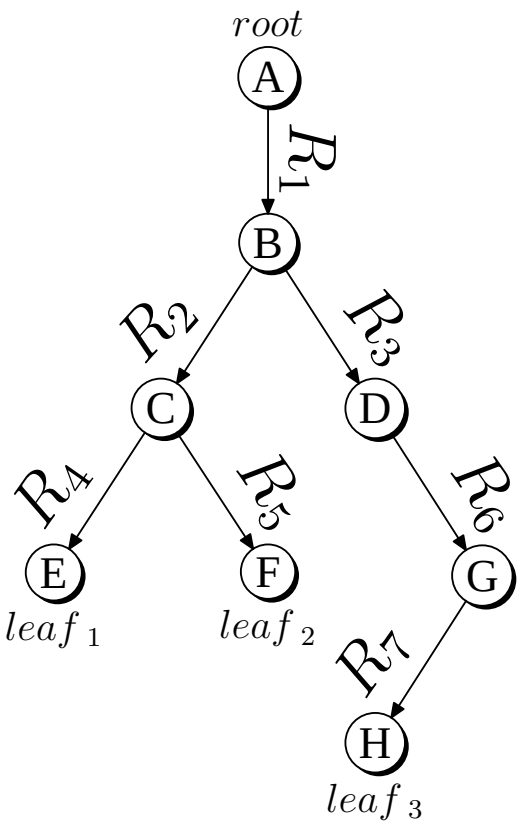

Figure 9: The complete rule tree.

A set of indirect rules may be interpreted as a rule tree. Indirect rules create edges of the rule tree. In our approach this tree is implemented with the set of representing indirect rules molecules. In nodes of the tree, a process of indirect rules self-hybridization is performed. In Figure 9, for example, seven rules as edges of a graph have created the complete rule tree with a root at the top and leaves at the bottom.

In our method, the knowledge base is represented by a water solution of DNA molecules for particular indirect rules. There can be different sets of DNA fragments in a vessel. If one or more rules are absent, then a tree is incomplete as depicted in Figure 10, where it is without edges: R_2 and R_6 between adequate nodes representing premises. Thus, molecules describing these edges are absent in water solution during biochemical operations.

In order to retrieve information about fired rules, we proceed as follows. We add DBFs to our test tube. The structure of DBF is similar to the rule structure with this exception that the first single-stranded part is complementary to the ending conclusion represented by a given leaf of the decision tree and the second single-stranded part to the first rule - the tree root. So each of the DBFs can hybridize like any rule, but only with the root and the leaves. In such a way after their hybridization, the DBFs change a complete inference paths into circular DNA strings as shown in Figure 11. There is also a possible detection of ending conclusions with DNA chips.

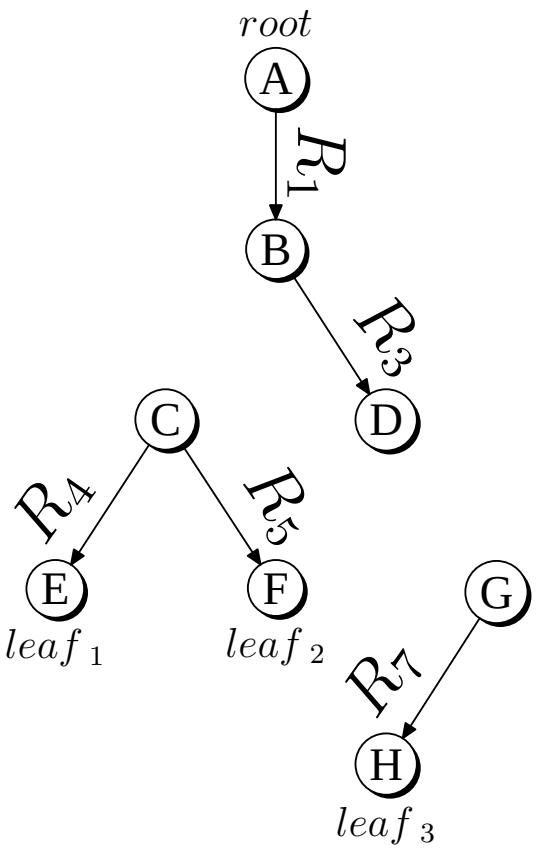

Figure 10: The incomplete rule tree.

In our case as initial facts, we prepare water solution with a complete set of the DBFs, which anneal to particular leaves. The inference mechanism depends on these fragments. Representing rules, molecules are added to our test tube solution. As a result circular molecules can be formed. Thus, for M leaves M circular molecules may be created. If they are detected with use of PCR and primers, some inference is acknowledged. As explained in Figure 11, the primer number 3 can be used to detect an adequate leaf conclusion.

To get more familiar with our approach the simplified inference system has been constructed and tested. As an example, consider a small inference system composed of three rules, which can be written as follows:

R_1: A =¿ B
R_2: B =¿ C
R_3: B =¿ D

Next, the set of proper molecules was prepared as depicted in Figure 12. This set may be represented by the graph as is provided in Figure 13, where respective links between nodes are implemented with DNA molecules. Complete sequences of the DBFs and their primers with places for

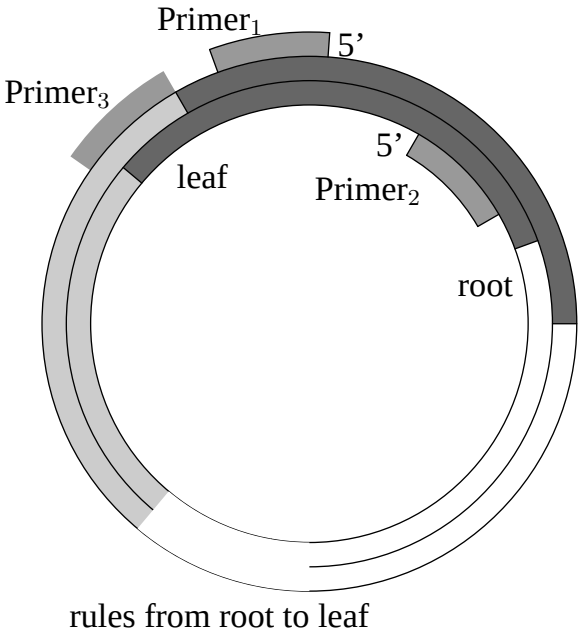

Figure 11: The circular molecule created from rules and one DBF.

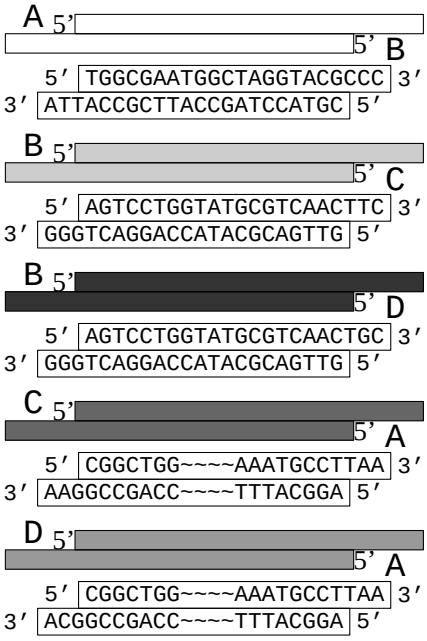

Figure 12: Molecular rules: $R_1, R_2, R_3,$ Molecular bases: B_1, B_2 (from top to bottom) and their sequences (aside).

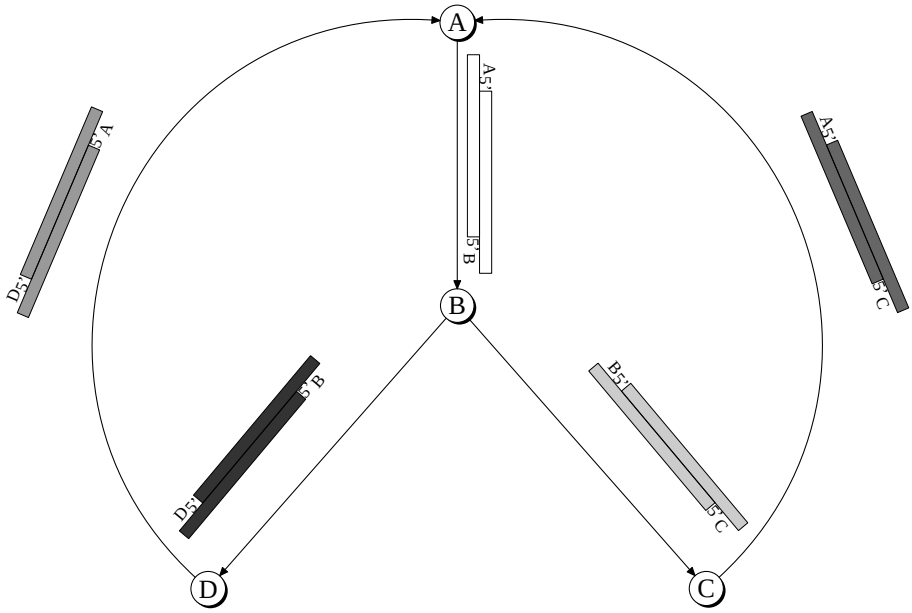

Figure 13: The graph of DNA inference system.

Primer

5′ CTCCAGATTTATCAGCAATA 3′

Primer

5′ GTCAGACCAAGTTTACTCATA 3′

DBFs: B or B

```
          5′ CGGCTGGCTGGTTTATTGCTGATAAATCTG 3′
3′ ACG or AAGGCCGACCGACCAAataacgactatttagac 5′
5′ GAGCCGGTGAGCGTGGGTCTCGCGGTATCATTGCAGCACT 3′
3′ ctcGGCCACTCGCACCCAGAGCGCCATAGTAACGTCGTGA 5′
5′ GGGGCCAGATGGTAAGCCCTCCCGTATCGTAGTTATCTAC 3′
3′ CCCCGGTCTACCATTCGGGAGGGCATAGCATCAATAGATG 5′
5′ ACGACGGGGAGTCAGGCAACTATGGATGAACGAAATAGAC 3′
3′ TGCTGCCCCTCAGTCCGTTGATACCTACTTGCTTTATCTG 5′
5′ AGATCGCTGAGATAGGTGCCTCACTGATTAAGCATTGGTA 3′
3′ TCTAGCGACTCTATCCACGGAGTGACTAATTCGTAACCAT 5′
5′ ACTgtcagaccaagtttactcataTATACTTTAGATTGAT 3′
3′ TGACAGTCTGGTTCAAATGAGTATATATGAAATCTAACTA 5′
5′ TTAAAACTTCATTTTTAATTTAAAAGGATCTAGGTGAAGA 3′
3′ AATTTTGAAGTAAAAATTAAATTTTCCTAGATCCACTTCT 5′
5′ TCCTTTTTGATAATCTCATGACCAAAATGCCTTAA 3′
3′ AGGAAAAACTATTAGAGTACTGGTTTTACGGA 5′
```

Figure 14: Primers for DBFs: $primer_1$, $primer_2$, DBFs: B_1, B_2 (from top to bottom).

Figure 15: Primers for pBR322: $primpbr_1$, $primpbr_2$, $primpbr_3$.

these primers, marked with small letters in the DBF code, are depicted in Figure 14. The sequences of our DBFs in Figure 12 are not complete. Their two variants were obtained by PCR amplification of pBR322 plasmid using primers, which sequences are shown in Figure 15. Sticky ends of the DBFs were obtained after digestion with the enzyme BglI.

Plasmids usually transport genes into a host cell, but this ability will be used in the future experiments. Circular molecules were multiplied with PCR and adequate primers as depicted in Figure 16 and 17. Two complete circular inference paths are shown there.

EXPERIMENTAL VERIFICATION

To verify the concept of simplified universal molecular reasoning engine some experiments were performed (Ausubel & Struhl, 1995; Sambrook et al., 1989). We constructed two double-stranded DNA fragments representing the DBFs B_1 and B_2. They are 305 base pairs long and they have 3 nucleotide sticky ends. One of the sticky ends is the same for B_1 and B_2: TAA. The second sticky end for B_1 is GAA and for B_2 is GCA. In fact, this is the only difference between fragments representing the DBFs. In addition primers $Primer_1$ and $Primer_2$ are complementary to some parts of these fragments. These primers are used in the PCR reaction that is used to amplify the amount of correctly formed circular oligos representing complete paths.

The interpretation of our computation is as follows: given fact A (represented by the TAA sticky end of B_1 or B_2) and a set of rules R (represented by short double-stranded DNA fragments with three nucleotide sticky ends at both sides) we can derive conclusion C or D (represented by the second sticky end of B_1 or B_2). In our small knowledge base there are only three rules described in Figure 12-13.

The computation consists of several steps:

- Preparation of fragments representing rules. Fragments representing rules were annealed (37^oC for 5 minutes) and phosphorylated by T4 polinucleotide kinase. Reactions were performed according to manufacturer recommendation written by Sambrook et al. (1989).

- Annealing. Correct circles are formed when B_1, R_1, R_2 or B_2, R_1, R_3 are present during annealing step. We call these test tubes B_1+ and B_2+, respectively. To check that our imple-

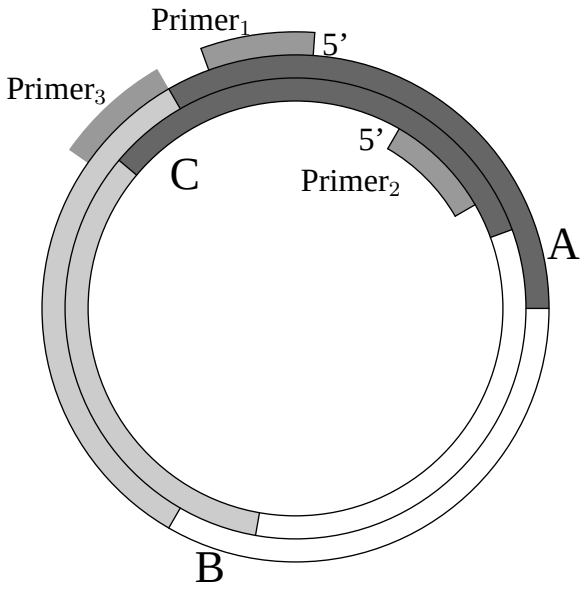

Figure 16: The circular DNA inference path $A \Rightarrow B \Rightarrow C \Rightarrow A$.

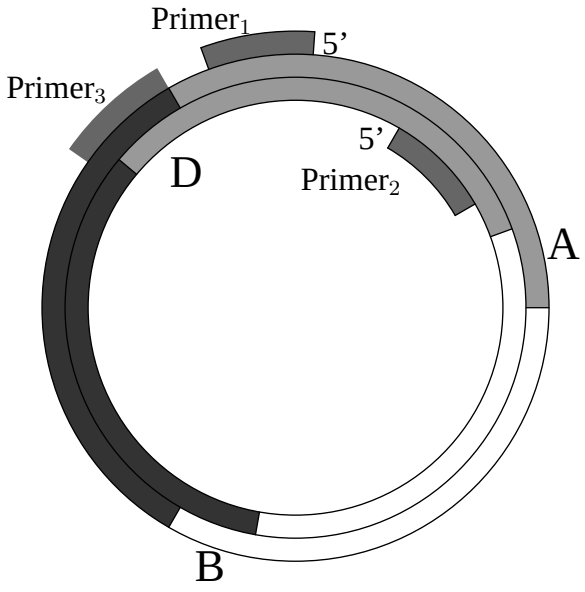

Figure 17: The circular DNA inference path $A \Rightarrow B \Rightarrow D \Rightarrow A$.

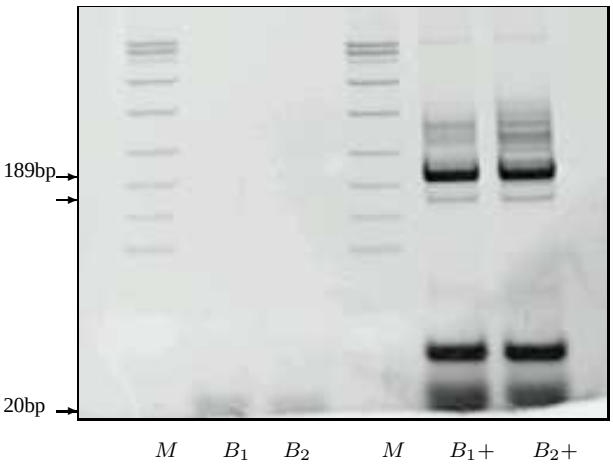

Figure 18: DNA electrophoretogram of the first experiment: M - marker lane, B - experiment lane, ligation 24h in room temperature, PCR - 25 cycles.

mentation of reasoning engine works fine, we make 4 test tubes that should not form correct circles. We call them: B_1 (B_1 only), B_2 (B_2 only), B_1- (B_1, R_1, and R_3) and B_2- (B_2, R_1, and R_2). In the annealing step, we mix fragments that represent appropriate rules and the DBFs in test tubes and let the annealing reaction occur for 5 minutes in 37^oC.

- Ligation. Process of ligation was performed in different conditions and for several time periods. Our goal was to determine perfect ligation conditions in which circles would be created only for test tubes B_1+ and B_2+. Results of some experiments are shown in Figure 18, 19, and 20. In all experiments, 1 Weiss unit of T4 DNA ligase was added to the test tubes. Ligation time varied from 1h to 24h. Ligation temperature was 37^oC or just room temperature.

- PCR. After renaturation (2 minutes in 95^oC), 2 units Taq DNA polymerase were added to each test tube. After that PCR was performed. Number of cycles was different in each experiment. After 18, 21, 24, or 25 cycles the test tubes were kept for 30 seconds in 25^oC.

- Electrophoresis and identification of the solution. PCR products were resolved in 6% acrylamid gel. Correct strings are 189 base pair long. The identification of such a fragment in proper lanes was possible because standard marker was also added. In most of the experiments correct circles were observed in tubes B_1+ and B_2+. The number of correct circles observed in other test tubes was much smaller (or there were none) (see Figures 18, 19, and 20).

In Figure 18, 19, and 20, the correct string size is equal to 189bp. It is seen in Figure 18 that after 24h ligation and 25 cycles of PCR in the lanes, B_1+ and B_2+ are strong 189bp bands, but DBFs sometimes anneal with themselves creating bands below those correct ones. In addition, 20bp and 21bp oligos were detected. They represent primers not used during PCR reaction. In the lanes B_1 and B_2 oligos corresponding to the correct product were not detected. This fact agrees with our expectations. In the later experiments, shown in Figure 19 after 1h ligation, 21 cycles worse correct

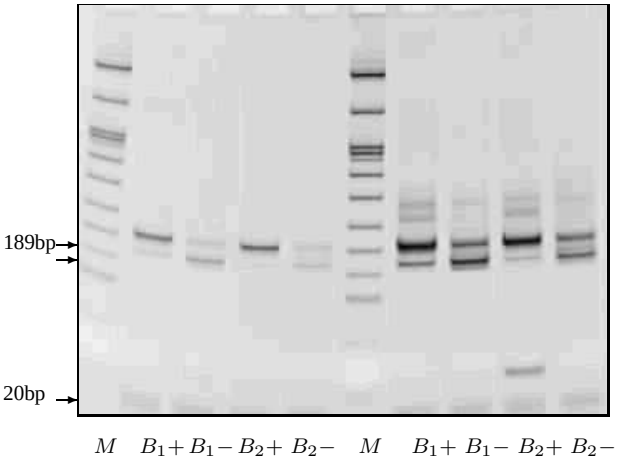

Figure 19: DNA electrophoretogram of the second experiment: M - marker lane, B - experiment lane, ligation 1h in room temperature, 21 cycles of PCR in four first exper. lanes, 24 cycles of PCR in four last exper. lanes.

bands were obtained (with stronger wrong ones). In the lanes B_1- and B_2- other rules are inserted in the place of the correct ones, e.g., in the B_1- lane - R_1, R_3, B_1 DNA fragments are used and in the B_2- lane - R_1, R_2, B_2 DNA fragments are used. In Figure 20, after 1h ligation and 24 cycles of PCR there was no improvement, but after 24 cycles there was a slight improvement in comparison with previous experiment. In general, B_2 lanes are better and this is caused by different sticky ends of DBFs.

Thus, it is quite sure that in spite of short ligation, good correct bands can be obtained. Mismatches in hybridization can be avoided by better design of oligos sequences and by carefully chosen, more expensive, better quality products. Analysis of the results of experiments leads to conclusion that the best ligation conditions are: 1h in room temperature. The number of cycles should be between 19 and 21 (however, more tests are needed to determine better conditions).

OPPORTUNITIES OF FURTHER INFERENCE SYSTEM DEVELOPMENT

In this point the idea first mentioned in work of Mulawka et al. (1998a), and modified in the previous points, is developed and extended. Our objective is to construct a special DNA molecule representing the rule with several premises, which may be conclusions of other rules. So far, every rule is permitted to have only one premise, which might also be the another rule conclusion. This has been the first premise in each rule. In the proposed below molecule, it is possible for the rule to have several such premises.

The ordinary premise is denoted by the symbol S_k for $k \in \{1, K\}$ (K - a number of such premises), and their set by $\mathbf{S}$. The ordinary conclusion (always as the first premise of the next rule) is depicted by the symbol Z_l for $l \in \{1, L\}$ (L - a number of all such conclusions), and their set by $\mathbf{Z}$. The new type conclusion (and also the premise) is described by Y_n for $n \in \{1, N\}$ (N - a number of all modified

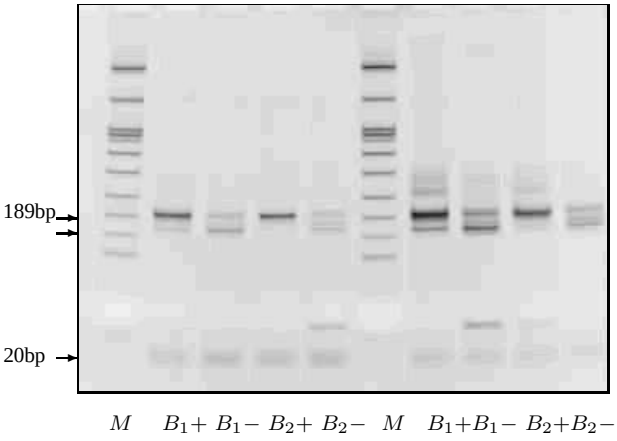

Figure 20: DNA electrophoretogram of the third experiment: M - marker lane, B - experiment lane, ligation 2h in room temperature, 21 cycles of PCR in four first exper. lanes, 24 cycles of PCR in four last exper. lanes.

conclusions), and their set by $\mathbf{Y}$. The rule is represented by connected oligonucleotides, which may be written:

$$R =< \sum_{m=1}^{m=M} P_j^m X_i| =< P_j^1 \dots P_j^M X_i|$$

where $P_j^1 \wedge P_j^2 \wedge \ldots \wedge P_j^M \Rightarrow X_i$; the symbol $\sum$ depicts hybridization, and after it concatenation for $m \in \{1, M\}$, where M is a number of rule premises, and $i \in \{1, L+N\}$, $j \in \{1, K+N\}$; $\mathbf{P} \in \mathbf{S} \cup \mathbf{Y}$ is a set of all possible premises, and $\mathbf{X} \in \mathbf{Z} \cup \mathbf{Y}$ is a set of all possible conclusions.

As is seen, premises and conclusions are DNA strings with orientation $3' \rightarrow 5'$ and are connected by complementary to their "halves" DNA fragments denoted by s_h for $h \in \mathbf{N}$:

$$\hat{R} = \left[\begin{array}{cccccccccc} + & s_h & + & s_{h+1} & \dots & s_{h-1+M} & + & s_{h+M} & + \\ \underline{P_j^1} & + & P_j^2 & + & \dots & + & P_j^M & + & X_i \end{array} \right]$$

The first premise P_j^1, and further premises with indices m, can be distinguished up to the final hypothesis at the end X_i. Indices of s strings are not connected with indices of premises and conclusions. Hybridized by them, premises and conclusions may be concatenated.

The added premise, $Y_n = Y_{(n)}$, and the rule R_t changes the structure of inference path:

$$R_1 \cdot R_W = \left[\begin{array}{ccccccccccc} + & s_{h,1} & \dots & + & s_{g,u} & + & s_{g+1,u} & \dots & + & s_{h+M_W,W} & + \\ \underline{S^1_{k,1}} & + & \dots & S^g_{k,u} & + & Y^r_{(n)} & + & \dots & S^{M_W}_{k,W} & + & Z_{l,W} \end{array} \right]$$

$$R_t = \left[\begin{array}{cccccccc} + & s_{h,t} & + & \dots & + & s_{h+M_t,t} & \overset{(n)}{|} \\ \underline{S^1_{k,t}} & + & S^2_{k,t} & \dots & S^{M_t}_{k,t} & + & + \end{array} \right]$$

where for $h, t \in \mathbf{N}$, $h < g < h + M_W$, $w \in \{1, W\}$, before inserting $Y_{(n)}$ in the place of the premise $S^{g+1}_{k,u}$ the inference path could be denoted by:

$$R_1 \cdot R_W = \sum_{w=1}^{w=W} R_w =< S^1_k \sum_{w=1}^{w=W} \left\{ \sum_{m=2}^{m=M_w} S^m_{k,w} + Z_{l,w} \right\} |.$$

The sequence of symbols $< + - - (n)|$ represents the left part of the DNA string $Y_{(n)} = Y^l_{(n)} * Y^r_{(n)} =< + - - (n)| * < Y^r_{(n)}|$, which as the conclusion connects its left part with the rule R_t, and as the premise - its right part with the rule R_u. Its left part is turn up with $++$, and is finished by an index (n) of the premise $Y_{(n)}$ in the following way: $\overset{(n)}{|}$ showing the firm connection between them. Thus, after hybridization two rules R_t and R_u are connected by one premise $Y_{(n)}$. The concatenation process reveals no connection between the premise strings $S^g_{k,u}$ and $Y_{(n)}$:

$$R_1 \cdot R_W = \left[\begin{array}{ccccccccccc} + & s_{h,1} & \dots & * & s_{g,u} & * & s_{g+1,u} & \dots & * & s_{h+M_W,W} & + \\ \underline{S^1_{k,1}} & * & \dots & S^g_{k,u} & + & Y^r_{(n)} & * & \dots & S^{M_W}_{k,W} & * & Z_{l,W} \end{array} \right]$$

$$R_t = \left[\begin{array}{cccccccc} + & s_{h,t} & * & \dots & * & s_{h+M_t,t} & \overset{(n)}{|} \\ \underline{S^1_{k,t}} & * & S^t_{k,t} & \dots & S^{M_t}_{k,t} & * & + \end{array} \right]$$

Note that in the system of $w + 1$ rules, two inference paths $|\widetilde{S^1_{k,1}} \cdot \widetilde{Z_{l,W}} >$ and $< S^1_{k,t} \cdot Z_{l,W}|$ are created and two other $< S^1_{k,1} \cdot Z_{l,W}|$ and $|\widetilde{S^1_{k,t}} \cdot \widetilde{Z_{l,W}} >$ are not created on the whole. The truth of the conclusion $Z_{l,W}$ from the rule R_W depends on the integrity of the inference path $|\widetilde{S^1_{k,1}} \cdot \widetilde{Z_{l,W}} >$ and on the truth of the premise $Y_{(n)}$ from the rule R_u. This premise is also a conclusion of the rule R_t. The truth of the conclusion $Y_{(n)}$ depends on the integrity of the inference path $< S^1_{k,t} \cdot Y_{(n)}|$, which is the left part of the path $< S^1_{k,t} \cdot Z_{l,W}|$. Thus, the truth of the final conclusion (hypothesis) $Z_{l,W}$ after inserting $Y_{(n)}$ depends on the integrity of the inference path $|\widetilde{S^1_{k,1}} \cdot \widetilde{Z_{l,W}} >$ and on the integrity of the path $< S^1_{k,t} \cdot Y_{(n)}|$, and in the consequence on the integrity of the inference path $< S^1_{k,t} \cdot Z_{l,W}|$.

In order to insert a next premise $Y_{(n+1)}$, for example, in the place of the premise $S^2_{k,1}$, the integrity of the inference path $< S^1_{k,1} \cdot Z_{l,W}|$ should be assured (of course, the path $|\widetilde{S^1_{k,1}} \cdot \widetilde{Z_{l,W}} >$ has to be concatenated, too), what is described in the following expression:

$$R_1 \cdot R_W = \left\lceil \begin{array}{ccccccccccc} + & s_{h,1} & \ldots & + & s_{g,u} & + & s_{g+1,u} & \ldots & + & s_{h+M_W,W} & + \\ \underline{S^1_{k,1} \quad + \quad \ldots \quad S^g_{k,u} \quad + \quad T_{(n)} \quad + \quad \ldots \quad S^{M_W}_{k,W} \quad + \quad Z_{l,W}} \end{array} \right\rfloor$$

and concatenation is denoted by:

$$R_1 \cdot R_W = \left\lceil \begin{array}{ccccccccccc} + & s_{h,1} & \ldots & * & s_{g,u} & * & s_{g+1,u} & \ldots & * & s_{h+M_W,W} & + \\ \underline{S^1_{k,1} \quad * \quad \ldots \quad S^g_{k,u} \quad * \quad T_{(n)} \quad * \quad \ldots \quad S^{M_W}_{k,W} \quad * \quad Z_{l,W}} \end{array} \right\rfloor$$

As is seen in order to obtain the mentioned integrity, the DNA string $T_{(n)} =< Y^r_{(n)}|$ with the length of the ordinary fact S_k together with the premise $Y_{(n)}$ is added to the reaction vessel. This enables concatenation of the mentioned path and after it insertion of the new type premise $Y_{(n+1)}$ before the DNA string $T_{(n)}$, which has been put in the place of the DNA fragment $Y_{(n)}$. Now in the place of the premise $S^2_{k,1}$ the premise $Y_{(n+1)}$ can be inserted and detected as in the case of the premise $Y_{(n)}$.

Next, in order to insert the new type premise $Y_{(n+1)}$, for example, in the place of the premise $S^{M_W}_{k,W}$, the integrity of the inference path $|\widetilde{S^1_{k,t}} \cdot \widetilde{Z_{l,W}} >$ should be assured (of course, the path $< S^1_{k,t} \cdot Z_{l,W}|$ has to be concatenated, too). Thus, together with the premise $Y_{(n)}$, the string $\tilde{T}_{(n)}$ is added. The latter oligonucleotide length enables concatenation of the mentioned path and inserting of the new type premise $Y_{(n+1)}$ after the premise $Y_{(n)}$. Hybridization of premises with the string $\tilde{T}_{(n)}$ looks like this:

$$R_t + R^r_W = \left\lceil \begin{array}{ccccccccccc} + & s_{h,t} & \ldots & + & s_{h+M_t,t} & \tilde{T}_{(n)} & s_{g+1,u} & \ldots & + & s_{h+M_W,W} & + \\ \underline{S^1_{k,t} \quad + \quad \ldots \quad S^{M_t}_{k,t} \quad + \quad Y_{(n)} \quad + \quad \ldots \quad S^{M_W}_{k,W} \quad + \quad Z_{l,W}} \end{array} \right\rfloor$$

and concatenation is denoted by:

$$R_t * R^r_W = \left\lceil \begin{array}{ccccccccccc} + & s_{h,t} & \ldots & * & s_{h+M_t,t} & \tilde{T}_{(n)} & s_{g+1,u} & \ldots & * & s_{h+M_W,W} & + \\ \underline{S^1_{k,t} \quad * \quad \ldots \quad S^{M_t}_{k,t} \quad * \quad Y_{(n)} \quad * \quad \ldots \quad S^{M_W}_{k,W} \quad * \quad Z_{l,W}} \end{array} \right\rfloor$$

Now in the place of the premise $S^{M_W}_{k,W}$, the premise $Y_{(n+1)}$ can be inserted and detected as in the case of the premise $Y_{(n)}$.

Thus, with every premise $Y_{(n)}$, strings $T_{(n)}$ and $\tilde{T}_{(n)}$ are added to the reaction vessel and they may create the following, double-stranded DNA molecule:

$$\left\lceil \begin{array}{cc} \tilde{T}_{(n)} & + \\ \underline{- \quad T_{(n)}} \end{array} \right\rfloor = \left\lceil \begin{array}{cc} \tilde{T}_{(n)} & \\ \underline{T_{(n)} \quad +} \end{array} \right\rfloor .$$

These strings enable self-assembling of inference paths (from any starting premise to conclusions) to assure their integrity. The detection of the truth of the hypotheses can be done now.

The decision tree changes now to the decision graph, because it has several starting premises called roots $K_i = S^1_{k,1}$ and each its node $Z_{l,w} = S^1_{k,v}$ for $v \neq w$ can have several predecessors $S^1_{k,w}, S^1_{k,w-1}, \ldots$ and can have several successors. Nodes without successors are final nodes called leaves and represent hypotheses.

$$\left\lceil \begin{array}{cccc} + & s_1 & + & s_2 \\ A & + & C^r_{(1)} & + \end{array} \right\rceil, \left\lceil \begin{array}{cc} s_2^r & + \end{array} \right\rceil = \left\lceil \begin{array}{c} \tilde{D} \end{array} \right\rceil$$
$$\left\lceil \begin{array}{ccc} + & s_3 & {\scriptstyle (1)}\,| \\ B & + & + \end{array} \right\rceil$$

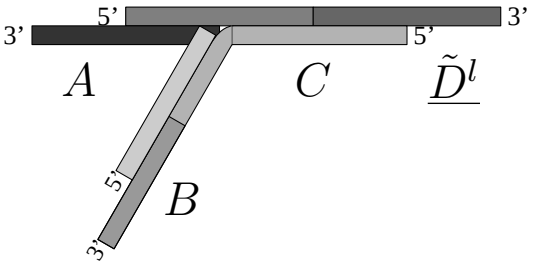

Figure 21: The rule representation with two premises A, C ($B \Rightarrow C$) and one conclusion D.

In order to explain our new method, we present exemplary inference systems consisted of DNA molecules given in Figure 21-24. It is very important to remember that the upper oligos $5' \rightarrow 3'$ represent hybridized or concatenated, complementary to premises and conclusions strings s and the lower oligos $3' \rightarrow 5'$ - these premises and conclusions. In Figure 21, the represented rule consists from the premises A, C and the conclusion D. The premise C is the conclusion of another rule $B \Rightarrow C$. DNA fragments s hybridizing to premises and conclusions create adequate structures.

$$\left\lceil \begin{array}{cccccc} + & s_1 & + & s_2 & + & s_4 \\ A & + & C^r_{(1)} & + & D & + \end{array} \right\rceil, \left\lceil \begin{array}{cc} s_4^r & + \end{array} \right\rceil = \left\lceil \begin{array}{c} \tilde{E} \end{array} \right\rceil$$
$$\left\lceil \begin{array}{ccc} + & s_3 & {\scriptstyle (1)}\,| \\ B & + & + \end{array} \right\rceil$$

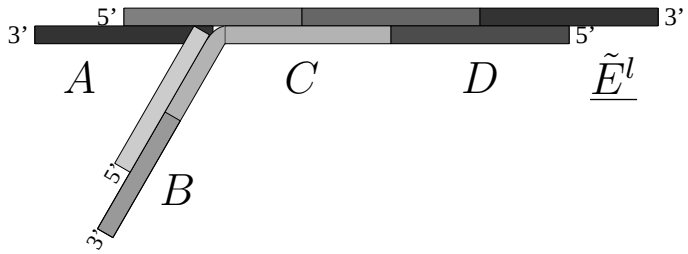

Figure 22: Hybridized two rules $A \wedge C \wedge D \Rightarrow E$ and $B \Rightarrow C$.

It is obvious that one inference system implementation may have several logical descriptions. Each of them is with different set of rules. The system in Figure 22 can be described in several ways:

$$A \wedge C \wedge D \Rightarrow E, B \Rightarrow C$$
$$A \wedge C \Rightarrow D, D \Rightarrow E, B \Rightarrow C$$

In order to assure the correct inference process, the conclusion C has to be true (it is the premise of another rule, so checking the truth of its premises is very necessary).

$$\left[\begin{array}{cccccccccc} + & s_5 & + & s_1 & + & s_2 & + & s_4 & + & B_{1;2} \\ \underline{K_1} & + & A & + & C^r_{(1)} & + & D & + & E & + \end{array}\right], \left[\begin{array}{c} B^r_{1;2} \\ \underline{\quad} \end{array}\right] = \left[\begin{array}{c} \tilde{K}^l_{1;2} \\ \underline{\quad} \end{array}\right]$$
$$\left[\begin{array}{cccc} + & s_6 & + & s_3 \\ \underline{K_2} & + & B & + \end{array}\right]$$

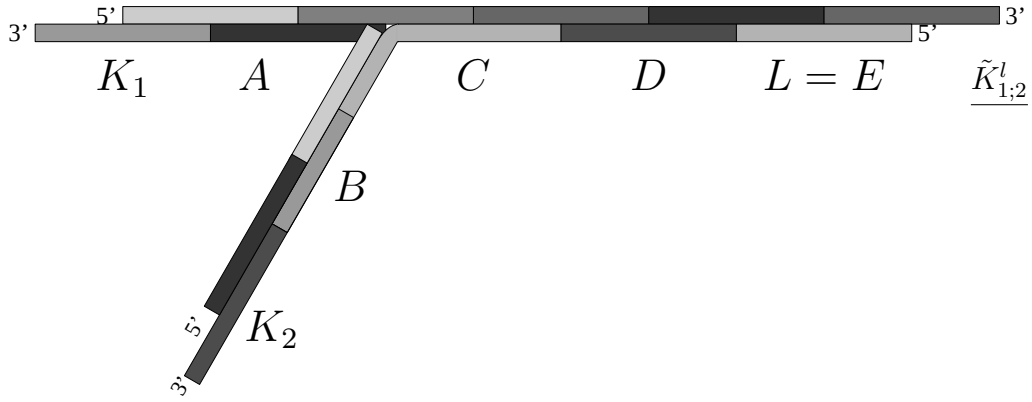

Figure 23: The decision graph with the joint in the place of the premise C from Figure 22 creating two roots K_1 i K_2.

To demonstrate the inference process, consider the following example. Let the proposed inference system consist of four rules and two DNA fragments DBFs written below:

$$R_1 : A \wedge C \wedge D \Rightarrow E$$
$$R_2 : B \Rightarrow C$$
$$R_3 : K_1 \Rightarrow A$$
$$R_4 : K_2 \Rightarrow B$$
$$B_1 : \widetilde{E^r} * \widetilde{K^l_1}$$
$$B_2 : \widetilde{E^r} * \widetilde{K^l_2}$$

The rule R_1 represented in Figure 22 has three premises A, C, and D, and the second premise is the conclusion of second rule R_2. After hybridizing of these two rules and ligation the upper oligos $5' \rightarrow 3'$ of the first rule and the lower oligos $3' \rightarrow 5'$ of the second rule are concatenated. Detection of these two oligo paths proves the truth of the first rule conclusion. In the decision graph during detection only one new type conclusion can exist (in order to insert another new type conclusion it is necessary to insert in the place of the previous new type conclusion special fragments $T_{(1)}$ or $\tilde{T}_{(1)}$, which assure integrity of the truth paths), as is depicted in Figure 23. In the case of the root K_1

detection with the help of the DNA string B_1 and creation of the circular, inference molecule, the upper oligos $5' \rightarrow 3'$ are completely concatenated. Their path can be amplified and detected. Alike in the case of the root K_2 detection with the help of the DNA string B_2 and creation of the circular, inference molecule the lower oligos $3' \rightarrow 5'$ are completely concatenated. Their path can be amplified and detected. Of course, in order to check the truth of the conclusion C, the detection of the root K_1

$$\left[\begin{array}{cccccccccc} + & s_6 & + & s_3 & \tilde{T}_1 & s_2 & + & s_4 & + & B_{2;3} \\ \underline{K_2} & + & B & + & C^{(1)} & + & D^r_{(2)} & + & E & + \end{array}\right], \left[\underline{B^p_{2;3}}\right] = \left[\underline{\tilde{K}^l_{2;3}}\right]$$

$$\left[\begin{array}{cccc} + & s_8 & + & s_7 \\ \underline{K_3} & + & F & + \end{array}\begin{array}{c} {}^{(2)} \\ | \\ + \end{array}\right]$$

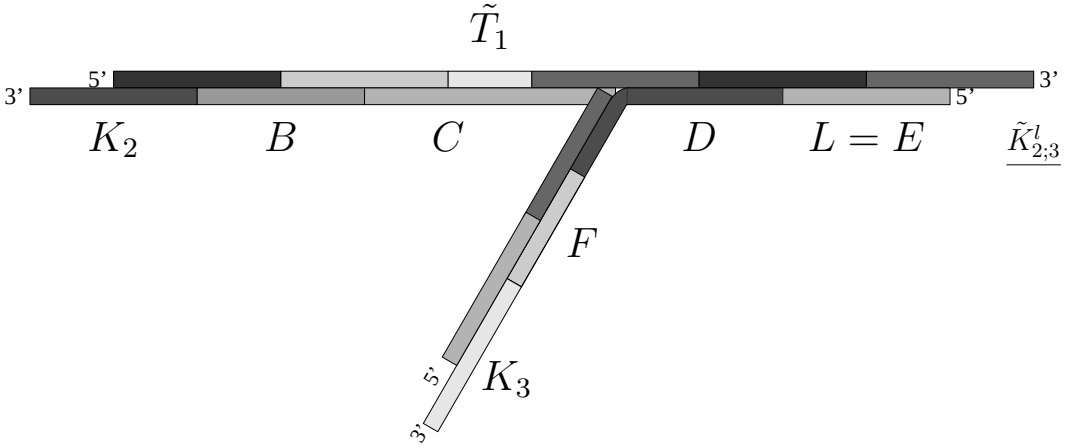

Figure 24: The decision graph with the joint in the place of the premise D from Figure 23 creating two roots K_2 i K_3.

has to be proceeded after the detection of the root K_2, because the truth path from K_1 to E depends on the integrity of the truth path from K_2 to C.

If the next new type conclusion is inserted (together with the whole truth path from the root K_3 with rules $K_3 \Rightarrow F$ and $F \Rightarrow D$) in the place of the ordinary premise D, the branch with the root K_1 should be removed and replaced by DNA string $\tilde{T}_1$, as is depicted in Figure 24. Changes are denoted by the following actions: in the symbol of the premise C the expression (1) is moved from the lower index to the upper index; then to the new type conclusion D lower index the expression (2) is added. During detection it is easy to distinguish the complete, circular inference paths from roots K_i to DNA strings DBF denoted by the symbol B_n, which right "halves" are complementary to some roots K and which left "halves" are complementary to some leaves L. Of course, like in the previous case the truth of the conclusion C, now the truth of the conclusion D depends on the integrity of the inference path from K_3 to L, and the truth of the inference path from K_2 to L depends on the truth of the conclusion D. Thus, the integrity of the main inference path from K_1 to L depends on the integrity of the inference path from K_2 to L and in the consequence, on the integrity of the inference path from K_3 to L. In this way, next new type premise-conclusions can be added to the inference system.

Complete inference paths can be created after adding to the reaction vessel all necessary premises, conclusions, and complementary to them DNA strings s_i, and DNA strings $Y_{(n)}$ together with adequate DNA strings $\tilde{T}_n$, T_n assuring integrity of the truth paths. During chemical reactions, correct conclusion premise joints are formed. After involving of n new type conclusions in the reaction, $n+1$ circular truth paths should be created.

The mentioned inference system with three roots K_i for $i \in 1, 2, 3$ would consist of six rules and three DBF strings.

$$R_1 : A \wedge C \wedge D \Rightarrow E$$
$$R_2 : B \Rightarrow C$$
$$R_3 : F \Rightarrow D$$
$$R_4 : K_1 \Rightarrow A$$
$$R_5 : K_2 \Rightarrow B$$
$$R_6 : K_3 \Rightarrow F$$
$$B_1 : \widetilde{E^r} * \widetilde{K_1^l}$$
$$B_2 : \widetilde{E^r} * \widetilde{K_2^l}$$
$$B_3 : \widetilde{E^r} * \widetilde{K_3^l}$$

It should be noted that in the system provided there may be more leaves (final conclusions-hypotheses), because from one premise we can have several conclusions. In this case dependencies would be denoted by several additional DNA strings representing rules, DBF fragments for each pair root-leaf $K_i - L_j$, and primers for amplification.

CONCLUSIONS

In this paper, novel approach to implementation of inference engine based on molecular computing paradigm has been discussed. It has been shown how knowledge can be structured, and stored by means of DNA circular molecules to implement an inference engine. Such a system can store knowledge for a narrowly defined subject area and solve problems by making logical deductions. The inference mechanism performs these tasks by manipulating on DNA circular molecules. It provides the problem-solving methods by which the rules are processed. The inference algorithm is simple. Standard genetic engineering DNA techniques such as annealing, ligation, and electrophoresis are required. Several experiments have been conducted to assess the performance of inference engine realized by biochemical reactions. The results of these experiments indicate interesting feature of the method. By using circular fragments derived from plasmids, the drawn inferences can be ”read” after the experiments with higher precision and efficiency. To achieve reliable performance, some parameters of reactions - temperatures, concentrations of oligos, times of reactions, etc. (Langohr, 1997; Lipton et al., 1996; Roweis & Winfree, 1999) have been experimentally tested. Our results show that molecular approach has a potential value for building the inference engine.

This approach reveals a number of advantages over traditional electronic machine. Self-assembling of molecules mimics the properties of associative memory (Jonoska et al., 1998; Winfree, 1998). This technique has the potential value to analyse large knowledge bases, thereby in principle, paves the way

to new methods of programming in logic (Mihalache, 1997). Thus, molecular inference engine seems to be a good choice for implementation of an expert system with a huge knowledge base in the parallel hardware where both data retrieval and inference process would be relatively fast.

In future experiments, plasmids with inference paths can be multiplied in bacteria cells after transformation into these cells. More sophisticated inference systems with rules having several premises and conclusions should be developed and improved.

ACKNOWLEDGMENTS

This work was supported by the Polish State Committee for Scientific Research, through the KBN grant number 8T11F00816. This article was processed with the LaTeX 2_ε macros package.

BIBLIOGRAPHY

Adleman, L.M. 1994. Molecular computation of solutions to combinatorial problems. *Science* 266:1021-1024.

Adleman, L.M., P.W.K. Rothemund, S. Roweis, and E. Winfree. 1999. On applying molecular computation to the data encryption standard. *Journal of Computational Biology* 6(1):53-63.

Amos M.. 1997. *DNA Computation*. Ph.D. Thesis, Department of Computer Science, University of Warwick UK.

Amos M., and P.E. Dunne. 1997. *DNA Simulation of Boolean Circuits*. Technical Report CTAG-97009, Department of Computer Science, University of Liverpool, UK.

Ashoori R.C. 1996. Electrons in Artificial Atoms. *Nature* 379:413-417.

Ausubel F., and K. Struhl. 1995. *Short Protocols in Molecular Biology. A Compendium of Methods from Current Protocols in Molecular Biology*, 3rd Ed. New York: John Wiley & Sons.

Baum E.B. 1995. Building an associative memory vastly larger than the brain. *Science* 268:583-585.

Baum E.B. 1996. Will Future Computers Be Made of DNA, *Windows Mag.*, 6, http://www.winweb.winmag.com/

Beth T. 1997. Quantum computers - A new concept in nanoelectronics. In *Proc. ECCTD'97*, Budapest 271.

Biswas N.N. 1993. *Logic Design Theory*. Prentice-Hall Int. Editions, New York.

Collier C.P., E.W. Wong, M. Belohradsky, F.M. Raymo, J.F. Stoddart, P.J. Kuekues, R.S Williams, and J.R. Heath. 1999. Electronically Configurable Molecular-Based Logic Gates. *Science* 285:391-393.

Csuhaj-Varjú E., L. Kari, G. Pun. 1996. Test tube distributed systems based on splicing. *Computers and AI* 15(2-3):211–232.

Dassen R. 1999. *A Bibliography of Molecular Computation and Splicing Systems*, http://-liinwww.ira.uka.de/bibliography/Misc/dna.html

Freund R., G. Pun, G. Rozenberg, and A. Salomaa. 1997. Watson-Crick Finite Automata. In *Proc. of the Third DIMACS Workshop on DNA-based Computers*, Philadephia, PA, 305-317.

Gehani A., and J. Reif. 1998. Micro Flow Bio-Molecular Computation. In *Proc. of the Fourth DIMACS Workshop on DNA-based Computers*, Pennsylvania, USA, 253-266.

Goldberg D.E.. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Gupta V., S. Parthasarathy, and M. Zaki. 1997. Arithmetic and Logic Operations with DNA. In *Proc. of the Third DIMACS Workshop on DNA-based computers*, Philadelphia, PA, 212-220.

Head T. 1987. Formal language theory and DNA: an analysis of the generative capacity of recombinant behaviors. In *Bulletin of Mathematical Biology* 49:737-759.

Head T., G. Pun, and D. Pixton. 1997. Language theory and molecular genetics. Generative mechanisms suggested by DNA recombination In *Handbook of Formal Languages*, eds. G. Rozenberg, and A. Salomaa. Heilderberg: Springer-Verlag.

Heath J.R., P.J. Kuekes, G.S. Snider, and R.S. Williams. 1999. A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology. *Science* 280:1716-1721.

Hill F.J., and G.P Peterson. 1974. *Switching Theory and Logical Design*. New York: Wiley.

Jagielska A., L.Z. Stolarczyk, and L. Piela. 1998. Mnemon - a Hypothetical Molecule with Bistable Electronic Ground State. In *Proc. of the Int. Conf. on Rough Sets and Current Trends in Computing - Spec. Session on DNA Comp.*, Warszawa, 6-16.

Jonoska N., S.A. Karl, and M. Saito. 1998. Three Dimensional DNA Structures in Computing. In *Proc. of the Fourth DIMACS Workshop on DNA-based Computers*, Pennsylvania, USA, 189-200.

Kolata G.. 1995. A Vat of DNA May Become the Fast Computer of the Future. In *New York Times*.

Koza J.R.. 1992. *Genetic Programming*. MIT Press, Cambridge, MA.

Langohr K.. 1997. *Sources of Error in DNA Computation*. University of Western Ontario.

Leete T., J. Klein, and H. Rubin. 1997. Bit operations using a DNA template. In *Proc. of the Third DIMACS Workshop on DNA-based Computers*, Philadephia, PA, 159–166.

Li Z. 1998. Algebraic properties of DNA operations. In *Proc. of the Fourth DIMACS Workshop on DNA-based Computers*, PA, 57-70.

Lipton R.J. 1995. DNA Solution of Hard Computational Problems. *Science* 268:542–545.

Lipton R., D. Boneh, C. Dunworth. 1996. *Making DNA Computers Error Resistant*. Princeton: Princeton University Press.

Liu Q., A. Frutos, L. Wang, A.J. Thiel, S.D. Gillmor, T. Strother, A.E. Condon, R.M. Corn, M.G. Lagally, and L.M. Smith. 1998. Progress Toward Demonstration of a Surface Based DNA Computation: a One Word Approach to Solve a Model Satisfiability Problem. In *Proc. of the Fourth DIMACS Workshop on DNA-based Computers*, Pennsylvania, USA, 15-25.

Malone M.S. 1995. Beyond semiconductors. In *The Microprocessor: A Biography*. New York: Springer-Verlag.

McCaskill J.S. 1997. *Open Flow Microreactors*. Tech. Report, IMB Company, Jena, Germany.

McCaskill J.S., T. Maeke, U. Gemm, L. Schulte, and U. Tangen. 1997. NGEN: A massively parallel reconfigurable computer for biological simulation: towards a self-organizing computer. *Lecture Notes Comp. Sci.* 1259:260-276.

Mihalache V. 1997. Prolog approach to dna computing. In *Proc. of the IEEE International Conference on Evolutionary Computation (ICEC97), Special Session on DNA-based Computation*, Indianapolis, IA.

Montemerlo M.S., J.C. Love, G.J. Opiteck, D. Goldhaber-Gordon, and Ellenbogen. 1996. *Technologies and Designs for Electronic Nanocomputers*. Mitre, McLean, VA.

Montemerlo M.S., J.C. Love, G.J. Opiteck, D. Goldhaber-Gordon, and Ellenbogen. 1997. Overview of Nanoelectronic Devices. In *Proc. IEEE* 85/4:521-540.

Mulawka J.J. 1996. *Expert Systems (in Polish)*. Warsaw: Wydawnictwa Naukowo-Techniczne (WNT).

Mulawka J.J. 1997. Molecular Computing Promise for New Generation of Computers. In *Proc. of the Polish-Czech-Hungarian Workshop on Circuit Theory, Signal Processing and Applications*, Budapest, 94-99.

Mulawka J.J., P. Borsuk, and P. Węgleński. 1998. Implementation of the Inference Engine Based on Molecular Computing Technique. In *Proc. IEEE Int. Conf. Evolutionary Computation (ICEC'98)*, Anchorage, AL, 493-498.

Mulawka J.J., and M.J. Oćwieja. 1998b. Molecular Inference via Unidirectional Chemical Reactions. In *Proc. of II Inter. Conf. on Evolvable Systems (ICES'98)*, Lausanna, Switzerland, 372–379.

Mulawka J.J., T. Janczak, P. Borsuk, and P. Węgleński. 1998. Reasoning via DNA Based Decision Trees. In *Proc. of First Inter. Conf. on Rough Sets and Current Trends in Computing (RSCTC'98)*, Warsaw, 17-26.

Mulawka J.J., P. Wąsiewicz, and A. Płucienniczak. 1998. Logical Operations with DNA Strands. In *Proc. of the Int. Conf. on Rough Sets and Current Trends in Computing - Spec. Session on DNA Comp.*, Warsaw, Poland, 27-36.

Mulawka J.J., and P. Wąsiewicz. 1998. Molecular Computing - New challenge of information technology (in Polish). *Informatyka Polish Journal* 7/8:36-39.

Mulawka J.J., P. Wąsiewicz, and A.P. Płucienniczak. 1999a. Another Logical Molecular NAND Gate System. In *Proc. of the 7th Int. Conf. on Microelectronics for Neural, Fuzzy, and Bio-Inspired Systems (MicroNeuro'99)*, Granada, Spain, 340-345.

Mulawka J.J., P. Wąsiewicz, and K. Piętak. 1999. Virus-enhanced genetic algorithms inspired by DNA computing. *Lecture Notes Art. Int. - Subseries LNCS* 1609:527-537.

Mulawka J.J., T. Janczak, A. Malinowski, and R. Nowak. 1999. DNA computing - Promise for information processing. *Universitatis Jagellonicae Acta Informaticae*, Krakow.

Mulawka J.J., and R. Nowak. 1999. Molecular Implementation of Logical Operations via Electrophoretical methods. In *Proc. Nat. Conf. On Evol. Algorithms and Global Optimization*, Potok Zloty.

Murphy R.C., R. Deaton, D.R. Franceschetti, S.E. Stevens, and M. Garzon. 1997. A new algorithm for DNA based computation. In *Proc. IEEE Int. Conf. Evolutionary Computation (ICEC'97)*, Indianapolis, IN, 207–212.

Ogihara M., and A. Ray. 1996. *Simulating Boolean Circuits On a DNA Computer*. TR 631, University of Rochester, Computer Science Department, NY.

Ogihara M., and A. Ray. 1998. *The Minimum DNA Computation Model and Its Computational Power*. TR 672, University of Rochester, NY.

Pun G., G. Rozenberg, A. Salomaa. 1998. *DNA Computing - New Computing Paradigms*. Berlin, Heidelberg: Springer-Verlag.

Pederson K. 1989. *Expert System Programming*. New York: Wiley.

Pisanti N. 1997. *A survey on DNA computing*. Technical Report TR-97-07, University di Pisa, Pisa Italy.

Rooß D., and K.W. Wagner. 1995,1996.*On the power of DNA-computers*. Technical reports, University of Wurzburg, Germany.

Roweis S., E. Winfree, R. Burgoyne, N. Chelyapov, M. Goodman, P. Rothemund, and L.M. Adleman. 1998. A Sticker-Based Model for DNA Computation. *Journal of Computational Biology* 5(4):615-629.

Roweis S., and E. Winfree. 1999. On the reduction of errors in DNA computation. *Journal of Computational Biology* 6(1):65-75.

Sakamoto K., D. Kiga, K. Komiya, H. Gouzu, S. Yokoyama, S. Ikeda, H. Sugiyama, and M. Hagiya. 1998. State Transitions by Molecules. In *Proc. of the Fourth DIMACS Workshop on DNA-based Computers*, Pennsylvania, USA, 87-100.

Sambrook J., E.F. Fritsch, and T. Maniatis. 1989. *Molecular Cloning. A Laboratory Manual.*, 2nd Ed. Cold Spring Harbor Laboratory Press, New York.

Stańczak J.T., and J.J. Mulawka. 1999. Evolutionary Algorithms with an Improved Selection of Individuals. In *Proc. of Int. Processing and Manufacturing of Materials*, July 11-15, Honolulu, Hawaii.

Wąsiewicz P., J.J. Mulawka, and B. Verma. 1997. Global Optimization and Genetic Methods. In *Proc. of the 1st Int. Conf. on Computational Intelligence and Multimedia Applications (ICCIMA-97)*, Gold Coast, Australia, February 10-12, 30-36.

Wąsiewicz P., and J.J. Mulawka. 1997. Genetic Programming in Optimization of Algorithms. In *Proc. of the 5th Fuzzy Days International Conference on Computational Intelligence*, Dortmund, Germany, April 28-30, 581-582.

Wąsiewicz P., and G. Klebus. 1997. Genetic Programming Approach to CMAC Parameters Tuning. In *Proc. of he Third Conference on Evolutionary Computation (Evolution Artificielle 97)*, Nimes, France, October 22-24, 287–296.

Wąsiewicz P., P. Borsuk, J.J. Mulawka, and P. Węgleński. 1999. Implementation of Data Flow Logical Operations via Self-Assembly of DNA. *Lect. Not. Comp. Sci.* 1586:174–182.

Wąsiewicz P., T. Janczak, J.J. Mulawka, and A. Płucienniczak. 1999. The Inference Via DNA Computing. In *Proc. Congress on Evolutionary Computation (CEC'99)*, July 2, Washington, D.C., 988-993.

Wąsiewicz P., A. Malinowski, R. Nowak, J.J. Mulawka, P. Borsuk, P. Węgleński, and A. Płucienniczak. 2001. DNA Computing: Implementation of Data Flow Logical Operations. *Future Generation Computer Systems Journal* 17/4:361-378.

Węgleński P. 1995. *Molecular Genetics (in Polish)*. Warsaw: Polish Scientific Publishers (PWN).

Winfree E. 1998. Whiplash PCR for $O(1)$ Computing. In *Proc. of the Fourth DIMACS on DNA-based Computers*, Pennsylvania, USA, 175-188.

Winfree E., X. Yang, and N.C. Seeman. 1996. Universal computation via self-assembly of DNA, Some theory and experiments. *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, ISSN, 1052–1798.