Autonomous Programmable DNA Nanorobotic Devices Using DNAzymes*

John H. Reif† Sudheer Sahu †

Abstract

A major challenge in nanoscience is the design of synthetic molecular devices that run *autonomously* (that is, without externally mediated changes per work-cycle) and are *programmable* (that is, their behavior can be modified without complete redesign of the device). DNA-based synthetic molecular devices have the advantage of being relatively simple to design and engineer, due to the predictable secondary structure of DNA nanostructures and the well-established biochemistry used to manipulate DNA nanostructures. However, ideally we would like to minimize the use of protein enzymes in the design of a DNA-based synthetic molecular device. We present the design of a class of DNA-based molecular devices using DNAzyme. These DNAzyme based devices are autonomous, programmable, and further require no protein enzymes. The basic principle involved is inspired by a simple but ingenious molecular device due to Mao et al [34] that used DNAzyme to traverse on a DNA nanostructure, but was not programmable in the sense defined above (it did not execute computations).

Our DNAzyme based designs include (1) a finite state automata device, *DNAzyme FSA* that executes finite state transitions using DNAzymes, (2) extensions to it including probabilistic automata and non-deterministic automata, and (3) its application as a *DNAzyme router* for programmable routing of nanostructures on a 2D DNA addressable lattice. Furthermore, we give a medical-related application, *DNAzyme doctor* that provide transduction of nucleic acid expression: it can be programmed to respond to the under-expression or over-expression of various strands of RNA, with a response by release of an RNA (The behavior of our nucleic acid transduction devices is similar to those of the prior paper of Shapiro[3], but ours have the advantage that they operate without use of any protein enzymes.) In addition, we describe some background theory and mathematical models essential to the operation of our devices, including stochastic models for simulation of strand displacement, and the operation of DNAzyme.

1 Introduction

1.1 Prior Autonomous Molecular Computing Devices

In the last few years the idea of constructing complex devices at the molecular scale using synthetic materials such as DNA has gone from theoretical conception to experimental reality.

(a) **DNA Tiling Assemblies.** One theoretical concept that had considerable impact on experimental demonstrations was that of Wang Tiling; this is an abstract model that allows for a finite set of 2D rectangles with labeled sides to assemble 2D lattices by appending together tiles at their matching sides. Winfree first proposed the use of DNA nanostructures to achieve Wang Tiling computations; the DNA nanostructures known as *DNA tiles* that self-assemble into 2D lattices as determined by the tiles pads (ssDNA on the sides of the tiles that can hybridize to other tile's pads). The last decade has seen major successes in experimental demonstrations of the use of such DNA tiling assemblies to construct patterned lattices and tiling computations. DNA tiling assemblies have been used effectively in construction of periodic two-dimensional lattices, such as those made from double-crossover (DX) DNA tiles [38], rhombus tiles [18], triple-crossover (TX) tiles [15], and "4x4" tiles [40], as well as triangle lattices [17] and hexagonal lattices [8]. They have also

*The work is supported by NSF EMT Grants CCF-0523555 and CCF-0432038.

†Department of Computer Science, Duke University, Durham, NC, USA. {reif,sudheer}@cs.duke.edu

been used for the construction of patterned lattices [39] by designing the DNA tile pads to program computations. The use of DNA tiling assembly has two major advantages over most other methods for molecular computation, since it: (i) operates entirely autonomously, without outside mediated changes, and (ii) does not require the use of protein enzymes.

DNA tiling assemblies do have limitations: in particular, in general as currently conceived, they do not allow for the molecular devices (the tiles in their case) to transition between multiple states (except of course for their free or assembled states). In contrast, many complex molecular mechanisms found in the cell can transition into multiple states, allowing far more flexibility of application.

(b) **Autonomous Molecular Computing Devices that Execute Multiple State Transitions:** There are only two other known methods for DNA computation that operate autonomously. Both use ingenious constructions, but require the use of enzymes.

(i) The *whiplash PCR machines* of [20, 21, 24, 37]. These however, can only execute a small number of steps before they require changes in the environment to execute further steps. Also, they require the use of polymerase enzyme.

(ii) The autonomous DNA machines of Shapiro[4, 2, 3], which execute finite transitions using restriction enzymes. The autonomous DNA machine [3] demonstrated molecular sensing and finite state response capabilities for that could be used for medical applications (though the demonstrations were made in test tubes only, rather than in natural biological environments as would be required for their medical applications). Their paper was important motivational factor in the work described here.

1.2 Our Main Contribution

This paper provides the first known design for a DNA-RNA based devices that (a) operates autonomously, (b) do not require the use of protein enzymes, and (c) allow for the execution of multiple state transitions. Our designs make use of certain prior DNA nanomechanical devices, which will be discussed below.

1.3 DNA Nanomechanical Devices

1.3.1 Prior Nonautonoumous Nanomechanical DNA Devices

A variety of DNA nanomechanical devices have been constructed that exhibit motions such as open/close [29, 30, 43], extension/contraction [1, 11, 16], and rotation [19, 35, 41]. The motion of these devices is mediated by external environmental changes such as the addition and removal of DNA fuel strands [1, 11, 16, 29, 30, 35, 41, 43] or the change of ionic strength of the solution [19]. For example, non-autonomous progressive walking devices, mediated by the addition and removal of DNA strands, were constructed both by Seeman [27] and Pierce [28]. Although in many cases ingeniously designed, these devices need external (human or automation-based) intervention for each step of their motions. These synthetic DNA devices are in sharp contrast with cellular protein motors and machines on macroscale that operate autonomously, without requiring any interference.

1.3.2 Prior Autonoumous DNA Nanomechanical Devices

Recent times have seen significant progress in construction of DNA nanomechanical devices that execute autonomous, progressive motions. Reif [23] gave two designs for autonomous DNA nanomechanical devices that traverse bidirectionally along a DNA nanostructure. Turberfield et al proposed using DNA hybridization energy to fuel autonomous free-running DNA machines [36]. Peng et al [42] was the first to experimentally demonstrate an autonomous DNA walker, which is an autonomous DNA device in which a DNA fragment translocates unidirectionally along a DNA nanostructure. It used DNA ligase and restriction enzymes.

Recently Mao demonstrated two autonomous DNA nanomechanical devices driven by DNA enzymes (non-protein), namely (a) a tweezer [10, 9] which is a DNA nanostructure that open and closes autonomously

and (b) a DNA crawler [34] using DNA enzyme, which traverses across a DNA nanostructure.

Their crawler device contains a DNA enzyme (DNAzyme) that constantly extracts chemical energy from its substrate molecules (RNA) and uses this energy to fuel the motion of the DNA device. This DNAzyme-based crawler integrates DNAzyme activity and strand-displacement reaction. They use 10-23 DNAzyme, which is a DNA molecule that can cleave RNA with sequence specificity. The 10-23 DNAzyme contains a catalytic core and two recognition arms that can bind to a RNA substrate. When the RNA substrate is cleaved, the short fragment dissociate from the DNAzyme and that provides a toehold for another RNA substrate to pair with short recognition arm of the DNAzyme. The crawler device traverses on a series of RNA stators implanted on a nanostructure as shown in Figure 1.

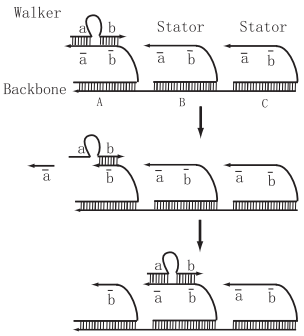


Figure 1: *Overview of Mao's crawler [34] constructed using DNA enzyme*

Their crawler is the primary inspiration to our designs. While an ingenious device, there are a number of limitations of Mao's DNAzyme-based crawler: (1) it did not demonstrate the loading and unloading of nanoparticles (2) it only traverses along a one dimensional sequence of ssRNA strands (stators) dangling from a DNA nanostructure, and its route is not programmable (3) it does not execute finite state transitions beyond what are required to move (that is, it does not execute computations).

1.4 Overview of this Paper and Results

The goal of this paper is to address the above limitations, providing substantially enhanced functionalities to the prior DNAzyme-based crawler previously developed. All the devices described in this paper are based on selective cleaving activity of DNAzyme and strand displacement processes. For the purpose of modeling these devices, it becomes imperative to model the strand displacement and cleaving activity of DNAzymes first. In the Section 2, we describe the kinetic models for these processes. We present the design of *DNAzyme FSA*: a finite state machine based on the activity of DNAzyme and strand displacements in Section 3. DNAzyme FSA can be easily extended to non-deterministic finite state automata and probabilistic automata as described in Section 3.7 and 3.8. In Section 4 we present a medical related application of DNAzyme FSA referred to as *DNAzyme doctor*. DNAzyme doctor is a molecular computer for logical control of RNA expression using DNAzyme. Another application of DNAzyme FSA, *DNAzyme router*: a DNAzyme based system for programmable routing of the walker on a 2D lattice is described in Section 5.

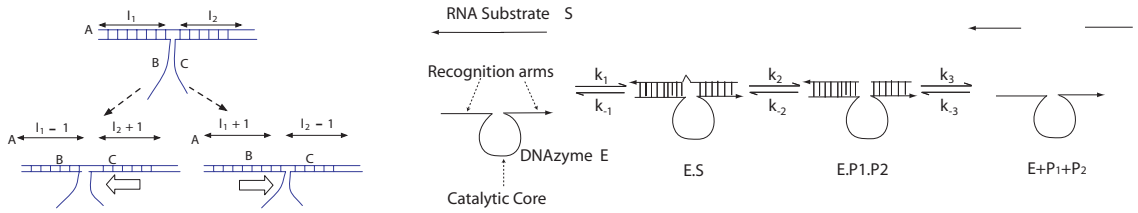
2 Strand Displacement and DNAzyme

The devices described in this paper are based on selective cleaving activity of DNAzyme and strand displacement processes. For the purpose of modeling these devices, it becomes imperative to model the strand displacement and cleaving activity of DNAzymes first. In the next subsections we describe the kinetic models for these processes.

2.1 Strand Displacement

In a strand displacement process two strands compete against each other to hybridize with a third strand. Figure 2.1 (a) shows a strand displacement process where strand B and C are competing against each other to hybridize with strand A. This ultimately results in removal of one of the competing strands, and hence the term strand-displacement.

Strand displacement can be modeled as a random walk of the junction where the two strands are competing against each other. For every step, the direction of migration of this junction is chosen probabilistically independent of its previous movements. It has been shown that the strand displacement is a biased random



(a) (b)

Figure 2: *(a) Strand displacement: molecule B and C compete against each other to hybridize with molecule A. (b) Mechanism of the cleaving of RNA substrate by DNAzyme*

walk in case of base-pair mismatches in these strands [5]. In other words, migration probability towards the direction with base-pair mismatches is substantially decreased.

Let us denote the nanostructure shown on top in Figure 2.1 (a) as molecule ABC. Let G°_{ABC} be its free energy. Denote G°_{rABC} and G°_{lABC} as the free energy of ABC after 1 base pair migration towards right, and left, respectively. Let $\Delta G^{\circ}_r = G^{\circ}_{rABC} - G^{\circ}_{ABC}$ and $\Delta G^{\circ}_l = G^{\circ}_{lABC} - G^{\circ}_{ABC}$. Let p_r be the probability of the right-directional migration and p_l be the probability of the left-directional migration. It has been shown in [5] that $p_r \propto \exp(-\Delta G^{\circ}_r/RT)$, similarly $p_l \propto \exp(-\Delta G^{\circ}_l/RT)$, where the change of free energies can be computed by the NN model [14]. Thompson et al[33] calculated the average time taken per base-pair migration (time per step) to be of the order of 100 μ sec. Strand displacement processes can be modeled as discrete time Markov chain processes using the above mentioned parameters.

2.2 DNAzymes

DNAzyme (also known as deoxyribozymes, DNA enzymes, and catalytic DNA) is a DNA molecule with a catalytic action. One of the widely used DNAzyme is 10-23 DNAzyme. It can cleave RNA with sequence specificity. The 10-23 DNAzyme contains a catalytic core and two recognition arms that can bind to a RNA substrate as shown in Figure 2.1 (b). The recognition domains provide both the sequence information necessary to specify RNA substrate and the binding energy needed to hold the substrate within the active site of enzyme. When the RNA substrate is cleaved, the short fragments dissociate from the DNAzyme. Another well studied DNAzyme is 8-17 deoxyribozyme. It also contains a catalytic core and two recognition arms. It is comparatively less flexible as compared to 10-23 DNAzyme in terms of target choice: 10-23 can cut an RNA phosphodiester bond located at any purine-pyrimidine site, while 8-17 requires an AG or GG site[12, 13].

Kinetics and thermodynamic characterization for cleaving activity of 8-17 DNAzyme and 10-23 DNAzyme are described in details in [6] and [32], respectively. RNA-cleaving activity of DNAzyme can be usually described into three reversible steps as shown in the Figure 2.1 (b). First step is the hybridization of the enzyme with the substrate. The second step is the cleaving of the substrate by the enzyme, which always requires metal ions as cofactor. This is usually the rate determining step in the reaction. Third step is the release of the cleaved product. k_1, k_2, and k_3 are the respective forward rate constants, and k_{-1}, k_{-2}, and k_{-3} are the respective reverse rate constants for the above mentioned three reversible steps. Substrate cleavage rate, $k_2 >> k_{-2}$ (ligation rate) suggests that enzyme has a strong preference for substrate cleavage over ligation. Enzyme substance association rate, $k_1 >> k_{-1}$ (dissociation rate) which is responsible for high enzyme-substrate association and hence high catalytic efficiency. Rate of product release step, $k_3 >> k_2$, which shows that substrate cleavage (rate constant k_2) is the rate determining step. It has been shown that DNAzymes show a high degree of sequence specificity. Catalytic rate increases log-linearly with increasing pH and linearly with various metal divalent cations. Under higher pH and modified divalent cation conditions, the 10-23 DNAzymes can cleave with k_{cat} of ~ 10 min^{-1} and a catalytic efficiency, k_{cat}/K_M, of 10^9 M^{-1}min^{-}1[32].

3 DNAzyme FSA: DNAzyme Based Finite State Automata

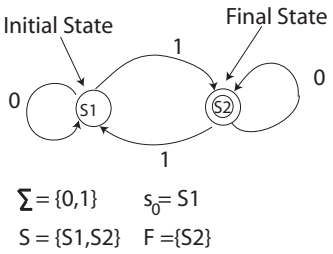


Figure 3: *A finite state automata*

A *finite state automata* can be described as a 5-tuple $(\Sigma, S, s_0, \delta, F)$, where Σ is a finite non-empty set of symbols called input alphabet, S is a finite non-empty set of states, $s_0 \in S$ is an initial state, δ is the state transition function ($\delta : S \times \Sigma \rightarrow S$), and $F \subset S$ is the set of final states.

Figure 3 illustrates an example of a finite state automata that accepts a binary string containing odd number of 1s.

In this section, we describe a DNAzyme based finite state automata, referred to as DNAzyme FSA. At any time an RNA sequence encoding an input symbol is examined by the DNAzyme FSA, then an appropriate state transition takes place, and then the RNA sequence encoding the next input symbol is examined. This process continues till all the input symbols are scanned and the output of the DNAzyme FSA is its state at the end of process.

Figure 4: *(a) Encoding of 0 and 1 in DNAzyme FSA. (b) Protector strand partially hybridizes with the input strand to form bulge loops. The sticky end formed at the end of the input strand outside of the bulge loops represents the active input symbol. This scheme protects the input symbols other than the currently active symbol from becoming active.*

3.1 Encoding the Input Symbols

First of all, we describe the way the input is encoded for the DNAzyme FSA. Input symbols 0 and 1 are encoded as the RNA sequences $x_1 \cdot a_1 \cdot x_2 \cdot a_2$ and $x_1 \cdot b_1 \cdot x_2 \cdot b_2$, respectively, where a_1, a_2, b_1, b_2, x_1, and x_2 are RNA sequences, and $\cdot$ represents concatenation. Figure 3 (a) illustrates this encoding of the input symbols. It should be noted that 0 and 1 share common subsequences x_1 and x_2. Also, there is a special subsequence x at the end of the input subsequence. This is central to the working of the DNAzyme FSA as will be explained later.

3.2 Active Input Symbol

While encoding the input for DNAzyme FSA, it is essential to have a mechanism to detect the current input symbol that is being scanned by DNAzyme FSA. We will refer to this symbol as *active input symbol*. In order to implement this feature in DNAzyme FSA only a small segment of the RNA strand encoding the input symbols is kept active. Most part of it is kept protected by hybridization with a partially complementary sequence, referred to as *protecting sequence*. It has not been shown in the Figure 3 (b) but the protecting sequence should not be one continuous strand. Instead it should contain nicks at various positions. This is

necessary for the working of device and will be explained later. The active input symbol is represented by the sticky end of the RNA sequence encoding the input. We refer to this nanostructure as *input nanostructure*. Figure 3 (b) illustrates the idea. The input nanostructure encodes the input 010. The active input symbol is rightmost 0 (in 010), and it is encoded by the sticky end of the input nanostructure, and hence is active. However, the leftmost 0 and the 1 are encoded in the protected portion of the input nanostructure. They have been protected by hybridization with a protecting sequence. Since the protecting sequence is partially complementary to the RNA sequence encoding the input symbols, it results in the formation of bulge loops. In the Figure 3 b) a_2, a_1, b_2, and b_1 contain a subsequence complementary to t_2, while x_2 and x_1 contain subsequence complementary to t_1. Since the RNA sequence encoding input is partially complementary to the protecting sequence $t_2.t_1.t_2.t_1...$ it forms the bulge loop structure as shown in the Figure 3 (b). Each input symbol is hence represented by two bulge loops. It should be noted that the special sequence x at the end of the input sequence and $\bar{x}$ at the end of protecting sequence ensure that only the desired alignment of protecting sequence with input sequence is favored. As a result, only the desired input nanostructure as shown in Figure 3 (b) is formed.

3.3 States and Transitions

After the description of the input, next we describe the design of states and transitions in finite state machine. In DNAzyme FSA, a network of DNAzymes is embedded on a two-dimensional plane, and the input nanostructure is routed over it. The state of the DNAzyme FSA at any time is indicated by the DNAzyme that holds the input nanostructure at that time. During each state transition of DNAzyme FSA, the segment of input nanostructure encoding the active input symbol is cleaved, the next bulge loop opens up exposing the segment encoding next input symbol, thereby making it new active input symbol, and the input nanostructure jumps to another DNAzyme that indicates the new state of DNAzyme FSA. In subsequent paragraphs, we will explain in details the complete process of state transition in DNAzyme FSA. As shown in Figure 5

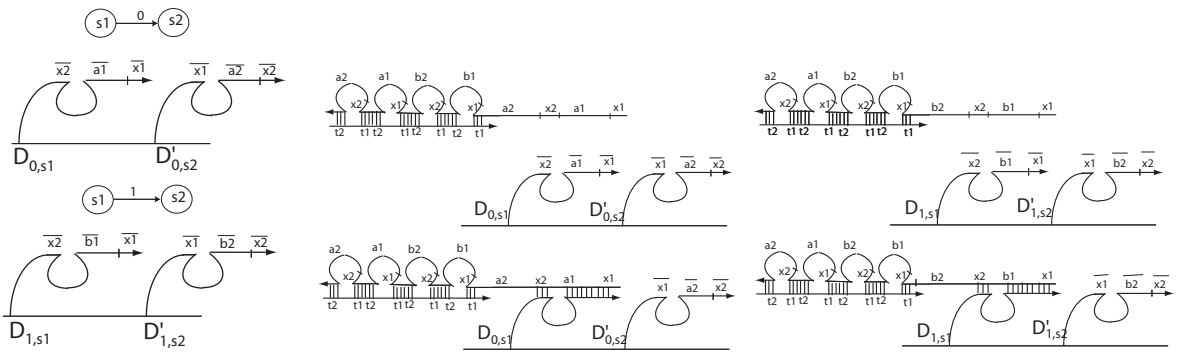


Figure 5: *(a) Figure illustrates the implementation of a state transition through DNAzymes. (b) D_{0,s_1} in the transition machinery for state transition at 0 combines with input nanostructure when active input symbol encoded by the sticky end is 0. When the active input symbol encoded by the sticky end is 1, D_{1,s_1} in the transition machinery for state transition at 1 combines with the input nanostructure.*

(a), a state transition from one state to another is implemented as two evenly spaced DNAzymes, referred to as *transition machinery* for that state transition. Each of these DNAzymes is tethered to another DNA nanostructure, which forms part of the backbone of the DNAzyme FSA. DNAzyme D_{0,s_1} and D'_{0,s_2} form the transition machinery for state transition from state s_1 to state s_2 for input 0. Similarly, DNAzyme D_{1,s_1} and D'_{1,s_2} form the transition machinery for state transition from state s_1 to state s_2 for input 1. It should be noted that in our nomenclature the first subscript of the DNAzyme specifies the active input symbol and the second subscript specifies the states for a transition machinery.

The foremost thing to ensure in DNAzyme FSA is that if the active input symbol is 0, then the state transition for input 0 should be taken. Similarly, if the active input symbol is 1, then the state transition for input 1 should be taken.

In the transition machinery for state transition for input 0, the DNAzymes D_{0,s_1} and D'_{0,s_2} contain DNA subsequences $\overline{x_2} \cdot \overline{a_1} \cdot \overline{x_1}$ and $\overline{x_1} \cdot \overline{a_2} \cdot \overline{x_2}$ respectively, at their free ends. The DNA subsequences of D_{0,s_1} is partially complementary to the RNA sequence that encode the symbol 0 ($x_1 \cdot a_1 \cdot x_2 \cdot a_2$). This ensures that only when the sticky end of input nanostructure is $x_1 \cdot a_1 \cdot x_2 \cdot a_2$, it can hybridize with the DNAzyme D_{0,s_1}. Thus a state transition for 0 is not taken in DNAzyme FSA, unless the active input symbol is 0.

Similarly, in the transition machinery for state transition for input 1, the DNAzymes D_{1,s_1} and D'_{1,s_2} contain DNA subsequences $\overline{x_2} \cdot \overline{b_1} \cdot \overline{x_1}$ and $\overline{x_1} \cdot \overline{b_2} \cdot \overline{x_2}$ respectively, at their free ends. These subsequences are partially complementary to the RNA sequence that encode the symbol 1 ($x_1 \cdot b_1 \cdot x_2 \cdot b_2$). As explained earlier, this ensures that a state transition for 1 is not taken in the DNAzyme FSA, unless the active input symbol is 1. Figure 5 (b) further illustrates the idea.

3.4 Description of State Transition

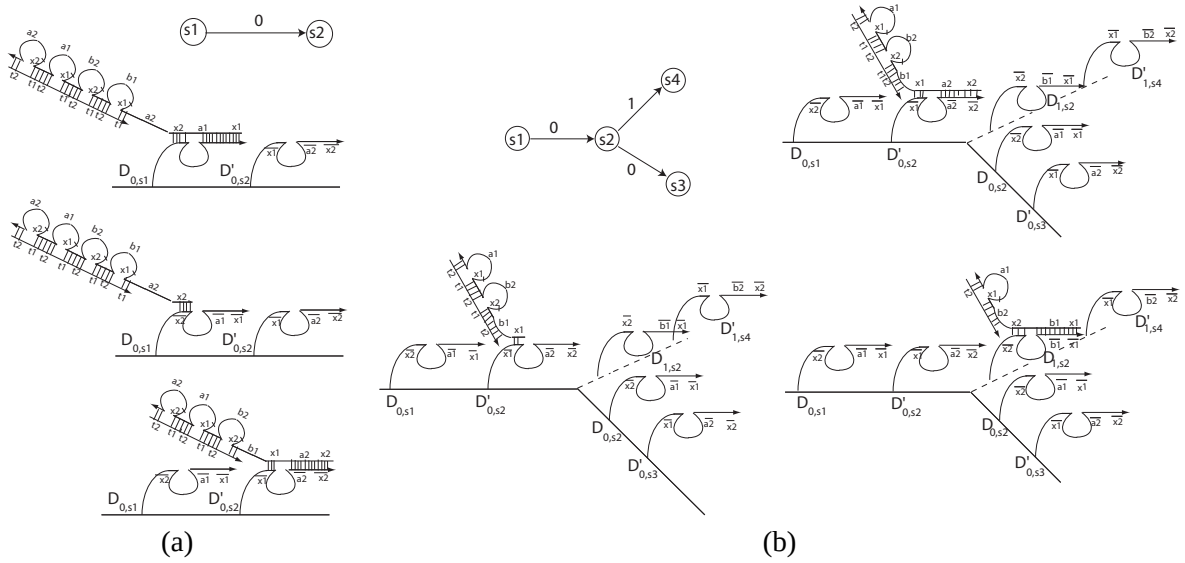


Figure 6: *(a) First half of a state transition by DNAzyme FSA from s_1 to s_2 at input 0 is illustrated. Sequence encoding active input symbol 0 gets cleaved by DNAzyme D_{0,s_1}, input nanostructure moves to next DNAzyme D'_{0,s_2} by strand displacement, and the next bulge loop in the input nanostructure opens up in the process. (b) Second half of a state transition by DNAzyme FSA from s_1 to s_2 at input 0 is shown. The mechanism is similar to the first half. However, in this part the next input symbol and next state transition of DNAzyme FSA is determined, and the input nanostructure lands up on the appropriate transition machinery for the next state transition to begin correctly.*

In this section, we will describe the movement of the input nanostructure over the DNAzymes in a transition machinery to carry out the state transition in DNAzyme FSA. Figure 6 (a) shows a transition machinery for input 0. Initially, the input nanostructure is hybridized with the DNAzyme D_{0,s_1}. The sticky end of the input nanostructure represents the active input symbol 0, and therefore, the transition at input 0 is to be performed. First, the DNAzyme D_{0,s_1} cleaves the input nanostructure as shown in Figure 6 (a). Now the sticky end of input nanostructure has only x_2 as complementary subsequence to the subsequence $\overline{x_2} \cdot \overline{a_1} \cdot \overline{x_1}$ at the free end of DNAzyme D_{0,s_1}. However, the longer subsequence $x_2 \cdot a_2$ in its sticky end is complementary with the subsequence $\overline{a_2} \cdot \overline{x_2}$ of DNAzyme D'_{0,s_2}. Therefore, a strand displacement process takes place with the free ends of DNAzymes D_{0,s_1} and D'_{0,s_2} competing against each other to hybridize with sticky end ($x_2 \cdot a_2$) of the input nanostructure. Since D'_{0,s_2} provides a longer complementary subsequence,

ultimately D_{0,s_1} is displaced and the input nanostructure is now hybridized with D'_{0,s_2} as shown in Figure 6 (a). It should be noted that the next bulge loop gets opened in this process. An input symbol is encoded across two bulge loops in the input nanostructure. As the first half of the sticky end ($x_1 \cdot a_1$) encoding the half of the active input symbol 0 got cleaved, the current sticky end is $x_2 \cdot a_2 \cdot x_1 \cdot b_1$, that contains half of the sequence encoding symbol 0 and half of the sequence encoding the symbol 1. This completes the first half of the state transition by DNAzyme FSA.

The second half of the transition in DNAzyme FSA takes place in exactly similar manner. Half of the sticky end ($x_2 \cdot a_2$) of the input nanostructure that encodes the remaining half of the active input symbol 0 gets cleaved, thus leaving only x_1 as complementary to free end of DNAzyme D'_{0,s_2} ($\overline{x_1} \cdot \overline{a_2} \cdot \overline{x_2}$). At this point the sticky end of the input nanostructure is $x_1 \cdot b_1$ which is half of the sequence that encodes the input symbol 1. It indicates that the next active input symbol is 1 and therefore, the next state transition should be from state s_2 at input 1. This is ensured by the DNAzyme FSA in the following way. Since the sticky end of the input nanostructure is ($x_1 \cdot b_1$), the DNAzyme D_{1,s_2} that has the sequence $\overline{x_2} \cdot \overline{b_1} \cdot \overline{x_1}$ at its free end gets involved in strand displacement with D'_{0,s_2} to hybridize with the sticky end ($x_1 \cdot b_1$) of input nanostructure. Because of the longer complementary sequence D_{1,s_2} ultimately displaces D'_{0,s_2} and hybridizes with the sticky end of nanostructure. This results in the opening of next bulge loop in input nanostructure as shown in Figure 6 (b).

It should be noted that D_{0,s_2} (with sequence $\overline{x_1} \cdot \overline{b_2} \cdot \overline{x_2}$ at its free end) does not have sequences complementary to the sticky end ($x_1 \cdot b_1$) of input nanostructure, so it can not get involved in any strand displacement. Therefore, the input nanostructure is guaranteed to move to the DNAzyme D_{1,s_2}. After the opening of the next bulge loop, the new sticky end ($x_1 \cdot b_1 \cdot x_2 \cdot b_2$) of input nanostructure encodes the input symbol 1. Thus, the input nanostructure lands up in the appropriate transition machinery for the next state transition, and the next state transition at input 1 can begin correctly.

It can be argued in a similar manner that during the second half of the transition, if the next active input symbol was to be 0, the input structure would have moved from DNAzyme D'_{0,s_2} to D_{0,s_2} instead of moving to D_{1,s_2}. We omit the explanation here for the sake of brevity.

Figure 6 (b) illustrates the second half of the state transition of DNAzyme FSA.

It should be noted that the strand displacement of the protector strand also takes place during the process. But since it contains nicks, its fragments just wash away in the solution when they get completely displaced.

3.5 Complete State Machine

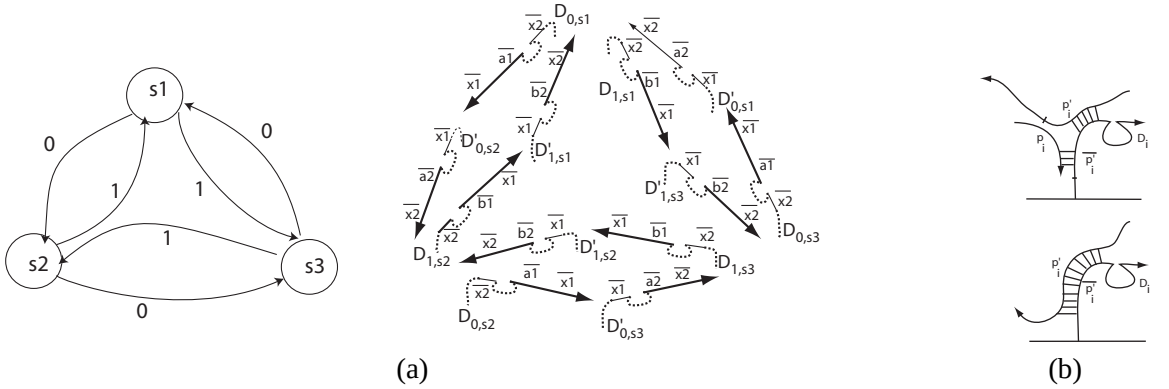


Figure 7: *(a) The DNAzyme implementation of the finite state machine shown on left. (b) Reporting sequence displaces the probe strand from the stem of the DNAzyme that indicates the output state of DNAzyme FSA. Thus, the output can be detected using fluorescent in-situ hybridization technique.*

The components described above can be integrated to implement the complete finite state automata. Any

state transition in the DNAzyme FSA can be implemented by two DNAzymes as described earlier. These DNAzymes are embedded on a nanostructure that forms the backbone of the DNAzyme FSA. The addressable nanostructures formed by DNA origami [25] or fully-addressable DNA tile lattices [22] might provide useful nanostructures for this backbone. Hence, the state machine can be laid out on this nanostructure by implanting a network of DNAzymes on it. The input nanostructure traverses over them in a programmable way and keeps getting cleaved in the process.

Figure 7 (a) shows an implementation of a DNAzyme FSA (at the right) for the finite state automata (at the left). It should be noted that the DNAzymes shown in the Figure 7 (a) are actually implanted on a backbone nanostructure. The dashed lines represent the sides of these DNAzymes that are embedded in the backbone nanostructure.

The output of the DNAzyme FSA is detected using insitu hybridization techniques. The details of the protocol are described in Section 3.6.

3.6 Detecting the Output State

In this section we describe the technique to detect the output of the DNAzyme FSA after completion of the computation on a given input. The state of DNAzyme FSA at the end of the computation is the output state. A special sequence is incorporated inside the last bulge loop in the input nanostructure. We call it as *reporting sequence*. In the DNAzyme FSA described above, as the state transitions take place, the input nanostructure gets cleaved and the bulge loops open up one by one as explained earlier. When the input gets cleaved upto the reporting sequence, the computation is completed. At this time the reporting sequence becomes available, and its position on the DNAzyme FSA indicates the output state. The role of reporting sequence in the detection of final state is described below.

Fluorescent in-situ hybridization (FISH) [7, 26, 31] is a cytogenetic technique which can be used to detect and localize the presence or absence of specific DNA sequences on longer DNA strands. It uses fluorescent probes which bind only to those parts of the DNA strands with which they show a high degree of sequence complementarity. All unhybridized or partially hybridized probes disappear, and only the probes that hybridized to the target are visible in fluorescence.

In our DNAzyme FSA, the stems of the DNAzyme stators contains a unique DNA subsequence. Let us assume that DNAzyme D_i contains the sequence $\overline{p'_i}$. The reporting sequence of the input nanostructure is designed in such a way so that it has segments complementary to each of these $\overline{p'_i}$ subsequences. Hence, reporting sequence is essentially a concatenation of p'_is. At the same time we have different probes each corresponding to a different DNAzyme stator. Each of them is labeled with a different fluorescent dye. It should be noted that probe p_i that is attached to DNAzyme D_i is a subsequence of p'_i as shown in Figure 7 (b).

As mentioned earlier that the reporting sequence becomes available at the end of the computation. In case D_i is the DNAzyme with which the reporting sequence is hybridized at one end, D_i determines the the output state. Since p'_i of the reporting sequence is a better complement to $\overline{p'_i}$ of DNAzyme D_i as compared to the probe p_i. Therefore, the reporting sequence displaces the probe p_i that contains the fluorescent dye from DNAzyme D_i, and p_i disappears in the solution. Figure 7 (b) illustrates this process.

Hence, all other probes except the one that hybridized to the DNAzyme determining the output state are visible in fluorescence. This protocol can be used to detect the output state of the automata.

3.7 Non-deterministic DNAzyme FSA

A *nondeterministic finite state automata* is a 5-tuple $(\Sigma, S, s_0, \delta, F)$, where Σ is a finite set of input symbols, S is a finite set of states, δ is a state transition function $(\delta : S \times (\Sigma \bigcup \{\epsilon\}) \rightarrow P(S)$ where $P(S)$ is the power set of S), ϵ is the empty string, $s_0 \subset S$ is a set of initial states, and $F \subset S$ is a set of final states. Figure 8

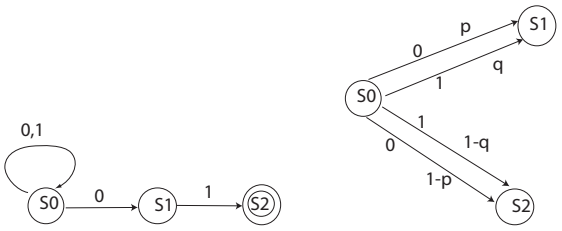


Figure 8: *(a) A non-deterministic finite automata that accepts* $(0+1)^*01$ *(b) Schematic of a probabilistic automata. The transition from state* S_0 *on input* 0 *takes place to state* S_1 *with probability* p *and to state* S_2 *with probability* $1-p$*. Similarly, the transition from state* S_0 *on input* 1 *takes place to state* S_1 *with probability* q *and to state* S_2 *with probability* $1-q$*.* p *and* q *are real numbers between 0 and 1.*

(a) shows one non-deterministic automata that accepts the language $(0+1)^*01$ (the set of binary strings that ends with "01").

The idea extends to the non-deterministic automata directly. Different DNAzyme-FSA described above will work in parallel inside a test-tube. Therefore, the above described scheme will work for non-deterministic automata as well. In case there are more than one transitions possible for one input from one state, each of them will be taken in one DNAzyme-FSA or the other inside the solution, and thus exhibiting non-deterministic nature of the automata. Regarding the output, if the output state in any of the DNAzyme-FSA in solution is an accepting state (or final state), it implies the acceptance of the input by the overall non-deterministic finite state automata.

3.8 Probabilistic DNAzyme FSA

A *probabilistic finite state automata* is a finite state automata in which the state transitions are probabilistic in nature. It can be described as a 5-tuple $(\Sigma, S, s_0, \delta, F)$, where Σ is a finite set of input symbols, S is a finite set of states, δ is a state transition function ($\delta : S \times \Sigma \times S \rightarrow [0,1]$), $s_0 \subset S$ is a set of initial states, and $F \subset S$ is a set of final states. Figure 8 (b) shows a probabilistic automata.

In case the sequences of all the DNAzymes are identical, then the DNAzyme-FSA described above becomes a probabilistic automata having equal probabilities of transitions from any state to any other state. However, to construct an arbitrary probabilistic finite state automata, the probabilistic transitions can be implemented by using partially complementary sequences in the designs. The sequences of the DNAzymes for transition are chosen in a way so that the ratios of probability of hybridization are in accordance with the transition probabilities.

4 DNAzyme Doctor: A Molecular Computer for Logical Control of RNA Expression using DNAzyme

The finite state automaton described in Section 3 can be used in various computational and routing applications. In this section we describe DNAzyme doctor, an application related to medical field. It is an autonomous molecular computer for control of RNA expression based on the overexpression and underexpression of other RNAs.Earlier Shapiro[3] had constructed a molecular computer using protein enzymes for logical control of RNA expression. Their molecular computer analyses the levels of messenger RNA species and in response produces a molecule capable of affecting levels of gene expression. DNAzyme doctor performs the same function, while completely eliminating the use of protein enzymes in the design. For the ease of illustration let us consider a similar example as given in [3]. Suppose a disease is diagnosed positive if RNAs R_1 is underexpressed, R_2 is underexpressed, R_3 is overexpressed, and R_4 is overexpressed. Thus,

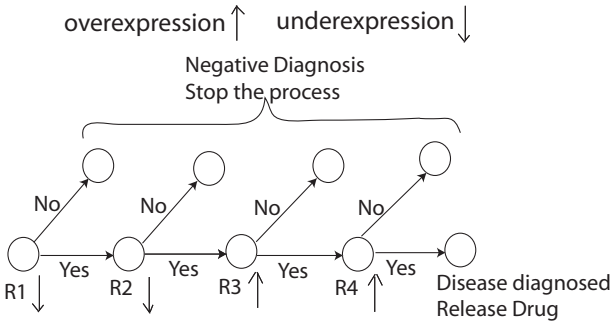


Figure 9: *A state diagram for DNAzyme doctor that controls the release of a drug RNA on the basis of the RNA expression tests for the a disease.*

the detection of the disease can be done by computing logical AND of the above mentioned four RNA expression tests. In case it is established that the disease exists, a curing drug should be released. While in any other case, the drug should not be released. Figure 4 illustrates the aforementioned logic in the form of a state diagram. The sequences y_1, y_2, y_3 and y_4 are characteristic sequences of RNAs R_1, R_2, R_3, and R_4

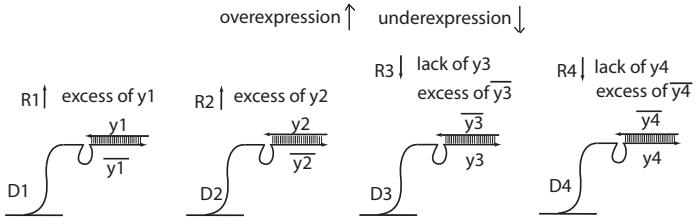


Figure 10: *The figure shows the consequences of overexpression and underexpression of different RNAs on the concentrations of the respective characteristic sequences. The overexpression of R_1 and R_2 results in excess of y_1 and y_2 respectively, and they block the path of input nanostructure by hybridizing with D_1 and D_2. Similarly underexpression of R_3 and R_4 results in excess of $\overline{y_3}$ and $\overline{y_4}$ respectively, to block the path of input nanostructure.*

respectively. The concentrations of the characteristic sequences y_1, y_2, y_3, y_4 as well as their complements $\overline{y_1}$, $\overline{y_2}$, $\overline{y_3}$, $\overline{y_4}$ are regulated at a threshold in accordance with the levels of RNA R_1, R_2, R_3, and R_4. If R_1 is overexpressed then y_1 is in excess, and if R_2 is overexpressed then y_2 is in excess. However, if R_3 is underexpressed, then lack of y_3 and if R_4 is underexpressed, then lack of y_4. But a threshold concentration of $\overline{y_1}$, $\overline{y_2}$, $\overline{y_3}$, $\overline{y_4}$ present in the the solution ensures that lack of y_3 causes excess of $\overline{y_3}$, and lack of y_4 causes excess of $\overline{y_4}$.

Since the DNAzyme doctor only needs to perform a logical AND, it can be implemented in a simple way. We make the input nanostructure walk over four DNAzyme stators implanted on a nanostructure in a straight path (as shown in Figure 11). Each DNAzyme stator represents one of the RNA expression test. In case the test is positive, the input nanostructure moves to next DNAzyme stator, otherwise it gets stuck and ultimately floats away in the solution. Therefore, the successful traversal of input nanostructure over all these DNAzyme stators implies that all tests are positive, and hence positive diagnosis of the disease.

In case the first test is negative (ie. overexpression of R_1), then excessively floating y_1 can bind to $\overline{y_1}$ part of the DNAzyme D_1. Similarly if second, third, or fourth tests are negative (ie.. overexpression of R_2, underexpression of R_3 or underexpression of R_4), then excessively floating y_2, y_3, or y_4 can bind to $\overline{y_2}$, $\overline{y_3}$, $\overline{y_4}$ portions of DNAzyme D_2, D_3, or D_4, respectively. The principle idea is illustrated in Figure 10.

Figure 11 shows the details of the sequences used in the design. It should be noted that $\overline{a_i} \cdot \overline{x_i}$ is a subsequence of $\overline{y_i}$. The input nanostructure traverses over the DNAzymes step by step as shown in Figure 11. The underlying mechanisms of these steps has been explained in Section 3. As explained earlier, when the input nanostructure moves to next DNAzyme, some portion of the sticky end is cleaved, and the next bulge

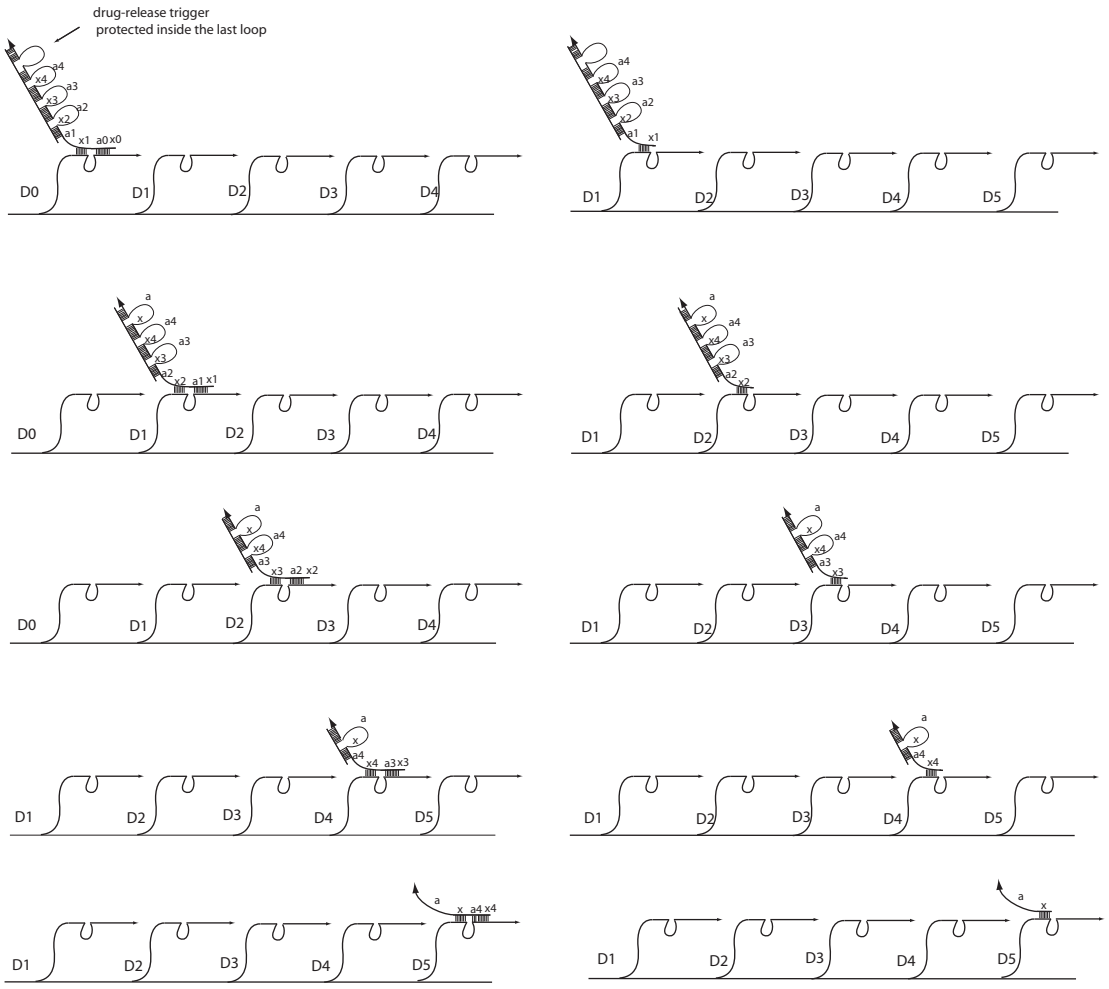


Figure 11: *The input structure walks over the DNAzyme structures D_1, D_2, D_3, and D_4 as explained in Section 3. The drug to be released in case of positive diagnosis of the disease is protected within the last bulge loop of input structure*

loop opens up to restore the length of the sticky end. As can be seen in Figure 11, after the DNAzyme D_3 cleaves the sticky end of input nanostructure, the input structure moves to DNAzyme D_4, and the last bulge loop in input nanostructure opens up. The last bulge loop in the input contains a *drug-release trigger*. After the cleaving action by DNAzyme D_4, the drug-release trigger part of input structure is loosely bound with D_4. The drug-release trigger is then released in the solution. The actual drug is kept protected in the solution, as shown in Figure 10. The drug-release trigger displaces the *lock* strand from the nanostructure that hides the drug as shown in Figure 10.

It should be noted that if any of the tests are negative then the traversal of input nanostructure over the path of DNAzymes is blocked. Hence, if the ith test fails, then the DNAzyme D_i is already hybridized with the DNA sequence y_is. It should be noted that $\overline{a_i} \cdot \overline{x_i}$ is a subsequence of $\overline{y_i}$. The DNAzyme D_i already hybridized with y_i would not participate in strand displacement with previous DNAzyme D_{i-1} to hybridize with sticky end of input nanostructure. Therefore, the input nanostructure can not traverse across this DNAzyme D_i and gets blocked at D_{i-1}.

After the cleaving of half of the sticky end of input nanostructure by DNAzyme D_{i-1}, its binding with D_{i-1} is not too strong either. So finally it detaches from the current DNAzyme and floats away in the solution. Hence the input structure is not cleaved upto the last bulge loop that contains the drug-release trigger, and therefore the drug-release trigger does not get released.

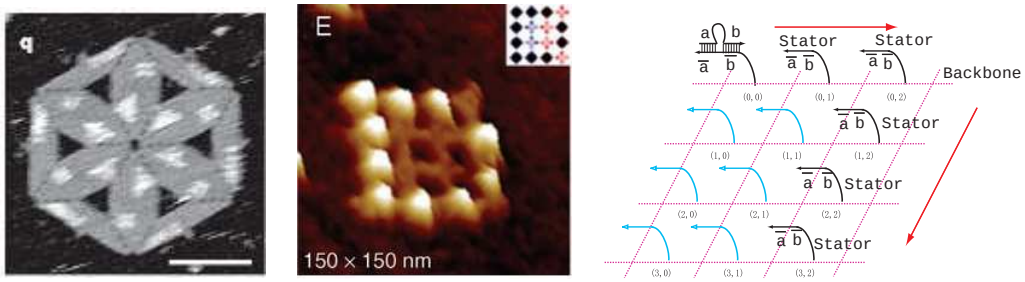
The ultimate goal in designing such a device will be to impart it the ability to perform inside a cell. It will require the device to be protected from other enzymes inside the cell. This protection can be imparted

by embedding the device inside artificial lipozymes.

5 Application of DNAzyme for Routing

DNAzyme crawler can be routed on a two-dimensional DNA lattice in a naive manner as described in Section 5.1. The limitations posed by this simple routing scheme are overcome by DNAzyme router: a DNAzyme based system for programmable routing of the walker on a 2D lattice described in Section 5.2. DNAzyme router is an application based on the design of DNAzyme FSA described earlier in Section 3.

5.1 Routing DNAzyme Crawler in Two Dimensional Lattice



(a) (b) (c)

Figure 12: *(a) A shape with pattern constructed using DNA origami by Rothemund [25](b) Letter D on a fully addressable 2D lattice constructed by Park et. al[22] (c) A predefined path on a fully addressable 2D DNA lattice for a DNAzyme crawler*

Rothemund [25] developed a method for using scaffolded origami to create arbitrary nanoscale shapes (Figure 12 a)) which may be decorated with arbitrary nanoscale patterns. Also, fully addressable two-dimensional DNA tile lattices (Figure 12 b)) have been demonstrated [22]. Specific DNA strands can be mounted at desired locations on these addressable nanostructures. Therefore, DNA stators can be embedded along any arbitrary path in a fully addressable 2D DNA lattice, as shown in Figure 12 c). DNAzyme crawler [34] can be made to travel along this predefined path of DNA stators, hence producing a motion in two-dimensions. However, in this scheme the path on which the DNAzyme crawler travels can be used only once.

The obvious advantage of this scheme is its simplicity. But the disadvantages are that the DNAzyme crawler can only travel on a predefined path and the path gets destroyed as the DNAzyme crawler moves along it. We present a more flexible and non-destructive scheme for two-dimensional routing, referred to as DNAzyme router in Section 5.2.

5.2 DNAzyme Router: DNAzyme based Programmable Routing in Two Dimensions

For any arbitrary path along the network of DNAzymes in a given DNAzyme FSA, an input nanostructure can be designed to traverse along that path. This principle can be used for the design of a programmable routing system. The input nanostructure that moves over the DNAzyme FSA is referred to as *walker* and the complete system as DNAzyme router. The path of the walker is programmed through the state transitions of the automata and the input symbols encoded in the walker. As an example, we can create a state machine on a rectangular grid (Figure 13), in which you move right if the input is 0, and towards bottom if the input is 1. Then by the state machine shown in Figure 13(a) and input shown in Figure 13(b) the walker can be made to travel along the path shown in Figure 13(b).

It should be noted that in a DNAzyme router the path does not get destroyed as a result of the motion of the walker. It is the input nanostructure (walker) that gets cleaved in the process, which is equivalent to

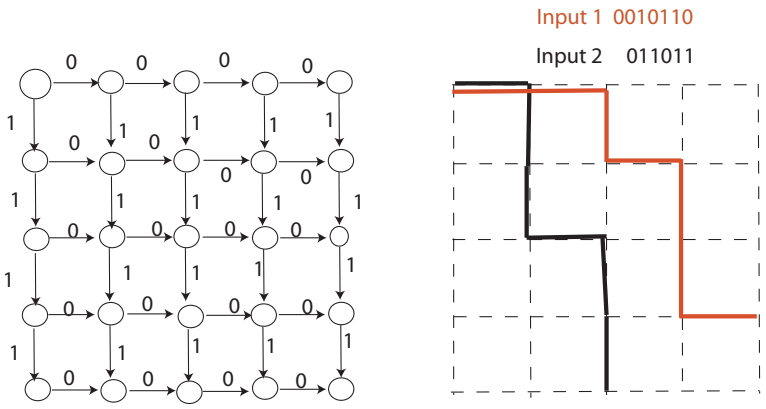


Figure 13: *Illustration of programmable routing in two dimensions*

exhaustion of fuel as a result of motion. Most remarkable feature of DNAzyme router is that we can have multiple walkers moving on the grid independently, each having its own programmed path.

6 Conclusion

We have described the construction of various devices based on the DNAzymes. In this article, we have focused more towards novel designs for the devices that can perform finite state transitions rather than the details of the laboratory implementation. However, it should be noted that among other implementational challenges, construction of the desired bulge loops for input nanostructure needs further investigation. DNAzymes evolve through invitro selection procedures, and these procedures can be designed to generate DNAzymes that cut distinct sequences. In the DNAzyme FSA, the number of DNAzymes required is proportional to the number of transitions in the automata. For binary-coded inputs the number of transitions is proportional to number of states. However, the implementation of finite state machines that do not have a planar layout might be challenging. Thus, it is a question for further research if this scheme can be extended easily to the design of finite state automata, whose layout is non-planar. The molecular computer for logical control of RNA expression can be useful in medical field if it can be used inside a cell, and the programmable walkers can be a really useful tool in nanopartical transportation systems at nanoscale. In conclusion, the designs provided in this paper might provide useful insight for research into many interesting problems in nanotechnology.

References

[1] P. Alberti and J. Mergny. DNA duplex-quadruplex exchange as the basis for a nanomolecular machine. *Proc. Natl. Acad. Sci. USA*, 100:1569–1573, 2003.

[2] 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.

[3] 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.

[4] 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.

[5] I. Biswas, A. Yamamoto, and P. Hsieh. Branch migration through dna sequence heterology. *J. Mol. Bio*, 1998.

[6] M. Bonaccio, A. Credali, and A. Peracchi. Kinetic and thermodynamic characterization of the rna-cleaving 8-17 deoxyribozyme. *Nucleic Acids Res*, 32:916 – 925, 2004.

[7] J. Borlido, S. Pereira, R. Ferreira, N. Coelho, P. Duarte, and J. Pissarra. Simple and fast in situ hybridization. *Plant Molecular Biology Reporter*, 20:219–229, 2002.

[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 and C. Mao. Putting a brake on an autonomous DNA nanomotor. *J. Am. Chem. Soc.*, 126:8626–8627, 2004.

[10] Y. Chen, M. Wang, and C. Mao. An autonomous DNA nanomotor powered by a DNA enzyme. *Angew. Chem. Int. Ed.*, 43:3554–3557, 2004.

[11] L. Feng, S. Park, J. Reif, and H. Yan. A two-state DNA lattice switched by DNA nanoactuator. *Angew. Chem. Int. Ed.*, 42:4342–4346, 2003.

[12] C. M. J., K. Andrew, and S. Lun-Quan. Optimisation of the 10-23 dnazyme-substrate pairing interactions enhanced rna cleavage activity at purine-cytosine target sites. *Nucleic acids research*, 31:2883–2889, 2003.

[13] L. J., W. Zheng, A. Kwon, and Y. Lu. In vitro selection and charcterization of a highly efficient zn(ii) dependent, rna-cleaving deoxyribozyme. *Nucleic Acids Res.*, 28:481–488, 2000.

[14] J. S. Jr. A unified view of polymer, dumbbell and oligonucleotide dna nearest-neighbor thermodynamics. *PNAS*, 95:1460–1465, 1998.

[15] T. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J. Reif, and N. Seeman. The construction, analysis, ligation and self-assembly of DNA triple crossover complexes. *J. Am. Chem. Soc.*, 122:1848–1860, 2000.

[16] J. Li and W. Tan. A single DNA molecule nanomotor. *Nano Lett.*, 2:315–318, 2002.

[17] 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.

[18] C. Mao, W. Sun, and N. Seeman. Designed two-dimensional DNA holliday junction arrays visualized by atomic force microscopy. *J. Am. Chem. Soc.*, 121:5437–5443, 1999.

[19] C. Mao, W. Sun, Z. Shen, and N. Seeman. A DNA nanomechanical device based on the B-Z transition. *Nature*, 397:144–146, 1999.

[20] D. Matsuda and M. Yamamura. Cascading whiplash pcr with a nicking enzyme. *Lecture Notes In Computer Science*, 2568:38 – 46, 2002.

[21] A. Nishikawa and M. Hagiya. Towards a system for simulating DNA computing with whiplash PCR. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 960–966, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.

[22] S. H. Park, C. Pistol, S. J. Ahn, J. H. Reif, A. R. Lebeck, C. Dwyer, and T. H. LaBean. Finite-size, fully addressable dna tile lattices formed by hierarchical assembly procedures. *Angew. Chem. Int. Ed.*, 45:735–739, 2006.

[23] J. Reif. The design of autonomous DNA nanomechanical devices: Walking and rolling DNA. *Lecture Notes in Computer Science*, 2568:22–37, 2003. Published in Natural Computing, DNA8 special issue, Vol. 2, p 439-461, (2003).

[24] J. A. Rose, R. J. Deaton, M. Hagiya, and A. Suyama. Pna-mediated whiplash pcr. *Lecture Notes In Computer Science*, 2340:104 – 116, 2001.

[25] P. Rothemund. Generation of arbitrary nanoscale shapes and patterns by scaffolded DNA origami. *Nature*, 2005.

[26] S. Schroder, M. Hain, and K. Sterflinger. Colorimetric in situ hybridization (cish) with digoxigenin-labeled oligonucleotide probes in autofluorescent hybhomycetes. *Internatl. Microbiol.*, 3:183–186, 2000.

[27] W. Sherman and N. Seeman. A precisely controlled DNA biped walking device. *Nano Lett.*, 4:1203–1207, 2004.

[28] J. Shin and N. Pierce. A synthetic DNA walker for molecular transport. *J. Am. Chem. Soc.*, 126:10834–10835, 2004.

[29] F. Simmel and B. Yurke. Using DNA to construct and power a nanoactuator. *Phys. Rev. E*, 63:041913, 2001.

[30] F. Simmel and B. Yurke. A DNA-based molecular device switchable between three distinct mechanical states. *Appl. Phys. Lett.*, 80:883–885, 2002.

[31] N. Simon, N. LeBot, D. Marie, F. Partensky, and D. Vaulot. Fluorescent in situ hybridization with rrna-targeted oligonucleotide probes to identify small phytoplankton by flow cytometry. *Appl. Environ. Microbiol.*, 61:2506–2513, 1995.

[32] S. S.W. and G. Joyce. Mechanism and utility of an rna-cleaving dna enzyme. *Biochemistry*, 37:13330–13342, 1998.

[33] B. J. Thompson, M. N. Camien, and R.C.Warner. Kinetics of branch migration in double-stranded dna. *Proc Natl Acad Sci U S A*, 73(7):2299–303, 1976 Jul.

[34] 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.

[35] Y. Tian and C. Mao. Molecular gears: A pair of DNA circles continously rolls against each other. *J. Am. Chem. Soc.*, 126:11410–11411, 2004.

[36] A. Turberfield, J. Mitchell, B. Yurke, J. A.P. Mills, M. Blakey, and F. Simmel. DNA fuel for free-running nanomachines. *Phys. Rev. Lett.*, 90:118102, 2003.

[37] E. Winfree. Whiplash pcr for $o(1)$ computing. Technical Report 1998.23, Caltech, 1998.

[38] E. Winfree, F. Liu, L. Wenzler, and N. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394(6693):539–544, 1998.

[39] H. Yan, T. LaBean, L. Feng, and J. Reif. Directed nucleation assembly of DNA tile complexes for barcode patterned DNA lattices. *Proc. Natl. Acad. Sci. USA*, 100(14):8103–8108, 2003.

[40] H. Yan, S. Park, G. Finkelstein, J. Reif, and T. LaBean. DNA-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301(5641):1882–1884, 2003.

[41] H. Yan, X. Zhang, Z. Shen, and N. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415:62–65, 2002.

[42] P. Yin, H. Yan, X. Daniell, A. Turberfield, and J. Reif. A unidirectional DNA walker moving autonomously along a linear track. *Angew. Chem. Int. Ed.*, 43:4906–4911, 2004.

[43] B. Yurke, A. Turberfield, J. A.P. Mills, F. Simmel, and J. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, 2000.