# String Displacement Systems and DNA Circuits Based on Isothermal Constrained Loop Extension DNA Amplification

Maurice Margenstern, Pascal Mayer, Sergey Verlan

# String Displacement Systems and DNA Circuits Based on Isothermal Constrained Loop Extension DNA Amplification

Maurice Margenstern[1], Pascal Mayer[2], and Sergey Verlan[3]

[1] Université Paul Verlaine - Metz, LITA, EA 3097, UFR MIM,
Ile du Saulcy, 57045 Metz Cédex, France
E-mail: margens@univ-metz.fr
[2] Alphanosos S.A.S.,
20-22 rue Henri et Gilberte Goudier,
63200 Riom, France Email: pascal.mayer@neuf.fr
[3] LACL, Département Informatique, Université Paris Est Créteil
61 av. Général de Gaulle, 94010 Créteil, France
E-mail: verlan@u-pec.fr

**Abstract.** In this paper, we first describe the isothermal constrained loop extension DNA amplification (ICLEDA), which is a variant of amplification combining the advantages of rolling circle amplification (RCA) and of strand displacement amplification (SDA). Then, we formalize this process in terms of the theory of formal languages, which yields two new operations of sticking and displacement on normal and circular strings. The obtained formalism is quite expressive and we show how to implement the computation of any boolean function on DNA strands using ICLEDA. We also discuss variants of the model that could eventually bring more computational power.

## 1 Introduction

The first attempt to use DNA for solving computational problems was done by Adleman [1]. Since that time several models of computation using DNA have been proposed, we refer to [3] for an overview. The structure and the properties of DNA molecules allow their natural representation as strings and operations on strings. So, abstracting from the physical constraints, it is possible to perform a language-theoretical study of the obtained models. This study can reveal some interesting properties and limitations that could eventually be translated back to the original DNA molecules. The book [11] contains several examples of such an approach.

In this article we follow the same line by formalizing the process of a special type of string amplification and displacement in terms of operations on strings. The most common in-vitro DNA strand replication method (also called "DNA amplification") is based on PCR. This method is based on a series of primer extension cycles with changing temperature conditions to allow for strand separation at the beginning of each cycle.

Rolling-circle amplification (RCA) is another method of strand replication based on circular DNA molecules [4] and is inspired from the natural replication mechanisms of several viruses [5]. The important observation is that this method does not require changing the conditions of the test tube for DNA amplification and produces long single stranded DNA molecules including multiple complementary copies of the circular template DNA fragment. This procedure was used in DNA computing as a basis for the simulation of the resolution refutation in [6].

The strand displacement amplification (SDA) is based on the ability of a restriction enzyme to nick a modified recognition site and the ability of a polymerase to initiate synthesis at the nick and displace a downstream DNA strand during replication [12]. Both above methods allow us to produce DNA strands in isothermal conditions. There are methods using both RCA and SDA, for example ramification-extension method (RAM) [13].

In this article we consider a new isothermal DNA replication method, called ICLEDA for "Isothermal Constrained Loop Extension DNA Amplification", described in [8]. It makes it possible to produce short linear and single stranded DNA strands in isothermal conditions. Importantly in the perspective of a practical application, the amplification is also possible when the template molecules are immobilized on a support. We formalize the amplification process in terms of formal languages. Such a formal system is constructed from a number of circular strings, which we call *amplification loop complexes* (or simply loop complexes) that can be in two states: blocked or unblocked. A loop complex in unblocked state produces infinitely the corresponding DNA strand (signal). The transition from a blocked to an unblocked state is done by annealing and primer extension, formalized as a string displacement operation. The resulting model is quite powerful, for example it is possible to simulate a signaling cascade whose nodes correspond to AND and OR gates. More precisely, using an approach similar to [9, 2], we consider asynchronous circuits with dual-rail encoding, i.e. a bit is encoded using 2 wires and determined by the transition happening on one of them, exclusively. The amplification to express the presence of signals which can further trigger the amplification of new signals following the circuit in an autonomous way.

A preliminary version of this article can be found in [7].

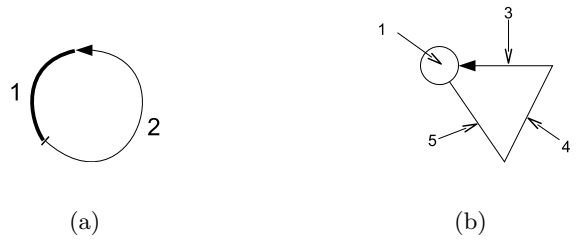
## 2 The biological mechanism

In this section, we first describe the ICLEDA amplification process defined in [8] on which the whole work is based. Then, we show how this mechanism allows us to devise a configuration, which we call the **loop complex**, which will later on allow us to implement logical gates in this context. Note that the word loop refers to the shape of the biophysical complex we consider rather than the computational device which is usually understood by this term, this is why the term *complex* is attached to *loop* in this denomination.

### 2.1 The amplification and the loop

The ICLEDA amplification method designed in patent[4] [8] is to some extend a combination of RCA and SDA amplification. We refer to this patent for more technical information.

The mechanism is represented on Fig. 1(a). The loop complex is a circular molecule composed of two parts: the amplifiable fragment (2) and the loop link (1). The arrow represents the 3' end of the amplifiable fragment. We represent this molecule schematically as on Fig. 1(b). For the sake of commodity we split the amplifiable fragment in 3 parts (3,4,5 on the picture) corresponding to the 3' end, middle and 5' end of the amplifiable fragment.



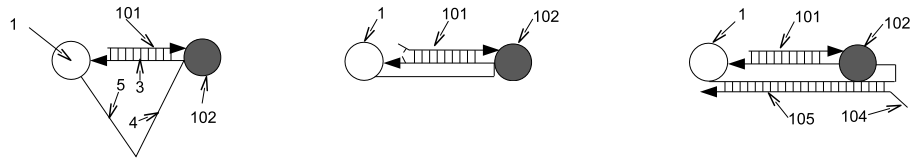

(a) (b)

**Fig. 1.** The loop complex (a): the amplifiable fragment (2) and the loop link (1). The schematic representation (b) highlighting the 3 components of the amplifiable fragment

The amplification mix contains primers (101) which hybridize to the 3' part of the amplifiable fragment (3). They can be further extended by DNA polymerase (102) present in the mix, see Fig. 2(a). The loop link (1) length is small compared to the length of the DNA fragment. The DNA fragment is also short in regards to the stiffness of double stranded DNA. In conditions where the biochemical replication reactions can take place, double stranded DNA molecules shorter than 300 – 500 nucleotides are too stiff for their extremities to come into close proximity. In other words, a circular DNA molecule shorter than 300 nucleotides cannot exist in full double stranded form, but is found as stretches of double stranded portions separated by single stranded portions. This is true also for the loop complexes used in ICLEDA. At some point the complex will be composed from a single stranded DNA having $n$ nucleotides from the 5' end of the amplifiable fragment, a double-stranded DNA corresponding to the 3' part of the amplifiable fragment, the extended primer and the linking loop of special length.

Since the two extremities of the amplifiable fragment are linked to each other this gives a geometric constraint for the loop. In order to continue the reaction either the single strand part should be extended to the maximum or the double stranded part should open at the opposite extremity. At some level of tension

[4] Presently, this patent is in the public domain.




(a) (b) (c)

**Fig. 2.** The amplification process: primer extension (a); maximum stretch of amplifiable fragment and the opening of the 5' end of the double strand (b); a second amplification started (c). Notation: link loop (1), amplifiable fragment (3,4,5), DNA polymerase (102), (extended) primer (101), single stranded fragment (104).

the energetic preference will be to continue the extension of the primer by DNA polymerase, while the opposite end will detach by Brownian motion. So, at the same time the double stranded fragment will be opened at 5' part and one nucleotide will be added by DNA polymerase. However it should be noted that the number of nucleotides on the double stranded part remains unchanged, due to the geometric constraints of the loop.

Since no more nucleotides are bound at the 3' end of the amplifiable fragment (3) at some point it becomes accessible for a hybridization with a new primer, see Fig. 2(b). The extension is blocked when it reaches the end on the amplifiable fragment (105) because of the presence of non-natural nucleotides in the link, see Fig. 2(c).

Now we remark that if in the mix a fragment of a single stranded DNA that matches the 3' part of the amplifiable fragment is present, then it can stick to the amplifiable fragment as shown on Fig. 3(a). We call such a strand a *trigger*. When a trigger is attached to the loop complex, no amplification can be done. A trigger can be detached from the loop complex by an *activator* that matches by its 3' end a part of the trigger strand as shown on Fig. 3(b). Once bound to the trigger the activator can be extended by DNA polymerase and this will release the trigger, so the loop complex will be able to start the amplification process.

### 2.2 Formalization

By a DNA-like alphabet $V$ we mean an alphabet with $2n$ letters, $n \geq 1$, of the form $V = \{a_1, \ldots, a_n, \bar{a}_1, \ldots, \bar{a}_n\}$. Letters $a_i$ and $\bar{a}_i$, $1 \leq i \leq n$ are said to be complementary letters. The terminology originates from the basic DNA alphabet $\{A, G, C, T\}$, where the letters $A$ and $G$ have their complements $T$ and $C$. By $h_w$ we denote the letter-to-letter endomorphism of a DNA-like alphabet $V$ mapping each letter to its complementary letter; $h_w$ is also called the Watson-Crick morphism. Let us denote by $rev$ the function yielding the reverse of a

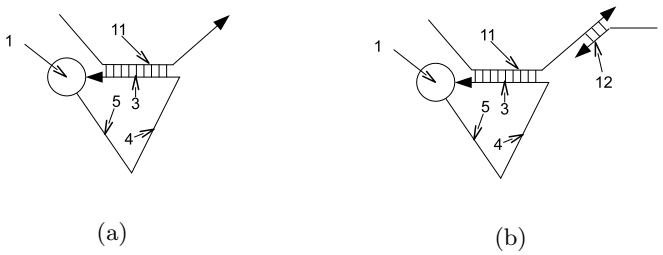



**Fig. 3.** The loop complex blocked by a trigger (a) and the hybridization with further extension used to remove the trigger (b). Notation: link loop (1), amplifiable fragment (3,4,5), trigger (11), DNA strand used to release the trigger (12)

word. Then the Watson-Crick complementarity function $\rho : V^* \to V^*$ is defined as $\rho(u) = rev(h_w(u))$.

We shall abstract DNA molecules as words over a DNA-like alphabet. By convention we will consider that they are read in $5'$ to $3'$ direction, i.e. the leftmost symbol of the word corresponds to the first nucleotide at the $5'$ end of the molecule. For the purpose of this paper the considered words will be partitioned into two kind of parts: sensitive and neutral. Sensitive parts of different words can eventually interact with each other, while the neutral parts can never interact with other sensitive or neutral ones. So in some sense, the neutral parts are just a separator for sensitive parts. We shall mark this difference between sensitive and neutral parts of words in the notation: sensitive parts will be denoted by capital letters and neutral parts will be denoted by lower case ones. As mentioned before we are not interested by the concrete symbols in words, but rather by the interleaving of sensitive and neutral parts.

Let $V$ be a DNA-like alphabet. We denote by $Sensitive(V)$ the letters that are used to label sensitive parts (by convention in latin uppercase) and by $Neutral(V)$ the letters used to label neutral parts (by convention in latin lowercase). We will assume that in any word two neutral letters (parts) cannot be adjacent to each other.

We shall formalize the partial annealing of DNA sequences. For this we consider that DNA molecules can bind together at some point if they share a complementary sequence of sensitive parts at that place. Now in order to be able to write the resulting object in a linear way, the binding points will be identified by a unique number and the corresponding sensitive parts of both molecules will be marked by this identifier. We shall use the notation $[A]_i$ to denote that $A$ is marked by $i$. For example, $[\![x[A]_1yAz, tBp[\bar{A}]_1]\!]$ indicate that molecules $xAyAz$ and $tBp\bar{A}$ are partially hybridized on the first sensitive part $A$ of $xAyAz$. Since indices are unique for each binding point, each complex molecular structure corresponds to an equivalence class grouping several indices, hence we can identify it by using only one of its representatives. We denote by $[V]$ the infinite extension of the alphabet $V$ containing normal and integer marked symbols.

Now we can define the operation *Glue* as follows.

$$
\begin{aligned}
Glue(u,v) = \{ [\![ u', v' ]\!] \mid \text{ where } \\
& u = \alpha_1 A_1 \dots A_k \alpha_2,\ v = \beta_1 \bar{A}_k \dots \bar{A}_1 \beta_2, \\
& u' = \alpha_1 [A_1]_i \dots [A_k]_i \alpha_2,\ v' = \beta_1 [\bar{A}_k]_i \dots [\bar{A}_1]_i \beta_2, \\
& i > 0,\ k \geq 1,\ \alpha_1, \alpha_2, \beta_1 \beta_2 \in [V]^*, \\
& \text{such that there is no } \alpha_1', \alpha_2', \beta_1', \beta_2' \in Sensitive(V)^*, \alpha_1' \alpha_2' \beta_1' \beta_2' \neq \lambda \\
& \text{satisfying } u = \alpha_1'' \alpha_1' A_1 \dots A_k \alpha_2' \alpha'', v = \beta_1'' \beta_1' \bar{A}_k \dots \bar{A}_1 \beta_2' \beta_2'' \\
& \text{and having } \alpha_1' A_1 \dots A_k \alpha_2' = \rho(\beta_1' \bar{A}_k \dots \bar{A}_1 \beta_2') \ \}.
\end{aligned}
$$

We remark that $Glue(u, v)$ returns all possible molecules that are obtained by a partial annealing of $u$ and $v$. The condition stated in the definition also requires the annealed part to be non-extensible, i.e. there should be no possibility to anneal more adjacent regions. We also remark that already marked symbols cannot be used for annealing. We will use the notation $u \otimes v$ to denote $Glue(u, v)$ and we will also extend this notation to circular strings and languages in a traditional way. We observe that $\otimes$ operation is commutative and in some cases associative, e.g. when there is no competition for gluing sites (there is no intersection between sensitive parts of different words), so it can be naturally extended to any number of operands. This corresponds to the fact that $\otimes$ models a situation in which the components are independent and may freely combine or not and in all possible combinations.

Now we shall define the operation that will simulate the strand displacement. We shall consider two cases: (1) when the displacing strand contains the full annealed sequence of the displaced one and (2) when the displacing strand starts with symbol(s) complementary to the free positions just before the attachment of the displaced strand. The latter case corresponds to a displacement by the sequence extended in the $5'$ (left) direction.

First, we define the *full displacement* operation $Displace_f$.

$$
\begin{aligned}
& Displace_f(u,v) = \{ u', v' \mid \text{ where there exist } i, j > 0 \text{ such that} \\
& u = [\![ \alpha_1 A_1 \dots A_n [B_1]_i \dots [B_m]_i \alpha_2,\ \beta_1 [\bar{B}_m]_i \dots [\bar{B}_1]_i \beta_2 ]\!], \\
& v = \gamma_1 \rho(A_1 \dots A_n B_1 \dots B_m) \gamma_2, \\
& u' = [\![ \alpha_1 [A_1]_j \dots [A_n]_j [B_1]_j \dots [B_m]_j \alpha_2,\ \gamma_1 [\bar{B}_m]_j \dots [\bar{B}_1]_j [\bar{A}_n]_j \dots [\bar{A}_1]_j \gamma_2 ]\!], \\
& v' = \beta_1 B_m \dots B_1 \beta_2 \ \}.
\end{aligned}
$$

Next, we define the *extension displacement* operation $Displace_e$.

$$
\begin{aligned}
& Displace_e(u,v) = \{ u', v' \mid \text{ where there exist } i, j > 0 \text{ such that} \\
& u = [\![ \alpha_1 A_1 \dots A_n [B_1]_i \dots [B_m]_i \alpha_2,\ \beta_1 [\bar{B}_m]_i \dots [\bar{B}_1]_i \beta_2 ]\!], \\
& v = \rho(A_1 \dots A_n) \gamma, \\
& u' = [\![ \alpha_1 [A_1]_j \dots [A_n]_j [B_1]_j \dots [B_m]_j \alpha_2,\ [\bar{B}_m]_j \dots [\bar{B}_1]_j [\bar{A}_n]_j \dots [\bar{A}_1]_j \gamma ]\!], \\
& v' = \beta_1 B_m \dots B_1 \beta_2 \ \}.
\end{aligned}
$$

Finally, we define the displacement operation *Displace*.

$$Displace(u, v) = Displace_f(u, v) \cup Displace_e(u, v).$$

We shall use the notation $u \bigoplus v$ to denote $Displace(u, v)$. We observe that $\bigoplus$ operation is commutative and in some cases associative (when there is no competition for the displacement positions), so it can be naturally extended to any number of operands.

The loop complex can be formalized as a circular word ${}^{\sim}\bar{F}u\bar{R}$ where $u$ is the neutral part and $\bar{F}$ with $\bar{R}$ are the sensitive ones, corresponding to the parts 4,3,5 on Fig. 1(b). Its functioning can be described by the following equation defining the unary operation *Produce*:

$$Produce({}^{\sim}\bar{F}u\bar{R}) = R\bar{u}F. \tag{1}$$

A trigger can be formalized as $t\bar{A}Fw$, where $w$ and $z$ are the neutral parts and $\bar{A}$, $F$ are the sensitive ones. We remark that glue and displacement operations can be applied for the case of circular strings exactly in the same manner as for the linear one. Hence, the attachment of the trigger to the loop complex can be described as follows:

$${}^{\sim}\bar{F}u\bar{R} \bigoplus t\bar{A}Fw \;\vdash\; {}^{\sim}\bar{F}u\bar{R} \otimes t\bar{A}Fw. \tag{2}$$

where $u$, $t$ and $w$ are neutral. From subsection 2.1, the result of (2) blocks the application of (1).

The loop can be unblocked by an activator of form $xA\bar{F}y$ or $xA$:

$${}^{\sim}\bar{F}u\bar{R} \otimes t\bar{A}Fw \bigoplus y\bar{F}Ax \;\vdash\; \{{}^{\sim}\bar{F}u\bar{R}, t\bar{A}Fw \otimes y\bar{F}Ax\}, \tag{3}$$

$${}^{\sim}\bar{F}u\bar{R} \otimes t\bar{A}Fw \bigoplus Ax \;\vdash\; \{{}^{\sim}\bar{F}u\bar{R}, t\bar{A}Fw \otimes \bar{F}Ax\}. \tag{4}$$

A *string displacement* system is the construct $D = (V, A_c, A_l)$, where $V$ is a DNA-like alphabet, $A_l \subseteq (V \cup \{\otimes\})^*$ is the set of initial partially annealed linear strings and $A_c \subseteq {}^{\sim}V(V \cup \{\otimes\})^*$ is the set of loop complexes eventually partially annealed with linear strings. The configuration of the system is a pair $C = (C_c, C_l)$, where as for axioms $C_l$ is the set of current partially annealed linear strings and $C_c$ is the current set of circular strings eventually partially annealed with linear ones. We pass from configuration $C$ to $C'$ (denoted as $C \vdash C'$) if
$C'_c = Displace(u, v), u \in C_c, v \in C_l$ and
$C'_l = Displace(u, v) \cup Produce(z), u \in C_c \cup C_l, v \in C_l, z \in C_c$.

The language generated by a displacement system $D$ is the set of all "pure" strings (not being part of a complex) that can be obtained from the axioms:

$$L(D) = \{w \in V^* \mid (A_c, A_l) \vdash^* (F_c, F_l) \text{ and } w \in F_l\}.$$

It is clear that for any string displacement system $D$ we have that $L(D)$ is finite.

## 3 Implementing boolean functions

In this section we show how it is possible to implement boolean functions using the operations introduced in the previous section.

We start by a remark that if a loop $^{\sim}\bar{F}u\bar{R}$ is unblocked, then there will be an **unbounded** number of copies of $R\bar{u}F$. This assumption results from the observation that once started, the amplification could produce a large enough number of resulting molecules, even if the loop is blocked again afterwards.

This implies that we can consider that initially all loops are blocked, otherwise we substitute them by a large number of DNA molecules corresponding to their result. So the computation in such a system consists in unblocking some loops in some order. This corresponds in a direct manner to asynchronous boolean circuits where the electrical impulses are propagated in the circuit. The signals we use are always identified by the part at the beginning of the molecule, *i.e.* a signal $\mathcal{A}$ will be given by the string $Aw$ for some $w \in V^*$.

It is known that every boolean function of $n$ variables can be implemented by a boolean circuit using AND, OR and NOT gates. It is possible to eliminate the NOT gate by considering the dual-rail encoding, i.e. that only "true" signals can circulate in the circuit, corresponding to the transition over the wire that we call *signal*. This encoding also allows to consider asynchronous circuits, especially important for the DNA computing area. The positional information can be handled by distinguishing signals (wires) and considering that each gate waits for concrete input signals and produces a unique output signal when all input signals are present.

The input of the circuit is not the true or false value for the same variable $x$, but rather a signal for $x$ or for $\neg x$. The output is also modified: instead of a single output having one of the values true or false, there are two outputs (marked by true and false) and a presence of a signal in some of the outputs indicate that the output value of the circuit is true or false. If we consider that the two output nodes are combined into a fictive output node then such a circuit is a DAG with the root being the output node and the leaves being the input variables and their negations. For a boolean formula $\phi$ such a modified circuit can be constructed by a superposition of two circuits, one computing $\phi$ and the other one computing $\neg\phi$.

We remark an important similarity between traditional electronic implementation of boolean circuits and our implementation: if a signal (represented by an electric charge in electronics and by DNA molecule in our case) appears at some moment during the computation, then it is sufficiently strong and does not disappear in the consequent steps. This allows us to make a direct analogy between two implementations and use similar construction techniques. This is different from other approaches of simulation of circuits by DNA computing, as we do not need additional amplification phase anymore.

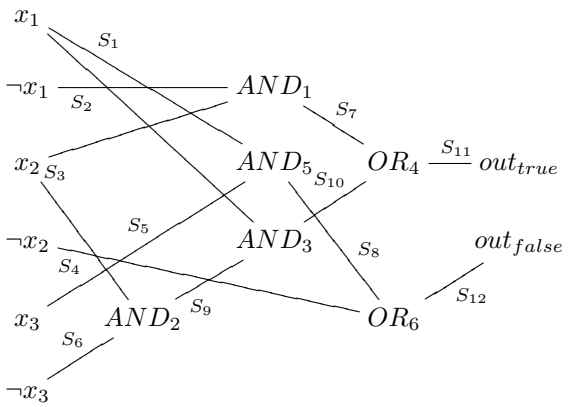
More precisely, let $f : B^n \to B$, $B = \{0, 1\}$ be a boolean function and let $D = (V, E, F)$ be the circuit implementing this function (where $V = \{1, \ldots, n\}$ is the set of vertices, $E \subseteq V \times V$ is the set of edges and $F : V \to \{x_p, \neg x_p, AND_m, OR_k, NOT_q, out\}$, $1 \leq p, q, m, k \leq n$ is the function that labels vertices of the circuit).

Then for any inner node $y$ such that $F(y) = OR_m$ (resp. $F(y) = AND_m$) we construct an OR loop (resp. AND loop) as discussed in 3.1 (resp. 3.2). The final gate will send the signal to the output node (the root of the circuit ($F(x) = out$)). A similar construction should be performed for $\neg f$. The final assembly is the union of these two circuits.

In order to compute the result in the initial configuration signals corresponding to $X_k$ (where $X_k$ is either $x_k$ or $\neg x_k$) should be introduced.

We give below an example of such a construction for the following function: $f(x_1, x_2, x_3) = (\neg x_1 \wedge x_2) \ \vee \ (x_1 \wedge x_2 \wedge \neg x_3)$.

The asynchronous dual-rail circuit computing $f$ is given below. We numbered the gates and labeled the edges going out from the same left node (corresponding to a concrete signal produced by the corresponding gate).




Having in mind that the true value of some node is represented by the presence of the corresponding signal (that labels the edge), it becomes clear that this construction can be directly implemented using loop complexes and corresponding signals by 4 AND gates and 2 OR gates.

The signals $S_1 - S_6$ correspond to the input values and the signals $S_{11}$ and $S_{12}$ to the output. So the computation starts by giving input signals (taking care of not having an input $x$ and $\neg x$ at the same time). Then the gates will act in cascade and one of two output signals ($S_{11}$ if $f$ is true or $S_{12}$ if $f$ is false) will be obtained.

### 3.1 The *OR*-gate

We base our construction of the fact that the outcome of rule (1) can be interpreted as the production of the signal $\mathcal{R}$.

In this condition, if the trigger $t\bar{A}Fw$ is initially attached to a loop complex $^{\sim}\bar{F}u\bar{R}$, forming $^{\sim}\bar{F}u\bar{R} \otimes t\bar{A}Fw$, rule (3) tells us that introducing the molecule $Ax$ we obtain $^{\sim}\bar{F}u\bar{R}$ and $t\bar{A}Fw \otimes \bar{F}Ax$. As now rule (1) applies to $^{\sim}\bar{F}u\bar{R}$: accordingly, we get the signal $\mathcal{R}$.

It is not difficult to obtain a similar sequence of deductions with $^{\sim}\bar{G}v\bar{R}$ and the trigger $r\bar{B}Gs$. Introducing the molecule $By$ we shall also get $\mathcal{R}$ by applying the rules. The final construction for the gate is shown on Fig. 4.

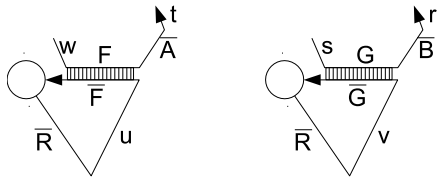



**Fig. 4.** The simulation of the OR gate.

We remark that the construction above can be extended to an $n$-ary OR gate. This gives the possibility to simulate semi-unbounded fan-in circuits.

### 3.2 The *AND*-gate

We can simulate an AND gate by considering two active regions on the loop, *i.e.* loops of form $^{\sim}\bar{F}_A u\bar{F}_B v\bar{R}$, where $\bar{F}_A$ and $\bar{F}_B$ are active zones for triggers having $A$ and $B$. Then the loop is blocked by two molecules as follows $^{\sim}\bar{F}_A u\bar{F}_B v\bar{R} \otimes t\bar{A}F_A w \otimes q\bar{B}F_B s$, see Fig. 5(a). Now if both activators $Ax$ and $By$ are present, then the loop complex can be unblocked:

$$
\begin{aligned}
^{\sim}\bar{F}_A u\bar{F}_B v\bar{R} \otimes t\bar{A}F_A w \otimes q\bar{B}F_B s \bigoplus Ax \bigoplus By \;\vdash \\
\vdash\; \{^{\sim}\bar{F}_A u\bar{F}_B v\bar{R} \otimes t\bar{A}F_A w \bigoplus Ax,\; q\bar{B}F_B s \otimes \bar{F}_B By\} \;\vdash \\
\vdash\; \{^{\sim}\bar{F}_A u\bar{F}_B v\bar{R},\; t\bar{A}F_A w \otimes \bar{F}_A Ax,\; q\bar{B}F_B s \otimes \bar{F}_B By\} \quad (5)
\end{aligned}
$$

In the above derivation the unblocking can start by the signal $Ax$, but following the commutativity of $\bigoplus$ it yields the same result.

It is clear that if only one of the activators $Ax$ or $By$ is present, then the loop complex is only partially unblocked (it has either the form $^{\sim}\bar{F}_A u\bar{F}_B v\bar{R} \otimes t\bar{A}F_A w$ or $^{\sim}\bar{F}_A u\bar{F}_B v\bar{R} \otimes q\bar{B}F_B s$) and cannot produce the resulting signal.

### 3.3 The initial *AND*-gate

Another variant of the AND gate is described in [8]. Like in the previous case it is also a complex of 3 molecules, however the loop complex is bound in only one place. Its construction is done in two stages: the loop complex $^{\sim}\bar{F}_A w\bar{R}$ is blocked by the trigger $F_A\bar{A}$. After that the molecule $\bar{B}A$ is added into the solution and it will stick to the $\bar{A}$ site. Hence, the complex $^{\sim}\bar{F}_A w\bar{R} \otimes \bar{A}F_A \otimes \bar{B}A$ will be

formed, see Fig. 5(b). We remark that since this complex is formed during the preparation stage, we can insure that no molecules $Ax$, $(x \neq B)$ or $tF_AAv$ are present in the solution.

Now during the computation the loop complex can be unblocked as follows:

$$\begin{aligned} {}^{\sim}\bar{F}_A w\bar{R} \otimes \bar{A}F_A \otimes \bar{B}A \bigoplus By \bigoplus Ax \ &\vdash \\ &\vdash \ \{{}^{\sim}\bar{F}_A w\bar{R} \otimes \bar{A}F_A,\ \bar{B}A \otimes \bar{A}By\} \bigoplus Ax \ \vdash \\ &\vdash \ \{{}^{\sim}\bar{F}_A w\bar{R},\ \bar{B}A \otimes \bar{A}By,\ \bar{A}F_A \otimes \bar{F}_A Ax\}. \quad (6) \end{aligned}$$

We remark that unlike the previous case this construction is not symmetric, *i.e.* first the signal $By$ is removing the molecule $\bar{B}A$ from the complex, freeing the site $\bar{A}$, which can be bound after that by the signal $Ax$ that finally unblocks the loop.


(a) (b)

**Fig. 5.** The simulation of the AND gate with two active regions on the loop (a) and with one active region on the loop (b)

## 4 Conclusions

In this article we present a formalization of a new method of DNA string amplification. This process lead us to the introduction of two operations on strings that permit to express the partial annealing and the strand displacement using a linear notation. The obtained model is very promising and it presents many mathematical challenges, especially because the introduced operations are not unary. Numerous extensions are possible, we cite [11] for a good overview of different possibilities. We would like to mention the possibility to use multisets or fuzzy sets instead of sets of strings. The motivation for such an extension is that the model allows us to produce a trigger and to stop the loop amplification. Another interesting idea is to consider molecules (strings) attached to a support

and use a washing procedure that would eliminate strings that are not attached or that will move unattached strings to some other compartment/membrane, allowing to further apply techniques from [10].

Another interesting outcome of this article is a new method for the simulation of boolean circuits. The use of ICLEDA offers many advantages like a single volume and unchanged reaction conditions. This implies that the corresponding implementation will not need any additional intervention. Moreover, since the loop complexes can be easily attached to the support it is possible to reuse the circuit by washing the tube and by introducing trigger molecules to block the loops. Another advantage of the method is that the signal molecules (corresponding to the true value of some gate) are of a small length; moreover, by introducing compartments it is possible to share some of the signals.

*Acknowledgements* Sergey Verlan would like to acknowledge the support of the ANR project SYNBOTIC.

## References

1. L. Adleman, Molecular computation of solutions to combinatorial problems. *Science*, **266**, 1994, 1021–1029.
2. H. Ahrabian, M. Ganjtabesh, A. Nowzari-Dalini, DNA algorithm for an unbounded fan-in Boolean circuit, *Biosystems*, **82**(1), 2005, 52-60.
3. M. Amos, Theoretical and experimental DNA computation. In: Natural Computing Series. Springer-Verlag, Berlin, Heidelberg, 2005.
4. A. Fire, S.Q. Xu, Rolling replication of short DNA circles. *Proc. Natl. Acad. Sci. USA.*, **92**(10), 1995, 4641–4645.
5. W. Gilbert, D. Dressler, DNA replication: the rolling circle model. *Cold Spring Harb. Symp. Quant. Biol.*, **33**, 1968, 473–484.
6. I.-H. Lee, J.-Y. Park, Y.-G. Chai, B.-T. Zhang, RCA-Based Detection Methods for Resolution Refutation. In J. Chen, J. H. Reif (Eds.), *DNA Computing, 9th Int. Workshop on DNA Based Computers, DNA9, Madison, WI, USA, June 1-3, 2003, Revised Papers*, *Lecture Notes in Computer Science*, 2943, 2004, 32-36.
7. M. Margenstern, P. Mayer, S. Verlan. DNA Circuits Based on Isothermal Constrained Loop Extension DNA Amplification. *CoRR*, **abs/1105.1424**, (2011).
8. P. Mayer, Méthode d'amplification d'acides nucléiques et ses applications, Brevet *INPI* N°2936246, September, 24, 2009. (*Method of amplification of nucleic acids and its applications*, French patent). `http://fr.espacenet.com/` as well as at `http://www.lacl.fr/verlan/data/brevetICEDA.pdf`
9. M. Ogihara, A. Ray, DNA-based self-propagating algorithm for solving bounded-fan-in Boolean circuits, In proceedings of *Third Conference on Genetic Programming, Morgan Kaufman Publishers, San Francisco*, (1998), 725–730.
10. Gh. Păun: *Membrane Computing. An Introduction.* Springer, 2002.
11. G. Păun, G. Rozenberg, and A. Salomaa. *DNA Computing: New Computing Paradigms.* Springer, 1998.
12. G.T. Walker, M.C. Little, J.G. Nadeau, D.D. Shank, Isothermal in vitro amplification of DNA by a restriction enzyme/DNA polymerase system. *Proc. Natl. Acad. Sci. USA*, **89**(1), 1992, 392-396.
13. D.Y. Zhang, M. Brandwein, T.C. Hsuih, H. Li, Amplification of target-specific, ligation-dependent circular probe. *Gene*, **211**(2), 1998, 277-85.