Chain Reaction Systems Based on Loop Dissociation of DNA

Keiichiro Takahashi[⋆], **Satsuki Yaegashi**[∗], **Atsushi Kameda**[∗] **and Masami Hagiya**[∗†]

[⋆]NovusGene Inc., 2-3 Kuboyama-cho, Hachioji-shi, Tokyo 192-8512, Japan
Olympus Corporation Utsugi Technical Research Institute
takahashi-k@novusgene.co.jp

[∗]Japan Science and Technology Corporation (JST-CREST)
yaegashi@lyon.is.s.u-tokyo.ac.jp, kameda@complex.eng.hokudai.ac.jp

[†]Department of Computer Science, Graduate School of Information Science and Technology,
University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
hagiya@is.s.u-tokyo.ac.jp

Abstract

In the field of DNA computing, more and more efforts are made for constructing molecular machines made of DNA that work in vitro or in vivo. States of some of those machines are represented by their conformations, such as hairpin and bulge loops, and state transitions are realized by conformational changes, in which such loops are opened. The ultimate goal of this study is to implement not only independent molecular machines, but also networks of interacting machines, called *chain reaction systems,* where a conformational change of one machine triggers a conformational change of another machine in a cascaded manner. A chain reaction system would result in a much larger computational power than a single machine in the number of states and in the complexity of computation. As a simple example, we propose a general-purpose molecular system consisting of logical gates and sensors. As a more complex example, we present a new idea of constructing a DNA automaton by a chain reaction system, which can have an arbitrary number of states.

1 Introduction

Seeman and Winfree's tile assembly model has made it possible to construct large structures of DNA by programmed self-assembly of basic components. The structures constructed so far include DNA nanotubes and planar patterns such as Sierpinski's triangle. Meanwhile, DNA nanomachines like the molecular tweezers [11] have come under the spotlight of nanorobotics. Those nanomachines realize finite state machines, where states are represented by their conformations, and conformational changes make state transitions.

The ultimate goal of this study is to construct general-purpose molecular systems consisting of interacting molecular machines. In contrast to self-assembly of DNA tiles, where static components only hybridize together, each component of such a system is a DNA machine that changes its state through interactions with other machines. In other words, we aim at constructing networks of DNA machines interacting with one another, where a conformational change of one machine triggers a conformational change of another machine in a cascaded manner. We call such networks of machines *chain reaction systems* in this paper. By a chain reaction system, it would be possible to realize information processing at the molecular level. Recently, molecular systems involving a chain reaction have been investigated by several other groups. For example, Dirks et al. have already demonstrated self-assembly of two stable hairpin species, triggered by an initiator strand, which may contain a DNA aptamer and become active when ATP binds the aptamer and exposes a sticky end [19]. In the chain reaction, one hairpin opening mutually triggers another hairpin opening and long linear complexes are yielded.

In the next section, we briefly introduce the design method that we have suggested to construct chain reaction systems. And in the subsequent sections, we introduce two examples of chain reaction systems. The first one is the chain reaction system simulating AND-OR circuits, where an AND gate is represented by a DNA machine including two bulge loops and an OR gate is composed of two hairpin structure molecules. The thermo- and photo-regulated hairpins [18], can be used as sensors that produce inputs to such logical circuits. As the second example, we present DNA automata, where both the transition rules and states would be implemented by DNA machines.

The figures are in colors. The pdf file of this paper, which shows the colors, is available at the following URL.

http://hagi.is.s.u-tokyo.ac.jp/pub/staff/hagiya/dna11/crs.pdf

2 Design Method of Chain Reaction Systems

There have been developed various methods for designing DNA sequences [4]. However, those methods are not enough for designing DNA machines and their networks as mentioned above. Developing a new and systematic design framework for constructing DNA machines is also a goal of our study [1, 2].

The framework for designing and implementing DNA machines advocated in this paper consists of the following three steps. We first pre-design candidate DNA sequences that fold into intended initial conformations. Secondly, we predict thermodynamic properties of the pre-designed sequences and select optimal ones. Finally, we actually verify behaviors of machines in laboratory experiments.

In the first step, we adopt the template method [5,6] that can be used to systematically generate a set of DNA sequences in which different sequences are guaranteed to have a certain number of mismatches. Briefly, in the template method, a DNA library X can be derived as $X = \tau \cdot E := \{\tau \cdot w \mid w \in E\}$, where τ is a mismatch-guaranteed template and E is an error-correcting code. The $\cdot$ operator stands for a bit-wise product as 1·1=G, 1·0=C (or 1·1=C, 1·0=G), 0·0=A and 0·1=T (or 0·0=T, 0·1=A). However, since the original template method requires all sequences to have the same length, we have modified the method so that templates of different lengths can be mixed to generate a set of DNA sequences of different lengths. By concatenating DNA sequences of different lengths generated by the extended template method, we can obtain candidate sequences that fold into target structures.

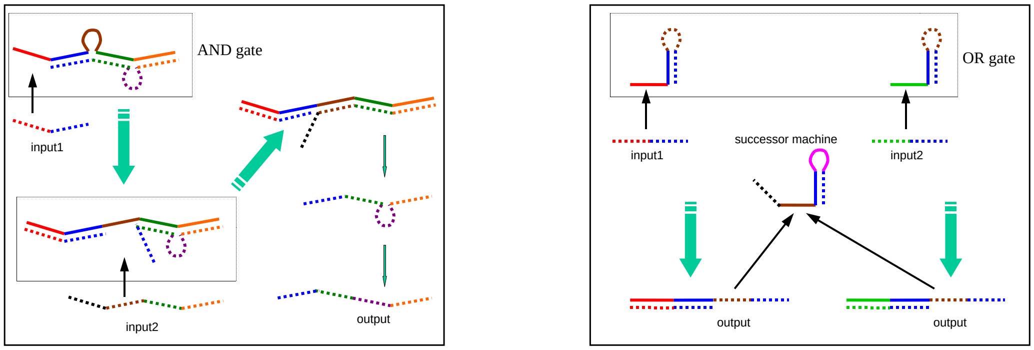
As for the second step, many existing programs for thermodynamic analysis of DNA, such as the Vienna Package [7], can only handle a single DNA sequence. Therefore, we have extended the Vienna Package to handle multiple sequences for both minimum free energies and partition functions [1–3]. Using the extended Vienna Package, we select optimal sequences from the candidates.

3 DNA logical Circuits

Logical circuits consisting of wired logic elements are one of the simplest models of computation, and hence DNA-based simulation of logical circuits has been investigated by many researchers in DNA computing. For example, the first DNA-based simulation of logical circuits is reported by

Ogihara and Ray [8], Amos et al. have proposed a DNA simulation of logical circuits using NAND gates in time proportional to the depth of the circuit [9], Carbone et al. have applied DNA tiles to the evaluation of circuits [10], and Seelig et al. have proposed the way of simulating circuits using hybridization catalysts [12].

Our chain reaction system realizing a DNA logical circuit consists of AND gate machines having two bulge loops and OR gate machines composed of two hairpin molecules. Figure 1 presents these logical gates. The AND gate contains regions that can hybridize with its inputs on the topside chain and its output as the downside chain. If the gate receives two input signals in the form of single-stranded molecules, it produces its output which will become an input to successive gates. The dotted black region on the second input prevents it from interacting with the first bulge loop (solid line in brown) before the first input breaks the stem region (blue), constructing a robust kinetic wall that prevents the system from rapidly moving to the equilibrium [17]. Meanwhile, the OR gate consists of two hairpin machines that have the same hairpin loops, which work as a converter of the input. An input that has the lead section complementary to that of either hairpin machine opens the hairpin and exposes the lead section of the output. This means that the gate converts the lead section of the input to that of the output.



(a) (b)

Figure 1: [AND Gate and OR Gate] The panel (a) shows the behavior of an AND gate. The AND gate is implemented by two strands that fold into a double-bulge structure. The panel (b) shows how an OR gate works. The OR gate consists of two hairpin molecules.

Inputs to those logical gates can be thermo- and photo-regulated hairpins, which we are now developing [18]. So far, we have verified that an input with a small hairpin that covers its lead section can be used as a thermo-sensor because the small hairpin is dissociated in high temperature. A small hairpin bearing azobenzenes in its stem can be used as a photo-sensor because UV light isomerizes the azobenzenes and dissociates the hairpin. In this way, we can construct a general-purpose molecular system consisting of logical gates and sensors, which combines various inputs and produces their boolean combination.

3.1 Design of an AND Gate

Hereafter, the term *unit* is used for a subsequence of a specified length on a given target sequence. When the given target sequence is scanned from $5'$ to $3'$ end, the pair of lengths of adjacent two units appearing on the sequence is called a *concatenation pattern,* denoted by $n * m$, where n is the length of the unit on the $5'$-end side and m is the length of the unit on the $3'$-end side. The set of all concatenation patterns of a given chain reaction system is used as the only information to generate optimal template tuples.

We are currently conducting some preliminary experiments of an AND gate. The actual sequences for the AND gate and its inputs are shown in Table 1. These sequences are designed using the extended template method. We selected $10*15$, $15*10$ and $15*15$ as the concatenation patterns for the system and we chose the template tuple $(0001001011, 111011101000111)$ by the extended template method, taking their GC-content into account. While the template 00010010011 is used for the loop regions and overhanging sections, the template 111011101000111 is relevant to the stem regions. Since the stems of length 15 on the AND gate have to stabilize the two bulge loops of length 10, the stem regions have a little extra GC-content of 67%. The actual DNA libraries of lengths 10 and 15 are obtained with BCH code, where the T_m values are about $55 \pm 1°$C and about $25 \pm 1°$C in our experimental conditions, respectively. Table 1 shows the sequences of the AND gate and the inputs.

TopB	:	5’-AAACTTCACC-GGGTCCGTGAATGGG-ATACTCTACTGC -CGCACCCTGATACCG-CGCTGGCAGATTCCG-(BHQ-1)-3’
DownB	:	5’-(FAM-)CGGAATCTGCCAGCG-AATCTAGACC -CGGTATCAGGGTGCG-CCCATTCACCGGACCC-3’
FO	:	5’-CCCATTCACGGACCC-GGTGAAGTTT-3’
SO	:	5’-CGGAATCTGCCAGCG-CGGTATCAGGGTGCG -GCAGTAGTAT-CCGTGCGTCTTAGCG-3’

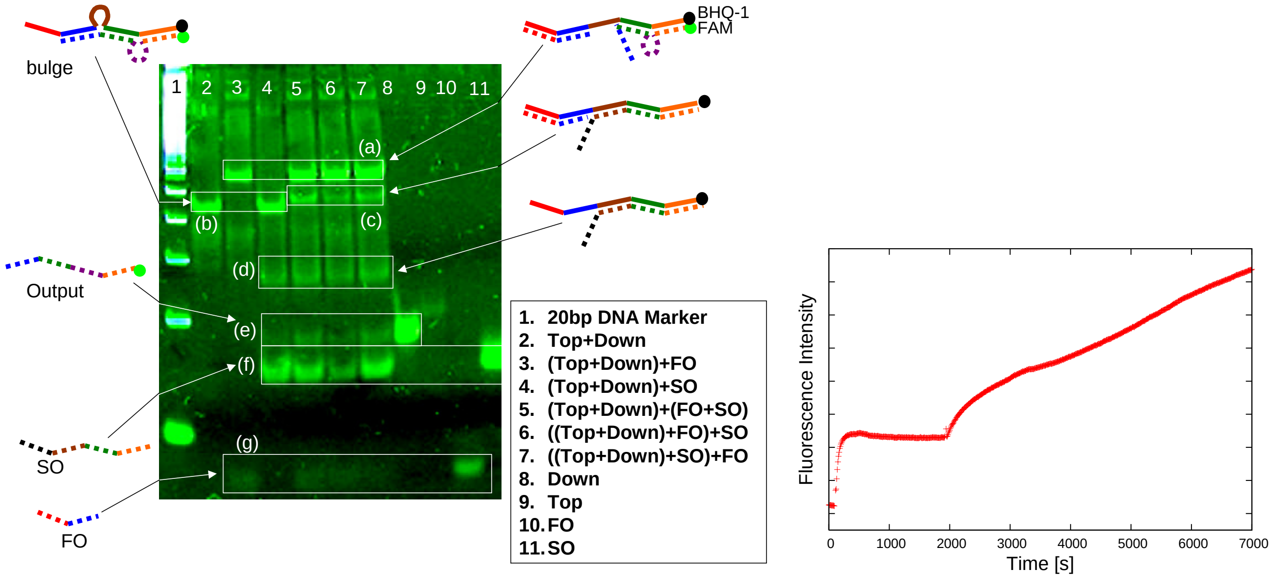
Table 1: [Sequences for AND Gate]: The sequences correspond the sequences in Fig. 1(a) in the same color. The upper two strands fold into a double bulge structure that functions as an AND gate when they hybridize together. For the sake of detecting behavior in the fluorescence experiment mentioned in the next subsection, the topside strand *TopB*is labeled at $3'$ end with the black hole quencher dye BHQ-1 and the downside strand *DownB* is labeled at $5'$ end with the reporter dye FAM. *FO* and *SO* are input molecules to the AND gate.

The sequences *TopBulge* and *DownB* compose the AND gate, where *TopB* has regions that can hybridize with the two inputs, and *DownB* works as the output molecule after receiving the inputs. The sequence named *FO* is the first input and the sequence named *SO* is the second one.

3.2 Experimental Results of an AND gate

By gel electrophoresis and fluorescence experiments, we have verified that the AND gate releases the output signal only when the two inputs exist.

Figure 2(a) shows the result of electrophoresis on 10% PAGE (non-denaturing gel). The boxes labeled with a lower-case letter on the gel classify the enclosed bands into the structures that we predict. The band on the cross of Lane 2 and the box (b) corresponds to the double-bulge structure of the AND gate. And the clear band on the cross of Lane 3 and the box (a) is the AND gate whose first bugle loop is opened by *FO*. Importantly, Lane 4 shows whether the output is produced only by the second input or not. In fact, the output molecules are rarely released from the AND gate, judging from the box (e) on which the output molecules are present. Through Lanes 5 to 7, the AND gate and the two inputs are applied to the gel. Each lane has almost the same appearance, but the order of mixing *FO* and *SO* is different in each lane. Lane 5 shows the result of the AND gate on receiving the two inputs at a time. In Lane 6, after hybridization between the AND gate and the first input, the second input is mixed. Conversely, after interaction between the AND gate and the second input, the first input is mixed in Lane 7. The important result is that the output is produced only when both of the inputs exist.



(a) (b)

Figure 2: [Experimental Results]: (a) The composition of each lane is given in the bottom right box, where each + means a hybridization process for thirty minutes. (b) The fluorescence intensity change from $0s$ to $2000s$ is by the reaction between the AND gate and the second input SO. FO is put at $2000s$, and all the molecules exist from $2000s$ to $7000s$.

The electrophoresis result is in good agreement with the fluorescence experiment. In order to confirm that the output molecules are derived only if the AND gate receives the first and second input molecules, we mixed the AND gate with the second input before giving the first one (Fig. 2(b)). While the reaction between the AND gate and the second input reaches saturation relatively fast in a small change of fluorescence intensity, the output molecules are released in a larger quantity when the two inputs exist.

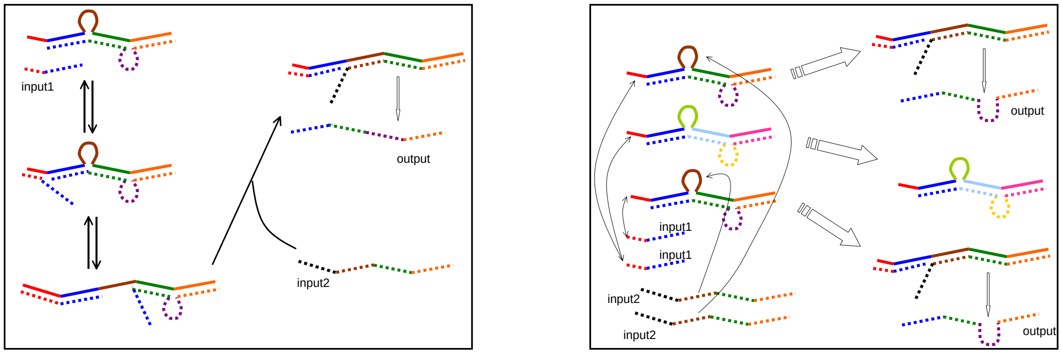
3.3 Discussion

We verified the function of the AND gate made of DNA molecules. However, the experiment unfortunately results in that most of the molecules of the AND gate hybridizing with the first input molecules are still intact even if the second input molecules exist as seen in the box (a) in Figure 2(a). The main reasons of this problem are as follows.

- There is a thermodynamic limitation in the system. Namely, since the system does not contain strands that the output molecules can hybridize with, the total energy change of the system is relatively small. Therefore, by preparing strands that are complementary to the final output molecules of a logical circuit, we should lower the energy of the system after the reaction.

- At the time of designing the sequences, our design framework did not take the structure of each sequence into consideration. Therefore, each sequence adopts a secondary structure that has many base pairs by itself. For example, the second input SO adopts the structure ((((..(((((.((((.........))))))))).....)))).((......)). (-8.86 kcal/mol) as the optimal one at 35°C.

The first problem does not arise in the approach using hybridization catalysts [12]. However, there is another problem concerning AND gates, which even the catalyst approach does not escape: the output will be attenuated if other AND gates that can receive the first signal exist. In our case, although the second input signal can be accepted by the AND gate only after the gate receives the first signal, the first input molecules can hybridize with any other AND gates that have the complementary lead section even if their second inputs do not exist.



(a) (b)

Figure 3: [Improvement of the AND Gate]: (a) Keeping the three states on the left equilibrium, the decay of signals will be prevented. (b) This figure shows the improved AND gates.

A possible solution to this attenuation problem is to keep hybridization and dissociation between the AND gate and the first signal under certain equilibrium as shown in the left part of Figure 3(a). If the second input signal does not exist in the system, the equilibrium will be kept. And if the second input molecule exists in the system, the AND gate with its first bulge loop opened by the first input will take the second input, moving the equilibrium towards hybridization. Figure 3(b) depicts the example of the improved strategy. The AND gate situated in the middle can accept the first input, but the reaction is in the equilibrium as mentioned above since the second input to the gate does not exit. Therefore, only the intended reaction will gradually proceed.

4 DNA automata

A wide variety of ideas for implementing finite automata by DNA have been reported. Among them, Gao et al. have proposed to use double-stranded molecules including one bulge loop which encode transition rules and several kinds of enzymes [13,14]. And Benenson et al. has actually implemented finite automata using more sophisticated encoding techniques [15]. And more recently, they have succeeded in analyzing mRNA levels of gene expression using their molecular computer *in vitro* [16].

Formally, a deterministic finite automaton (DFA) is a structure

$$M = (Q, \Sigma, \delta, s, F),$$

where Q is a finite set of *states*, Σ is a finite set called *input alphabet*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $s \in Q$ is the *initial state* and $F (\subset Q)$ is a set of *final states*.

Here we propose a new kind of deterministic finite automaton $(Q, \{0, 1\}, \delta, s, F)$ comprised of only DNA molecules, based on our chain reaction system. The automata receives external input molecules one by one, which are manually put into the solution, and makes transitions by chain reaction. The design of our DNA automaton is flexible in that it can have arbitrarily many states.

To construct such a chain reaction system which simulates a DFA, three kinds of component molecules encoding the corresponding components of the DFA are required: state molecules, transition rule molecules and state activation molecules. Each state molecule uniquely encodes a state $q \in Q$ and also contains the hairpins corresponding to the acceptable input symbols 0 and 1 (Fig. 5(a)(1)). A transition rule consists of two modules of DNA molecules, where one is comprised of a bulge loop, an interior loop and a hairpin loop, and the other consists of a hairpin loop and a 3-loop (Fig. 6(b)). The former module, triggered by the state molecule, hybridizes with the latter one, which then releases a state activation molecule at the completion of the transition. The state activation molecule identifies the next state and contains the hybridization site of input molecules (Fig. 5(a)(2)). Only state molecules which hybridize with the activation molecule represent the current state, and other state molecules remain inactive, that is, they cannot read any input symbols (Fig. 5(b)).

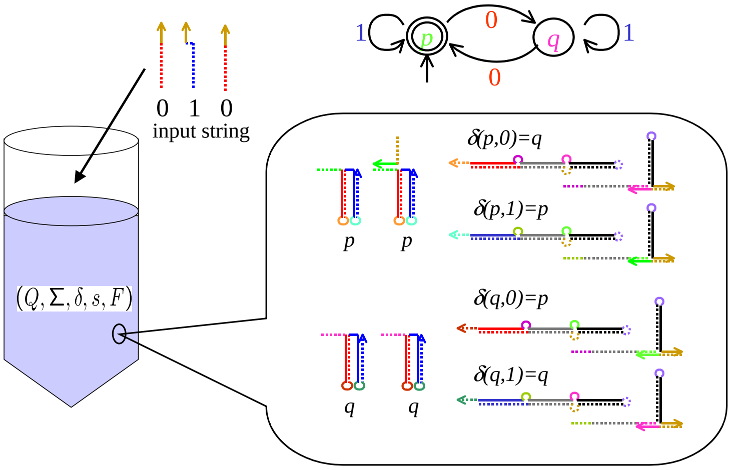


Figure 4: [DNA Automaton]: The solution dissolving state molecules and transition rule molecules processes an input binary string.

In order to explain how the chain reaction proceeds, let us show a simple DNA automaton (Fig. 4: for simplicity, blocker subsequences that hinder unwanted interference as found in [17] are omitted) which is found in the work of Benenson et al. [15]:

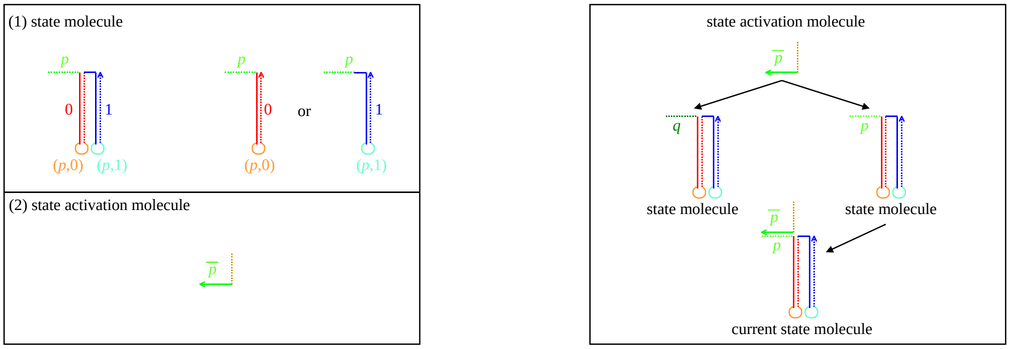
$$M = (Q, \Sigma, \delta, s, F),$$

where $Q = \{p, q\}$, $\Sigma = \{0, 1\}$, $s = p$, $F = \{p\}$ and $\delta : Q \times \Sigma \longrightarrow Q$ is specified by $\delta(p, 0) = q$, $\delta(p, 1) = p$, $\delta(q, 0) = p$ and $\delta(q, 1) = q$, as shown on the upper right panel of the figure.

At first, since the initial state p is the current state, the state activation molecule, which should be kept equimolar to the first input molecule, stochastically hybridizes with the state p molecule. If the molecule encoding the input symbol 0 is put into the solution, the initial state p takes in the symbol 0 via branch migration, opening the hairpin stem and exposing the region encoding the label $(p, 0)$ on the hairpin loop as in Figure 6(a). The region encoding $(p, 0)$ works as a specifier for the transition rule modules representing $\delta(p, 0) = q$. Note that the label $(p, 0)$ is hidden from the transition modules until the hairpin structure opens.

In the next step, the molecular interaction between the initial state p and the transition rule $\delta(p, 0) = q$ proceeds through the label $(p, 0)$ as in Figure 6(b). Figure 7 depicts the interaction of the two modules after the hybridization with the initial state. The first module hybridizing with the initial state uniquely communicates with the other module via the opened bulge structure (purple line), which encodes the label $\delta(p, 0)$ and which is complementary to the end part of the overhang on the second module.

At the end of the transition reaction, the state activation molecule for the next state q is released. And then, the activation molecule will also stochastically hybridize with the state q molecule and represent the current state. At this point, the next input molecule can be mixed. For the detection of acceptance, we can label the final state molecule with fluorescent dye.



(a)

(b)

Figure 5: [State Molecules and Current State Molecules]: (a) The panel (1) shows state molecules encoding state p. Each state molecule encodes the state (green) and the acceptable inputs 1 and 0 (red and blue, respectively). Orange and light blue lines on the hairpin loops encode the elements of $Q \times \{0, 1\}$. If the transition rule applied to the state p is a partial function, the state molecule consists of one hairpin stem and overhang (the two molecules on the right hand side). And the panel (2) describes a state activation molecule, which contains the subsequence $\overline{p}$ complementary to the state p (green). The molecule also has the hybridization site of input molecules (ocher). The state activation molecule is released from a transition rule molecule (Fig. 6(b)) if the state transition succeeds. (b) A state molecule that an state activation molecule hybridizes with works as the current state and receives an input molecule.

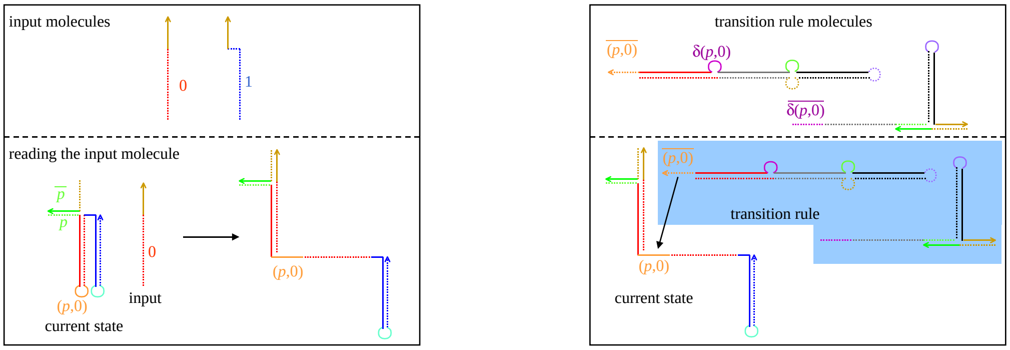
Although the lengths of stems and loops (or sticky ends) have to be carefully chosen, we expect that the DNA automaton can be stably constructed with units of lengths 10 and 25 for stems and loops, respectively, where concatenation patterns are 10*25, 25*10 and 25*25.

5 Conclusion and Future Work

In this paper, we proposed the notion of chain reaction systems based on interaction of multiple DNA machines. And as examples of chain reaction systems, we proposed and explained the DNA logical circuit and the DNA automaton. The former system would realize a multi-sensor system and the latter is expected to have more states than existing DNA automata.

We also showed the results of preliminary experiment for the AND gate. Implementation of an actual multi-sensor system consisting of multiple AND gates, OR gates and sensor machines remains as future work. We are currently refining AND gates so that they efficiently receive the pair of input signal molecules and release the output molecule. And also, the actual design of the DNA automaton is future work.

In order to construct a robust chain reaction system, it is essential to establish techniques to keep the current state of each machine stable unless it is given an input signal that takes part in



(a) (b)

Figure 6: [Input Molecules and Transition Rule Molecules]: (a) Input molecules are shown in the upper box. The red region encodes the input symbol 0 and the blue region encodes the input symbol 1. The ocher region is common to both the input molecules, which is complementary to the counterpart of state activation molecules. The bottom box shows the current state reading input 0. (b) The pair of two molecules in the upper box represents the transition rule $\delta(p, 0) = p$. The transition rule is only applied to the current state through the hybridization of $(p, 0)$ and $\overline{(p, 0)}$.

the cascading chain of transitions, and it is also important to make each cascading reaction rapid. For the first problem, we proposed the combination of

1. preventing loop opening by blocker substrands,
2. mismatches systematically guaranteed by the (extended) template method, and
3. optimal selection of sequences by the extended Vienna Package.

We expect that this combination of techniques is effective in constructing large chain reaction systems, although we are currently dealing with a single gate.

For the second problem, however, the current gate is very slow because of the inefficient reaction between SO and the gate. According to Figure 2(b) from $0s$ to $2000s$ and Lane 4 of Figure 2(a), we can observe that a certain amount of SO rapidly completes a reaction with the gate without FO (the cross of Lane 4 and the box(d) in Figure 2(a)), while the rest of SO remains independent of the gate as intended (the cross of Lane 4 and the box(f) in Fig. 2(a)). And some of the latter SO seems allowed to push away the output from the gate quite rapidly as soon as FO completely hybridizes with the gate (from $2000s$ to $4000s$ in Figure 2(b)). The efficiency of hybridization between FO and the gate is indicated in the boxes (a) and (g) of Figure 2(a). After $4000s$ in Figure 2(b), the remaining SO seems allowed to react with the gate very slowly, due to some kinetic trap in which SO is considered to form a certain structure with the gate and FO. As mentioned in Subsection 3.3, we consider that in order to avoid unwanted kinetic traps and facilitate the reaction we should enhance the driving force of the system and use small structure openers as well as select proper lengths of stems and loops.

Acknowledgments

The authors would like to thank anonymous reviewers for valuable comments. The work reported here is also supported in part by Grand-in-Aid for Scientific Research on Priority Area No.14085202, Ministry of Education, Culture, Sports, Science and Technology, Japan. And the authors deeply

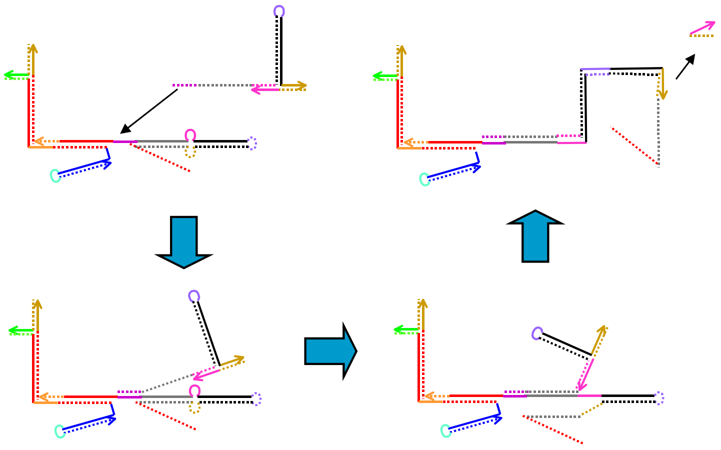


Figure 7: [State Transition]: The transition rule releases a state activation molecule for the next state.

thank Professor Satoshi Kobayashi (University of Electro-Communications) for designing DNA libraries that we used in our early experiments and advices on the extended template method.

References

[1] H. Uejima *et al.* Secondary Structure Design of Multi-state DNA Machines Based on Sequential Structure Transitions. *Ninth International Meeting on DNA Based Computers, LNCS, Springer* **2943**, 74–85 (2004)

[2] M. Kubota *et al.* Branching DNA Machines Based on Transitions of Hairpin Structures. *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*, 2542–2548 (2003)

[3] K. Takahashi *et al.* On Computation of Minimum Free Energy and Partition Function of Multiple Nucleic Acid Sequences. *FIT2004, Forum on Information Science and Technology*, 91–92 (2004)

[4] R. M. Dirks *et al.* Paradigms for computational nucleic acid design. *Nucleic Acids Res.*, **32**, 1392–1403 (2004)

[5] M. Arita *et al.* DNA Sequence Design Using Templates. *New Generation Computing*, **20**, 263–277 (2002)

[6] S. Kobayashi *et al.* On Template Method for DNA Sequence Design. *DNA Based Computers (DNA8), LNCS, Springer*, **2568**, 205–214 (2003)

[7] I. L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Res.* **31**, 3429–3431 (2003)

[8] M. Ogihara *et al.* Simulating Boolean circuits on a DNA computer. *Algorithmica,* **25**, 239–250 (1999)

[9] M. Amos *et al.* DNA simulation of Boolean circuits. *Proc. of the Third Annual Conference on Genetic Programming,* Morgan Kaufmann, 679–683 (1998)

[10] A. Carbone *et al.* Circuits and programmable self-assembling DNA structures. *PNAS*, **99**, 12577–12582 (2002)

[11] B. Yurke *et al.* A DNA-fuelled molecular machine made of DNA. *Nature*, **406**, 605–608 (2000)

[12] G. Seelig *et al.* DNA Hybridization Catalysts and Catalyst Circuits *DNA10, Tenth International Meeting on DNA Based Computers, Preliminary Proceedings*, 202–213 (2004)

[13] Y. Gao *et al.* DNA implementation of nondeterminism. *DNA Based Computers III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **48**, 137–148 (1999)

[14] M. Garzon *et al.* In vitro Implementation of Finite-State Machines. *Proc. Workshop on Implementing Automata (WIA '97)*, 56–74 (1998)

[15] Y. Benenson *et al.* Programmable and autonomous computing machine made of biomolecules. *Nature* **414**, 430–434 (2001)

[16] Y. Benenson *et al.* An autonomous molecular computer for logical control of gene expression. *Nature* **429**, 423–429 (2004)

[17] K. Takahashi *et al.* Preliminary Experiments on Hairpin Structure Dissociation for Constructing Robust DNA Machines. *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS'04)*, 285–290 (2004)

[18] K. Takahashi *et al.* Photo- and Thermo-Regulation of DNA Nanomachines. *submitted*

[19] R. M. Dirks *et al.* Triggered amplification by hybridization chain reaction. *PNAS* **101**, 15275–5278 (2004)