Compact Self-Repairing DNA Lattices

Urmi Majumder, John H. Reif
Department of Computer Science, Duke University,
Durham, NC 27705, USA.
{urmim, reif}@cs.duke.edu

Abstract

Self-repair is essential to all living systems, providing the ability to remain functional in spite of gradual damage. In the context of self-assembly of self-repairing synthetic biomolecular systems, recently Winfree developed a method for transforming a set of DNA tiles into its self-healing counterpart at the cost of increasing the lattice area by a factor of 25. The overall focus of this paper, however, is to develop *compact* designs for self-repairing tiling assemblies with reasonable constraints on crystal growth. Specifically, we use a special class of DNA tiling designs called *reversible* tiling which when carefully designed can provide inherent self-repairing capabilities to patterned DNA lattices. We further note that we can transform any irreversible computational DNA tile set to its reversible counterpart and hence improve the self-repairability of the computational lattice. But doing the transform with an optimal number of tiles, is still an open question. However, for every DNA tile encoding some computation, irrespective of its type, we can modify its design such that it can force only forward reassembly and hence improve the self-repairability of the resultant lattice.

Keywords: self-assembly, self-repair, DNA, cellular automata, reversible computation

1 Introduction

1.1 Motivation

Currently, many scientists are in the process of substituting existing top-down techniques used in conventional manufacturing processes with bottom-up assembly techniques. This involves, among other things, developing self-assembly methods for patterning nano-materials as an alternative to using lithographic techniques. However, eventually, nanostructures can be damaged. What can we do when a self-assembled nanostructure is damaged?

1.2 The Challenge of Self Repairing Biomolecular Systems

This question leads us to realize that nature's capability to self-repair still far exceeds the self-healing capability of synthetic biochemical systems. As nanoscientists are building more complex systems at the molecular scale each day, this challenge of *Self-Repairing Biomolecular Systems* will become increasingly important. In fact, an interdisciplinary team at the University of Illinois at Urbana-Champagne has already developed a polymer composite that has the ability to self heal microcracks.[24] In the context of self-assembled nano-structures, such a system will provide a transition from the existing simple one-time assemblies to self-repairing systems, yielding capabilities that have broad impact to nano-engineering and provide numerous feasible practical applications.

One interesting specific challenge in the area of self-repair to be addressed in this paper is to develop a molecular architecture for self-repairing memory. The interesting feature of this architecture is that, in spite of partial destruction of the nanostructure storing the memory, its bits can be restored.

1.3 Use of DNA Lattices to Demonstrate Self-Repairing Processes

While the ultimate goal here is to build and experimentally demonstrate self-repairing capabilities for a variety of biomolecular systems, we need to begin first with well-understood chemistries. DNA has emerged as an ideal material for constructing self-assembled nanostructures because of its well defined structural properties, immense information encoding capacity and excellent Watson-Crick pairing. Exciting progress has been made on many frontiers of DNA self-assembled structures recently, especially in constructing *DNA Lattices* formed of DNA nanostructures known as *DNA Tiles*.[1–4,6] Thus we feel this provides an ideal platform on which self-repair at the molecular scale can be demonstrated.

1.4 Programmable Self-Assembly and Self-Repairability

One of the important goals of nanotechnology is to develop a method for assembling complex, aperiodic structures. Algorithmic self-assembly(AA) achieves this goal. AA has a strong theoretical foundation due to Winfree[4] but its experimental demonstration is quite limited by assembly errors, because while the theoretical model assumes a directional growth, crystals in reality can grow in all possible directions. This results in ambiguities at the binding sites. Consequently mismatch errors prevent further growth of the computational lattices. This is evident from the few experimental demonstrations of algorithmic assembly we have so far.[20,21]

There have been several designs of error-resilient tile sets[7,14,15] that perform "proofreading" on redundantly encoded information[7] to decrease assembly errors. However, they too assume the notion of forward directional growth of the tiling lattice. Hence self-repair is not always feasible with such tile sets because of errors due to possible re-growth of the lattice in reverse direction.

Winfree,[8] however, recently proposed an ingenious scheme that makes use of modified DNA tiles that force the repair reassembly to occur only in a forward manner. He converted an original tile set into a new set of self-healing tiles that perform the same construction at a much larger scale (5-fold larger scale in each direction and hence the new lattice requires a multiplicative factor of $5 \times 5 = 25$ more area) However, the much larger scale appears to make his construction more of theoretical interest than of practical use. The challenge is to limit the number of new tiles required, so that such a procedure can be applied in practice.

Additionally, we should mention a study that intentionally induces a hole and characterizes the hole in the context of self-repair. In fact it shows that puncturing can be a very effective process for error tolerance.[29]

1.5 Our Paper's Results and Organization

The goal of this paper is to use a class of DNA tile sets with a certain property we call *reversibility* which will allow the reassembly of DNA tiles within a damaged lattice to occur in all possible directions without error with respect to at least two adjacent binding sites.

In section 2 of this paper we discuss how carefully designed reversible computations can improve self-repairing capability of the tiling with a specific instance called *Reversible XOR*. We observe that this lattice allows the first known molecular architecture for a self-repairing memory by storing bits in a two dimensional(2D) spatial domain which due to its self-healing properties is capable of restoring the bits in case of its partial destruction. We further introduce a new measure for computing the self-repairability of a tile set.

In section 3, we discuss techniques from theory of computation to transform irreversible CA to reversible CA that in theory improves the self-repairability of the corresponding computational DNA lattice. We also observe that doing the transformation with minimum number of tiles is still an unsolved problem. However, we conclude with a discussion on how we can improve self-repairability of the damaged lattice by changing the tile design minimally and without exploding the tile set size.

2 Reversible Tiling Lattices and their Self-Repairing Properties

2.1 Reversible Computations and Reversible Tiling Lattices

A computation is said to be *reversible* if each step of the computation can be reversed, so that the computation can proceed in both forward or reverse manner. What is the relevance of reversible computation to the problem of self-repairing molecular assembly? Reversible computations have some unique properties, since they allow a partial computation to complete in a unique manner. A molecular assembly using DNA tiles can be viewed as a computation, where each step is executed by a given tile that is assembled adjacent to other previously assembled tiles. Each tile takes as its inputs the matching tiles of adjacent already placed tiles and provides as output the available free pads. Essentially, the tile computes an individual step mapping the values of its attached pads to the values of its still unattached pads. In general, forward-only tilings assume we are adding to a growing aggregate that started from within a concavity, and where further tiles can only be added to the lattice within the concavity. In contrast, some of the reversible tilings discussed here are also able to extend via tiles added to convex edges of the growing lattice. A careful tile design along with the reversibility property, allows a partially destroyed tiling lattice to be easily repaired, so long in the repair reassembly tiles are added with at least two adjacent matching binding sites. The reversible XOR tile set described just below is an interesting example of reversible self-assembly. It

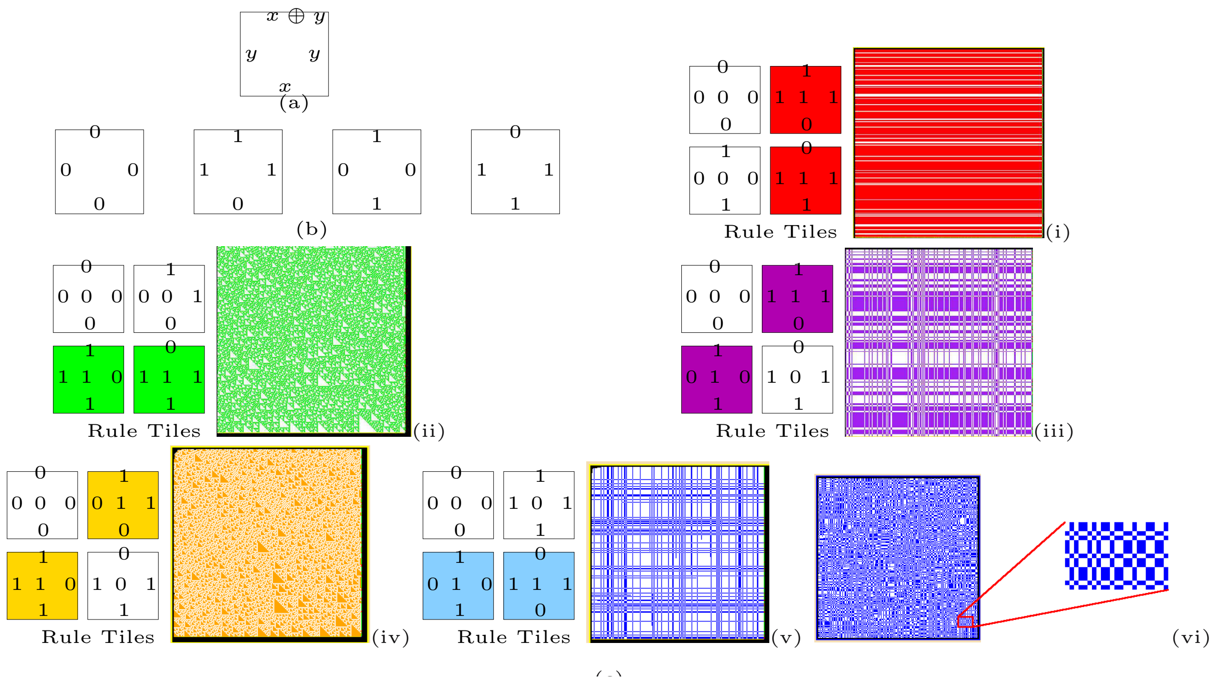


Figure 1: *(a)RXOR template, (b)Four Rule Tiles for RXOR, (c)Assembly of rule tiles within a frame defining the boundary of lattice growth according to aTAM:Rule tiles(left)+ resultant lattice(right):(i) propagation of the input y, (ii) propagation of the input x, (iii) propagation of input y but coloring of the tile based on the xor value in the tile, (iv) propagation of input x but coloring of the tile based on the xor value in the tile, (v)propagation of both inputs but coloring of the tile based on the xor value in the tile, (vi)Assembly of only the rule tiles, portion of error-free lattice(inset).*

realizes a complex pattern that can achieve self-healing without increasing assembly time or number of tile types. In addition, such a self-healing assembly can act as a scaffold for other elements, for e.g. protein and would ensure self healing of the substance to which the self-assembled lattice acts as a scaffold. We now formally define self-repair in the context of self-assembly with square abstract tiles with four sticky ends on four sides, before discussing how reversibility can improve self-repairability.

Definition 1 *We call a tile set self-repairing, if any number of tiles are removed from a self-assembled aggregate to generate convex hole(s) such that all the remaining tiles still form a connected graph*[I]*, then subsequent growth is guaranteed to restore every removed tile without error so long as repair reassembly happens with respect to at least two adjacent binding sites*

Note: This is a more restricted version of self-repairing tile set compared to the one that is described elsewhere.[8] Throughout the paper, we'll use this definition of self-repairing tile set and we also use the terms self-healing and self-repairing interchangeably.

2.2 The Reversible XOR Operation

We will now consider an interesting example of a reversible operation known as *Reversible XOR(RXOR)*.

[I]In the context of tile assembly, each tile is a vertex and the sticky end connections among the tiles denote the edges. An aggregate is connected if every tile can be reached from every other tile in the aggregate following the sticky end connections

The exclusive OR (known as XOR) operation takes as input two Boolean arguments (each can be *true(T)* or false (F)) and returns T if one, and only one, of the two inputs is true. Also, the XOR operation combined with one further simple Boolean operation (that does not alter values between input and output) makes the unit reversible and is known to provide a logical basis to do any Boolean computation. We call this *Reversible XOR(RXOR).*

2.3 RXOR: A Family of Reversible Tiling Lattices

We describe a family of lattices called *RXOR lattices* that uses the XOR operation at each tile to form an interesting patterned lattice with reversible tiles. In the DNA tile implementation, each tile has two sticky ends on each side which is central to the lattice formation. Figure 1(a+b) gives the template and the set of rule tiles for one instance of reversible XOR.

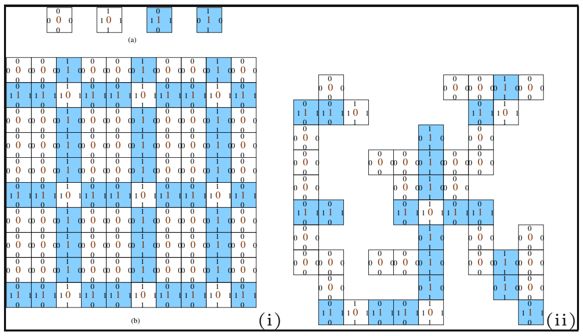
2.3.1 Periodic and Nonperiodic Patterns RXOR Tiling Lattices.

The figures in 1(c) illustrate some of the great variety of (periodic and nonperiodic) patterns that can be generated via RXOR operations at each tile. The rule tiles, the coloring scheme and the ideal lattice formed (when there are no errors) in each case is given in figure 1(c). (**Note:** *All the simulations assume a tile assembly model with $\tau = 2$, where τ is the number of binding sites for a tile to bind to the growing tiling lattice.*) The lattices with triangular patterns(figure 1(c):(ii)+(iv)) are interesting and complex, but it is difficult to determine errors in this lattice. The lattices with band structure(figure 1(c):(i)+(iii)+(v)) are interesting since they can *redundantly store bits* in one/two dimension (by this we mean a bit is propagated through the linear lattice). In general such a $n \times n$ lattice can store n bits (by this we mean n bits are propagated through the $n \times n$ lattice)(figure 1(c):(i)+(iii)) and $2n$ bits as in figure 1(c):(v). Note that although figure 1(c):(ii)+(iv), demonstrate reversible computation they are not self-repairing. However, in figure 1(c):(i)+(iii)+(v) if some of the lattice tiles are removed, the self-repair will restore the lattice and preserve the bits. Error in the lattices from figure 1(c):(i)+(iii) occurs whenever there is a discontinuity in the horizontal bands. Error analysis of the lattice in figure 1(c):(v) is also quite simple(except when two blue bands intersect and the color reverses, any discontinuity in the band structure corresponds to a mismatch error).

2.3.2 Various RXOR Tiling Lattices with Errors

Although all the tiling lattices shown in figure 1 are reversible, we will use the tiling lattice given in Figure 1(figure 1(c):(v)) as the example RXOR lattice in our further discussions below of self-repair. Without a frame self-assembly is error-prone, the lattice formed is not ideal. Hence, to estimate error rates, one can observe the largest portions of the lattice which are error-free[Figure 1(figure 1(c):(vi))]

(a)



(b)

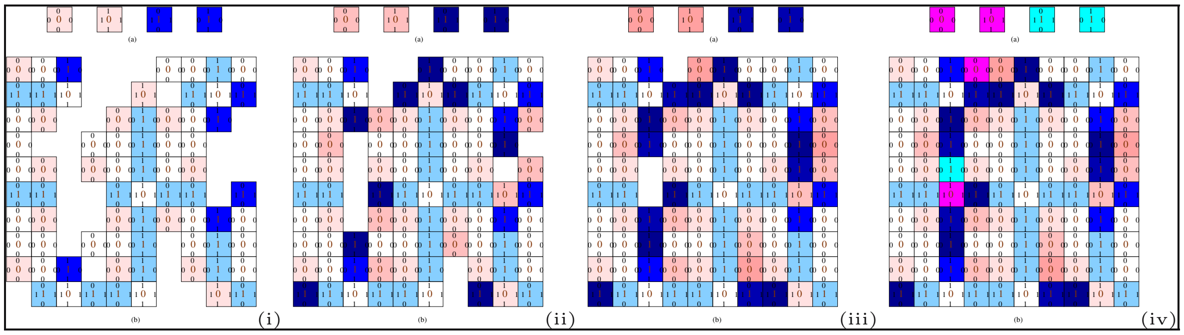


Figure 2: *(a) Original self-assembled lattice(i) and damaged lattice(ii), (b): A possible Self Healing Lattice Growth from (i) to (iv).*

2.3.3 RXOR Tiling Lattices as an Example of Self-Healing Patterned Lattices

Previous work suggested that reversible self assembly can perform "proofreading" on redundantly encoded information.[7] Carefully designed RXOR is an interesting example of reversible self-assembly that achieves self-healing, without increasing assembly time or number of tile types as required by Winfree's Self-Healing construction.[8] For instance, consider the original 10×10 lattice in figure 2(a(i)). Suppose this lattice is damaged and the resulting structure looks like the one in figure 2(a(ii)). Since the tile set is self-healing, so one can recover the original lattice gradually. Ideally in the first step, all the tiles in the damaged lattice with at least two free binding site are available for attaching new tiles which are shown in different shades of the original color scheme[Figure 2(b(i))]. In the subsequent steps, the lattice grows further based on the newly incorporated tiles[Figure 2(b):(ii)+(iii)] and finally one obtains the original lattice[Figure 2(b):(iv)].

2.3.4 Use of RXOR Tiling Lattices to Redundantly Store/Copy Bits

Note that a row or a column in any rectangular window of the lattice in figure 1c(v) is entirely determined by a single cell. For instance, if a tile propagates 0 in the north-south direction and 1 in the east-west direction, then the corresponding column will have tiles with 0 in the north-south direction and the corresponding row will have tiles with 1 in the east-west direction. Thus a $m \times n$ lattice can store a total of $m + n$ bits and can be used as a self-healing memory because if damaged the $m \times n$ memory is capable of recovering all the $m + n$ bits as is shown in figure 2.

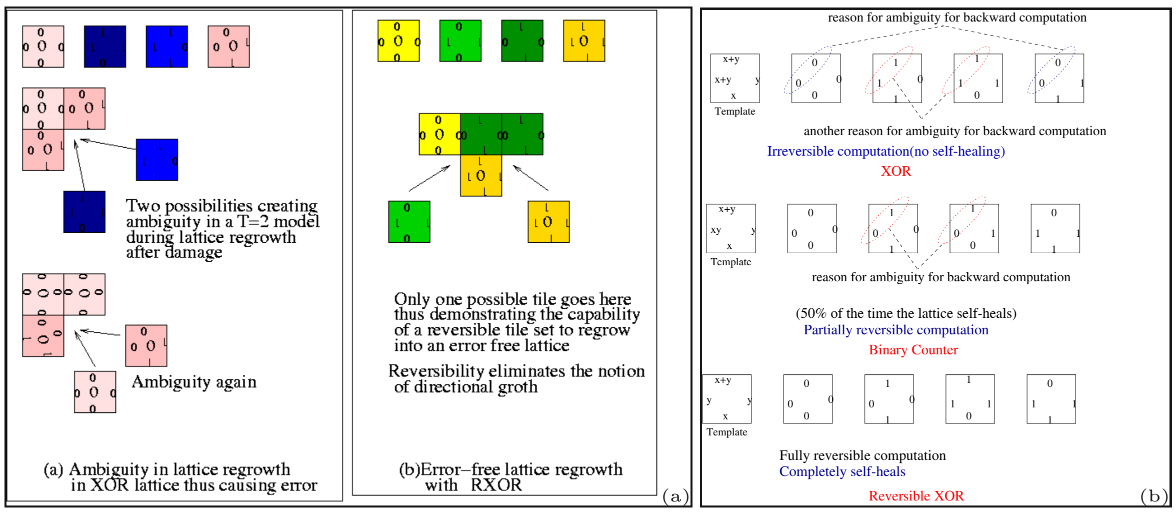

Two possibilities creating ambiguity in a T=2 model during lattice regrowth after damage
Ambiguity again
(a) Ambiguity in lattice regrowth in XOR lattice thus causing error
Only one possible tile goes here thus demonstrating the capability of a reversible tile set to regrow into an error free lattice
Reversibility eliminates the notion of directional groth
(b)Error−free lattice regrowth with RXOR
(a)
reason for ambiguity for backward computation
Template
another reason for ambiguity for backward computation
Irreversible computation(no self-healing)
XOR
Template
reason for ambiguity for backward computation
(50% of the time the lattice self-heals)
Partially reversible computation
Binary Counter
Template
Fully reversible computation
Completely self-heals
Reversible XOR
(b)

Figure 3: *(a) Concrete Self-healing comparison during lattice growth for completely reversible assembly and completely irreversible assembly, (b) Degree of reversibility comparison based on ambiguity of adjacent binding sites for the rule tiles of traditional assemblies.*

2.4 Reversibility improves self-repairability

In figure 3b we present three examples of computation in the increasing order of their reversibility. While computations for RXOR and Sierpinski Triangle pattern generation(ST) are self-explanatory, Binary Counter(BC) computation is reversible 50% of the time since out of the four possible combinations of the two inputs, we can retrieve them from the output sum and output carry only in two cases(when sum is zero and carry is either zero or one). However, in terms of self-repairability, RXOR tiling lattice completely self-heals, since for every possible open site in the latter lattice, there is a unique tile that can be bound to it given the constraints on crystal growth. But both BC and ST tiling lattices create ambiguity for tile attachment in a convex lattice site[Figure 3a]

Based on,[25] we conclude that in general a tile set is self-healing, if the following constraints are satisfied. Let the inputs be x and y and the corresponding outputs be $f(x, y)$ and $g(x, y)$ with the arrangement of the input output ends in an abstract square tile starting from the north end in a clockwise direction as $g(x, y), x, y, f(x, y)$. The tile set is self-repairing if and only if

- if $f(x, y)$ is input sensitive to x if y is constant and $g(x, y)$ is input sensitive to y if x is constant and
- if both change then at least one of $f(x, y)$ or $g(x, y)$ also changes, the tile set is self-repairing

2.5 A Measure for Error-Resilience and Self-Repairability

In general, when we design a tile set for algorithmic self-assembly it would be very useful if we can estimate its robustness against mismatch errors so long crystal growth occurs with respect to at least two adjacent binding sites. This also applies to the self-healing of a damaged lattice. Thus, we introduce a new measure for self-repairability of a tile set which we call ”corner site ambiguity”. This is inspired by Hönberg *et al.*[23] where the authors address the question of how the properties of a tile system are related to the periodicity of the resultant self-assembled nanostructures. We now define a *corner site* and *corner site ambiguity*

Definition 2 *A corner site is a pair of adjacent binding sites in a growing aggregate or in a convex hole in a damaged lattice.*

To reiterate, our abstract tiles are squares as in the original tile assembly model and the inputs are at the south and east ends. Thus the total number of possible corner sites with the number of tiles in the tile set T as w is $4w$.

Definition 3 *We define corner site ambiguity $C(T)$ as the average number of tiles in a tile set T that can bind to an available corner site.*

To measure $C(T)$, we compute the number of tiles that can bind to each corner site first and then compute the average.

2.5.1 Corner Site Ambiguity and Self-Repairability

A tile set is self-repairing if and only if,

$$C(T) = \forall T, min\{C(T)\}$$

In other words, if all of the $4w$ corner sites are distinct, then exactly one tile can bind to it and hence $C(T) = 1$. In terms of concrete examples, the RXOR tile set has $C(T) = 1$ while each of ST and BC tile set has $C(T) = 1.25$. Obviously, the higher the value of $C(T)$, the more error-prone is the resultant assembly and re-assembly after lattice damage.

3 Self-Repairing Transformation for any Tile Set

One of the major goals of algorithmic self-assembly is to provide compact designs for self-repairing error-resilient tile set. In this paper we demonstrated that a carefully designed tile set performing reversible computation can be self-repairing. Unfortunately, not all reversible tile sets are self-repairing. However, since reversibility ensures uniqueness for the adjacent pair of outputs, it definitely does improve the self-repairability of the tile set. So transforming an irreversible tile set into a reversible tile set improves its error-resilience. In fact we can show that reversible tiling is Turing Universal and thus any tile set will benefit from such a transformation.

3.1 Transforming Irreversible Computational Lattices into Reversible Tiling Systems

3.1.1 1D Cellular Automata

We first define some of the concepts involved in discussing the methods that would transform an irreversible computational lattice into its reversible counterpart.

Definition 4 *A 1D cellular automaton is a discrete model of computation that comprises of a regular grid (of arbitrary finite dimension) of finite state automata known as cells, where each cell can be in one of a finite number of states. The state of a cell at time t(discrete), is a function of the state of a finite number of cells called the neighborhood at time $t - 1$. At each time step, the state functions or the rules are applied to the whole grid and a new generation is produced. This assignment of states to all cells results in a new configuration.*

In other words, the dynamics of a cellular automaton is given by its local map, which is used at every time step by each cell to determine its new state from the current state of certain cells in its vicinity. Formally, the local map is the composition of two operators, viz. the *neighborhood* which enumerates the cells affecting the given cells and the *table* which specifies how those cells affect it.

In order to be able to simulate reality, cellular automata should abide by a number of laws of physics viz *locality* and *reversibility*. By locality we mean that the smaller the computational elements, the more densely they can be packed in a given volume and hence faster is the computation. The other feature is *reversibility* which implies that information can neither be created or destroyed and thus the second law of thermodynamics is valid in this case. Cellular automata are also known to be *Turing universal*, which implies they are capable of any computation that any other standard model of computation (such as a Turing machine or a conventional random access computer) is capable of.

Definition 5 *A cellular automaton is reversible if and only if for every current configuration of the CA there is exactly one immediately prior configuration. Formally, by applying the local next step mapping to every cell of the array, from any configuration q one can obtain a new configuration $q^{'}$. This transformation is called the global next state map on the set of configurations. A cellular automaton is reversible if the global map is invertible.*

Reversible Cellular Automaton from Irreversible Cellular Automaton The main observation in creating a reversible cellular automaton (CA) is that the rule for the system remains unchanged if all its elements have inputs and outputs reversed. There are known algorithms for finding pre-images for one dimensional(1D) CA. In fact, any 1D rule can be proved to be reversible or irreversible. However determining reversibility of a CA in higher dimensions is an undecidable problem, so has no finite algorithm that always halts.

Several heuristic methods for constructing a reversible CA have also been proposed. However, the most useful ones for constructing one with a predefined set of properties are a)*Second order* *techniques* and b)*Partitioning schemes* . The literature (e.g., see Wolfram[13]) gives several examples of reversible CA. See Toffoli and Margolus,[11] for a list of applications of reversible CA.

3.1.2 Simulation of 1D Cellular Automaton by 2D DNA Lattices

It was shown by Winfree[18] that a two dimensional tiling lattice can be used to simulate any one dimensional cellular automaton. In particular, each horizontal row of n tiles of the tiling lattice can simulate a given step of the one dimensional cellular automaton, and the values from each automata configuration are communicated from a prior row of tiles to the next row of tiles above it. We can use similar argument to prove that if the given CA is reversible, then the resulting tiling lattice is reversible.

Theorem 3.1.1 *A two dimensional DNA lattice with reversible tiles and size $n \times T$ can be used to simulate a one dimensional reversible CA with n cells running in time T.*

Proof Following Winfree[18] we use a special kind of CA : *Blocked Cellular Automaton* where for each row, cells are read in pair and two symbols are written guided by the rule table. For each rule, $(x, y) \longrightarrow (u, v)$, we create a tile whose sticky ends on the input side(south and east sides of a cross tile) are $\overline{x}$ and $\overline{y}$ and that at the output ends(west and north sides of a cross tile) are u and v. We also have an initial assembly of tiles simulating the initial BCA tape. We add the rule tiles to the solution containing the initial tape. As figure 4 demonstrates, rule tiles anneal into position if and only if both sticky ends match. Thus we can simulate forward computation with DNA assembly. As described in[18] we access the output using a special "halting" tile gets incorporated in the lattice.

For a reversible CA, by definition, there's exactly one prior configuration for every current configuration. In terms of DNA assembly this implies that if we treat u and v as our inputs(north and west ends of a cross tile) and $\overline{x}$ and $\overline{y}$ as our outputs(south and east ends of the cross tile), then also the rule tiles will anneal into position abiding by the sticky ends match constraint. Thus we can simulate backward computation with reversible DNA assembly. For instance, if we remove the tiles which are crossed in Figure 4, since the two functions are invertible, so the correct rule tiles will reassemble, thus demonstrating reversible computation.

In particular, each horizontal row of n tiles of the tiling lattice simulates a given step of the CA, and the values from each automata step to step are communicated from a prior row of tiles to the next row of tiles above it. Thus, a two dimensional DNA lattice with reversible tiles and size $n \times T$ can be used to simulate a one dimensional reversible CA with n cells running in time T.

Theorem 3.1.2 *A 2D DNA lattice performing reversible computation and size $(2n + 7)T$ can simulate a DNA lattice performing irreversible computation and size nT*

Proof Morita[22] outlined a transformation technique for converting a 1D 3 neighbor CA to a 1D 3 neighbor partitioned CA. We can directly map this technique to DNA self-assembly.

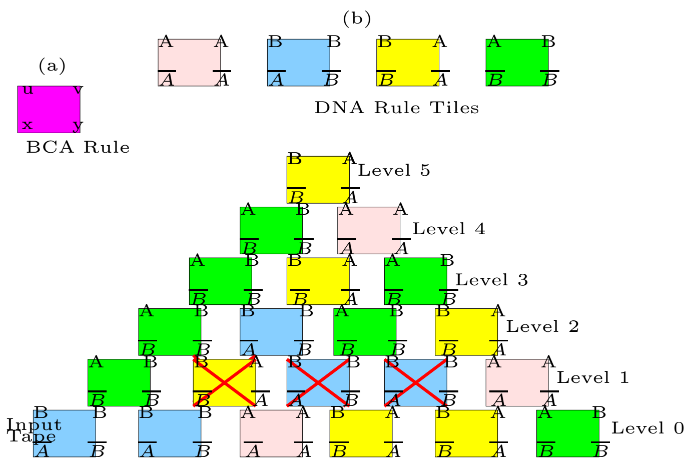


Figure 4: *((a)BCA rule in an abstract DNA tile, (b)DNA Rule Tiles for RXOR Computation, (c)Both forward and backward computation takes place starting from the initial tape.*

If we restrict ourselves to one function instead of two, as in ordinary CA, one can have an abstract tile which has six sticky ends, three for inputs and three for outputs. In case of the ordinary CA, the three inputs denotes the values of the cell and its left and right neighbors while the three outputs contain the same output value which serve as inputs to the triplet directly above in the next computation step in a $\tau = 3$ Abstract Tile Assembly Model[17][Figure 5a].

In case of a partitioned CA, the inputs are right output of the left cell, center output of the middle cell and the left output of the right cell directly below(corresponds to the previous computation step) and the three outputs are respectively the left, center and right outputs of the cell. The latter serve as the right input of the left cell, the center input of the middle cell and the left input of the right cell directly above(corresponds to the next computation step)[Figure 5b].

Morita[22] further showed that the number of steps $T(n)$ to simulate one step of an ordinary 1D 3 neighbor CA A by a partitioned D 3 neighbor CA P is $2n + 7$ in the worst case. When combined with the previous theorem, this implies the result.

One observation here is that, although the size of the transformed tile set is still asymptotically the same as before(A CA with alphabet C and $|C|^3$ rules will have $O(|C|^3)$ for its reversible counterpart crystal growth in a $\tau = 3$ model), the blow up for all practical purposes is fairly high. For instance, if we consider a binary alphabet then a tile set with 8 tiles yield a set of 58 tiles in its reversible counterpart. Accommodating this seven fold increase in a biomolecular implementation is not very practical yet.

This motivates us to investigate redundancy based self-repairing schemes where redundancy is created by encodings in the pads of the tiles with no scale up of the assembly. Unfortunately this direction is no more promising.

Definition 6 *A redundancy based compact error resilient scheme is an error reduction scheme,*

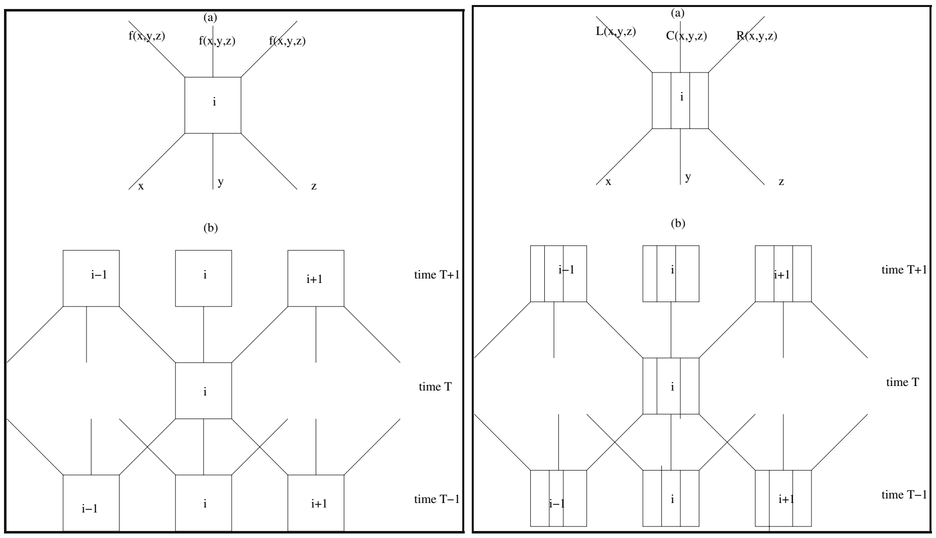


Figure 5: *(a)Abstract Tile for original CA and its tiling simulation, (b)Abstract Tile for corresponding PCA and its tiling simulation.*

where redundancy is created by encodings in the pads with absolutely no scale up of the assembly. Hence, the computation at position (i, j) is still performed at the same position. However, there is an increase in the number of type of pads for tiles.

Theorem 3.1.3 *There is no compact reversible transformation to generate a self-healing tile set for any irreversible computation using redundancy based scheme.*

Proof Following the notation proposed by Sahu *et al.*,[25] in order to make any tile set reversible we can incorporate the inputs at the corresponding output ends. In order to maintain consistency on the choice of inputs, we need to transform the original tile $V(i, j+1), U(i, j), V(i, j), U(i+1, j)$(from north in a clockwise fashion) [Figure 6a] to its reversible counterpart with $V(i, j+1), V(i, j), V(i-1, j)$ at the north end, $U(i, j), U(i-1, j), U(i-1, j-1)$ at the east end, $V(i, j), V(i, j-1), V(i-1, j-1)$ at the south end and $U(i+1, j), U(i, j), U(i, j-1)$ at the west end[Figure 6b] . However, then the $U(i, j)$ and the $V(i, j)$ at the input ends become a function of the rest of the inputs and hence this tile does not represent a valid computational unit. Even if we accept the dependency the transformation incorporates 4 sets of computation in the new tile as opposed to one in the original tile and reversibility only helps for a single level[Figure 6c]. For instance say keeping the values of $(U(i-1, j), V(i, j-1))$ constant, two choices for the tuple $(U(i-1, j-1), V(i-1, j-1))$ yield the same output pair $(U(i, j-1), V(i-1, j))$. Now the latter are input to the $(i-1, j)$ and $(i, j-1)$ original modules. Since the values of $(U(i-1, j), V(i, j-1))$ pair is constant, the outputs from the (i, j) original module will also be the same and hence such a construction will not be self-repairing.

Thus it still remains to be answered whether an irreversible tile set can be transformed to its reversible counterpart using a minimal tile set. However, reversible computation has its own

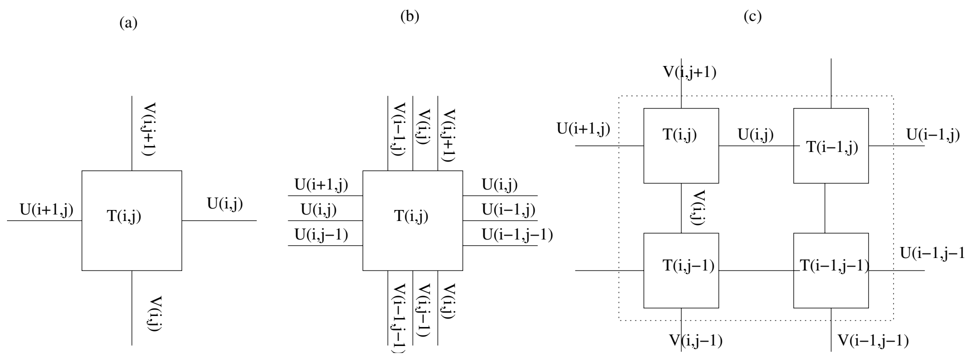


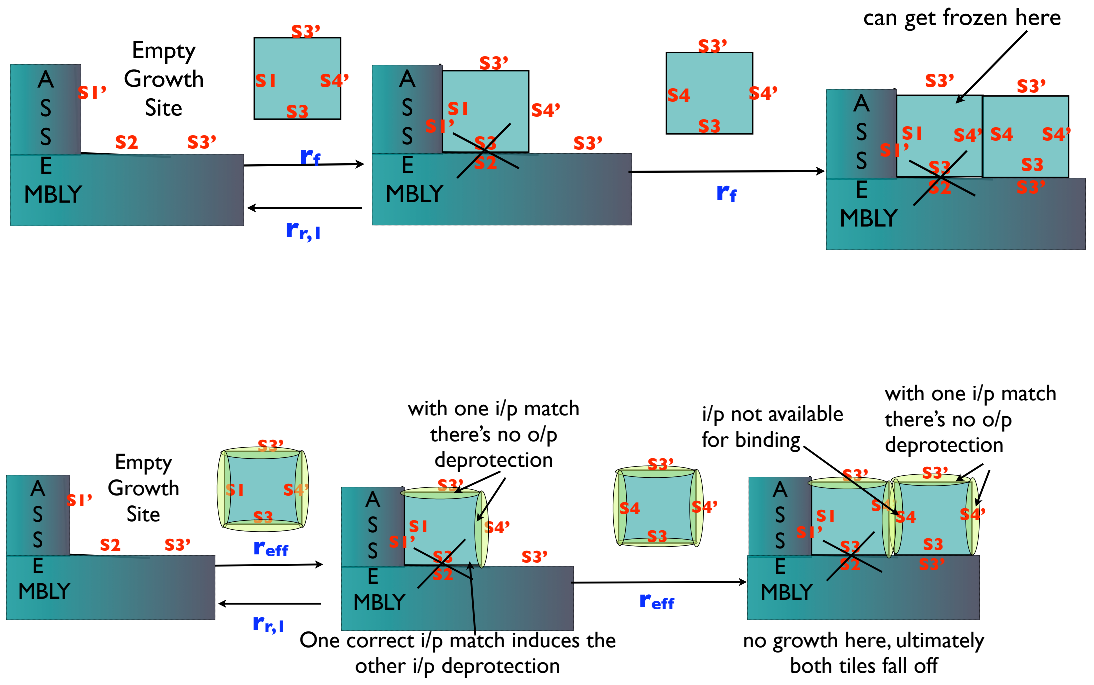
Figure 6: *(a)Original Abstract Tile,*[25] *(b)Transformed Abstract Tile capable of reversible computation, (c)A single computational unit in the transformed tile corresponds to four sets of computation in the original tile.*

merits in quantum computing, optical computing, nanotechnology and low power CMOS design. In fact, if we can have 3D DNA assembly, that would allow us to propagate the redundant bit in the third dimension, we can hope to improve the self-repairability of the resultant assembly given that crystal growth occurs with respect to at least *three* adjacent matching binding sites.

3.2 Modified DNA Tiles that Force only Forward Reassembly

Although improving self-repairability using reversible computation is not quite promising for all kinds of computation, we can, however, make minimal changes to the tile design and improve the self-healing capability of the damaged lattice. Activatable tiles[28] proposed by Majumder *et al.* can be used for repair reassembly in DNA lattices. An activatable tile system works in the following manner: We start with a set of "protected" DNA tiles, which we call *activatable tiles*; these tiles do not assemble until an initiator nanostructure is introduced to the solution. The initiator utilizes strand displacement to "strip" off the protective coating on the input sticky end(s) of the appropriate neighbors.[30] When the input sticky ends are completely hybridized, the output sticky ends are exposed. DNA polymerase enzyme can perform this deprotection,[31] since it can act over long distances (e.g: across tile core) unlike strand displacement. The newly exposed output sticky ends, in turn, strip the protective layer off the next tile along the growing face of the assembly.

In essence, if we have an available growth site and an activatable tile binds to it, the former has to be completely deprotected (i.e. correctly matched) before another activatable tile can bind to it. Hence, a wrong tile cannot be "frozen" in an assembly. This process has been explained in Figure 7. However, it is not a zero probability event. We assume that output deprotection is an irreversible event. Hence if the first tile leaves the growth site after it is deprotected completely, then it will be equivalent to an "unprotected" tile and can cause all the same problems caused by the latter. Fortunately, we can bound the probability of error by tweaking the various free parameters like sticky end length, type of polymerase, protection strand and primer length and others. Hence the tiles in solution are mostly protected. Consequently after a hole has been punctured in the lattice, re-growth takes place using mostly forward accretion. There is, however, a small probability of



ASSEMBLY
Empty Growth Site
S1'
S2
S3'
S3'
S1
S4'
S3
rf
rr,1
S1
S1'
S4'
S3
S2
S3'
S3'
S4
S4'
S3
rf
can get frozen here
S3'
S3'
S1
S4'
S4
S4'
S1'
S3
S3
S2
S3'
reff
rr,1
with one i/p match there's no o/p deprotection
One correct i/p match induces the other i/p deprotection
reff
i/p not available for binding
with one i/p match there's no o/p deprotection
no growth here, ultimately both tiles fall off

Figure 7: *(Top) Markov Chain representation of a tile with no protection binding to a growth site.* r_f *denotes the rate of tile association while* $r_{r,b}$ *denotes the rate of tile with* b *matches dissociating. The glues on the tile are shown in red. A glue* s *matches with a glue on the interfacing tile if the latter is* $s^{'}$*. The Markov Chain shows how a tile with single binding site match can bind to the growth site and before it can leave since one binding site matching is not as favorable as two binding site matching, another tile with two binding site matching comes and "freezes" the former in its growth site (Bottom). With activatable tiles such freezing cannot happen since the output protection (depicted in yellow) are not removed until both inputs correctly match. Hence the second tile cannot bind with enough binding energy and both tiles will fall off eventually. This would ensure correct forward accretion of tiles in the damaged growth area of the lattice.*

backward growth from the unprotected tiles that was once part of the original tiling assembly and dissociated after outputs are deprotected. The likelihood is comparatively small since the forward reaction rate depends on the concentration of the monomers and the protected tiles are much more abundant than their unprotected counterparts.

4 Discussion

Although molecular self-assembly appears to be a promising route to bottom-up fabrication of complex objects, to direct growth of lattices, error-free assembly cannot be assumed. Thus in this paper with our compact design of self-repairing tile sets, we addressed the basic issue of fault tolerant molecular computation by self-assembly. Our design exploited the reversibility property to provide inherent self-repairing capabilities with some constraints on crystal growth. We observed that this design will allow the first known molecular architecture for self-repairing memory. Finally we observed that although in theory we can construct 2D reversible computational DNA lattices for 1D irreversible CAs and hence improve the self-healing capability of resultant computational lattice. Doing the transformation, however, with a minimal tile set is still an open question. Nevertheless, for every computation, irrespective of whether it is reversible or irreversible, we can encode the information in DNA tiles and transform these DNA tiles to their activatable counterparts and use them in the assembly instead to improve self-repairability of damaged DNA lattices.

5 Acknowledgments

The authors are supported by NSF EMT NANO grants CCF-0829798 and CCF-0523555.

References

[1] D. Liu, M. S. Wang, Z. X. Deng, R. Walulu, and C. D. Mao, *J. Am. Chem. Soc.*, 126:2324–2325, 2004.

[2] 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.

[3] C. Mao, W. Sun, and N. C. Seeman, *J. Am. Chem. Soc.*, 121:5437–5443, 1999.

[4] 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.

[5] 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.

[6] 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.

[7] Erik Winfree and Renat Bekbolatov, Proofreading tile sets: Error correction for algorithmic self-assembly, *DNA Computing* , 2943:126-144

[8] Erik Winfree Self Healing Tile Sets *Nanotechnology: Science and Computation, pages 3-21, 2006*

[9] T Toffoli Computation and construction universality of reversible cellular automata *J. Comp. Syst. Sci*, 15:213-231, 1977

[10] Morita, Kenichi and Masaterus Computation universality of one dimensional reversible (injective) cellular automata *Trans. IEICE E 72*, 72:758-762, 1989

[11] T Toffoli and N Margolus Invertible Cellular Automata: A Review *Physica*, 1994

[12] J-C Dubacq How to simulate Turing machines by invertible one-dimensional cellular automata

[13] S Wolfram A New Kind of Science *Wolfram Media Inc*, 2002

[14] J H Reif, S Sahu and P Yin, Compact error-resilient computational DNA tiling assemblies, *Tenth International Meeting on DNA Based Computers (DNA10)*, 2004

[15] HL Chen and A Goel, Error Free Self-Assembly using error-prone tiles, *DNA Computing 10, 2004*

[16] R Schulman, E Winfree, Controlling nucleation rate in algorithmic self-assembly, *DNA Computing 10*, 2004

[17] Winfree E, Simulations of Computing by Self-Assembly, Caltech CS Tech Report 1998.22

[18] Winfree E, On the Computational Power of DNA Annealing and Ligation, DNA Based Computers, pgs 199-221, 1996

[19] Toffoli T Reversible Computing, Automata, Languages and Programming, Springer Verlag, pp.632-644

[20] Rob D. Barish, Paul W. K. Rothemund, and Erik Winfre,e Two Computational Primitives for Algorithmic Self-Assembly: Copying and Counting, Nano Letters 5(12): 2586-2592

[21] Paul W.K. Rothemund, Nick Papadakis and Erik Winfree. Algorithmic Self-Assembly of DNA Sierpinski Triangles, PLoS Biology 2 (12) e424, 2004

[22] Morita K, Reversible simulation of one-dimensional irreversible cellular automata, Theoret. Comput. Sci., 148:157-163, 1995

[23] Hönberg B and Olin H, Programmable Self-Assembly-Unique Structures and Bond Uniqueness, J. Comput. Theor. Nanosci. 2006, Vol 3, 1-7, 2006

[24] S.R. White, N.R. Sottos, P.H. Geubelle, J.S. Moore, M.R. Kessler, S.R. Sriram, E.N. Brown and S. Viswanathan, Autonomic healing of polymer composites, Nature, 409, 794-797, 2001

[25] Sudheer Sahu and John Reif, Capabilities and Limits of Compact Error Resilience Methods for Algorithmic Self-Assembly in Two and Three Dimensions, Twelfth International Meeting on DNA Based Computers (DNA12), Seoul, Korea, June 5-9, 2006.

[26] Seeman, N. C, J. Biomol. Struct. Dyn. 1990, 8, 573.

[27] E Winfree, F Liu, LA Wenzler and NC Seeman, Nature, Vol: 394, 1998

[28] U Majumder, T.H. LaBean and J.H. Reif Activatable Tiles: Compact, Robust, Programmable Assembly and other Applications DNA13, LNCS 4848, 2008.

[29] M Hashempour, Z M Arani and F Lombardi Error Tolerance of DNA Self-healing Assemblies by Puncturing 22nd IEEE Intl. Symp. on Defect and Fault Tolerance in VLSI Systems, 2007.

[30] B J Thompson, M N Camien and R C Warner PNAS, 73(7), pp: 2299-2303, July, 1976.

[31] http://en.wikipedia.org/wiki/Polymerization.