Intensive *In Vitro* Experiments of Implementing and Executing Finite Automata in Test Tube

Junna Kuramochi[1] and Yasubumi Sakakibara[2]

[1] Softbank BB Corporation, Japan
[2] Keio University, Department of Biosciences and Informatics,
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan
yasu@bio.keio.ac.jp

Abstract. We report our intensive *in vitro* experiments in which we have implemented and executed several finite-state automata in test tube. First, we employ the length-encoding technique proposed and presented in [4, 3] to implement finite automata in test tube. In the length-encoding method, the states and state transition functions of a target finite automaton are effectively encoded into DNA sequences, a computation (accepting) process of finite automata is accomplished by self-assembly of encoded complementary DNA strands, and the acceptance of an input string is determined by the detection of a completely hybridized double-strand DNA. Second, we design and develop practical laboratory protocols which combine several *in vitro* operations such as annealing, ligation, PCR, and streptavidin-biotin bonding to execute *in vitro* finite automata based on the length-encoding technique. We have carried laboratory experiments on various finite automata of from 2 states to 6 states for several input strings. To our knowledge, this is the first *in vitro* experiments that have succeeded to execute 6-states automaton in test tube.

1 Introduction

The finite-state automata (machines) are the most basic computational model in Chomsky hierarchy and are the start point to build universal DNA computers. Several works [1, 2, 3, 4] have attempted to develop finite automata *in vitro*. Benenson et al. [1] have successfully implemented the two state finite automata by the sophisticated use of the restriction enzyme (actually, *Fok*I) which cut outside of its recognition site in a double-stranded DNA. However, their method has some limitations for extending to more than 2 states. Yokomori et al. [4] have proposed a theoretical framework using length-encoding technique to implement finite automata on DNA molecules. Theoretically, the length-encoding technique has no limitations to implement finite automata of any larger states.

In this paper, we attempt to implement and execute finite automata of a larger number of states *in vitro*, and carry intensive laboratory experiments on various finite automata of from 2 states to 6 states for several input strings. To our knowledge, this is the first *in vitro* experiments that have succeeded to compute 6-states automaton in test tube.

A. Carbone and N.A. Pierce (Eds.): DNA11, LNCS 3892, pp. 193–202, 2006.

2 Methods

2.1 Length-Encoding Method to Implement Finite-State Automata

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a (deterministic) finite automaton, where Q is a finite set of states numbered from 0 to k, Σ is an alphabet of input symbols, δ is a state-transition function such that $\delta : Q \times \Sigma \longrightarrow Q$, q_0 is the initial state, and F is a set of final states. We adopt the length-encoding technique [4] to encode each state in Q by the length of DNA subsequences.

For the alphabet Σ, we encode each symbol a in Σ into a single-strand DNA subsequence, denoted $e(a)$, of fixed length. For an input string w on Σ, we encode $w = x_1 x_2 \cdots x_m$ into the following single-strand DNA subsequence, denoted $e(w)$:

$$5'\text{-}\ \ e(x_1)\underbrace{\mathtt{X}_1\mathtt{X}_2\cdots\mathtt{X}_k}_{k\text{ times}}e(x_2)\underbrace{\mathtt{X}_1\mathtt{X}_2\cdots\mathtt{X}_k}_{k\text{ times}}\cdots e(x_m)\underbrace{\mathtt{X}_1\mathtt{X}_2\cdots\mathtt{X}_k}_{k\text{ times}}\ \ \text{-}3',$$

where $\mathtt{X}_i$ is one of four nucleotides $\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}$, and the subsequences $\mathtt{X}_1\mathtt{X}_2\cdots\mathtt{X}_k$ are used to encode $k+1$ states of the finite automaton M. For example, when we encode a symbol '1' into a ssDNA subsequence GCGC and a symbol '0' into GGCC, and encode three states into TT, a string "1101" is encoded into the following ssDNA sequence:

$$5'\text{-}\ \ \overbrace{\mathtt{GCGC}}^{1}\mathtt{TT}\overbrace{\mathtt{GCGC}}^{1}\mathtt{TT}\overbrace{\mathtt{GGCC}}^{0}\mathtt{TT}\overbrace{\mathtt{GCGC}}^{1}\mathtt{TT}\ \ \text{-}3'$$

In addition, we append two supplementary subsequences at both ends for PCR primers and probes for affinity purifications with magnetic beads which will be used in laboratory protocol:

$$5'\text{-}\ \ \underbrace{\mathtt{S}_1\mathtt{S}_2\cdots\mathtt{S}_s}_{\text{PCR primer}}e(x_1)\mathtt{X}_1\mathtt{X}_2\cdots\mathtt{X}_k\cdots e(x_m)\mathtt{X}_1\mathtt{X}_2\cdots\mathtt{X}_k\underbrace{\mathtt{Y}_1\mathtt{Y}_2\cdots\mathtt{Y}_t}_{\text{probe}}\underbrace{\mathtt{R}_1\mathtt{R}_2\cdots\mathtt{R}_u}_{\text{PCR primer}}\ \ \text{-}3'.$$

For a state-transition function from state q_i to state q_j with input symbol $a \in \Sigma$, we encode the state-transition function $\delta(q_i, a) = q_j$ into the following complementary single-strand DNA subsequence:

$$3'\text{-}\ \ \underbrace{\overline{\mathtt{X}}_{i+1}\overline{\mathtt{X}}_{i+2}\cdots\overline{\mathtt{X}}_k}_{k-i\text{ times}}\overline{e(a)}\underbrace{\overline{\mathtt{X}}_1\overline{\mathtt{X}}_2\cdots\overline{\mathtt{X}}_j}_{j\text{ times}}\ \ \text{-}5'$$

where $\overline{\mathtt{X}}_i$ denotes the complementary nucleotide of $\mathtt{X}_i$, and $\overline{y}$ denotes the complementary sequence of y. Further, we put two more complementary ssDNA sequences for the supplementary subsequences at both ends:

$$3'\text{-}\ \ \overline{\mathtt{S}}_1\overline{\mathtt{S}}_2\cdots\overline{\mathtt{S}}_s\ \ \text{-}5',\qquad 3'\text{-}\ \ \underbrace{\overline{\mathtt{Y}}_1\overline{\mathtt{Y}}_2\cdots\overline{\mathtt{Y}}_t\overline{\mathtt{R}}_1\overline{\mathtt{R}}_2\cdots\overline{\mathtt{R}}_u}_{\text{biotinylated}}\ \ \text{-}5',$$

where the second ssDNA is biotinylated for streptavidin-biotin bonding.

Now, we put all those ssDNAs encoding an input string w and encoding state-transition functions and the supplementary subsequences of probes and PCR primers. Then, a computation (accepting) process of the finite automata M is accomplished by self-assembly among those complementary ssDNAs, and the acceptance of an input string w is determined by the detection of a completely hybridized double-strand DNA.

The main idea of length-encoding technique is explained as follows. Two consecutive valid transitions $\delta(h, a_n) = i$ and $\delta(i, a_{n+1}) = j$ are implemented by concatenating two corresponding encoded ssDNAs, that is,

$$3'\text{-}\ \underbrace{\mathtt{AAA}\cdots\mathtt{A}}_{k-h}\overline{e(a_n)}\underbrace{\mathtt{AAA}\cdots\mathtt{A}}_{i}\ \text{-}5',$$

and

$$3'\text{-}\ \underbrace{\mathtt{AAA}\cdots\mathtt{A}}_{k-i}\overline{e(a_{n+1})}\underbrace{\mathtt{AAA}\cdots\mathtt{A}}_{j}\ \text{-}5'$$

together make

$$3'\text{-}\ \underbrace{\mathtt{AAA}\cdots\mathtt{A}}_{k-h}\overline{e(a_n)}\underbrace{\mathtt{AAA}\cdots\mathtt{A}}_{k}\overline{e(a_{n+1})}\underbrace{\mathtt{AAA}\cdots\mathtt{A}}_{j}\ \text{-}5'.$$

Thus, the subsequence $\underbrace{\mathtt{AAA}\cdots\mathtt{A}}_{k}$ plays a role of "joint" between two consecutive state-transitions and it guarantees for the two transitions to be valid in M.

2.2 Designing Laboratory Protocols to Execute Finite Automata in Test Tube

In order to practically execute the laboratory experiments for the method described in the previous section, we design the following experimental laboratory protocol, which is also illustrated in Fig. 1:

0. Encoding: Encode an input string into a long ssDNA, and state-transition functions and supplementary sequences into short pieces of complementary ssDNAs.

1. Hybridization: Put all those encoded ssDNAs together into one test tube, and anneal those complementary ssDNAs to be hybridized.

2. Ligation: Put DNA "ligase" into the test tube and invoke ligations at temperature of 37 degree. When two ssDNAs encoding two consecutive valid state-transitions $\delta(h, a_n) = i$ and $\delta(i, a_{n+1}) = j$ are hybridized at adjacent positions on the ssDNA of the input string, these two ssDNAs are ligased and concatenated.

3. Denature and extraction by affinity purification: Denature double-stranded DNAs into ssDNAs and extract concatenated ssDNAs containing biotinylated probe subsequence by streptavidin-biotin bonding with magnetic beads.

4. Amplification by PCR: Amplify the extracted ssDNAs with PCR primers.

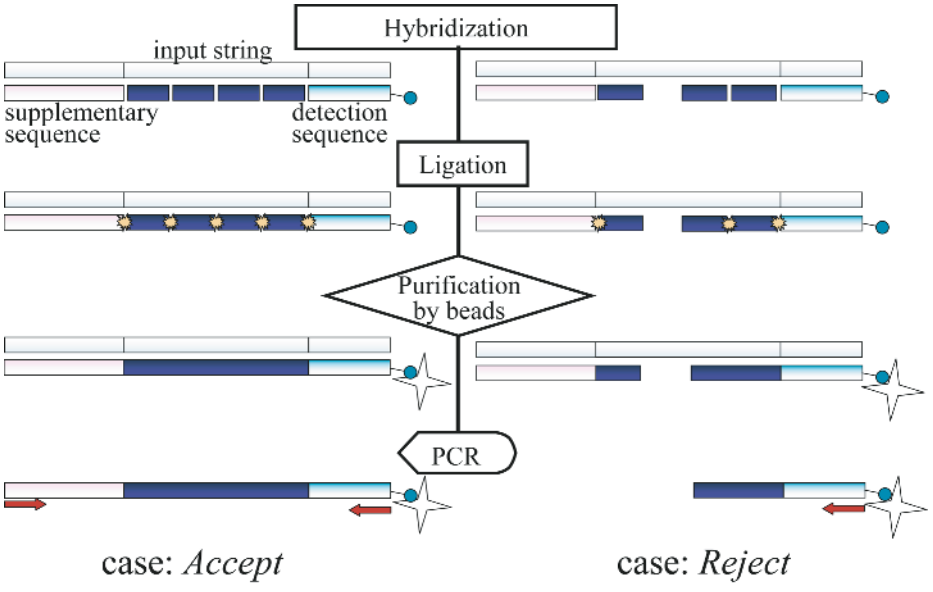

Fig. 1. The flowchart of laboratory protocol to execute *in vitro* finite automata which consists of five steps: hybridization, ligation, denature and extraction by affinity purification, amplification by PCR, and detection by gel-electrophoresis. The acceptance of the input string by the automata is the left case, and the rejection is the right case.

5. Detection by gel-electrophoresis: Separate the PCR products by length using gel-electrophoresis and detect a particular band of the full-length. If the full-length band is detected, that means a completely hybridized double-strand DNA is formed, and hence the finite automaton "accepts" the input string. Otherwise, it "rejects" the input string. In our laboratory experiments, we have used a "capillary" electrophoresis microchip-based system, called Bioanalyser 2100 (Agilent Technologies), in place of conventional gel-electrophoresis. The capillary electrophoresis is of higher resolution and more accurate than gel electrophoresis such as agarose gel.

Further, we have carefully designed DNA sequences encoding symbols, states, probes for affinity purification, PCR primers as follows. Two main factors for designing those DNA sequences are (1) Tm (melting temperature) to avoid mis-hybridizations and (2) to avoid forming secondary structures:

	DNA sequence	Tm
symbol '0'	GACGTTGGATGTGGG	50.165
symbol '1'	GCGTGTACGATGCAG	51.523
state	AAGCAGTTTT	23.641
probe	CTGGTTGCTTGTCCC	50.344
PCR primers	GCGTCTTGGTTGCTGAAATG	58.521
	CCGACTTCGTACGAGATTAG	55.481

3 Experiments

We have carried laboratory experiments on various finite automata of from 2 states to 6 states for several input strings.

3.1 2-States Automaton with Two Input Strings

Our experiments begin with a simple two-states automaton shown in Fig. 2 (left) with two input strings, (a) "1101" and (b) "1010". The language accepted by this automaton is $(10)^+$, and hence the automaton accepts the string 1010 and rejects the other string 1101.

The results of electrophoresis by Bioanalyser are displayed in the form of electropherogram (as shown in Fig. 3) which plots standard curve of migration time against DNA size where the x-axis is migration time and the y-axis is fluorescence intensity. They can also be displayed in gel-like image (as shown in Fig. 2 (right)). For these two input strings, the full-length DNA is of 190 bps (mer). Hence, if a band at position of 190 mer is detected in the result

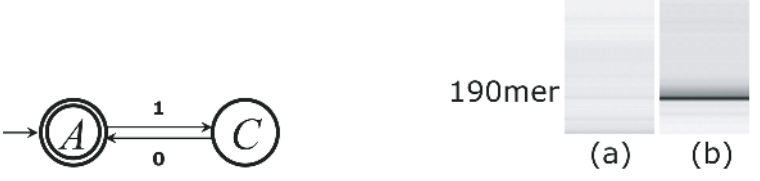

Fig. 2. (Left:) A simple 2-states automaton used for this experiment. (Right:) The results of electrophoresis are displayed in gel-like image. Lane (a) is for the input string 1101, and lane (b) for 1010. Since the full-length band (190 mer) is detected only in lane (b), we determine the automaton accepts only the input string (b) 1010.

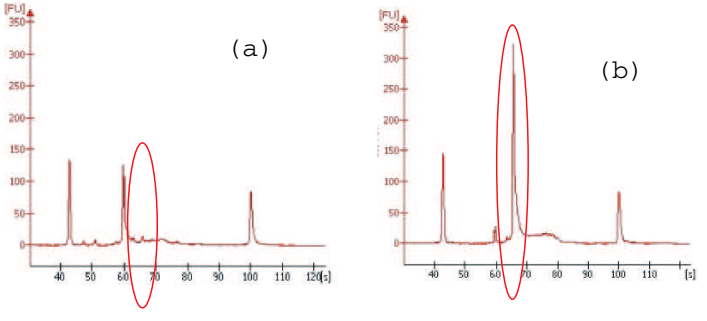

Fig. 3. The results of electrophoresis are displayed in the form of electropherogram where the vertical axis is fluorescence intensity. Peaks at the full-length position are marked with circles. Plot (a) is for the input string 1101, and plot (b) for 1010. A strong peak at the full-length position is detected only in plot (b), and hence we determine the automaton accepts the input string (b) 1010.

of electrophoresis, that means a completely hybridized double-strand DNA is formed, and hence the finite automaton "accepts" the input string.

Both figures 2 and 3 clearly show that our *in vitro* experiments have successfully identified the correct acceptance of this automaton for two input strings, and hence correctly executed the computation process of the automaton *in vitro*.

3.2 4-States Automaton with Three Input Strings

Our second experiment attempts 4-states automaton shown in Fig. 4 (upper left) for the three input strings (a) 1101, (b) 1110, and (c) 1010. This 4-states automaton accepts the language $(1(0\cup1)1)^*\cup(1(0\cup1)1)^*0$, and hence it accepts 1110 and 1010 and rejects 1101.

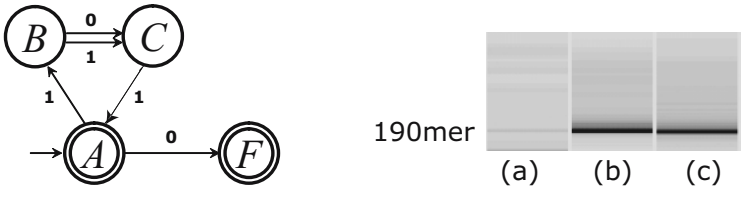

Fig. 4. (Left:) A 4-states automaton used for this experiment. (Right:) The results of electrophoresis are displayed in gel-like image. Lane (a) is for the input string 1101, lane (b) for 1110, and lane (c) for 1010. Since the full-length band (190 mer) is detected in lane (b) and (c), we determine the automaton accepts two input strings (b) 1110 and (c) 1010.

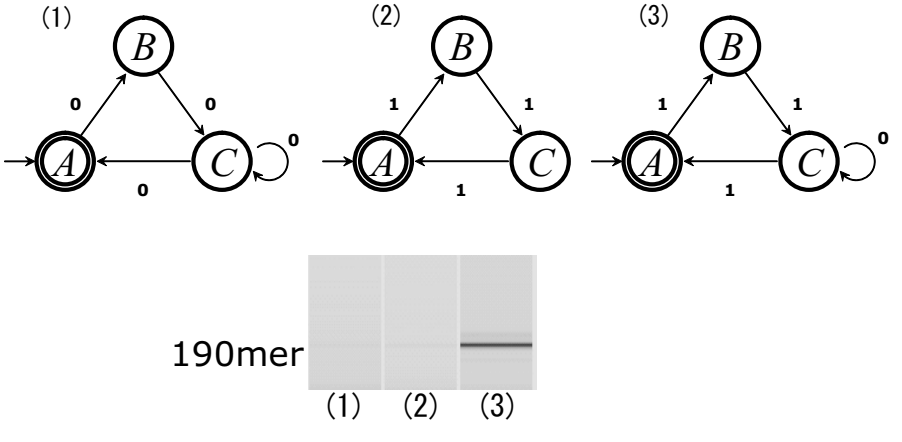

Fig. 5. (Upper:) Three different types of 3-states automata used for this experiment. (Lower:) The results of electrophoresis are displayed in gel-like image. Lane (1) is for the automaton (1), lane (2) for the automaton (2), and lane (3) for the automaton (3). Since the full-length band (190 mer) is detected only in lane (3), we determine the automaton (3) accepts the input string 1101.

The results are shown in Fig. 4 (upper right) in gel-like image. As in the first experiment, the full-length DNA is of 190 bps (mer). Bands at position of 190 mer is detected in lane (b) and lane (c). Hence, our *in vitro* experiments have successfully detected that the automaton accepts two input string (b) 1110 and (c) 1010.

3.3 Three 3-States Automata with One Input String

In this experiment, we execute three different types of 3-states automata shown in Fig. 5 (upper) for one input string "1101". The automaton (1) accepts the language 000*0, (2) accepts (111)*, and (3) accepts (110*1)*. Hence, the automaton (3) only accepts the input string 1101.

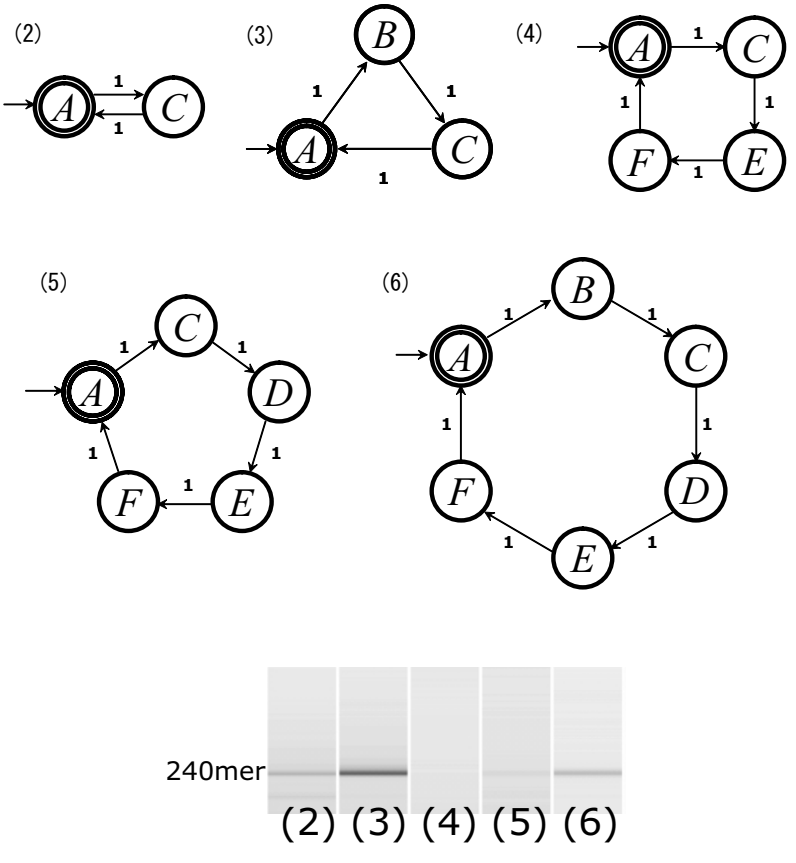

Fig. 6. (Upper:) Five different automata of from 2 states to 6 states used for this experiment. (Lower:) The results of electrophoresis are displayed in gel-like image. Lane (2) is for the automaton (2), (3) for (3), (4) for (4), (5) for (5), and (6) for (6). Since the full-length bands (240 mer) are detected in lane (2), (3) and (6), we determine the automata (2), (3) and (6) accepts the input string 111111.

The results are shown in Fig. 5 (lower) in gel-like image. Again, the full-length DNA is of 190 bps (mer). A band at position of 190 mer is detected only in lane (3). Hence, in our *in vitro* experiments, the automaton (3) has correctly accepted the input string 1101 and the automaton (1) and (2) have correctly rejected 1101.

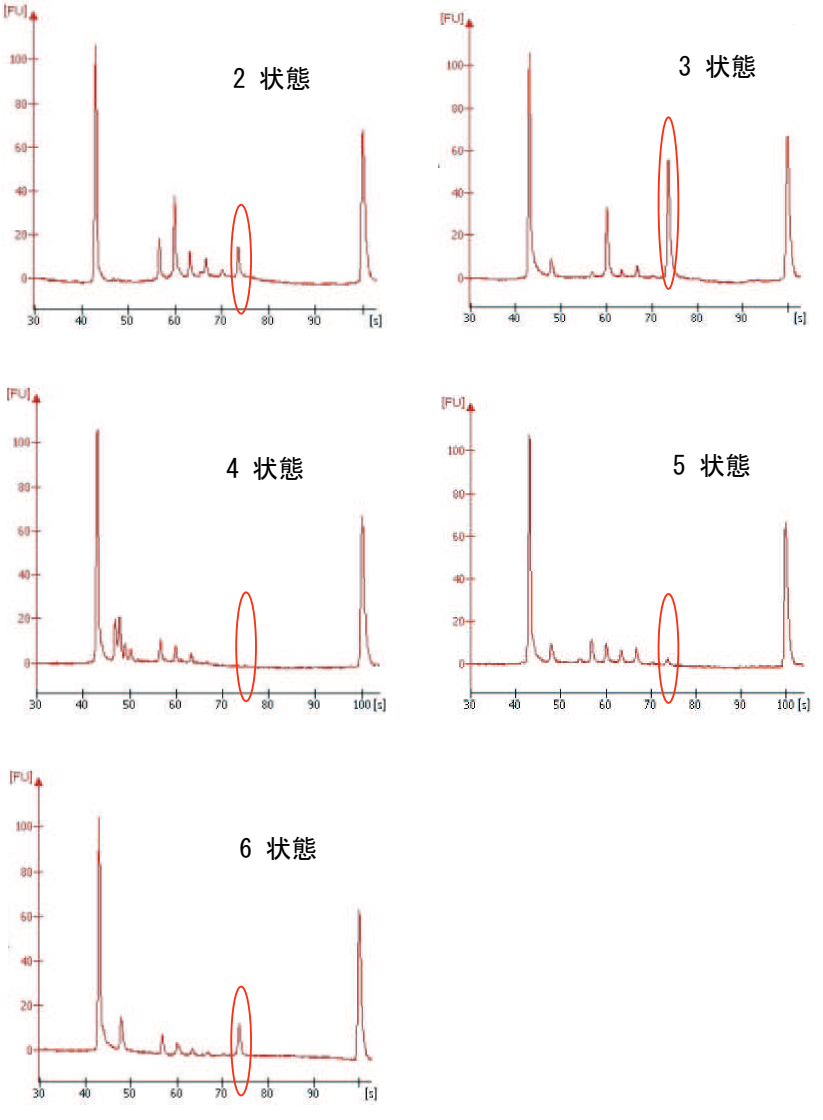

Fig. 7. The results of electrophoresis are displayed in the form of electropherogram. Peaks at the full-length position are detected in plot (2), (3) and (6).

3.4 From 2-States to 6-States Automata with One Input String "111111" of Length 6

Our final experiments are 5 different automata of from 2 states to 6 states shown in Fig. 6 (upper) for one input string "111111" of length 6. The automaton (2) accepts the language $(11)^*$, that is, strings with even numbers of symbol '1', (3) accepts the language $(111)^*$, strings repeating three times of 1s, (4) accepts the language $(1111)^*$, strings repeating four times of 1s, (5) accepts the language $(11111)^*$, strings repeating five times of 1s, (6) accepts the language $(111111)^*$, strings repeating six times of 1s. Since 6 is a multiple of 2, 3 and 6, the automata (2), (3) and (6) accept the input string 111111 of length 6.

The results are shown in Fig. 6 (lower) in gel-like image and in Fig. 7 in the form of electropherogram. For the input string 111111, the full-length DNA is of 240 bps (mer). Bands at position of 240 mer are detected in lanes (2), (3) and (6) in Fig. 6, and strong peaks at the full-length position are also detected in plot (2), (3), and (6) in Fig. 7. Hence, in our *in vitro* experiments, the automaton (2), (3) and (6) have correctly accepted the input string 111111 and the automaton (4) and (5) have correctly rejected 111111.

4 Discussions

Since the full-length ssDNAs contain repeated patterns on DNA sequences, PCR amplifications with such templates produce many unexpected and unnecessary products which become obstacles for the precise detections using electrophoresis. The use of fluorescence tags is a possible solution for this problem.

An interesting future work is execute our *in vitro* automata for multiple input strings in parallel.

Acknowledgements

This work is supported in part by Grant-in-Aid for Scientific Research on Priority Area No. 14085205. This work was also performed in part through Special Coordination Funds for Promoting Science and Technology from the Ministry of Education, Culture, Sports, Science and Technology, the Japanese Government, and a grant of Keio Leading-edge Laboratory of Science and Technology (KLL) specified research projects.

References

1. Benenson, Y., T. Paz-Ellzur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414, 430–434, 2001.
2. Păun, Gh., G. Rozenberg, and A. Salomaa. *DNA Computing*. Springer-Verlag, Heidelberg, 1998.

3. Sakakibara, Y. and T. Hohsaka. In Vitro Translation-based Computations. *Proceedings of 9th International Meeting on DNA Based Computers*, Madison, Wisconsin, 175–179, 2003.
4. Yokomori, T., Y. Sakakibara, and S. Kobayashi. A Magic Pot : Self-assembly computation revisited. *Formal and Natural Computing*, LNCS 2300, Springer-Verlag, 418–429, 2002.