

Design of Autonomous DNA Cellular Automata ^{*}

Peng Yin¹, Sudheer Sahu¹, Andrew J. Turberfield², and John H. Reif¹

¹ Department of Computer Science, Duke University
Box 90129, Durham, NC 27708-0129, USA.
{py, sudheer, reif}@cs.duke.edu

² University of Oxford, Department of Physics, Clarendon Laboratory
Parks Road, Oxford OX 1 3PU, UK.
a.turberfield@physics.ox.ac.uk

Abstract. Recent experimental progress in DNA lattice construction, DNA robotics, and DNA computing provides the basis for designing DNA cellular computing devices, *i.e.* autonomous nano-mechanical DNA computing devices embedded in DNA lattices. Once assembled, DNA cellular computing devices can serve as reusable, compact computing devices that perform (universal) computation, and programmable robotics devices that demonstrate complex motion. As a prototype of such devices, we recently reported the design of an Autonomous DNA Turing Machine, which is capable of universal sequential computation, and universal translational motion, *i.e.* the motion of the head of a single tape universal mechanical Turing machine. In this paper, we describe the design of an Autonomous DNA Cellular Automaton (ADCA), which can perform *parallel* universal computation by mimicking a one-dimensional (1D) universal cellular automaton. In the computation process, this device, embedded in a 1D DNA lattice, also demonstrates well coordinated parallel motion. The key technical innovation here is a molecular mechanism that synchronizes pipelined “molecular reaction waves” along a 1D track, and in doing so, realizes parallel computation. We first describe the design of ADCA on an abstract level, and then present detailed DNA sequence level implementation using commercially available protein enzymes. We also discuss how to extend the 1D design to 2D.

1 Introduction

DNA has recently been used, with great success, to fabricate nanoscale lattices and tubes [3, 8, 11, 14–16, 18, 20–22, 29, 31, 32], to construct nanomechanical devices [2, 4, 9, 10, 24, 25, 27, 23, 28, 33, 34, 36, 37], and to perform computation [1, 5–7, 13, 17, 19, 26]. The progress in these three fields together provides the basis for the next step forward: designing and constructing autonomous DNA computing devices embedded in well defined DNA lattices, which are capable of (universal) computation. We call such devices *DNA cellular computing devices*.

Once assembled, DNA cellular computing devices can serve as reusable, compact computing devices that perform (universal) computation, and programmable robotics

^{*} The work is supported by NSF ITR Grants EIA-0086015 and CCR-0326157, NSF QuBIC Grants EIA-0218376 and EIA-0218359, and DARPA/AFSOF Contract F30602-01-2-0561.

devices that demonstrate complex motion. First, DNA cellular computing devices represent a step forward beyond the prior molecular computing schemes, such as algorithmic self-assembly of DNA tiles, which performs a one-time computation during the assembly. In contrast, DNA cellular computing devices, once assembled, can perform multi-round computations. The output, embedded in the DNA lattices, is preserved and can serve as input for future computations. In addition, DNA cellular computing devices are more compact than the tiling scheme. A 1D Autonomous DNA Turing machine holds equivalent computational power as a 2D tiling lattice. Second, DNA cellular computing devices can demonstrate sophisticated programmable motion, for example, universal translational motion, which we define as the motion of the head of a single tape universal mechanical Turing machine. As such, DNA cellular computing devices may promise interesting applications in nanorobotics and nano-computation, as well as, nano-fabrication, nano-sensing, and nano-actuated electronics.

As one prototype of the DNA cellular computing devices, we previously reported the design of an autonomous unidirectional DNA mechanical device capable of universal sequential computation, termed as Autonomous DNA Turing Machine [35]. Here, we extend our previous work and obtain the design of an Autonomous DNA Cellular Automaton (ADCA). By mimicking a 1D universal cellular automaton, the ADCA can perform *parallel* universal computation, and in the process, demonstrate well coordinated parallel motion. The parallel computation and motion is realized using a novel molecular mechanism that synchronizes pipelined “molecular reaction waves” along a 1D track.

A cellular automaton is a set of “colored” cells on a grid of specified shape that evolve through discrete time steps according to a set of transition rules based on the colors of neighboring cells [30]. If the lattice is a one (resp. two) dimensional lattice, the cellular automaton is called a one (resp. two) dimensional cellular automaton. Figure 1 (a) shows the cells of an example one-dimensional cellular automaton. Each cell of this automaton can have one of two states, or equivalently two colors, WHITE and BLACK. The evolving process of a cellular automaton is specified by the transition rules. Figure 1 (b) illustrates one example rule set for the cellular automaton shown in Figure 1 (a). The rule set consists of 8 transition rules (numbered (1) - (8) in the figure). For example, according to rule (1), if the current cell and both of its neighbors have color BLACK, at next time step, the middle cell will change to color WHITE. This rule is denoted as BLACK, BLACK, BLACK \rightarrow WHITE. Applying the rules in Figure 1 (b) to the initial configuration in Figure 1 (a), we obtain the transition history table depicted in Figure 1 (c). Cellular automaton can hold universal computing power. The cellular automaton depicted in Figure 1 is one such universal cellular automaton, known as rule 110, as described in [30]. This cellular automaton can be simulated by the ADCA described in this paper.

The rest of the paper is organized as follows. In Sect. 2, we describe the structural design and operational process of our ADCA. In Sect. 3, we give a detailed molecular implementation of the design presented in Sect. 2. We then briefly describe how to extend the design of ADCA to two-dimensions in Sect. 4 and close in Sect. 5.

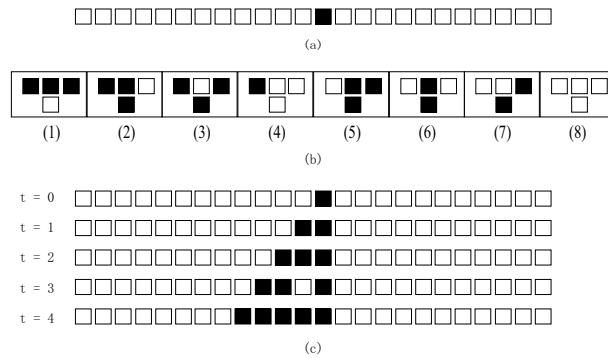


Fig. 1. A universal cellular automaton with two colors: Rule 110. The figure is adapted from [30]

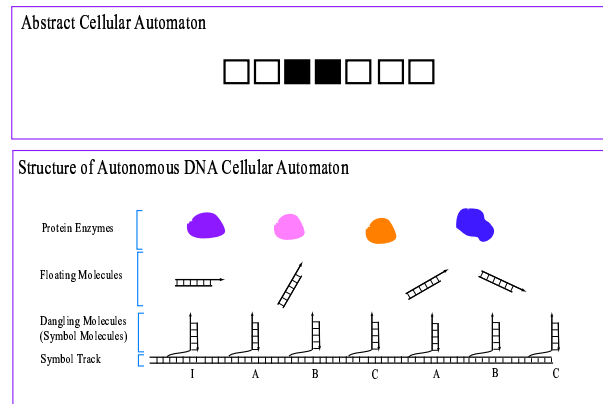


Fig. 2. Top panel: the initial configuration of an abstract cellular automaton. Bottom Panel: schematic drawing of the structure of an ADCA corresponding to the abstract cellular automaton in the top panel. The backbones of DNA strands are depicted as directed line segments. The short bars represent base pairing between DNA strands

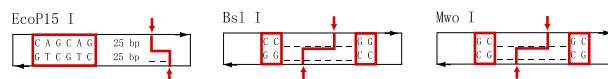


Fig. 3. Three endonucleases used in the molecular implementation of the ADCA. The recognition site of an enzyme is bounded by a box and the cleavage site indicated with a pair of arrows. The symbol “—” indicates the position of a base that does not affect endonuclease recognition

2 Design

2.1 Structure

The ADCA operates in a solution system. Figure 2 illustrates an example abstract cellular automaton in the top panel, and the structure of the corresponding ADCA in the

bottom panel. The ADCA is composed of four parts: a rigid *symbol track*, a linear array of *dangling DNA molecules* tethered to the symbol track, a set of *floating DNA molecules*, and a group of floating *protein enzymes*.

- **Symbol track.** The symbol track provides a rigid structural platform on which the dangling-molecules are tethered. It can be implemented, for example, as a rigid addressable DNA lattice, such as the barcode DNA lattice reported in [31].
- **Dangling DNA molecules.** The array of dangling-molecules, also called symbol-molecules, tethered to the symbol track represent the array of cells (symbols) in the cellular automaton (and hence the name symbol-molecule). A dangling-molecule is a duplex DNA fragment, with one end tethered to the symbol track via a flexible single strand DNA fragment and the other end possessing a single strand DNA extension (the sticky end). Due to the flexibility of the single strand DNA linkage, a dangling-molecule moves rather freely around its joint on the symbol track. We require that the only possible interactions between two dangling-molecules are those between two immediate neighbors. This requirement can be ensured by properly spacing the dangling-molecules along the rigid track.
- **Floating DNA molecules.** In addition to the array of dangling-molecules, the system contains *floating-molecules*. A floating-molecule is a free floating (unattached to the symbol track) duplex DNA segment with a single strand overhang at one end (sticky end). A floating-molecule floats freely in the solution and thus can interact with another floating-molecule or a dangling-molecule provided that they possess complementary sticky ends. There are two kinds of floating-molecules: the rule-molecules and the assisting-molecules. The rule-molecules collectively specify the computational rules and are the programmable part of the ADCA, while the assisting-molecules assist in carrying out the operations of the ADCA, which we describe in detail in Sect. 2.2 and Sect. 2.3.
- **Protein enzymes.** The system also contains floating DNA ligase and three types of DNA endonucleases. The enzymes perform ligations and cleavages on the DNA molecules to effect the designed structural changes and hence the information processing. The cleavage patterns of the endonucleases are described in Figure 3.

2.2 Structural Changes

Figure 4 illustrates the structural changes during the operation of ADCA.

Initial configuration. Figure 4 (a) depicts an example abstract cellular automaton in its top panel, and a corresponding ADCA in its bottom panel. For simplicity and clarity, the floating enzymes and the floating DNA molecules in the ADCA are omitted from the figure; the symbol track, as well as the duplex and sticky end portions of a dangling-molecule, is depicted as a thick line segment; the flexible hinge of a dangling-molecule as a thin curve. The leftmost symbol-molecule is a special *initiator* dangling-molecule, I , representing the cell 0 in the abstract cellular automaton (see Figure 4). To the right of I , three types of dangling-molecules, A , B , and C , are positioned evenly along the track in a periodic order such that cells $3i + 1$, $3i + 2$, and $3i + 3$, where i is a non-negative integer, in the abstract cellular automaton are represented in the ADCA by symbol-molecules A , B , and C , respectively. The symbol-molecules differ

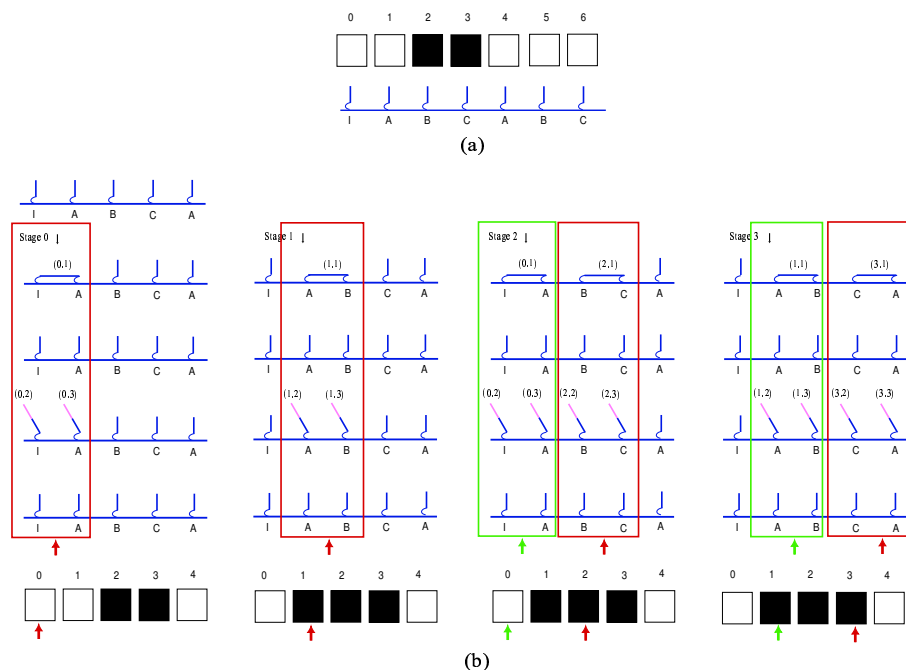


Fig. 4. Structural changes during the operation of an ADCA. In (b), red(dark) and green(grey) boxes indicate two pipelined reaction waves

in their *default sticky ends*, *i.e.* the sticky ends they possess in their respective initial configurations before the reaction starts. As we shall see later, this is essential for the synchronization of the operation of the ADCA. The color of each cell in the abstract cellular automaton is encoded in a corresponding symbol-molecule in the ADCA.

Pipelined reaction waves. Figure 4 (b) illustrates structural changes. During the operation of the ADCA, the initiator molecule I sends out a “reaction wave” that travels down the track from left to right. A critical novel property of the ADCA is that multiple reaction waves can travel down the track in a “pipelined” fashion. However, we have carefully engineered a “synchronization” mechanism so that a reaction wave that starts at a later stage can never overtake one that starts earlier. This ensures the synchronization of the state changes of the ADCA, and hence its correct operation.

Figure 4 (b) illustrates two consecutive reaction waves, respectively indicated with red (dark) and green (grey) boxes and arrows. Next, we focus on the first reaction wave (indicated by dark boxes and arrows), and describe the structural changes of ADCA.

In Stage 0, the reaction wave starts at the initiator I at position 0, then travels sequentially to A in Stage 1, B in Stage 2, and C in Stage 3. The reaction wave finishes one full cycle in Stages 1, 2, and 3, and thus goes on inductively down the track.

In Stage i , where $i = 0, 1, 2$, and 3, three types of reactions occur, namely reactions $i.0$, $i.1$. and $i.2$.

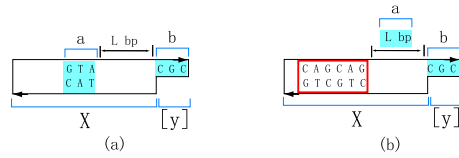


Fig. 5. Encoding state information in DNA molecules. The backbones of DNA strands are depicted as directed line segments. X and $[y]$ represent the duplex portion and the sticky end of the DNA molecule, respectively. The bases shaded in blue are used to encode state information, a or b . “ L bp” indicates L DNA base pairs, where L is a non negative integer. In Figure (b), the number L is used to encode state information a and is thus shaded in blue. The red (dark) box indicates the recognition site for an endonuclease, in this case, EcoPI5 I

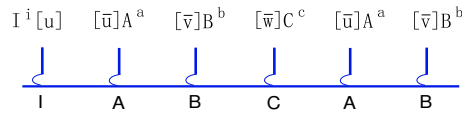


Fig. 6. Initial configuration

- In Stage 0, I has a complementary sticky end to its right neighbor A and is thus ligated to A , and the ligation product is subsequently cleaved by an endonuclease (Reaction 0.1). Next, I is “modified” by an assisting-molecule, depicted as a pink (grey) line segment, and restored to its default configuration (Reaction 0.2). The “modification” will be implemented as ligation and cleavage events and will be described in detail in Sect. 3. In a parallel reaction 0.3, A is also modified by another assisting-molecule such that A will possess a complementary sticky end to B , and thus the reaction wave is ready to enter Stage 1 (Reaction 0.3).
- In Stage 1, similar structural changes occur as in Stage 0. However, after reaction 1.1, A will possess a sticky end that encodes the state, *i.e.* color, information of itself, its left neighbor I , and its right neighbor B . In the ensuing reaction 1.2, a rule-molecule corresponding to a transition rule in Figure 1 recognizes A ’s sticky end and effects a state transition of molecule A . A will then be modified by an assisting-molecule and restored to its default configuration, encoding its new state. In the example shown in Figure 4 (b), a rule-molecule corresponding to rule (7) in Figure 1 changes the color encoded in A from WHITE to BLACK. In the parallel reaction 1.3, B will be modified to possess a complementary sticky end to C .
- In Stages 2 and 3, reactions of the same nature as in Stage 1 will occur. Details are omitted for brevity.

2.3 Information Flow

Notation. We next describe the information flow during the operation of the ADCA. For ease of exposition, we first introduce some notation. An information encoding DNA molecule is denoted as $X^a[y]^b$, where X is its duplex portion, $[y]$ is its sticky end portion, and a and b respectively represent the state information encoded in X and $[y]$. This

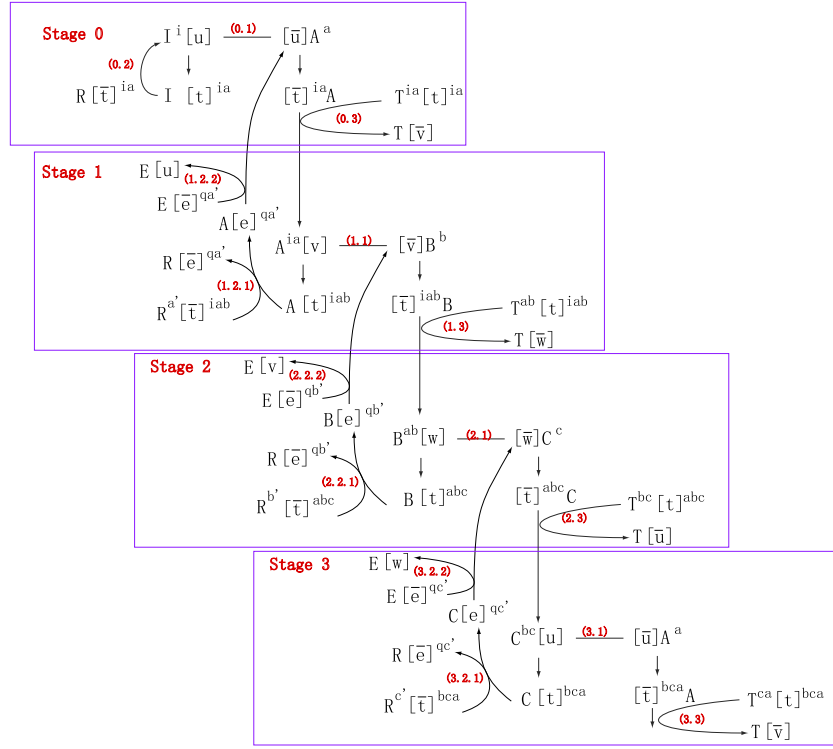
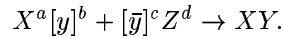
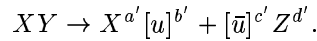


Fig. 7. Information flow

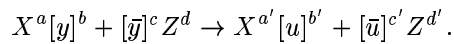
is illustrated in Figure 5. As shown in the figure, there are two ways to encode information a in the duplex X . In Figure 5 (a), a is encoded as a unique DNA sequence GTA ; in Figure 5 (b), a is encoded as the number of base pairs (L bp in the Figure) between an endonuclease recognition site and the sticky end of DNA molecule. The sequence of the sticky end $[y]$, in this case CGC , encodes the state information b . Furthermore, we use $[\bar{y}]$ to denote the complementary sticky end of $[y]$. Finally, we describe how to represent ligation and cleavage events. The ligation of two molecules $X^a[y]^b$ and $[\bar{y}]^c Z^d$ is described by the equation



Suppose XY incorporates an endonuclease recognition site and is cut into $X^{a'}[u]^{b'}$ and $[\bar{u}]^{c'} Z^{d'}$. This is represented as



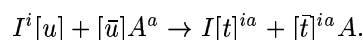
We can combine the above two equations and obtain,



Initial configuration. Figure 6 shows the ADCA in its default configuration before the reaction starts. As mentioned in Sect. 2.2, molecules A , B , and C possess different default sticky ends: $[\bar{u}]$, $[\bar{v}]$, and $[\bar{w}]$. Note that the state information a , b , and c are encoded in the duplex portions of A , B , and C , *not* their sticky ends. This is essential to ensure that repeated reactions between neighboring symbol-molecules can occur for multiple rounds, as described below.

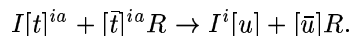
Information flow. Figure 7 illustrates the information flow. We follow the framework of four-stage structural changes presented in Sect. 2.2 and enumerate the involved reactions below.

1. *Reaction 0.1.* Initiator molecule $I^i[u]$ and its immediate right neighbor $[\bar{u}]A^a$ share complementary sticky ends u and $[\bar{u}]$, and result in reaction,

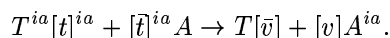


Note that the sticky end $[\bar{t}]$ of product A encodes both the state information i from reactant I and the state information a from reactant A .

2. *Reaction 0.2.* The rule-molecule $[\bar{t}]^{ia}R$ restores $I[t]^{ia}$ to its default configuration in reaction,

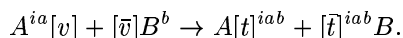


3. *Reaction 0.3.* Molecule $[\bar{t}]^{ia}A$ is modified by assisting-molecule $T^{ia}[t]^{ia}$ in reaction



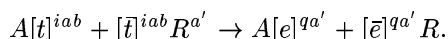
Now A is transformed to $[v]A^{ia}$. This essentially transduces the state information ia initially encoded in the sticky end of A to its duplex portion. Hence we term the assisting-molecule $T^{ia}[t]^{ia}$ as a *transducer-molecule*. The above reaction also modifies A 's sticky end to $[v]$, which is complementary to the default sticky end of A 's immediate right neighbor B . This makes A ready to interact with B .

4. *Reaction 1.1.* Molecule $A^{ia}[v]$ interacts with its right neighbor $[\bar{v}]B^b$ in reaction,



Now the sticky end of the product A encodes state information iab , *i.e.* the current state of A 's left neighbor, the current state of A , and the current state of A 's right neighbor. This suffices to specify a transition rule shown in Figure 1 and results in Reaction 1.2 below.

5. *Reaction 1.2.* Reaction 1.2 has two steps. In step 1.2.1, $A[t]^{iab}$ interacts with a rule-molecule $[\bar{t}]^{iab}R^{a'}$ in reaction,



This effects a state transition of molecule A , as specified by the rule $iab \rightarrow a'$. However, to enable A to repeatedly perform computation, we need to further restore A to its default configuration, *i.e.* a configuration with a default sticky end $[\bar{u}]$ and encoding its new state a' in its duplex portion. This task is carried out by another kind of assisting-molecule called *extension-molecule* E . However, as

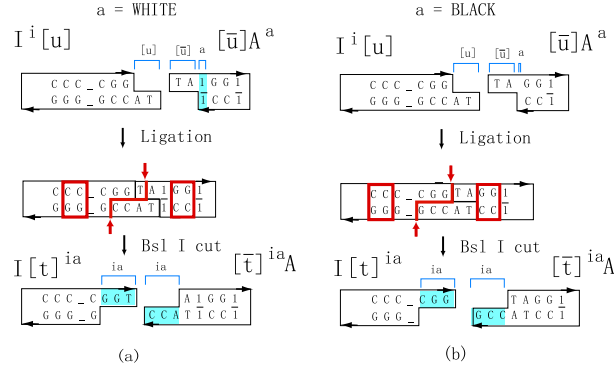
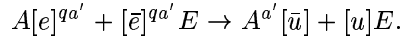
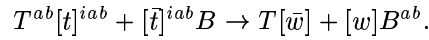


Fig. 8. Example molecular implementation of reaction 0.1. The red (dark) box and red (dark) arrows respectively indicate the recognition and cleavage sites for endonuclease Bsl I. The encoded state information is indicated with blue (grey) region. Base pair $1/\bar{1}$ is a pair of unnatural bases

a floating molecule, E needs to not only recognize A 's current state but also distinguish A from the other two types of symbol-molecules, B and C . As such, we require the sticky end of A encodes not only its state information but also its type information q , where $q \in \{q_A, q_B, q_C\}$. Hence, in the above equation, the product A possesses a sticky end $[e]$ encoding both its type information q and its new state a' (a' not shown in Figure 7 (b)). This molecule $A[e]^{qa'}$ is then modified by extension-molecule $E[\bar{e}]^{qa'}$ in reaction 1.2.2,



6. *Reaction 1.3.* Molecule $[\bar{t}]^{iab}B$ is modified by transducer-molecule $T^{ab}[\bar{t}]^{iab}$ in reaction,



Note that now B encodes state ab in its duplex portion (state i is not kept since it is not required for effecting B 's transition), and possesses sticky end $[w]$, which is complementary to the default sticky end of C .

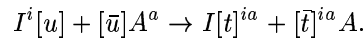
7. *Other reactions.* Similar to reactions 1.1, 1.2, and 1.3.

3 Molecular Implementation

3.1 Step-by-step Implementation

To demonstrate the practicality of our design, we next give a detailed description of the molecular implementation of the ADCA. The complete DNA molecule set will be described in Sect. 3.2.

1. *Reaction 0.1.* Figure 8 depicts an example molecular implementation of reaction 0.1,



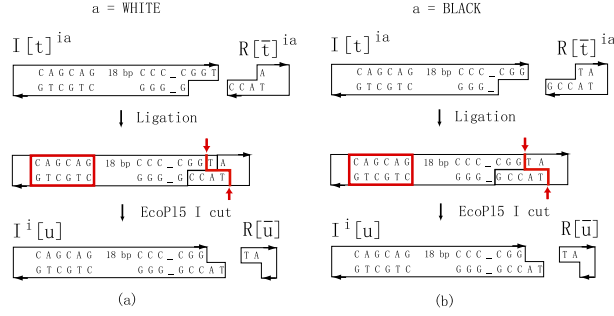
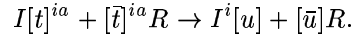


Fig. 9. Example molecular implementation of reaction 0.2

For simplicity, only the end sequences of dangling-molecule A are depicted; for full sequences, see Figure 15 (a) in Appendix. Panels (a) and (b) respectively illustrate cases when $a = \text{WHITE}$ and $a = \text{BLACK}$. In molecule $[\bar{u}]A^a$, the state information $a \in \{\text{WHITE}, \text{BLACK}\}$, is encoded by the presence or absence of a DNA base pair between the sticky end $[\bar{u}]$ (sequence TA) and the half recognition site for endonuclease Bsl I (sequence GG/CC) in the duplex portion. This is further indicated in Figure 8 by the shaded blue (grey) region. In the case $a = \text{WHITE}$, the cleavage of ligation product IA by Bsl I produces a sticky end sequence GGT for molecule I , and CCA for molecule A . Both these unique sticky end sequences encode state information ia . When $a = \text{BLACK}$, a different pair of sticky ends $[t]/[\bar{t}]$ are generated (CGG/GCC).

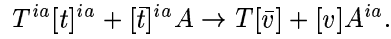
Molecule A contains a pair of unnatural bases, *i.e.* synthetic bases other than the natural bases A, C, G, and T. They are required because the four-letter ACGT natural vocabulary does not provide sufficient encoding space for our construction. For a survey on experimental synthesis of unnatural bases, see [12].

2. *Reaction 0.2.* Figure 9 depicts an example molecular implementation of reaction 0.2,



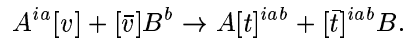
The endonuclease involved is EcoP15 I. This restores I to its default configuration with sticky end sequence $[u] = \text{AT}$.

3. *Reaction 0.3.* Figure 10 depicts an example molecular implementation of reaction 0.3,



Here, we only illustrate the case $a = \text{WHITE}$, and omit the similar case $a = \text{BLACK}$ for brevity. Note that the state information ia initially encoded in the sticky end $[\bar{t}]$ (sequence CCA) of A is now encoded in the blue (grey) region in its duplex portion (sequence CGA/GCT).

4. *Reaction 1.1.* Figure 11 depicts an example molecular implementation of reaction 1.1,



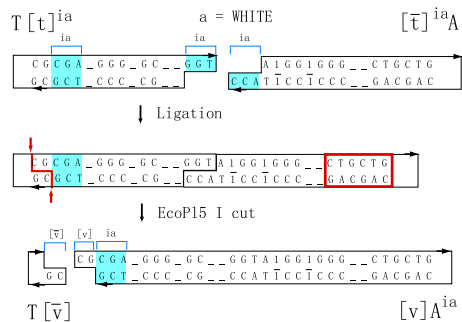


Fig. 10. Example molecular implementation of reaction 0.3

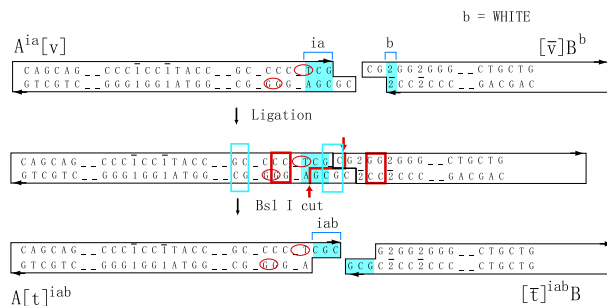
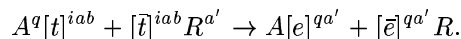


Fig. 11. Example molecular implementation of reaction 1.1. The red (dark) box and red (dark) arrows respectively indicate the recognition and cleavage sites for endonuclease Bsl I. The bases labeled with red (dark) circles contain phosphorothioate bond and are hence resistant to enzyme cleavage. The light blue (grey) box indicates the Mwo I recognition site

Again, we only illustrate the case $a = \text{WHITE}$ and $b = \text{WHITE}$ for brevity. This reaction is similar to reaction 0.1. Both the sticky ends $[t]$ and $[\bar{t}]$ encode state information iab .

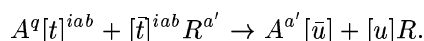
Two technical points warrant explanation in this reaction. First, the bases labeled with red (dark) circles contain phosphorothioate bond and are hence resistant to enzyme cleavage. This modification of the bases is required to prevent the unwanted cleavage of the DNA duplex by Mwo I, whose recognition sites are indicated with blue (grey) boxes in the figure. This trick will be used again in reaction 1.2.2. Second, we assume here that the cleavage by endonuclease Bsl I will occur, but the cleavage by EcoP15 I will not occur (note that both molecule A and molecule B contain EcoP15 I recognition site, *i.e.* CAGCAG/CTGCTG). This assumption is based on the fact that Bsl I, a Type II endonuclease, can act far more efficiently than EcoP15 I, a Type III endonuclease.

5. *Reaction 1.2.1.* Figure 12 (a) depicts an example molecular implementation of reaction 1.2.1,



Recall that this reaction effects a state transition for molecule A , as specified by the rule $iab \rightarrow a'$. Here, the rule molecule incorporates a spacer region, the length of which (L bp) encodes the new state information a' . In particular, when $L = 2$, $a' = \text{BLACK}$; when $L = 6$, $a' = \text{WHITE}$.

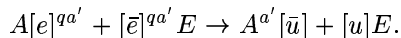
When $a' = \text{WHITE}$ (the EcoP15 I cleavage step not shown in Figure 12 (a)), molecule A is restored to the desired target configuration $[\bar{u}]A^{a'}$, with the default sticky end $[\bar{u}] = \text{AT}$. In this case, the next step, reaction 1.2.2, is not required. The reaction can thus be rewritten as,



However, when $a' = \text{BLACK}$ (the EcoP15 I cleavage step shown in the figure), molecule A is modified to $A[e]^{qa'}$, with a unique sticky end $[e] = 1\text{G}$ that encodes both A 's type information $q = q_A$ and A 's new state a' (Recall that $q \in \{q_A, q_B, q_C\}$ encodes type information, in the case illustrated here, $q = q_A$. This information is initially encoded in the blue (grey) duplex portion of A , in the form of sequence $1/\bar{1}$). Then the reaction proceeds to the next step, reaction 1.2.2, which will finish the state transition for A .

In our molecular implementation, in a transition $xyz \rightarrow y'$, the values of y , z , and z' cooperatively determine the spacer length L , which in turn decides the result of the transition. For detail, see Figure 15 (c) in Appendix.

6. *Reaction 1.2.2.* Figures 12 (b) depicts the case $a = \text{WHITE}$, $a' = \text{BLACK}$, which follows from the case illustrated in Figures 12 (a).



Here, extension-molecule $E[\bar{e}]^{qa'}$ restores A to its default configuration with sticky end $[\bar{u}] = \text{AT}$. However, now A encodes new state a' in its duplex portion.

7. *Other reactions.* Similar to the above reactions, and hence omitted for brevity.

3.2 Complete Molecule Set

The complete set of DNA molecules constituting ADCA are described in Figure 15 and Figure 16 in the Appendix. The dangling-molecules (Figure 15 (a)) and the floating rule-molecules (Figure 15 (c)) are the programmable parts of the ADCA: the selections of dangling-molecules and rule-molecules respectively determine the initial configuration and the transition rules of the ADCA. Note that all the sticky ends of rule-molecules are unique. In contrast, the floating assisting-molecules, *i.e.* transducer-molecules (Figure 16) and extension-molecules (Figure 15 (b)), only assist in the proper operation of the ADCA and are non-programmable.

3.3 Futile Reactions

Besides the main reactions described above, there exist *reversible futile reactions* in the system. These futile reactions are carefully engineered such that they will not block the main reactions. Futile reactions, however, can also be used to maintain a dynamic

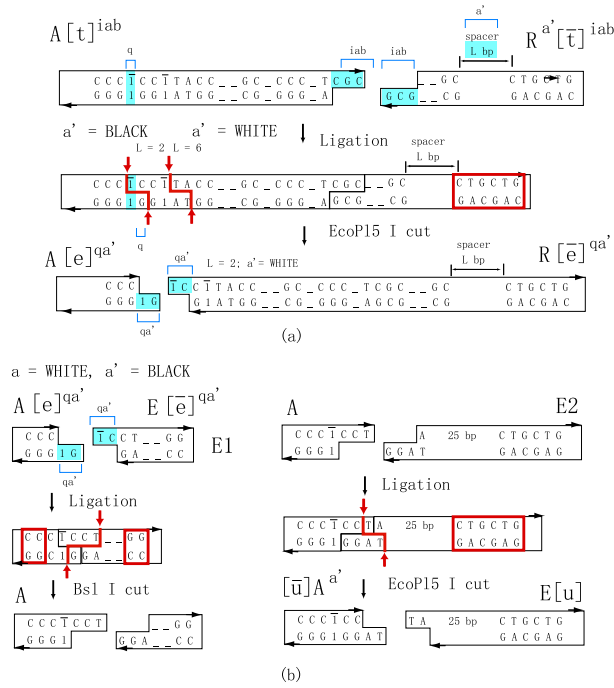


Fig. 12. Example molecular implementation of reaction 1.2. Panel (a): reaction 1.2.1. Panels (b): reaction 1.2.2

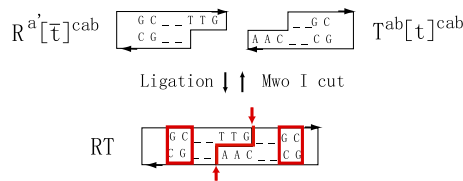


Fig. 13. An example futile reaction

balance among the floating molecules, and, in doing so, ensure the proper operation of the devices. Figure 13 shows an example where futile reactions are used to maintain a balance between R and T molecules that have complementary sticky ends. For a more detailed discussion of futile reactions, see [35].

3.4 Computer Simulation

Fully debugging the above molecular implementation requires meticulous inspection of every step of the ADCA operation, which can become exceedingly tedious. We thus developed a computer simulator and used it to test and debug the ADCA. The simulator takes as input the DNA sequences which specifies an ADCA instance, simulates the

operations of ADCA, and gives graphical output. For detail, see <http://pengyin.org/paper/dnaCA/>.

4 Two-Dimensional ADCA

To illustrate the operational principle of 2D ADCA, we first present an abstract view of the 1D ADCA in Figure 14 (a) and (b). Figure 14 (a) illustrates a reaction wave of the 1D ADCA: the reaction wave starts at initiator I and travels sequentially down the one-dimensional track. Figure 14 (b) examines one individual dangling-molecule X , where $X = A, B, C$. Assume w.l.o.g., $X = B$. As shown in Figure 14 (b), B in a 1D ADCA undergoes the following four phases in one full reaction cycle.

1. *Phase 1.* B has a sticky end that is complementary to its left neighbor A (indicated by a solid square to the left of B in Figure 14 (b)). This is before reaction 1.1 as depicted in Figure 7. In this phase, B encodes in its duplex portion its own state information denoted by \mathcal{C} (\mathcal{C} for center).
2. *Phase 2.* In reaction 1.1, B interacts with its left (*i.e.* west) neighbor, and enters phase 2. Now B encodes in its sticky end both the state information of itself, \mathcal{C} , and the state information of its west neighbor, \mathcal{W} . This sticky end is complementary to a floating transducer-molecule, indicated by a circle around B .
3. *Phase 3.* After reaction 1.3, B now encodes state information \mathcal{CW} in its duplex portion, and possesses a sticky end complementary to its right (*i.e.* east) neighbor (indicated by a solid square to the right of B).
4. *Phase 4.* In reaction 2.1, B interacts with its east neighbor, and enters phase 4. Now B encodes in its sticky end the state information of itself \mathcal{C} , its west neighbor \mathcal{W} , and its east neighbor \mathcal{E} . This sticky end thus encodes all the state information required to effect a state transition for B , and is recognized by a floating rule-molecule (indicated by a thick circle). And this state transition restores B to its default configuration, finishing a full cycle.

With the above understanding of the 1D ADCA, we can extend it to 2D as follows. First, we take care of reaction waves by positioning two arrays of initiators as shown in Figure 14 (c). Each initiator can send out a reaction wave that travels either horizontally or vertically. Next, we take care of the information flow and synchronization, by again examining one single molecule X . As shown in Figure 14 (d), we engineer the system such that molecule X undergoes 8 phases. During these 8 phases, X sequentially interacts with its west (\mathcal{W}), north (\mathcal{N}), east (\mathcal{E}), and south (\mathcal{S}) neighbors to garner the state information from each of them. As such, upon entering phase 8, X carries in its sticky end the state information \mathcal{CWNES} . This state information is sufficient to effect a state transition for X . As in the 1D case, X will undergo a state transition and re-enters phase 1, completing a full circle.

5 Conclusion

In this paper, we present the theoretical design and molecular implementation of 1D ADCA and describe how to extend it to 2D.

Open question: can we simplify the current complex design?

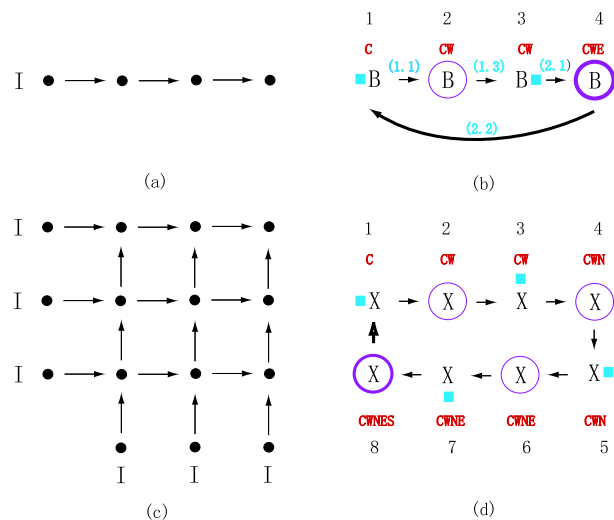


Fig. 14. (a) (b) Operational overview of 1D ADCA. (c) (d) Operational overview of two-dimensional ADCA. In panels (b) and (d), black numbers indicate the phases of X ; blue (grey) numbers indicate reactions corresponding to reactions depicted in Figure 7; red (dark) letters indicate the state information carried by X

References

1. L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.
2. P. Alberti and J.L. Mergny. DNA duplex-quadruplex exchange as the basis for a nanomolecular machine. *Proc. Natl. Acad. Sci. USA*, 100:1569–1573, 2003.
3. R. Barish, P.W.K. Rothmund, and E. Winfree. Algorithmic self-assembly of a binary counter using DNA tiles. 2005. In preparation.
4. J. Bath, S.J. Green, and A.J. Turberfield. A free-running DNA motor powered by a nicking enzyme. *Angew. Chem. Intl. Ed.*, 44:4358–4361, 2005.
5. Y. Benenson, R. Adar, T. Paz-Elizur, Z. Livneh, and E. Shapiro. DNA molecule provides a computing machine with both data and fuel. *Proc. Natl. Acad. Sci. USA*, 100:2191–2196, 2003.
6. Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, and E. Shapiro. An autonomous molecular computer for logical control of gene expression. *Nature*, 429:423–429, 2004.
7. Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414:430–434, 2001.
8. N. Chelyapov, Y. Brun, M. Gopalkrishnan, D. Reishus, B. Shaw, and L. Adleman. DNA triangles and self-assembled hexagonal tilings. *J. Am. Chem. Soc.*, 126:13924–13925, 2004.
9. Y. Chen, M. Wang, and C. Mao. An autonomous DNA nanomotor powered by a DNA enzyme. *Angew. Chem. Intl. Ed.*, 43:3554–3557, 2004.
10. L. Feng, S.H. Park, J.H. Reif, and H. Yan. A two-state DNA lattice switched by DNA nanoactuator. *Angew. Chem. Intl. Ed.*, 42:4342–4346, 2003.
11. Y. He, Y. Chen, H. Liu, A.E. Ribbe, and C. Mao. Self-assembly of hexagonal DNA two-dimensional (2D) arrays. *J. Am. Chem. Soc.*, 127:12202–12203, 2005.

12. A.A. Henry and F.E. Romesberg. Beyond A, C, G, and T: augmenting nature's alphabet. *Curr. Opin. Chem. Biol.*, 7:727–733, 2003.
13. J. Kuramochi and Y. Sakakibara. Intensive in vitro experiments of implementing and executing finite automata in test tube. In *Proc. 11th International Meeting on DNA Computing*, pages 59–67, 2005.
14. T.H. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J.H. Reif, and N.C. Seeman. The construction, analysis, ligation and self-assembly of DNA triple crossover complexes. *J. Am. Chem. Soc.*, 122:1848–1860, 2000.
15. D. Liu, M. Wang, Z. Deng, R. Walulu, and C. Mao. Tensegrity: Construction of rigid DNA triangles with flexible four-arm dna junctions. *J. Am. Chem. Soc.*, 126:2324–2325, 2004.
16. J. Malo, J.C. Mitchell, C. Venien-Bryan, J.R. Harris, H. Wille, D.J. Sherratt, and A.J. Turberfield. Engineering a 2D protein–DNA crystal. *Angew. Chem. Intl. Ed.*, 44:3057–3061, 2005.
17. C. Mao, T.H. LaBean, J.H. Reif, and N.C. Seeman. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, 407:493–496, 2000.
18. J.C. Mitchell, J.R. Harris, J. Malo J, J. Bath, and A.J. Turberfield. Self-assembly of chiral DNA nanotubes. *J. Am. Chem. Soc.*, 126:16342–16343, 2004.
19. P.W.K. Rothmund. A DNA and restriction enzyme implementation of Turing machines. In R. J. Lipton and E.B. Baum, editors, *DNA Based Computers: Proceedings of the DIMACS Workshop, April 4, 1995, Princeton University*, volume 27, pages 75 – 119, Providence, Rhode Island, 1996. American Mathematical Society.
20. P.W.K. Rothmund. Generation of arbitrary nanoscale shapes and patterns by scaffolded DNA origami. 2005.
21. P.W.K. Rothmund, A. Ekani-Nkodo, N. Papadakis, A. Kumar, D.K. Fygenson, and E. Winfree. Design and characterization of programmable DNA nanotubes. *J. Am. Chem. Soc.*, 126:16344–16353, 2004.
22. P.W.K. Rothmund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA sierpinski triangles. *PLoS Biology* 2 (12), 2:e424, 2004.
23. N.C. Seeman. From genes to machines: DNA nanomechanical devices. *Trends in Biochemical Sciences*, 30:119–125, 2005.
24. W.B. Sherman and N.C. Seeman. A precisely controlled DNA biped walking device. *Nano Lett.*, 4:1203–1207, 2004.
25. J.S. Shin and N.A. Pierce. A synthetic DNA walker for molecular transport. *J. Am. Chem. Soc.*, 126:10834–10835, 2004.
26. M.N. Stojanovic, S. Semova, D. Kolpashchikov, J. Macdonald, C. Morgan, and D. Stefanovic. Deoxyribozyme-based ligase logic gates and their initial circuits. *J. Am. Chem. Soc.*, 127:6914–6915, 2005.
27. Y. Tian, Y. He, Y. Chen, P. Yin, and C. Mao. Molecular devices - a DNAzyme that walks processively and autonomously along a one-dimensional track. *Angew. Chem. Intl. Ed.*, 44:4355–4358, 2005.
28. A.J. Turberfield, J.C. Mitchell, B. Yurke, Jr. A.P. Mills, M.I. Blakey, and F.C. Simmel. DNA fuel for free-running nanomachines. *Phys. Rev. Lett.*, 90:118102, 2003.
29. E. Winfree, F. Liu, L.A. Wenzler, and N.C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394(6693):539–544, 1998.
30. S. Wolfram. *A new kind of science*. Wolfram Media, Inc., Champaign, IL, 2002.
31. H. Yan, T.H. LaBean, L. Feng, and J.H. Reif. Directed nucleation assembly of DNA tile complexes for barcode patterned DNA lattices. *Proc. Natl. Acad. Sci. USA*, 100(14):8103–8108, 2003.
32. H. Yan, S.H. Park, G. Finkelstein, J.H. Reif, and T.H. LaBean. DNA-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301(5641):1882–1884, 2003.
33. H. Yan, X. Zhang, Z. Shen, and N.C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415:62–65, 2002.

34. P. Yin, A.J. Turberfield, and J.H. Reif. Designs of autonomous unidirectional walking DNA devices. In *Proc. 10th International Meeting on DNA Computing*, pages 119–130, 2004.
35. P. Yin, A.J. Turberfield, S. Sahu, and J.H. Reif. Design of an autonomous DNA nanomechanical device capable of universal computation and universal translational motion. In *Proc. 10th International Meeting on DNA Computing*, pages 344–356, 2004.
36. P. Yin, H. Yan, X.G. Daniell, A.J. Turberfield, and J.H. Reif. A unidirectional DNA walker moving autonomously along a linear track. *Angew. Chem. Int. Ed.*, 43:4906–4911, 2004.
37. B. Yurke, A.J. Turberfield, Jr. A.P. Mills, F.C. Simmel, and J.L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, 2000.

Appendix: Complete Molecule Set

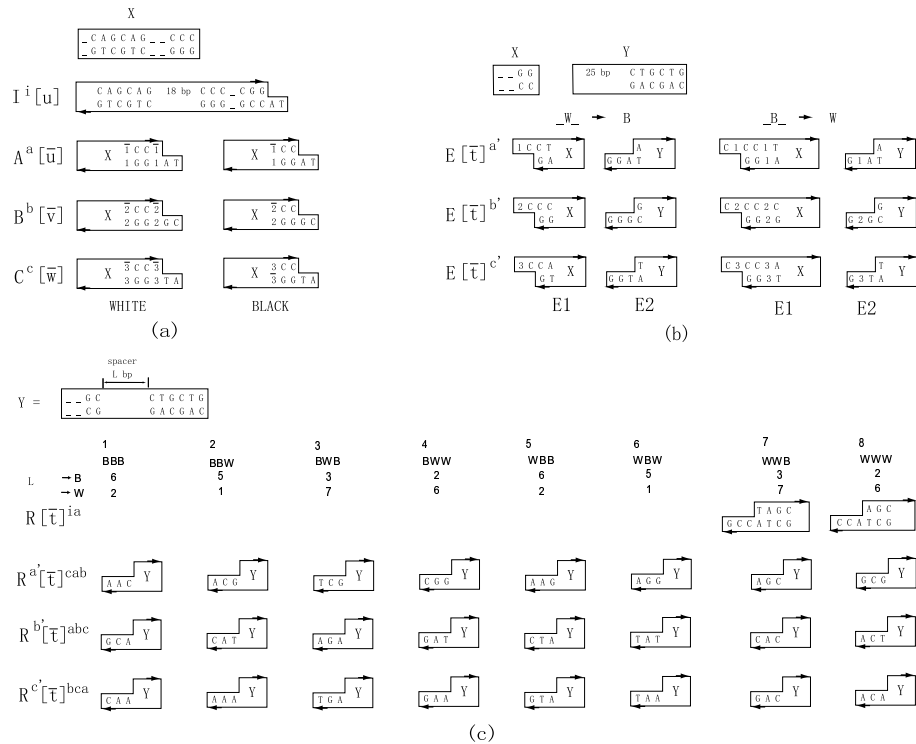


Fig. 15. (a) Dangling-molecules. Left and right panels respectively depict the cases when the encoded information $a/b/c = \text{WHITE}$ and BLACK . (b) Extension-molecules. Left and right panels respectively depict the cases when the transition is $x, \text{WHITE}, z \rightarrow \text{BLACK}$ and $x, \text{BLACK}, z \rightarrow \text{WHITE}$, where $x, z \in \{\text{BLACK}, \text{WHITE}\}$. (c) Rule-molecules. The eight columns (1-8) correspond to the eight possible configurations of xyz in the rule $xyz \rightarrow y'$, where $x, y, z, y' \in \{\text{BLACK}, \text{WHITE}\}$. The symbol BBW stands for the configuration $xyz = \{\text{BLACK}, \text{BLACK}, \text{WHITE}\}$. The value of L is determined cooperatively by y, z , and z'

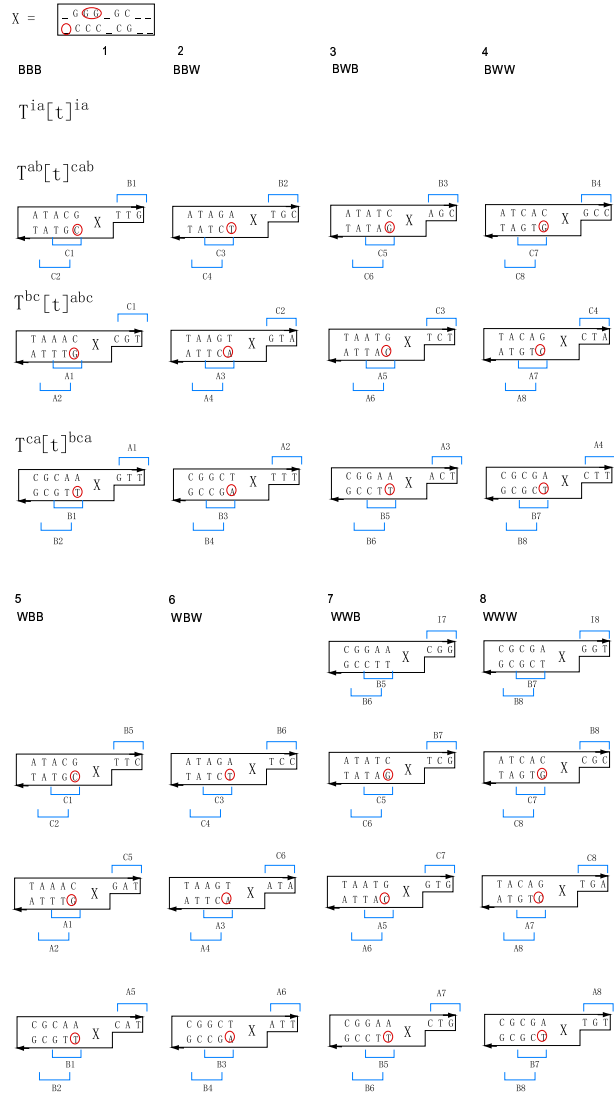


Fig. 16. Transducer-molecules. The bases labeled with red (dark) circles contain phosphorothioate bond and are hence resistant to enzyme cleavage (see reaction 1.1, Figure 11)