Activatable Tiles for Compact Robust Programmable Molecular Assembly and Other Applications*

Urmi Majumder† Sudhanshu Garg ‡ Thomas H. LaBean§and John H. Reif¶

Abstract: Algorithmic DNA self-assembly is capable of forming complex patterns and shapes, that have been shown theoretically, and experimentally. Its experimental demonstrations, although improving over recent years, have been limited by significant assembly errors. Since 2003 there have been several designs of error-resilient tile sets but all of these existing error-resilient tile systems assumed directional growth of the tiling assembly. This is a very strong assumption because experiments show that tile self-assembly does not necessarily behave in such a fashion, since they may also grow in the reverse of the intended direction. The assumption of directional growth of the tiling assembly also underlies the growth model in theoretical assembly models such as the TAM. What is needed is a means for enforce this directionality constraint, which will allow us to reduce assembly errors.

In this paper we describe a protection/deprotection strategy to strictly enforce the direction of tiling assembly growth so that the assembly process is robust against errors. Initially, we start with (i) a single "activated" tile with output pads that can bind with other tiles, along with (ii) a set of "deactivated" tiles, meaning that the tile's output pads are protected and cannot bind with other tiles. After other tiles bind to a "deactivated" tile's input pads, the tile transitions to an active state and its output pads are exposed, allowing further growth.

*This paper is a revised version of the conference proceedings extended abstract: Urmi Majumder, Thomas H. LaBean, and John H. Reif, Activatable Tiles: Compact, Robust Programmable Assembly and Other Applications, in DNA Computing: DNA13.

†Oracle Corporation, Washington D.C., USA (urmi.majumder@gmail.com)

‡Department of Computer Science, Duke University, Durham, NC, USA (sgarg@cs.duke.edu)

§Materials Science & Engineering Department at NC State, Raleigh, NC, USA (thlabean@gmail.com)

¶Department of Computer Science, Duke University, Durham, NC, USA (reif@cs.duke.edu) and Adjunct, Faculty of Computing and Information Technology (FCIT), King Abdulaziz University (KAU), Jeddah, Saudi Arabia

When these are activated in a desired order, we can enforce a directional assembly at the same scale as the original one. Such a system can be built with minimal modifications of existing DNA tile nanostructures. We propose a new type of tiles called *activatable tiles* and its role in compact proofreading. Activatable tiles can be thought of as a particular case of the more recent Signal Tile Assembly model, where signals transmit binding/unbinding instructions across tiles on binding to one or more input sites.

We describe abstract and kinetic models of activatable tile assembly and show that the error rate can be decreased significantly with respect to Winfree's original kinetic tile assembly model without considerable decrease in assembly growth speed. We prove that an activatable tile set is an instance of a compact, error-resilient and self-healing tile-set. We describe a DNA design of activatable tiles and a mechanism of deprotection using DNA polymerization and strand displacement. We also perform detailed stepwise simulations using a DNA Tile simulator Xgrow, and show that the activatable tiles mechanism can reduce error rates in self assembly. We conclude with a brief discussion on some applications of activatable tiles beyond computational tiling, both as (i) a novel system for concentration of molecules, and (ii) a catalyst in sequentially triggered chemical reactions.

Keywords: DNA self-assembly, error correction, Tile Assembly Model, strand displacement, DNA polymerization, programmable molecular machines, polymerase chain reaction, deprotection systems, state transition, self-healing, kinetic trapping model, compact proofreading, continuous Markov Chains, sensing systems, concentration systems, reaction catalyzation.

Abbreviations: PCR: Polymerase Chain Reaction; DNA: DeoxyriboNucleic Acid, ds-DNA: double stranded DeoxyriboNucleic Acid; TAM: Tile Assembly Model; aTAM: abstract Tile Assembly Model; kTAM: kinetic Tile Assembly Model; aATAM: abstract Activatable Tile Assembly Model; kATAM: kinetic Activatable Tile Assembly Model; LTM: Layered Tile Mechanism; PTM: Protected Tile Mechanism

1 Introduction

The potential of self-assembling DNA nanostructures is derived from the predictable properties of DNA hybridization as well as from the theoretical power of the assembly to instantiate any computable pattern [36]. Winfree [37] formalized this process

of tiling assembly growth with his proposed Tile Assembly Model (TAM) which describes how a complex structure can spontaneously form from simple components called "tiles"; this assembly can also perform computation. However, the main problem for a practical implementation of TAM based assemblies is that tile additions are very error-prone.

Optimizing physical parameters like relative stoichiometry of tiles can improve fabricated patterns, but this is not sufficient to fully solve the problem of the growth of errors in computational tiling assemblies. Experimental demonstration of complex tilings show that error rates have ranged in 1-D from 2-5%, [22] and in 2-D from 10% in 2004, [29] 1.7% in 2007, [8] 0.13% in 2009, [2] 0.05% in 2010, [4] to 0.02% in replicating bits of information. [33]

The primary kind of error often encountered in DNA tile assembly experiments is known as the *error by insufficient attachment* [5], which occurs when a tile violates the TAM rule stating that a tile may only be added if it binds strongly[1] enough. Figure 2a) gives an example of tiling error by insufficient attachment. Thus there is a mismatch between theoretical models of DNA tiles and reality, providing major challenges in applying this model to real experiments.

Prior Work: There have been several designs of error-resilient tile sets [26, 40, 5] that perform "proofreading" on redundantly encoded information [40] to decrease assembly errors. While Winfree et al. [40] and Reif et al. [26, 30] addressed the problem of decreasing growth errors[2] in assembly, Chen et al. [5] addressed both growth and facet nucleation errors[3] by investigating errors by insufficient attachment. Schulman et al. [32] addressed the spontaneous nucleation error[4] with their *zig-zag tile set*. Each of these works, however, addresses only certain types of errors and proposes a construction that works with limited classes of tile sets. Additionally, most of the constructions result in a blow up the tile set size by a multiplicative factor, greatly hindering practical implementation. This leads to a major open question in error-resilient self-assembly: Is it possible to design a compact tile set that can address all three kinds of errors simultaneously? Our *activatable tile set* is an effort towards achieving this ultimate goal.

[1] In the TAM for temperature $\tau = 2$, a tile binds strongly either using at least one strong bond or two weak bonds.

[2] *Growth error* happens when a tile with one weak bond (weakly binding tile) attaches at a location where a tile with two weak bonds could have, and should have, been placed.

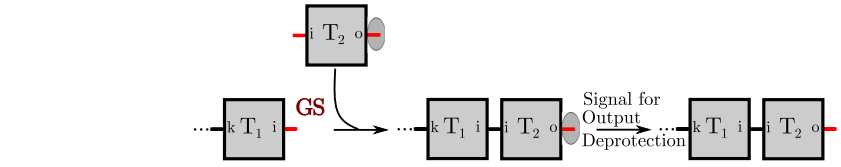
[3] *A facet nucleation error* happens when weakly binding tile attaches to a site where no tiles should attach at the moment.

[4] *Spontaneous nucleation errors* occur when a large assembly grows in absence of a seed tile.

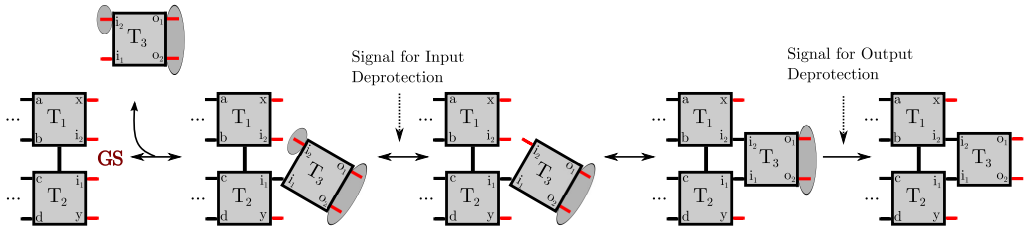
Limitations of Previous Approaches towards Robust Assembly: Existing error-resilient tile sets assume directional growth. This is a very strong assumption because experiments show that real tiles do not behave in such a fashion. The assumption, however, is key to assembly growth according to TAM. Thus, a potential solution to minimizing assembly errors is to enforce this directionality constraint. Observe that, if we start with a set of "deactivated" tiles which activate in a desired order, we can enforce a directional assembly at the same scale as the original one. Such a system can be built with minimal modifications on the rich repository of DNA nanostructures that we have designed and tested to date [20, 41].

Previous Approaches to Direct Tiling Assembly Procedures The main inspiration for the idea of *activatable tiles* has been snaked-proofreading technique of Chen et al. [5], which replaces each original tile by a $k \times k$ block of tiles. The assembly process for a block doubles back on itself such that nucleation error cannot propagate without locally forcing another insufficient attachment. Can such a growth order be enforced at the original scale of the assembly? Other motivating work has been from Dirks et al. [7], who designed a system where monomer DNA nanostructures, when mixed together, do not hybridize until an initiator strand is added. Can the idea of triggered self-assembly be used in the context of computational DNA tiling?

The answers to both questions are yes. The basic schemes in one and two dimensions are shown in Figure 1. The key idea is to 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. When the input sticky ends are completely hybridized, the output sticky ends are exposed. The newly exposed output sticky ends, in turn, strip the protective layer off the next tile along the growing face of the assembly. DNA polymerase enzyme can perform this deprotection, since it can act over long distances (e.g: across tile core) unlike strand displacement. The use of polymerase as a long range effector is justified because of its successful use in PCR, a biochemistry technique often used for exponentially amplifying DNA. Many repeated rounds of primer polymerization are required in conventional PCR. In contrast, we are using only a single round of primer polymerization (similar to a single round of PCR) to expose the desired sticky ends in our activatable tiles.



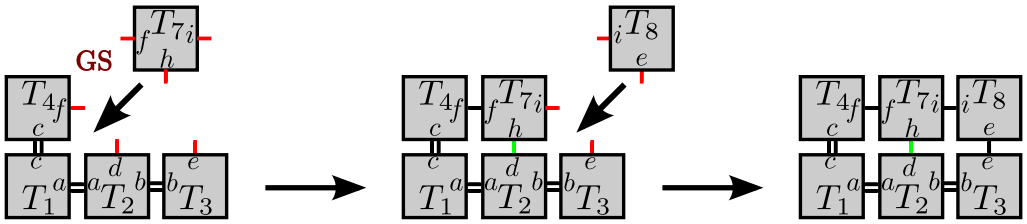
(a) Scheme of deprotection for 1D Assembly



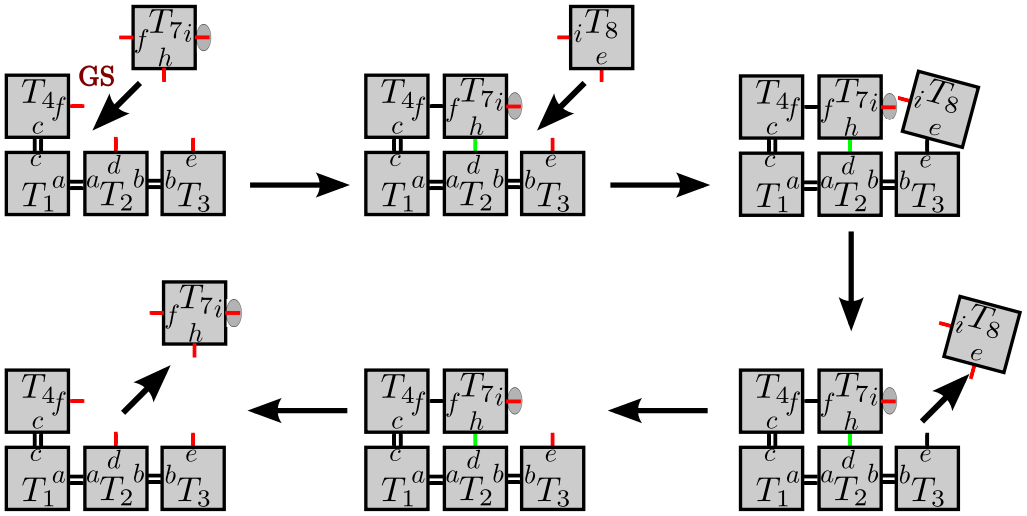
(b) Scheme of deprotection for 2D Assembly

Figure 1: GS denotes the growth site. Sticky pads in black denote hybridized, and red denote unhybridized. The oval padding over the input and output pads denotes protection from hybridization.

Enzyme-free Activated Tiles: Prior Protected and Layered Tile Mechanisms One of the most relevant previous works that is that of Murata [23] and Fujibayashi *et al.* [9, 10]: the Protected Tile Mechanism (PTM) and the Layered Tile Mechanism (LTM) which utilize DNA protecting molecules to form kinetic barriers against spurious assembly. In the PTM, a protection strand covers only the input strands of the base tile, while leaving the output strands open for attachment. Although this does induce a reduction in the error rates, spontaneous nucleation errors are still possible, where in the absence of a seed tile, active output strands can interact with inactive input strands, remove the protection strand, and initiate erroneous assemblies. The LTM is much more similar to our model, wherein a layered tile covers both the input and the output strands of the base tile. However, the two output strands are protected by 3 nt each, which is a weak protection mechanism and is highly sensitive to the temperature of assembly, constraining the temperatures at which assembly can be performed. A more recent improvement of the LTM [13], is a more robust model, where the output strands are protected by longer domains (10 nt), and the layered tile self-strand displaces its output protection from the base tile. However, since the two ends of the layered tile are complementary by design and can attach to each other, in addition to the constraint that a twist in the central double helix is required, experimental synthesis of such an enveloped tile could involve some effort.



(a) Insufficient attachment without Activatable Tiles



(b) This shows how the use of activatable tiles disallows a tiling error by insufficient attachment.

Figure 2: Errors of Insufficient Attachment are disallowed using Activatable tiles. GS denotes the growth site. Sticky pads in black denote hybridized, red denote unhybridized, and green denote mismatched edges. The oval padding over the input and output pads denotes protection from hybridization.

Active Tile Assembly Models : Our activatable tiles model can be considered to be a particular instance of the Signal Tile Assembly Model (STAM) [24, 25]. In the STAM, on successful binding of one or more input strands, signals transmit binding/unbinding instructions across tiles enabling/disabling the output sites. This model has been shown to be Turing Universal in 2D at temperature 1 [16], and has been the subject of detailed study [19, 14, 17]. In our activatable tiles model, on successful binding of input strands, only 1 instruction is transmitted across the tile, which instructs the output sites to become available for binding irreversibly. Other mechanisms of error control involve using staged assembly [6]. Our activatable tiles require a single enzyme, a strand displacing DNA polymerase, many of which (BST, ϕ29, klenow fragment) are known to work with high efficiency. Enzymatic [12] and

Enzyme-free [43] 1-D versions of activatable tiles have been experimentally shown to work, lending some credibility to the theoretical model of active tile self-assembly.

Our Results and the Organization of the Paper: Section 1 introduced the notion of deprotection and discussed the need for activatable tiles in computational assemblies. In Section 2, we provide abstract and kinetic models for activatable tiles that build on Winfree's original TAMs, with the primary difference being that each tile now has an associated finite state machine. We analyze potential sources of error in activatable TAM and compare both error rate and growth speed with that of the original TAM. In Section 3 we observe that since tiling assembly growth happens at the original scale of the assembly with low error rates, activatable tiles can provide compact error-resilience. In Section 3 we also prove that activatable tiles can provide compact self-healing by repairing a hole of certain size with high probability before backward assembly growth can start, assuming suitable values of kinetic parameters. In Section 5.1, we describe the DNA design of an example one dimensional activatable tile and its deprotection using strand displacement and DNA polymerization. In Section 5.2 we extend this design to the two dimensional case. In Section 6 we observe that the applications of activatable tiles are not limited to computational assemblies and discuss a novel concentration/sensing system based on activatable tiles. We also briefly describe how activatable tiles can be used for catalyzing chemical reactions. In Section 7 we conclude the paper with some open questions and future work.

2 The Activatable Tile Assembly Models

An abstract model is a theoretical abstraction from reality that is often easier to work with conceptually as well as mathematically. Thus developing an abstract activatable tile assembly model will help us describe the mechanism of tiling assembly growth with activatable tiles as well as analyze potential sources of error in the process. Since Winfree has already established the framework for tiling assembly models with his TAMs, we build our abstract Activatable Tile Assembly Model (aATAM) and the kinetic Activatable Tile Assembly Model (kATAM) discussed in this section on Winfree's abstract and kinetic TAMs respectively [37]. We use the term TAM to describe any models that fall under the broad umbrella of the Tile Assembly Model, and define our aATAM and kATAM under this umbrella as well.

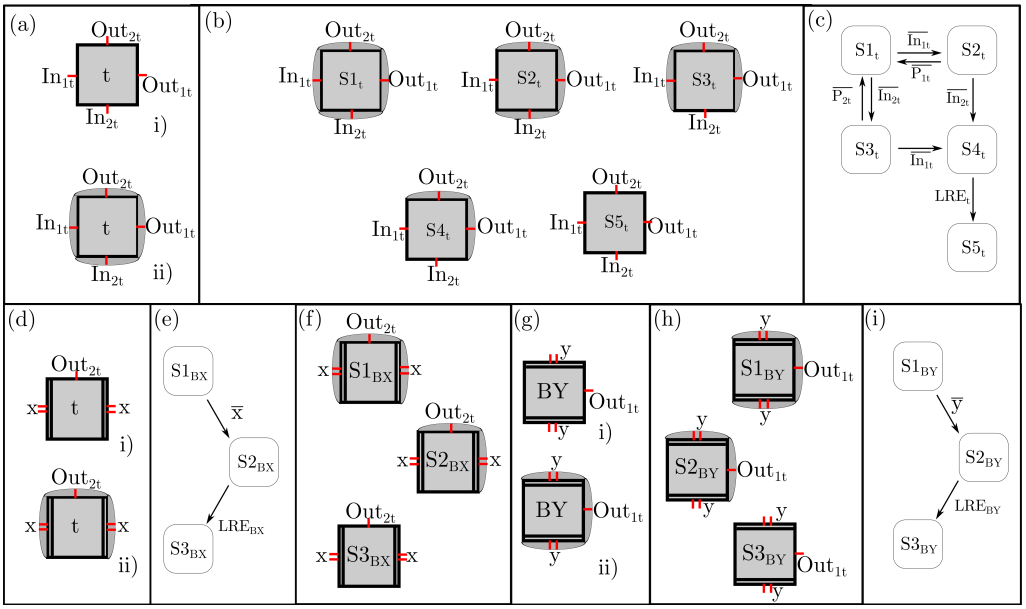


Figure 3: (a-i) Original Abstract Rule Tile, (a-ii) Protected Abstract Tile, (b) Different states associated with the activatable rule tile, (c) State Transition Diagram for Activatable Rule Tile, (d-i)Original Abstract Boundary Tile along x axis, (d-ii) Protected Abstract Boundary Tile along x-axis, (e)Different states associated with the Activatable Boundary Tile along x-axis, (f) State Transition Diagram for Activatable Boundary Tile along x-axis, (g-i)Original Abstract Boundary Tile along y axis, (g-ii)Protected Abstract Boundary Tile along y-axis, (h) Different states associated with the Activatable Boundary Tile along y-axis, (i) State Transition Diagram for Activatable Boundary Tile along y-axis. The oval padding on the sides of certain tiles in the Figure is used to denote protection of tile pads from hybridization. Sides of tiles without labels indicate absence of sticky ends.

2.1 The abstract Activatable Tile Assembly Model (aATAM)

In the simplest version of activatable tiles, the idea is to start with a set of "protected" *rule tiles* so that the tiles do not assemble until the pre-assembled initiator assembly consisting of a *seed tile* and multiple *boundary tiles* is introduced in the mixture. In the more complex version, the initiator is the seed tile alone and the boundary tiles have a protection-deprotection scheme similar to that of the rule tiles. Note that the seed tile does not have any protection, and its pads are free to start binding to input pads of other boundary and rule tiles. The boundary and rule tiles each have protection on their input/output pads.

The aATAM is similar to the original abstract TAM (aTAM) due to Winfree

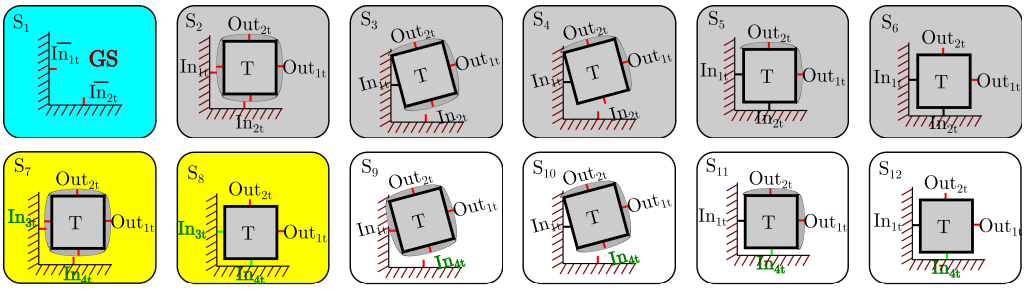


Figure 4: The different states associated with the transition diagram for the kATAM. State S_1 (highlighted light-blue) shows the growth site GS. The states highlighted in grey (S_2,S_3,S_4,S_5,S_6 show possible 2-input matches on the GS), non-highlighted (S_9,S_{10},S_{11}, S_{12} show possible 1-input matches on the GS), yellow (S_7,S_8 show 0-input matches on the GS). The complementary strands on the GS $I\bar{n}_{1t}$ and $I\bar{n}_{2t}$ are only shown in state S_1 for ease of display. The mismatched pad names (In_{3t} and In_{4t}) are shown in dark-green.

[37] except that each tile type t has an associated finite state machine (FSM) M_t and hence, each tile has a state. Refer to Figure 3 for the rest of this discussion. The new abstract rule tile is shown in Figure 3(a-ii). Unlike the original tile [Figure 3(a-i)], it has all of its sides protected. The states in the FSM M_t arise from the presence or absence of protection on the four sides of the tile type t (as shown in Figure 3(b)). The state transition diagram is shown in Figure 3(c). The idea is to start with a completely protected rule tile and at the end have a tile similar to one described in Winfree's aTAM once its input ends are properly bound to appropriate "neighbors" in a growing assembly. A tile of type t is inert in state $S1_t$ until it is activated on either one of its input pads (See states $S2_t$ and $S3_t$, with 1 pad bound). The bound input pad in turn activates the remaining input which can bind to an appropriate adjacent tile on the growing face of the crystal ($S4_t$). (Note that although the theoretical model shown in Figure 3(c) and Figure 5 describes either of the input pads can attach first, the experimental design in Figure 17 and the analysis in Section 2.3, are designed such that exactly one of the input pads can bind first, which in turn activates the other input. Alternate designs (not-shown) can be designed such that either input binds first (for e.g. using the co-operative strand displacement mechanism [42].)

In case there is no such neighbor available, the protection ($P1_t$ or $P2_t$), which is part of the tile until the outputs are deprotected, covers the inputs again and the tile leaves the assembly ($S1_t$). With at most one of the input pads bound, (recall that outputs are not available for binding until both input pads are matched) there can be at most one weak bond between the tile and the assembly. A tile in aATAM abides

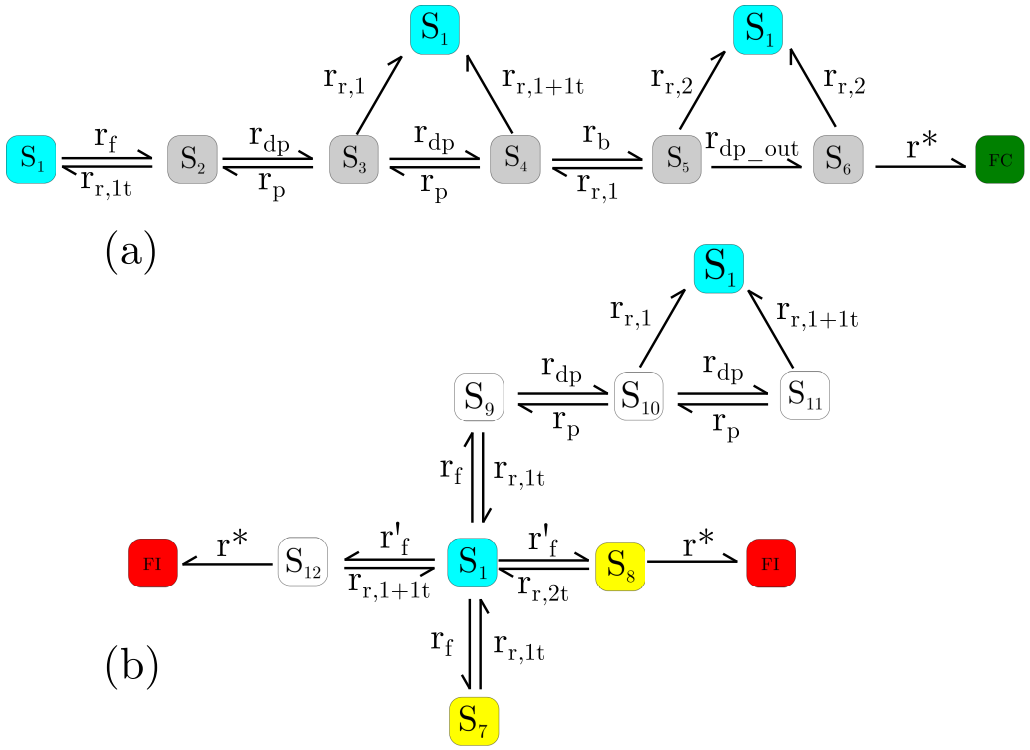


Figure 5: Continuous time Markov Chain associated with the kATAM (Specifically the Kinetic Trapping Model) (a) Shows the transition diagram for a growth site GS to have a tile frozen correctly at the site (b) Shows the transition diagram for a growth site GS to have a tile frozen incorrectly at the site. The colour code corresponds to the number of matches (Figure 4).

by the temperature $\tau = 2$ rule just as in aTAM and hence this tile dissociates. When both inputs are matched, the long range effector (LRE) deprotects the output pads ($S5_t$).[5]

[5]In the more complex version of activatable tile set, the seed is the initiator tile in the assembly experiments and consequently does not have any protection on any of its sides. Boundary tiles still need to be protected [Figure 3(d-ii) and 3(g-ii)]. The corresponding states are shown in Figures 3(e) and 3(h). Deprotection is simpler for boundary tiles. Since the bonds on the east side of the x-axis boundary tiles (B_x) and that on the south side of the y-axis boundary tiles (B_y) are strong bonds, matches on those pads can trigger the deprotection of the north and west pads for B_x as well as B_y [Figures 3(f) and 3(i)].

2.2 The kinetic Activatable Tile Assembly Model (kATAM)

The kATAM is based on Winfree's original model kTAM, but due to the the stochastic nature of the protection on all sides of the tile, additional errors need to be modeled. Therefore we need more free parameters than just r_f and $r_{r,b}$ for tiling assembly growth. Figure 4 shows the different states possible in the finite state machine for the kATAM and Figure 5 shows the state transition diagram. In addition to the assumptions of kTAM, the main assumptions of kATAM are: (i) The input protection is only reversible while the output pads are still protected, (ii) Output protection is irreversible, meaning once a tile is completely deprotected, it cannot return to the stage where every side of the tile has a protective cover. Monomers in solution are, thus, either entirely protected or entirely deprotected, iii) Universal toeholds are used, for ease of analysis and implementation. Universal toeholds implies that the toehold sequence is the same across all the input/output pads, even though the branch migration sequences might be different, iv) The system we model is the sierpinski 7 tiles system, as modelled by Winfree in the original kTAM.

We start with an empty growth site(S1). Completely protected tiles can be added to it at a rate r_f, proportional to their concentration (recall G_{mc} is the logarithm of the concentration). This event corresponds to state S7, S9 and S2, depending on whether the tile has 0, 1 or 2 input matches at its growth site. In kATAM, tiles binding at the growth site come in another flavor too: They may be completely deprotected (i.e. as the tiles in the original kTAM). The reason for modeling these deprotected tiles is that even a tile with both inputs correctly matched can be knocked off the growth site after output deprotection. These tiles, however, are added to the growth site, at a different rate $r_f^{'}$ that will be shown later to be much smaller than r_f. This is the transition probability to states S8, S12 and S6 from S1. Later in this section we will discuss how we can derive $r_f^{'}$ from the free parameters. Further, tiles in states S7, S9 and S2 fall off at a rate $r_{r,1t}$ since they are only bound by one input toehold (toehold of In_{1t}) to the assembly. The rate of dissociation $r_{r,b}$ from states S8, S12 or S6 depends only on the extent of input matches (just as in original kTAM) and hence are $r_{r,2t}$, $r_{r,1+1t}$ [6] and $r_{r,2}$ respectively. With one input match, the tile in S9 (S2) transitions to S10 (S3) at the rate of r_{dp} (deprotection) and returns to S9 (S2) at the rate of r_p (protection). This tile state corresponds with monomers with one deprotected input and one protected input. After the first match, the input

[6] Dissociation from state S8 involves unbinding of 2 toeholds, and state S12 involves unbinding of 1 bound input pad and 1 toehold, since the toeholds have been assumed to be universal.

deprotects the adjacent input at the rate of r_{dp}. If the second input is also matched, then it binds (very fast) at a binding rate r_b. If, however, there is a mismatch for the second input, either the protective cover[7] falls back on the input at the rate r_p (S11 $\rightarrow$ S10) or the tiles come off the growth site at the rates depending on the extent of binding. Note that r_{dp}, r_p, r_b are free parameters whose value depends on the experimental situation.

When both inputs are matched, the output pads(S5) are deprotected at the rate r_{dp_out}. Just as with r_{dp} and r_p, r_{dp_out} is a free parameter that depends on the experimental situation. Tiles can, however, fall off from the growth site, while in any state at a rate that depends on the extent of binding. Tiles with more than two bindings (three or four) can fall off the growth site, too, but at a considerably lower rate of $r_{r,3}$ or $r_{r,4}$. Thus we do not show these transitions in Figure 4.

2.3 Kinetic Parameters for an Example Deprotection System

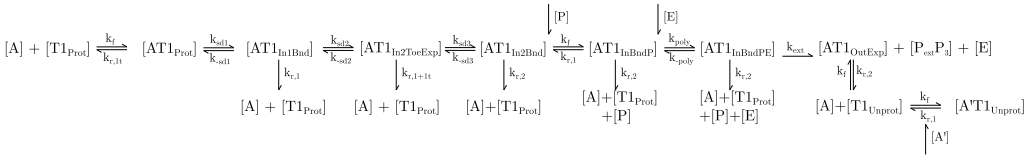


Figure 6: Kinetic Parameters for an example deprotection system.

The chain of equations in Figure 6 shows how to derive the various parameters for a certain implementation of deprotection using a DNA polymerase enzyme (modeled as an irreversible process) and strand displacement technique (modeled as a reversible process).[8] Consider the growth site GS in an assembly A and suppose tile $T1$ can bind at GS. $T1_{Prot}$, the protected version of $T1$, arrives at GS at a rate r_f (corresponding rate constant k_f). Since all its pads are protected, and only one input toehold is available for binding, it can leave the growth site at a rate $r_{r,1t}$ (corresponding rate constant $k_{r,1t}$). Without loss of generality, suppose the south end input (e.g. input 1) binds, and triggers a signal making the east end input (input 2) available for binding and, if this hybridization is successful, the outputs become uncovered. Input 1 has an exposed toehold, facilitating displacement of the protection strand P_1. Input 1 of

[7]The design of these tiles are such that the protection is part of the nanostructure until the outputs are deprotected.

[8] For the subsequent discussion, note that the backward rate is denoted by the negation of forward rate. For instance, if k_{poly} be the forward rate of association of the DNA polymerase enzyme to the primer, k_{-poly} denotes the rate of dissociation of the polymerase enzyme.

$T1_{Prot}$ hybridizes with A via strand displacement of P_1, to form $AT1_{In1Bnd}$ (forward and backward rate constants are k_{sd1} and k_{-sd1} respectively). P_1 is now free to displace P_2, the protection of input 2, using a toehold region forming $AT1_{In2ToeExp}$ (forward and backward rate constants are k_{sd2} and k_{-sd2} respectively). Once input 2 is exposed by strand displacement, it hybridizes with A, forming $AT1_{In2Bnd}$ (forward and backward rate constants are k_{sd3} and k_{-sd3} respectively); the complement $P^{'}$ of the primer P, which was held in a hairpin loop on the output protection strand P_3, is now made available. The primer P binds to P_3, forming $AT1_{InBnd}P$ at the rate r_f (It can dehybridize at the rate $r_{r,1}$); [9] DNA polymerase enzyme E next binds at the $3^{'}$ end of P forming $AT1_{InBnd}PE$ (forward and backward rate constants are k_{poly} and k_{-poly} respectively) and extends it to the output ends with a rate constant k_{ext}.[10] In the next step when $P_{ext}P_3$ dissociates, the outputs of tile $T1$ in the assembly $AT1_{OutExp}$ are exposed. Completely deprotected tiles ($T1_{UnProt}$) can fall off GS at a rate $r_{b,2}$ dictated by their concentration (rate constant $k_{b,2}$; *these tiles are the only source of errors in assembly.* They can cause an error by attaching to a growth site in another assembly $A^{'}$ with a single match ($A^{'}T1_{UnProt}$).

We derive the rates of the different reactions.[11] k_{sd1}, k_{sd2} and k_{sd3} are the forward rate constants of strand displacement while k_{-sd1}, k_{-sd2} and k_{-sd3} are the corresponding backward rate constants. In general,

$$\frac{k_{sdi}}{k_{-sdi}} = e^{\Delta G/kT} \tag{2.3.1}$$

for $i = 1, 2, 3$, where ΔG is the free energy change in the duplex formation for the toehold region and is calculated in a similar manner as G_{se}. Further, the average time taken per base-pair migration has been estimated from empirical data [44, 35] to be of the order of $10 - 100$ μsecs. Thus k_{sdi} and k_{-sdi}, for $i = 1, 2, 3$ can be estimated

[9] Note that if the concentration of a growing assembly front at any instant is [GS], and the primer concentration is [P], then the reaction rate is $k_f[GS][P]$. Likewise, the rate of monomer tile attachment is $k_f[GS][T]$, where [T] is the tile concentration. By keeping the concentration of primer and tiles the same, the reaction rate r_f can be assumed to be the same for both these reactions. Also, the dehybridization rate constant $r_{r,1}$ is kept the same by keeping the primer length the same as the sticky ends. Since a primary assumption of the kTAM is that monomer tile concentrations are kept constant, similarly primer concentrations are also kept constant.

[10] The polymerase extension of the primer hybridized to the protection strand, is modeled as an irreversible atomic process for simplicity. If the exonuclease activity during polymerization occurs at a reasonably low rate, then the assumption is quite justified.

[11] The rate constants k_{sd1} and k_{sd3} each correspond to the reaction rate r_{dp}, while the rate constants k_{-sd1} and k_{-sd3} each correspond to r_p. The rate constants k_{poly} and k_{ext} together contribute to the rate r_{dp_out}. The rate constant k_{sd2} is involved in the deprotection of the second input pad by the first activated input pad, and has not been modelled in Figure 5.

from the corresponding toehold length.

Once at least one of the inputs are hybridized, the tile can dissociate with a rate constant $k_{r,i}$, where $i = 1, 2$ depending on how many inputs are currently bound. Recall that, at this stage, we assume that $T1$ returns to its original protected state ($T1_{Prot}$) when it falls off the assembly. This corresponds to rate

$$r_{r,i} = k_f e^{-iG_{se}} \tag{2.3.2}$$

The primer P hybridizes at a rate

$$r_f = k_f[P] = k_f e^{-G_{mc}} \tag{2.3.3}$$

These equations are as modelled in the kTAM [38]. We assume that the polymerase enzyme initially binds with a rate constant of k_{poly} and dissociates with k_{-poly}, but the subsequent polymerization after binding is irreversible. The polymerization once begun occurs with a rate constant of k_{ext}. In general, k_{poly} and k_{-poly} are experimentally obtained depending on the polymerase. However, if we do not have that data at our disposal, it can also be derived analytically. Since mathematical treatment of pre-steady state kinetics is quite difficult when exonuclease activity is included, eukaryotic DNA polymerization is often studied in only steady state. In general, let k_{cat} and k_{exo} be the catalytic rate of DNA polymerization and exonuclease reactions, k_1 and k_{-1} represent the association and dissociation rates, respectively of nucleotide binding and n be the number of consecutive nucleotide incorporation allowed, then using steady state kinetic analysis [31], the concentration of the tiles whose both outputs are exposed due to the polymerization of the protection strand ($[AT1_{OutExp}]$) can be evaluated given the concentration of the tiles with primers bound to P_3, ($[AT1_{InBnd}P]$). Specifically,

$$k_{ext} = \frac{1}{1 + (1 + \frac{K_m}{N})\Sigma_{i=0}^{n-1}(\frac{k_{exo}}{k_{cat}})^{n-i}(\frac{K_m}{N})^{n-i-1}} \tag{2.3.4}$$

Here N is the nucleotide concentration and $K_m = \frac{k_{-1}+k_{cat}}{k_1}$. The derivation of k_{ext} is as follows: if D_i and $D_i^{'}$ represent the concentration of the polymerized primer, i bases long and the complex polymerized i-mer with the next nucleotide to be incorporated bound in its position but not yet catalyzed respectively (both are complexes with DNA polymerase), then

$$\frac{dD_i}{dt} = k_{cat}D_{i-1}^{'} + k_{-1}D_i^{'} + k_{exo}D_{i+1} - k_{exo}D_i - k_1 D_i N \tag{2.3.5}$$

and

$$\frac{dD_i^{'}}{dt} = k_1 D_i N - (k_{-1} + k_{cat})D_i^{'} \tag{2.3.6}$$

for $i = 0, \ldots, n-1$ and $\frac{dD_n}{dt} = k_{cat}D_{n-1}^{'} - k_{exo}D_n$. The last equation ensures that the polymerization stops at the end of n bases. Further $D_{-1} = 0$. Hence solving for these equations in steady state where each $\frac{dD_i}{dt}$ and $\frac{dD_i^{'}}{dt}$ is zero, we can obtain the equation for k_{ext}.

2.4 The Kinetic Trapping Model

In the context of the abstract tile system, the kinetic trapping model monitors a particular growth site. As tiles attach to the neighboring growth sites, the tile currently in the monitored growth site "freezes" there permanently at the effective growth rate r^* (even if it has one or more mismatches among its four binding sides). The kinetic trap model can be used to find the probability that the correct tile is in the growth site when the site freezes. In addition to the states described in Figure 4, the model has the sink states *Frozen Correct(FC)* and *Frozen Incorrect(FI)*[12] [Figure 5]. In this model, the probability of an error-less step in the assembly is the probability of a tile transitioning to FC at $t \to \infty$. We compare the growth speed and the error rate with that of the original Winfree model.

Since there are many free parameters in the kinetic model, such as $r_f, r_{r,b}, r_p$ etc we decrease the dimensionality of the parameter space by combining some of the parameters together e.g. r_p, r_{dp} and r_{dp_out}. This is done by computing the rate at which tiles become completely deprotected after reaching a growth site, thus neglecting the intermediate states in Figure 5. This corresponds to the rate at which a tile reaches state $S6$ if it is in $S1$. We call this rate r_{eff} and assume that r_{eff} is a function of G_{se} such that

$$r_{eff} = k_f e^{(-2+\epsilon_1)G_{se}} \tag{2.4.1}$$

where ϵ_1 is a constant and $0 < \epsilon_1 < 1$. Note that r_{eff} is similar to r_f in the original kTAM. Based on the continuous time Markov Chain (CTMC) in Figure 5, we can

[12] A growth site can only be frozen if the output pads of the tile sitting in that growth site are available for binding. Hence the transitions to FC and FI are only from S6, S8 and S12 and not from S5 or S11.

evaluate r_{eff} as

$$r_{eff} = r_f \frac{r_{dp}}{(r_{dp} + r_{r,1t})} \frac{r_{dp}}{(r_p + r_{dp} + r_{r,1})} \frac{r_b}{(r_b + r_{r,1+1t} + r_p)} \frac{r_{dp\_out}}{(r_{dp\_out} + r_{r,2} + r_{r,1})} \tag{2.4.2}$$

This formulation is derived as follows: the rate of moving from $S1$ to $S3$ is $r_f \frac{r_{dp}}{(r_{dp}+r_{r,1t})}$. Similarly, the rate at which tiles transition from $S3$ to $S4$, given the rate of transfer from $S1$ to $S3$ is $r_f \frac{r_{dp}}{(r_{dp}+r_{r,1t})} \frac{r_{dp}}{(r_p+r_{dp}+r_{r,1})}$. Similarly one can obtain the rate of transitioning from $S1$ to $S6$.

Here, r_b is the rate of binding of two unbound complementary sticky ends in close proximity (See transition from state $S4$ to $S5$). We assume that $r_b \gg (r_{r_{1+1t}} + r_p)$ and thus $\frac{r_b}{(r_b+r_{r,1+1t}+r_p)} \approx 1$. Hence, equation 2.4.2 is now

$$r_{eff} = r_f \frac{r_{dp}}{(r_{dp} + r_{r,1t})} \frac{r_{dp}}{(r_p + r_{dp} + r_{r,1})} \frac{r_{dp\_out}}{(r_{dp\_out} + r_{r,2} + r_{r,1})} \tag{2.4.3}$$

One primary assumption in kATAM is that

$$r_{r,1} > r_f > r_{eff} > r_{r,2} \tag{2.4.4}$$

Further, we assume

$$\begin{aligned}
r_{r,1t} &= k_f e^{-\frac{t}{s} G_{se}} && (2.4.5)\\
r_{r,2t} &= k_f e^{-\frac{2t}{s} G_{se}} && (2.4.6)\\
r_{r,1} &= k_f e^{-G_{se}} && (2.4.7)\\
r_{r,1+1t} &= k_f e^{-(1+\frac{t}{s}) G_{se}} && (2.4.8)\\
r_{r,2} &= k_f e^{-2G_{se}} && (2.4.9)\\
r_{eff} &= k_f e^{(-2+\epsilon_1) G_{se}} && (2.4.10)\\
r_f &= k_f e^{(-2+\epsilon_1+\epsilon_2) G_{se}} && (2.4.11)
\end{aligned}$$

for

$$\begin{aligned}
0 &< \epsilon_1, \epsilon_2 && (2.4.12)\\
\epsilon_1 + \epsilon_2 &< 1 && (2.4.13)
\end{aligned}$$

Here, t is the length of the toehold, and s is the length of the sticky ends. Observe that $r_f = k_f e^{-G_{mc}}$ as well.

These assumptions follow from the following set of intended interactions. Since

our system is being modelled at $\tau = 2$, a two sticky end bond detachment must be much slower. Hence, $\frac{r_f}{r_{r,2}} > 1$. Likewise, $\frac{r_{eff}}{r_{r,2}} > 1$. This implies $e^{(\epsilon_1+\epsilon_2)G_{se}} > 1$ and $e^{\epsilon_1 G_{se}} > 1$. On the other hand, a single sticky end bond detachment should be faster than the forward rate of tile attachment, otherwise aggregates can build up, and errors increase. Hence, $\frac{r_f}{r_{r,1}} < 1$. Similarly, $\frac{r_{eff}}{r_{r,1}} < 1$. This implies $e^{-(1-(\epsilon_1+\epsilon_2))G_{se}} \ll 1$ and $e^{-(1-\epsilon_1)G_{se}} \ll 1$. Combining these results in the constraints above.

For simplicity of the model, we can ensure that

$$r_{eff} \gg r_{r,2} \tag{2.4.14}$$

This implies the value of ϵ_1 is close to 1, which can be done by adjusting the kinetic parameters in the deprotection system (e.g. toehold length in the strand displacement events, nucleotide concentration and template length for polymerization etc). Another important assumption we make is that DNA polymerization has been modeled as irreversible and, hence, at equilibrium every tile is completely deprotected. Based on these assumptions we conclude the following claim:

Claim With respect to the original kTAM, the error rate in kATAM can be decreased without a considerable decrease in the speed of the growth of the assembly since $\frac{\epsilon}{\epsilon_{old}} = e^{-\epsilon_1 G_{se}}$ and $\frac{r^*}{r^*_{old}} > e^{-\epsilon_2 G_{se}}$.

Proof In order to prove this claim, we first need to estimate $r_f^{'}$, the rate at which completely deprotected tiles bind to a growth site. Observe that in Figure 5, there is a transition from state $S6$ to state $S1$ when the tile after output deprotection leaves the growth site. The state of the tile however, has changed as indicated in Figure 6 (from $T1_{Prot}$ to $T1_{UnProt}$). Thus, the transition from $S5$ to $S6$ is irreversible. This implies that at $t \to \infty$ (steady state), all protected monomers will pass through state $S6$. There is another leak pathway through which tiles can get deprotected, but we assume that this pathway has extremely low probability and hence do not include it in our analysis [13].

Hence $e^{-G_{mc}}$ is an upper bound on the fraction of tiles in state $S5$ at $t \to \infty$. [14]

[13] Another way of deprotected tiles forming is if a protected monomer tile and primer bind with a small chance, and the polymerase extends the primer deprotecting the tile. An explanation of this is given in a following note 2.4.1.

[14] This is a loose upper bound because at $t \to \infty$, most growth sites are frozen and the tiles in those growth sites cannot leave and contribute to the expected number of tiles leaving $S6$. The concentration of completely deprotected tiles, however, is maximum at steady state.

Based on the CTMC in Figure 5, the probability of a tile leaving state $S6$ is

$$p_{out\_S5} = \frac{r_{r,2}}{r^* + r_{r,2}} \tag{2.4.15}$$

where r^*, the effective growth rate is given by

$$r^* = r_{eff} + r_f^{'} - r_{r,2} \tag{2.4.16}$$

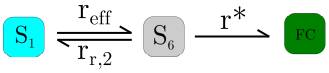


Figure 7: r^* derivation based on a correct tile attaching into place from state S1 to S6.

Note that r^* is the effective rate at which the tile gets locked into place by a correct neighbouring tile attaching to the assembly. We assume that the net rate at which a neighbouring tile attaches to lock the tile into place is independent of whether a tile has 0,1,2 bonds at the current growth site. Hence r^* is the same from states $S6$, $S8$ and $S12$. Also see Figure 7 to see how r^* is defined in terms of equation 2.4.16.

Also, the expected portion of completely deprotected tiles T_{deprot} that leaves $S6$ is

$$E[T_{deprot}] = \frac{r_{r,2}}{r^* + r_{r,2}} e^{-G_{mc}} \tag{2.4.17}$$

by linearity of expectation.

Recall that the rate of tile addition is solely dependent on the concentration of the tiles. Hence the rate at which a completely deprotected tile binds to a growth site is

$$r_f^{'} = k_f \frac{r_{r,2}}{r^* + r_{r,2}} e^{-G_{mc}} \tag{2.4.18}$$

Observe that we have

$$\begin{aligned} r_f^{'} &= r_f \frac{r_{r,2}}{r_{eff} + r_f^{'}} \\ &< r_f \frac{r_{r,2}}{r_{eff}} \end{aligned} \tag{2.4.19}$$

since $r_f^{'} > 0$. For simplicity, we will use $r_f \frac{r_{r,2}}{r_{eff}}$ as an estimate of $r_f^{'}$ since the bound

can be made tighter using the assumption in equation 2.4.14. [15] Thus

$$r_f^{'} = e^{(-2+\epsilon_2)G_{se}} \tag{2.4.20}$$

We now analyze the probability of an error-less assembly step based on the kinetic trapping model in Figure 5. Let $p_i(t)$ be the probability that i is the state t seconds after the growth site has appeared, assuming the site has not yet been frozen. Thus if we start with unit concentration of tile in $S1$, it accumulates differentially in FC and FI. Based on the continuous time Markov Chain (CTMC) in Figure 5, we compute the probability of an error-less step for an assembly as

$$\begin{aligned} p_{FC}(\infty) &= \frac{\frac{r_{eff}+r_f^{'}}{r^*+r_{r,2}}}{\frac{r_{eff}+r_f^{'}}{r^*+r_{r,2}} + \frac{2r_f^{'}}{r^*+r_{r,1}} + \frac{4r_f^{'}}{r^*+r_{r,1t}}} \\ &\sim \frac{1}{1+2\big(\frac{r_f^{'}}{r_f^{'}+r_{eff}}\big)\big(\frac{r^*+r_{r,2}}{r^*+r_{r,1}}\big)} \end{aligned} \tag{2.4.21}$$

since (i) there is only one correct tile for any growth site (this tile ends up in FC), two tiles with one binding site match and the remaining four tile types have no matching binding sites (these tiles end up in FI) and (ii) $r_{r,1} > r_f > r_{eff} > r_{r,2}$. Hence error rate is

$$\begin{aligned} \epsilon &= 1 - p_{FC}(\infty) \\ &\sim 2\big(\frac{r_f^{'}}{r_f^{'}+r_{eff}}\big)\big(\frac{r^*+r_{r,2}}{r^*+r_{r,1}}\big) \end{aligned} \tag{2.4.22}$$

for small ϵ. Simplifying,

$$\epsilon = 2\frac{e^{-(\epsilon_1-\epsilon_2)G_{se}}}{1+e^{-(\epsilon_1-\epsilon_2)G_{se}}} \frac{e^{-(1-\epsilon_1)G_{se}}(1+e^{-(\epsilon_1-\epsilon_2)G_{se}})}{1+e^{-(1-\epsilon_1)G_{se}}+e^{-(1-\epsilon_2)G_{se}}-e^{-G_{se}}} \sim 2e^{-(1-\epsilon_2)G_{se}} \tag{2.4.23}$$

neglecting $e^{-G_{se}}(e^{\epsilon_1 G_{se}} + e^{\epsilon_2 G_{se}} - 1)$. Recall that error in Winfree's model [38] is

$$\epsilon_{old} = 2e^{G_{mc}-G_{se}} = 2e^{-(1-\epsilon_1-\epsilon_2)G_{se}} \tag{2.4.24}$$

[15] Out of r_{eff} tiles reaching state $S6$ in unit time, only $r_{r,2}$ leave. Only those tiles which leave the growth site after complete deprotection can come back to any growth site with a rate constant of k_f. Hence if $r_{eff} \gg r_{r,2}$ then $r_f^{'}$ should be much less than r_{eff} and we can safely neglect the contribution from $r_f^{'}$ in the denominator $r^* + r_{r,2}$ while computing the value of $r_f^{'}$.

Thus,

$$\frac{\epsilon}{\epsilon_{old}} = e^{-\epsilon_1 G_{se}} \tag{2.4.25}$$

The growth speed in kATAM,

$$r^* = k_f e^{(-2+\epsilon_1)G_{se}}(1 + e^{-(\epsilon_1-\epsilon_2)G_{se}} - e^{-\epsilon_1 G_{se}}) \tag{2.4.26}$$

and growth speed in kTAM,

$$r^*_{old} = r_f - r_{r,2} = k_f e^{(-2+\epsilon_1+\epsilon_2)G_{se}}(1 - e^{-(\epsilon_1+\epsilon_2)G_{se}}) \tag{2.4.27}$$

Thus we have

$$\frac{r^*}{r^*_{old}} = \frac{e^{-\epsilon_2 G_{se}}(1 + e^{-(\epsilon_1-\epsilon_2)G_{se}} - e^{-\epsilon_1 G_{se}})}{(1 - e^{-(\epsilon_1+\epsilon_2)G_{se}})} > e^{-\epsilon_2 G_{se}} \tag{2.4.28}$$

We now have a reduction in the error rate as well as a reduction in the growth rate of the assembly. From equations 2.4.25 and 2.4.28, since ϵ_1 and ϵ_2 are controllable parameters, we conclude that the assembly error rate can be decreased with a trade-off in the growth speed. Note that there is a lot of slack in our analysis. Further, since there are multiple free parameters in addition to G_{mc} and G_{se} in kATAM, the exact correlation between error rate and growth speed is still an open question.

2.4.1 Deprotected Tiles forming due to leak pathway between Monomer Tiles and Primer interaction

There is a low probability with which a primer P and protected monomer tile (T_{Prot}) (Figure 13) can bind to each other, and the polymerase extends the primer (Figure 14) resulting in a deprotected tile. Here we analyze the average concentration of unprotected tiles due to the leak mechanism. In the original kTAM, the concentration of a protected monomer tile per growth site per aggregate, is kept constant. This concentration of monomer tiles is given by G_{mc}, thus $[T_{Prot}] = e^{-G_{mc}}$. In our assumption in the kinetic trapping model, the primer concentration per growth site per aggregate is kept constant (see footnote 9), and equal to the T_{Prot} concentration. Thus $[P] = e^{-G_{mc}}$. Now, lets say that the leak reaction is modelled by $\mathrm{P}+\mathrm{T}_{\mathrm{Prot}} \xrightarrow{k_{leak}} \mathrm{T}_{\mathrm{Unprot}}+\mathrm{W}$, where W is the waste product, then $\frac{d[T_{Unprot}]}{dt} = k_{leak} \times [P] \times [T_{Prot}] = k_{leak} e^{-2G_{mc}}$.

Now, this is the average leak rate of production of deprotected tiles per growth site per aggregate. The total number of deprotected tiles at any point of time, would

involve integrating the total number of growth sites across all fractional aggregates at time t, and multiplying this by the leak rate above to get an estimate of the concentration of deprotected tiles. This concentration would then be subtracted by the expected concentration of deprotected tiles that bound to some growth site at some aggregate. Since this analysis is a lot more complex, a simplifying assumption has been made, that the leak pathway is extremely slow, and does not contribute to the concentration of deprotected tiles.

3 Compact Proofreading with Activatable Tiles

Activatable tiles provide error-resilience to a growing assembly by enforcing directional growth. Ideally the output ends are never available until the corresponding input ends are completely hybridized, thus preventing both errors by insufficient attachment as well as nucleation errors. There is a small probability, however, of errors by insufficient attachment caused by tiles that leave a growth site after output deprotection. Furthermore, the computation still occurs at the original scale, unlike Chen's snaked proofreading technique [5] which increases the lattice size by a multiplicative factor of k^2. Hence, activatable tiles indeed provide compact error-resilience. Since the seed is the only completely unprotected tile when the assembly begins and the concentration of completely unprotected rule or boundary tiles existing in solution at any given time is very low, activatable tiles can also prevent spontaneous nucleation and enforce "controlled growth".[16]

We can formally prove that activatable tiles are indeed an instance of compact proofreading. Soloveichik et al. [34] gave a concise definition of compact proofreading and we adapt it to our ATAM:

Definition 3.0.1 *Given a small constant $0 < q < 1$, a sequence of deterministic tile systems $\{T_1, T_2, T_3, \ldots\}$ is a compact proofreading scheme for pattern P if (i) T_N produces the full infinite pattern P under the aATAM, (ii) T_N has poly$(\log N)$ tile types (poly(n) denotes $n^{O(1)}$) and (iii) T_N produces the correct $N \times N$ initial portion of the pattern P with probability at least q in time $O(Npoly(logN))$ in the kATAM for some value of the free parameters in the model.*

Theorem 3.0.1 *An activatable Tile System A_N is a compact proofreading scheme.*

[16] Controlled growth is defined to be the growth occurring for parameter values in a certain part of the kinetic parameter space, such that (i) growth does occur, (ii) errors are rare and (iii) growth not seeded by the seed tile is rare [38].

Proof Let the tile system in aTAM be T_N and the activatable tile system be A_N. A_N is same as T_N except that each tile type has an associated finite state automata. Since in aATAM activatable tiles can bind to a growth site only if they can bind strongly enough (just as in aTAM), A_N can produce the whole system correctly under aATAM so the first condition is satisfied.

Moreover, $|A_N| = |T_N|$, the only difference being that we start the assembly with "protected" version of T_N. Since this work is concerned with only deterministic tile systems, the argument of Soloveichik et al. [34] applies and we need only constant number of tile types so long the tile set has a locally deterministic assembly sequence.

The argument for the third condition is similar to that of Chen et al. [5]. In this model, errors are only caused by insufficient attachments; these errors are caused by tiles dissociating from growth sites after their output protection has been stripped off. In an insufficient attachment event, first an unprotected monomer (with a single binding site match) attaches at the rate of $r_f^{'}$. However, before this tile is knocked off at the rate of $r_{r,1}$, a second tile (protected/unprotected) can attach to the first tile at the rate $r_f^{'} + r_{eff}$. Thus, based on the corresponding CTMC [Figure 8] (also see Figure 2) we can say that the rate of an insufficient attachment is

$$r_{insuf} = \frac{r_f^{'}(r_f^{'} + r_{eff})}{r_{r,1} + r_f^{'} + r_{eff}} = e^{(-3+\epsilon_1+\epsilon_2)G_{se}} \frac{1 + e^{-(\epsilon_1-\epsilon_2)G_{se}}}{1 + e^{-(1-\epsilon_1)G_{se}} + e^{-(1-\epsilon_2)G_{se}}} \tag{3.0.29}$$

since an insufficient attachment happens as soon as the growth site transitions to state 2.

Our goal with respect to a particular growth site is to bury the correct tile k levels deep before an insufficient attachment event occurs.[17] In other words, if we have a $k \times k$ square whose left bottom corner location is occupied by this tile, then the $k \times k$ square completes before an insufficient attachment event occurs. This puts the tile under consideration into a "k-frozen" state. The process of tile attaching or detaching in a 2D assembly can be modeled as a random walk. [18] Note that the forward growth (tile association at the output ends of the current tile) happens at the rate of $r_{eff} + r_f^{'}$ while the backward growth (dissociation of the current tile) has a rate of $r_{r,2}$. Thus, the average rate of growth (the mean of forward and backward

[17] The time taken for single tile attachment is $O(\frac{1}{r_{eff}})$ which is less than $\frac{1}{r_{insuf}}$.

[18] The stochastic process of tile attachment and detachment in self -assembly has often been modeled as a random walk [5]. Further this is similar to the lattice gas model where modeling interactions as random walks is quite well established.

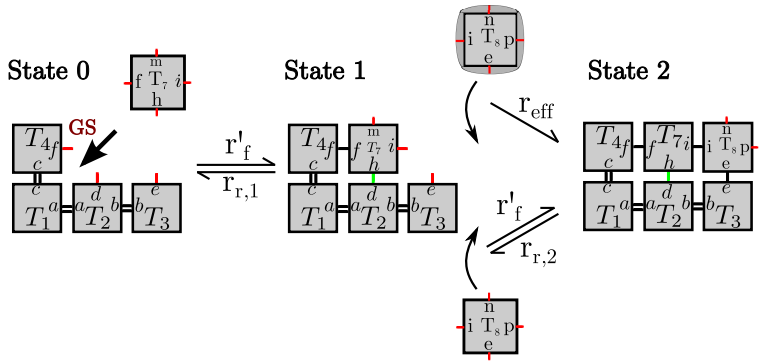


Figure 8: Continuous Time Markov Chain for error due to insufficient attachment in kATAM.

rates) is

$$\overline{r} = \frac{1}{2}(r_{eff} + r_f^{'} + r_{r,2}) \tag{3.0.30}$$

and the expected time taken for this $k \times k$ square to grow is $O(\frac{k^4}{\overline{r}})$ since in a 2D random walk, we have to take k^4 steps in expectation in order to cover k^2 locations.

Thus, for any small ϵ_{insuf}, we can find a constant c_{insuf} such that, with probability $1 - \epsilon_{insuf}$, no insufficient attachment happens at this specific location but a correct tile becomes k-frozen within time $O(\frac{k^4}{\overline{r}})$. In other words,

$$\frac{k^4}{\overline{r}} < \frac{c_{insuf}}{r_{insuf}} \tag{3.0.31}$$

where r_{insuf} is given by 3.0.29. Hence, for a given k, and constant kinetic parameters, this time is also constant. Thus we can use the same argument as Adleman et al. [1] that the $N \times N$ square is completed in expected $O(N)$ time.

□

Compact Self-healing with Activatable Tiles: The impact of activatable tiles goes beyond the compact error-resilience which is a primary concern for fault tolerant self-assembly. In case of gross external damage, e.g. a hole is created in a growing tiling assembly, activatable tiles can repair the damage with minimal error by enforcing directional growth. Since the original, self-assembled lattice was formed by algorithmic accretion in the forward direction, only forward re-growth is capable of rebuilding the correct structure. The protected monomers in the solution ensure a forward directional accretion. There is a small probability, however, of backward

growth from the unprotected monomers that were 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 concentration of the monomers and the protected tiles are much more abundant than their unprotected counterparts.

Theorem 3.0.2 *A damaged hole of size S (where S is small compared to the size of the desired pattern) is repaired before backward growth can occur in the kATAM with high probability in time $O(S^2)$ for appropriately set values of the free parameters in the model.*

Proof Observe that the maximum rate of error due to backward growth is bounded by $r_f^{'}$ while the forward rate of growth is $r_{eff} + r_f^{'}$. We will now show how to estimate the value of G_{se} required to repair a hole of size S, where size is defined to be the number of tiles. Observe that

$$\overline{r} > r_f^{'} \tag{3.0.32}$$

where $\overline{r}$, as previously defined is the average rate of growth. Using the same technique as in the previous theorem, the hole can be repaired in $O(\frac{S^2}{\overline{r}})$ (where $\overline{r}$ is the average rate of growth as previously defined) by a 2D random walk on the set of S tile positions on the 2D plane. Next, we need to guarantee no backward growth happens during this interval. We can argue that for any small ϵ_{heal} $(0 < \epsilon_{heal} < 1)$, we can find a constant c_{heal} such that with probability $1 - \epsilon_{heal}$,

$$\frac{S^2}{\overline{r}} < \frac{c_{heal}}{r_f^{'}} \tag{3.0.33}$$

From this, for a given S, we can compute G_{se} so that there is no backward growth when a hole of size at most S is repaired in $O(S^2)$ time assuming constant kinetic parameters. □

4 Computer Simulations

Winfree's original simulator, Xgrow [38] was modified to estimate error rates in assemblies with activatable tiles. The original simulator simulates two types of tile events tile addition and tile deletion. We introduced tile change events, which correspond to change of state in our model. Instead of introducing new states for each tile type into Xgrow, we opted to create new tile types for each different state of a tile. Thus, we divided each boundary tile into 3 tile types, and each rule tile into 5 tile

types. On the occurence of a tile change event, a tile of one tile type gets converted into a tile of another tile type. This allowed us to simulate the state change in the system. In this set of simulations, we model each individual event that occurs in the aKTAM.

All simulations are done assuming a single flake. A single flake in Xgrow refers to a growth assembly from a single seed tile. Note that this does not mean that only one seed tile exists. Multiple seed tiles were observed getting incorporated into the growing assembly at high concentrations ($G_{mc} = 1, 2, 3$). We modelled the original Sierpinski tiling set consisting of 1 seed tile, 2 boundary tiles, and 4 rule tiles. Thus, our activatable sierpinski tiling set contains 1 seed tile, 6 boundary tiles (3 "states" per original boundary tile), and 20 rule tiles (5 "states" per original rule tile).

Other running parameters of our tile set include: 1) $k_f = 3.5 \times 10^6 M^{-1} s^{-1}$ [44]. 2) Temperature was kept constant at 298.15K (25^{o}C); 3) Monomer tile concentration is dictated by G_{mc}, which was varied from 1-100; 4) Primer Concentration was set to 10 μM. However, in cases where the monomer tile concentration exceeds the primer concentration, the primer concentration was set as [P] = max(10 μM, $e^{-G_{mc}}$). This evaluated to the forward rate attachment of primer being 30 s^{-1}; [19] 5) Five different toehold lengths are simulated (1,3,5,7,9) nt, while the branch migration length was kept constant at 10 nt. The $G_{se} = (\frac{4000K}{T} - 11)s$ is calculated based on the sticky end length s. For each of the 5 toeholds, the G_{se} used were (26.57, 31.40, 36.24, 41.07, 45.90). The partial glue strength between a toehold and a sticky end is $\frac{toehold}{s}$. This partial glue strength is an input parameter into Xgrow, along with the glue strengths for other glues (strength 1 for all 4 edges of rule tiles and 1 edge of each boundary tile, and strength 2 for 2 edges of each boundary tile). 6) Branch migration length was kept constant at 10 nt, and the branch migration unimolecular rate constant r_{dp} = r_p was set to 4 s^{-1}.[20]; 7) Simulations were done on a 32X32 square, with the tile concentration being varied across multiple simulation runs (by varying G_{mc}). All the simulations were modelled for 10^9 events, or the formation of the complete square assembly, whichever ended first; The modified Xgrow-Activatable system is available online [11].

[19] Note that this is different from the assumption in section 2.3, which assumes that the primer concentration is maintained equal to the monomer tile concentration at all times. In simulations with very low concentrations of the monomer tile, if the primer concentration is also very low, then the primer binding reaction becomes the rate limiting step, and the simulation does not proceed. In order to avoid this, the primer was kept constant at a (relatively high) experimentally realizable concentration of 10 μM. Thus, $r_{primer} = k_f$ * [P] = 30 s^{-1}.

[20] The branch migration rate constant k_b [44], is given by $k_b = \frac{400}{x^2} s^{-1}$. In our case, $x = 10$.

Results: The phase diagram in Figure 9a shows that low-error assemblies can be formed at a higher concentration in the aKTAM model as compared to the original kTAM model. However, the growth rate in the aKTAM is 5-7 orders of magnitude lower than the growth rate of the kTAM (Figure 9b). Thus, the trade-off for lower error assemblies is the growth speed. In addition, we notice that at very low concentrations ($69 \leq G_{mc} \leq 72$, toehold = 5nt, $83 \leq G_{mc} \leq 91$, toehold = 9nt)), the kTAM forms successful assemblies, while the aKTAM is extremely slow and assemblies are unable to form. This is expected, since in addition to an extremely low concentration, it is likely that a collision event can unbind at a much higher rate in the aKTAM, thus resulting in much lower assembly growth.

Also, the toehold lengths of 5 and 7 nt, with a branch migration length of 10 nt each, seem to perform optimally for the aKTAM model, at concentrations and timelines that are realizable practically in experimental conditions. The lower toeholds of 1 and 3 nt give almost no assembly formation in the aKTAM model. The large toeholds of 9 nt can also form error free assemblies, but the growth rate is extremely low, and thus will take months to years to form assemblies. Thus, these simulations can indicate the right set of toehold and branch migration lengths to use in activatable/signal tile assembly experimental models.

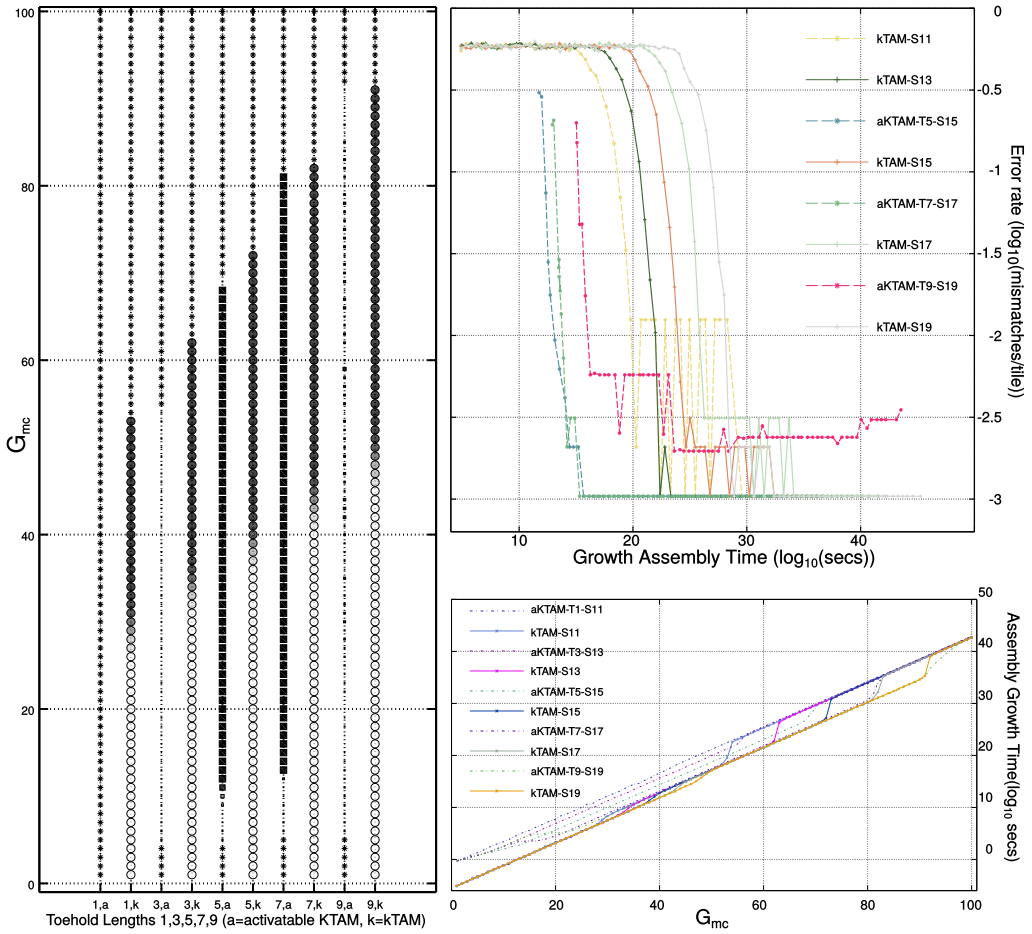


Figure 9: **(a) Left: Phase Diagram showing a side-by-side comparison of Activatable Tiles versus the original kTAM**. Five different sticky ends of lengths 11, 13, 15, 17 and 19 have been simulated, each with a toehold of 1,3,5,7,9 nt and branch migration domain of 10 nt respectively. The circles and squares refer to assemblies formed in the kTAM model and the aKTAM model respectively. The size of the polygon represents the size of the assembly formed, and the darker the polygon, the more error free the assembly is. Incase the number of tiles in the assembly $< 5\%$, it is marked by an asterix, denoting failure to form an assembly. **(b) Top Right: Growth Time of assembly versus error rates in the assembly**. As the growth times of assembly increase, the error rates decrease. This is true for both the kTAM and the aKTAM. However, the error rates in the aKTAM are far lower than the error rates of the kTAM. All the aKTAM data is represented by dashed lines, while the kTAM data is represented by solid lines. Note that there are some jagged edges, these occur when the simulation reports 0-5 mismatches in the assembly non-monotonously, and thus jagged edges appear. Also note the nomenclature, T9-S19 refers to toehold 9, sticky end 19 nt. The data for aKTAM-S1-T11 and aKTAM-S3-T13 is not shown, since the assemblies are either too small or couldn't form, and thus cannot be used for comparing growth and error rates. **(c) Bottom-Right: Growth Time of Asssemblies as a function of** G_{mc}. This graph shows that the growth speed of the aKTAM is slower than the growth rate of the kTAM, across the different toehold and sticky end lengths.

5 DNA Design of Activatable Tiles

This section provides details on a experimental design of Activatable Tiles. No experimental work has been done as part of this article. An alternate 1D design has been implemented [12]. We describe a 2D design that has been worked out to a high level of detail and can serve as a prototype to implement 2D experiments.

5.1 DNA Design of 1D Activatable Tiles

The DNA design of 1D activatable tiles is very helpful in understanding the more complex DNA design of 2D activatable tiles. Recent experimental work has been shown that was motivated by the activatable tiles model [12, 43]. It is also motivated by the need for a protection strategy for tiles that self-assemble into a 1D lattice, such as the boundary of a computational tiling. To ease understanding, we make use of three levels of abstraction to describe the DNA tile design: at the highest level of abstraction, we describe the deprotection strategy using a finite state machine; in the next level, we explain the same mechanism with an example; in the lowest level, the design description involves actual DNA sequences. The *1D abstract tile* is a square with a single input and single output. Every tile has a tile core (which is common to all tiles) and a unique pair of sticky ends encoding the input and the output respectively. An *unprotected tile* is the same as the original tile while its *protected counterpart* has a layer covering its sticky ends. The *toehold*, an exposed part of the input sticky end, facilitates strand displacement and consequently deprotection.

Consider a simple example system comprising of five tiles. There are two start tiles (seed tiles), $S1$ and $S2$, two intermediate tiles, $I1$ and $I2$ and one output tile O [Figure 10](a). Ideally in the presence of $S1$, the desired assembly order is $S1 \rightarrow I1 \rightarrow O$, while in the presence of $S2$, the desired assembly order is $S2 \rightarrow I2 \rightarrow O$ [Figure 10](b). In the absence of protection, however, such directional growth cannot be guaranteed [Figure 10](c). The goal is to prevent spontaneous nucleation by starting with protected tiles and deprotecting them only after they have been attached to the growing assembly. Errors in 1D and 2D are very different. Single tile mismatches, a major source of error in 2D tile assemblies, do not happen in 1D systems. Instead, errors are introduced when assembly does not occur in the desired fashion due to ambiguous binding sites as shown in Figure 10(c). The following subsections discuss how such sequential assembly is ensured.

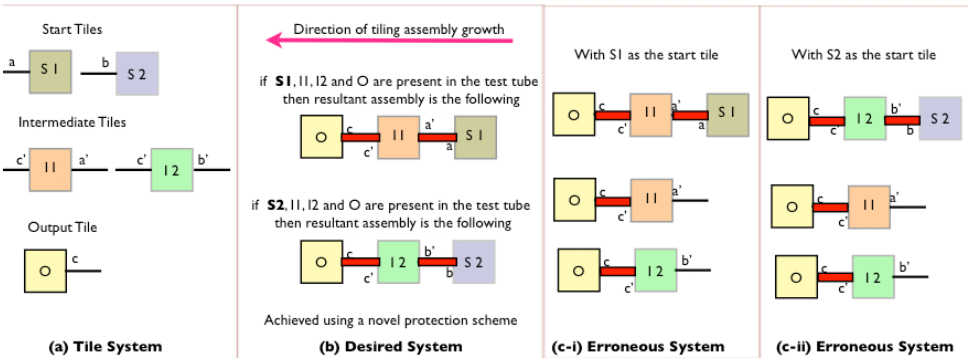


Figure 10: (a) 1D Abstract Tile System for testing sequential assembly. In the protected system, unlike the unprotected counterpart where none of the tiles have any protection, $I1, I2$ and O have protection layers, (b) The desired assembly with initiator tile $S1$ and $S2$ respectively, (c) Possible spontanous nucleation errors if the assembly is not forced to go from the initiator (seed) tile to the output tile.

High Level Description of the Design: The key idea is first presented in the form of a finite state machine [Figure 11].

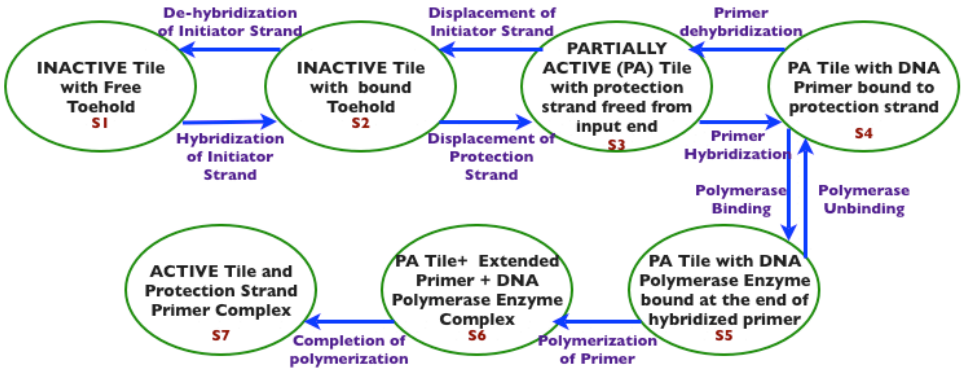


Figure 11: A Finite State Automata for the deprotection of the output sticky end of a 1D tile only when its input sticky end is correctly matched.

Since the input/output sticky ends are protected by a protection strand, an "*inactive*" set of tiles co-exist in the solution and do not self-assemble into one dimensional lattices (S1). When the initiator strand (analogous to a seed tile in the kTAM model, which may be part of a larger nanostructure) is introduced in the mixture, the former displaces the protection strand at the input end of the tile with matching sticky ends at any growth site, resulting in a "*partially active*" tile with a correctly bound input sticky end and protected output sticky end (S3). In presence of a suitable primer, the

protection strand (now free at the input end but still hybridized to the output sticky end) can act as the template for DNA polymerization (S4-S6). When the polymerization completes, the protection strand is stripped off the output sticky end leaving it to initiate another deprotection. Thus the assembly proceeds from input to output end at all times ("*active*" state (S7)).

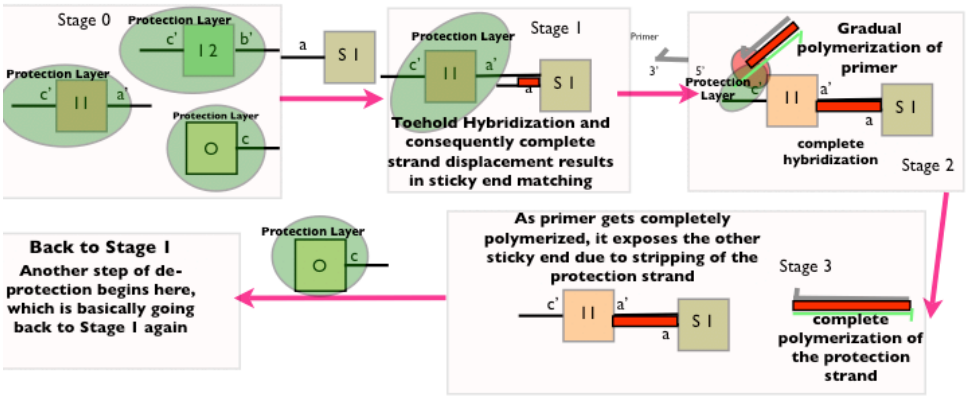


Figure 12: A high-level reaction pathway for deprotection of a 1D protected tile.

We further explain the deprotection strategy using the example system from above; Figure 12 shows the corresponding reaction pathway. Initially, all the tiles ($I1$, $I2$ and O) in the solution are protected (Stage 0) and they do not interact until the start tile $S1$ is introduced. $S1$ next displaces the protection layer at the input end of the intermediate tile $I1$ (Stage 1). Once the protection layer at the input end of $I1$ is freed, the primer in the solution can hybridize with it and DNA polymerase enzyme can extend it all the way up to the output end, thus exposing the output sticky end (Stage 2 and 3). Tile O is deprotected and hybridized to $I1$ in the next cycle.

Experimental Design: The simplest DNA sequence designs are shown in Figure 13. The tile core can be simply a double-stranded DNA with the sticky ends being the single stranded overhang extending out of the double-stranded portion. The sticky ends are protected by the protection strand M.

For adjacent tiles, the protection strand needs to be arranged in a different manner so as to satisfy both constraints on the direction for sticky end matching as well as template for polymerization [Figure 13], resulting in two kinds of tile types. Some of the key features of this tile design are (i) The 3 base portion (E) at the $3'$ end of the protection strand prevents polymerization of the toehold $H1$, (ii) The portion of the

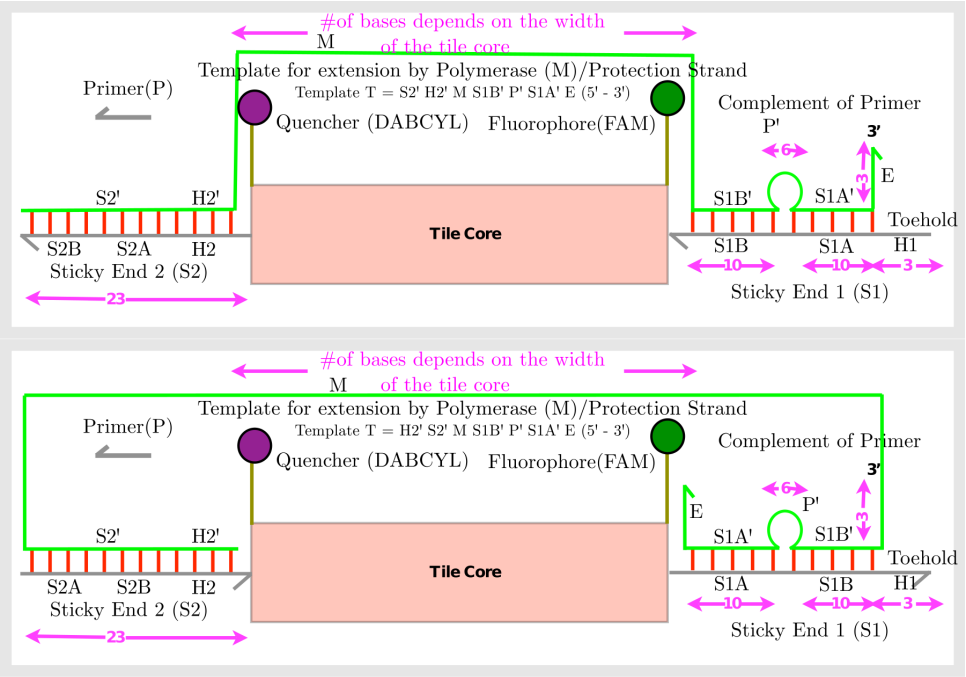


Figure 13: ((Top)Protection Strategy for Tile 1, (Bottom)Protection Strategy for its neighbor Tile 2. The number of bases for each section is shown in pink.

protection strand which hybridizes to the primer P is held tightly in a hairpin loop of six bases between two subportions of the input sticky end, (iii) The fluorophore and the quencher for detection purposes, are positioned such that the fluorophore is quenched only when correct tiles hybridize. Now suppose, Tile 2 is already deprotected and part of an assembly. Figure 14 describes a reaction pathway to describe how Tile 1 is deprotected by Tile 2. One can verify whether the tile system is assembling as desired by observing the pattern in the fluorescence data. Native gel electrophoresis can be used to find whether the dominant assembly in presence of the initiator tile is the one desired.

5.2 DNA Design of 2D Activatable Tiles

Our DNA design for 2D activatable tiles is a direct extension of our 1D activatable tiles. Before giving a domain design for activatable tiles, just as we did in the 1D description, we first describe the protection/deprotection strategy using an abstract version of DX tiles. The *2D abstract DX tile* is a square with two inputs and two outputs; every tile has a tile core (common to all tiles) and unique sticky ends encoding

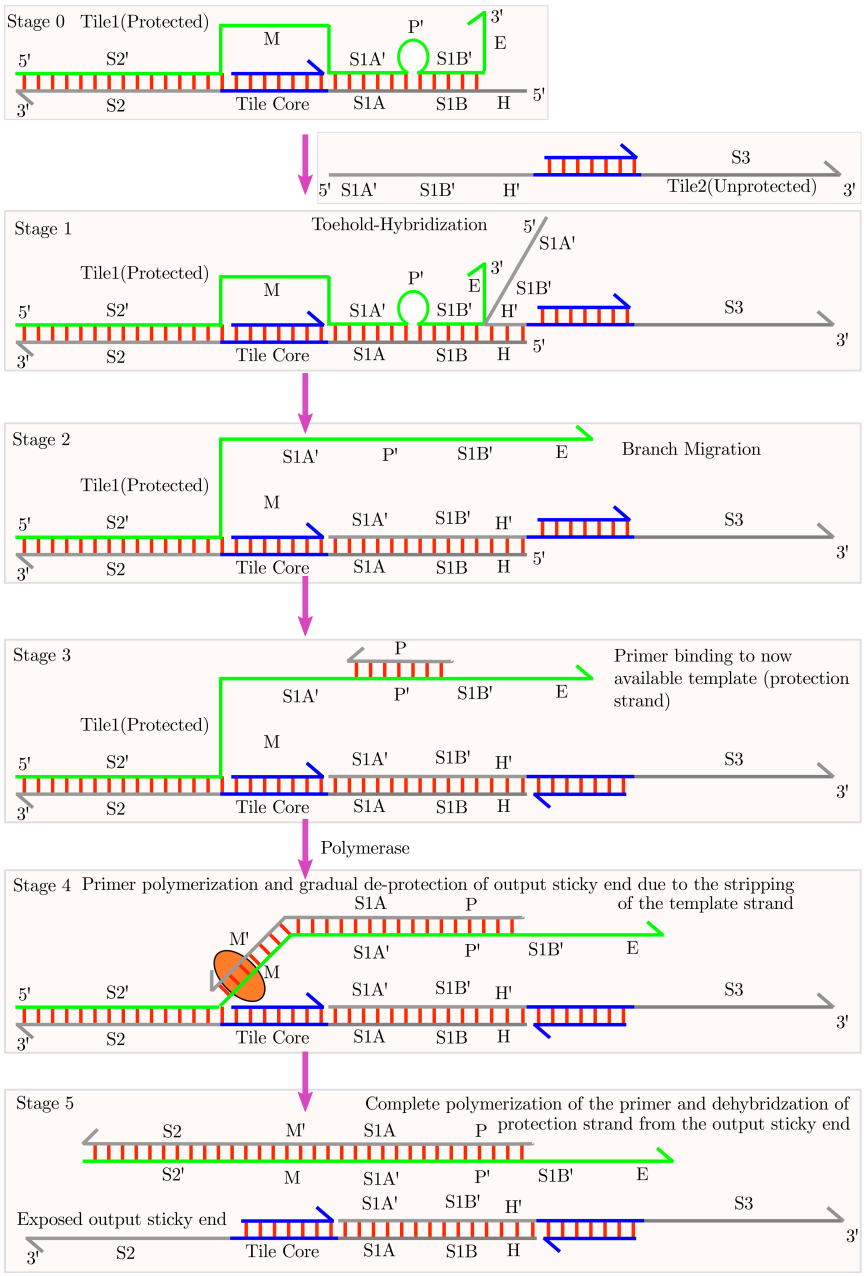
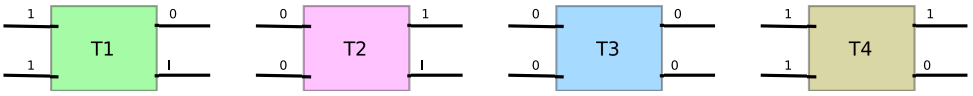
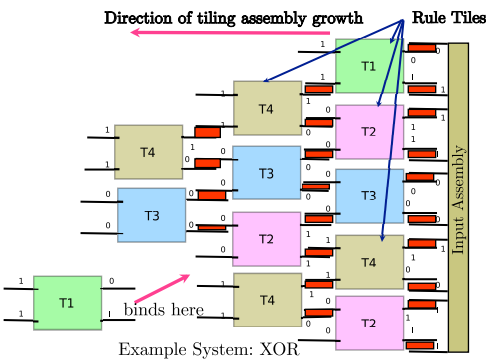


Figure 14: Reaction Pathway for deprotection of Tile 1 by Tile 2 at the experimental level.

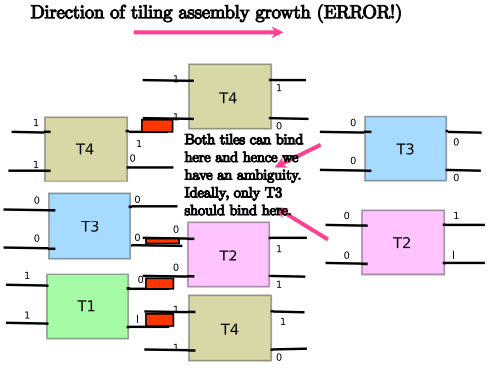
the input and the output. An *unprotected tile* is the same as the original tile whereas its *protected counterpart* has a layer covering its sticky ends. One of the tile inputs (input 1) has a few bases exposed at the end, facilitating strand displacement and, consequently, deprotection. This exposed portion is referred to as a toehold. The other input (input 2), however, is completely hybridized to a protection strand which is separate from the output protection strand. The idea is that the input 2 sticky end cannot hybridize until the input 1 is correctly hybridized. The toehold of input 2 is exposed only when its protection strand is displaced by the protection strand from input 1. When input 2 hybridizes completely, it frees the protection strand covering the outputs from the input end. The DNA polymerase enzyme then exposes the output sticky ends.



(a) Abstract DX Rule Tiles for XOR computation.



(b) Desired Directional Tiling Assembly starting with the input strand.



(c) Erroneous Growth when directionality constraint is not enforced.

Figure 15: 2D Tiles used in XOR Computation(a), described along with desired(b) and undesired assembly pathways(c).

Example computational system in 2D: XOR computation: For this computation, the output is 1 only when exactly one of the inputs is 1. Figure 15a) shows four DX computational rule tiles implementing the computation for XOR. Figure 15b) shows an example of desired directed tile assembly growth in the right to left direction. In absence of protection, however, such directional growth of tiling assembly

cannot be guaranteed. An instance of an erroneous tiling assembly growth is shown in Figure 15c). The goal is to enforce a sequential assembly to avoid ambiguities shown in Figure 15c) by using the novel protection-deprotection scheme.

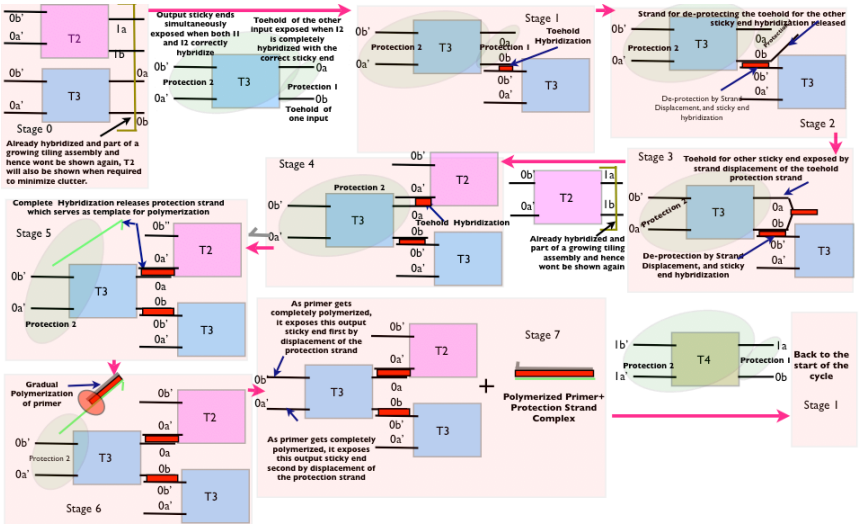


Figure 16: High Level reaction pathway for deprotection in two dimensions

High Level Description of the Design: We describe a high level version of deprotection using XOR computation as an example system (Figure 16). Assuming tiles $T2$ and $T3$ are part of a growing tiling assembly, ideally another $T3$ should bind, at their output ends. This results in the output sticky end $0b'$ of bound tile $T3$ (part of a tiling assembly) displacing the protection strand over input sticky end $0b$ of the protected monomer $T3$. The protection strand from the input $0b$ of tile $T3$ next displaces the toehold protection of the $0a$ input for tile $T3$. In the following step the output sticky end $0a'$ of bound tile $T2$ displaces the protection on $0a$ input of protected tile $T3$. Once the protection layer at this input end is freed, the primer in the solution can hybridize with it and DNA polymerase enzyme can extend it to the output ends, thus exposing the output sticky ends one by one (output 2 followed by output 1).

Experimental Design: A sequence design of such an activatable tile is not difficult. Figure 17 (right) shows the details of the protection strategy for an experimental validation. Here, one can use a DX or a TX tile as the tile core, since they are very compact; a compact tile will improve the likelihood of the strand displacement and

polymerization events that ultimately deprotect the tile and make it available for hybridization. Figure 17 also gives the protection strategy for the input and output sticky ends. Input 1 is protected by design, since it is designed as a hairpin, and one arm of the hairpin acts as the protector strand. The internal toehold of input 1 ($HIn1$) hybridizes with the correct sticky end (from the growth site) and eventually displaces the input 1 protector. The T_1 portion of the input 1 protector strand then hybridizes with the input 2 toehold protector strand, and via *toehold-exchange* [44], exposes the input 2 toehold $HIn2$. This facilitates another strand displacement by the output sticky end (part of a larger nanostructure) that binds with the input 2 sticky end. Once the protection strand 2 is freed from the input side, it hybridizes with the primer P. Next, DNA polymerase enzyme present in the solution extends the primer and eventually pulls the protection strand (template) first out of output 2, $SOut2$, and next out of output 1, $SOut1$.

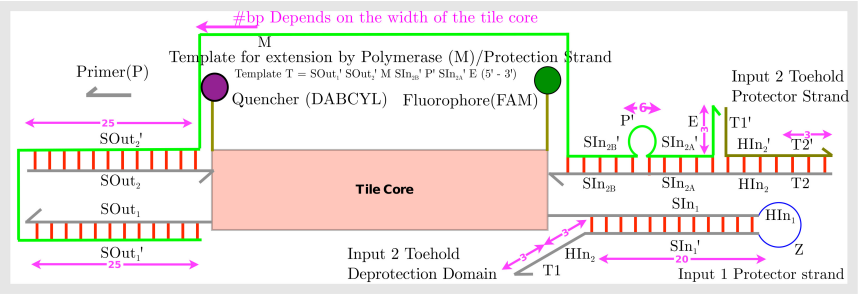


Figure 17: Details of a protected DNA tile for two dimensional tiling (Number of bases in each strand is shown in pink.).

5.3 Testing Processing and Displacing Power of ϕ 29

Even before we can incorporate the concept of activation in real system, we need to verify the process of deprotection via the use of suitable polymerase. Our chosen polymerase is polymerase ϕ29. Researchers have studied the structure of ϕ29 polymerase and have provided useful insights into its exceptional strand displacement and processing power, and have deduced its translocation mechanism [18]. We start with a DNA origami structure which is the upper left quadrant of the origami rectangle [28]. Note that it has a pair of fluorophores and quenchers, placed close enough that no signal in the fluorescence spectrometer can be observed. However, when the primer P is introduced in the solution, in presence of ϕ29 it transforms the origami structure into a mere double-stranded DNA, thus releasing the fluorophore bearing

FIRST STAPLE PAIR WITH MODIFICATION:

TGAGTTTCAAAGGAACAACTAAAGATCTCCAAT BLACK-HOLE QUENCHER

6-CARBOXYFLUORESCEIN TAAAAAAGGCTTTTGCGGGATCGTCGGGTAGCA

SECOND STAPLE PAIR WITH MODIFICATION:

AAGAGGAACGAGCTTCAAAGCGAAAGTTTCATT IOWA BLACK RQ

TEXAS RED-X NHS ESTER TTCCATATATTTAGTTTGACCATTAAGCATAAA

PRIMER:

GCGGGGTTTTGCTCAGTACCAGGCGGATAAGTGCCG

Figure 18: DNA sequences for staples with modifications and primer or testing processing and displacement ability of polymerase $\phi 29$. The other staple sequences are used directly from the publication [28].

staples. As the latter is no longer in close proximity of staples bearing quenchers, an increase in signal can be detected in the spectrometer. However, there is one difficult step in this process. Since annealing of origami involves excess of staple strands (most commonly in $10:1$ ratio [28]), the signal changes due to both staples incorporated in the origami and excess staples in solution. To obtain the portion of change due only to origami staples, the origami needs to be purified using Microcon centrifugal filter devices ((100,000 MWCO, 300 $\times$ g speed, 10 min)) before introducing the primer or the polymerase. The origami uses the commercially available M13mp18 viral genome as the scaffold and the sequences of the fluorophore and quencher bearing strands and the primer are shown in Figure 18.

6 Other Applications of Activatable Tiles

Beyond their applications to computational tiling, activatable tiles can be used for building sensing and concentration systems [Figure 19]. For instance, a type of modified activatable tile that has a docking site (e.g. a DNA or RNA aptamer binding site) specific to this target molecule can be designed. Initially, the tiles are in an inactive state; they are not bound to a target molecule nor are they assembled together. When a target molecule binds to the tile's docking site, the tile transitions from an inactive to an active state. Tiles in the active state can assemble. As the

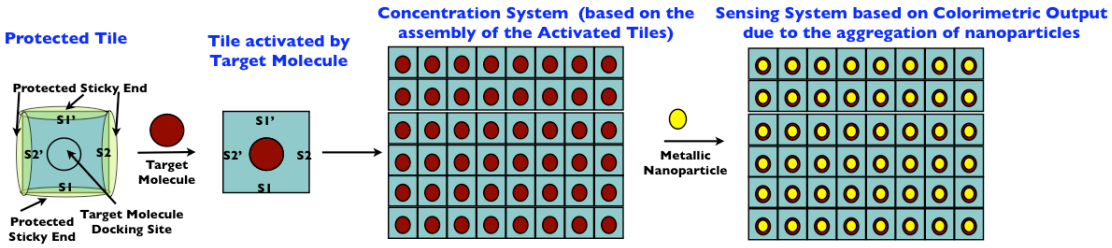


Figure 19: A concentration and sensing system with activatable tiles.

activated tiles assemble, the target molecules are concentrated making an excellent concentration system. For added functionality, one can attach metallic nanoparticles to the tiles or target molecules. With the nanoparticles, the assembly of activatable tiles detects the presence of a target molecule in solution (based on the colorimetric output) and behaves as a nano-scale sensor.

Activatable tiles can also be used for reaction catalyzation. The concept of DNA directed chemistry has been explored in recent years [27]. Suppose that for some small k, the goal is to place k distinct small, target molecules in close proximity, to initiate or catalyze a chemical reaction. Then k distinct modified activatable tiles can be designed with a docking location that provides a binding site for one of the distinct target molecules. The tiles undergo a state transition from inactive to active only when they are carrying their target molecules. Once activated, these k distinct tiles assemble into a small tiling lattice, putting the target molecules in close proximity, and allowing them to react. In addition, some of the reaction products can be used to make the tiles disassemble and return to the inactive state, allowing the tiles to be reused. Observe that the location of the binding site has a major role to play in this catalyzation process. The binding site on the same face of each tile type is so designed that after assembly, the molecules, bound to the tiles will be close to each other. They are never bound inside the lattice and therefore, the reaction can never become slower. Figure 20 shows such a reaction catalyzation for $k = 4$. Reaction catalyzation is an established subfield of chemistry. There are two types of reaction catalyzation: 1) Heterogeneous catalyst, where the catalyst holds the reactant molecules in close proximity to each other and thus increases the reaction rate ; 2) Homogeneous catalyst, where the catalyst reacts with the reactants to form intermediate products and is eventually released from the intermediates to form final products. An example of a heterogeneous catalyst is finely divided iron in the Haber

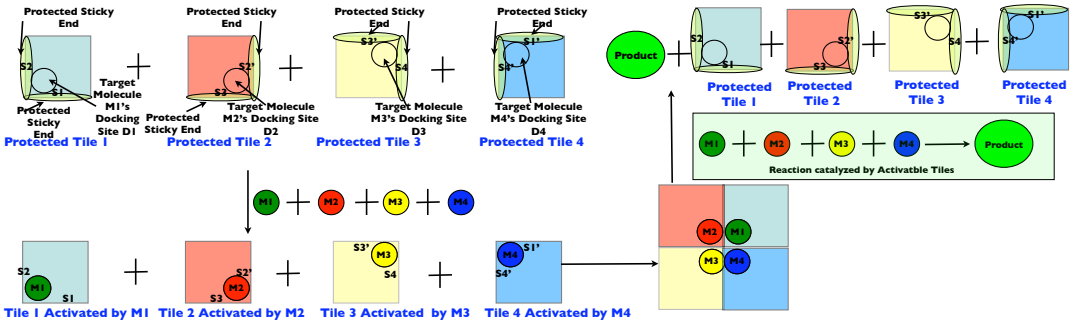


Figure 20: Reaction catalyzation with activatable tiles.

process of manufacturing ammonia while that of a homogeneous catalyst is chlorine free radicals in the breakdown of ozone. Observe that in our case, activatable tiles behave as both homogeneous and heterogeneous catalysts since not only do they participate in the reaction, but the reaction takes place on their surfaces.

7 Conclusion

The reduction of errors in computational tiling assemblies eliminates a major roadblock in the development of applications of DNA lattices, allowing, for example, the construction of complex nano-electronic circuits. In this paper, we have described a tile design which uses strand displacement and DNA polymerization to improve the robustness of computational assembly without increasing scale. One of the key features of our design is that although deprotection enforces sequentiality, the parallel, asychronous nature of self-assembly remains. We developed abstract and kinetic models for activatable tiles that allow us to compare error rates and growth speed with that of Winfree's original kinetic model. We showed that activatable tiles can provide robust assembly growth in the same scale as the original assembly and can even repair small amounts of damage assuming suitable values of the model's kinetic parameters. These results show that an activatable tile set is indeed a compact error-resilient and self-healing tile set. We further described a DNA design for activatable tiles based on these models. Additionally we observed that activatable tiles not only reduce errors in computational tiling assembly, they can be used for tasks including molecular sensing and reaction catalyzation.

Although it may be impossible to eliminate errors completely from the assembly process, the design for activatable tiles appears to be quite promising, and some re-

cent experimental work has been shown that provides some credibility to this model [12]. Our detailed computer simulations simulate each of the protection/deprotection steps, and use recent experimental rate constants that hopefully provide a valid approximation of a real experimental scenario. Future work would include comparing this to other existing error-correction techniques in more detail [15, 9, 10, 3], particularly Fujibayashi *et al.'s* enzyme-free activated tile model [9, 10]. Our hope is that our proof of concept system will make self-assembly experiments significantly more robust to assembly errors. We would also like to investigate one interesting open question: Can combining overlay redundancy techniques [30] with the idea of activatable tiles further improve the error-resilience of self-assembly experiments in the original scale?

A further challenging direction beginning to be investigated is the development self-repairing tiling assemblies [39, 21].

8 Acknowledgments

The authors thank the anonymous referees, whose suggestions have had a noticeable improvement on the quality of the article, in helping us better articulate ideas and by giving attention to detail. The work was supported by NSF Grants CCF- 1217457, CCF-1141847, CCF-0829797.

References

[1] Adleman, L., Cheng, Q., Goel, A., and Huang, M.-D. Running Time and Program Size for Self-Assembled Squares. *Symposium on Theory of Computing* (2001), 740–748.

[2] Barish, R. D., Schulman, R., Rothemund, P. W. K., and Winfree, E. An Information-bearing Seed for Nucleating Algorithmic Self-assembly. *Proceedings of the National Academy of Sciences 106*, 15 (2009), 6054–6059.

[3] Chen, H.-L., Cheng, Q., Goel, A., Huang, M.-D., and de Espanés, P. M. Invadable self-assembly: Combining robustness with efficiency. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 2004), SODA '04, Society for Industrial and Applied Mathematics, pp. 890–899.

[4] Chen, H.-L., and Doty, D. Parallelism and time in hierarchical self-assembly. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms* (2012), SODA '12, SIAM, pp. 1163–1182.

[5] Chen, H.-L., and Goel, A. Error Free Self-assembly Using Error Prone Tiles. *DNA Computing* (2004), 62–75.

[6] Demaine, E., Demaine, M., Fekete, S., Ishaque, M., Rafalin, E., Schweller, R., and Souvaine, D. Staged self-assembly: nanomanufacture of arbitrary shapes with o(1) glues. *Natural Computing 7*, 3 (2008), 347–370.

[7] Dirks, R., and Pierce, N. Triggered Amplification by Hybridization Chain Reaction. *Proceedings of the National Academy of Sciences of the United States of America 101*, 43 (2004), 15275–15278.

[8] Fujibayashi, K., Hariadi, R., Park, S. H., Winfree, E., and Murata, S. Toward reliable algorithmic self-assembly of dna tiles: A fixed-width cellular automaton pattern. *Nano Letters 8*, 7 (2008), 1791–1797. PMID: 18162000.

[9] Fujibayashi, K., and Murata, S. A method of error suppression for self-assembling dna tiles. *DNA Computing 3384* (2005), 113–127.

[10] Fujibayashi, K., Zhang, D. Y., Winfree, E., and Murata, S. Error suppression mechanisms for dna tile self-assembly and their simulation. 589–612.

[11] Garg, S. Xgrow modified for activatable tiles. Available at https://github.com/sudhanshugarg/projects/tree/master/xgrow-activatable.

[12] Garg, S., Chandran, H., Gopalkrishnan, N., LaBean, T. H., and Reif, J. Directed enzymatic activation of 1-d dna tiles. *ACS Nano 9*, 2 (2015), 1072–1079. PMID: 25625898.

[13] Gautam, V., Haddow, P., and Kuiper, M. Reliable self-assembly by self-triggered activation of enveloped dna tiles. In *Theory and Practice of Natural Computing*, A.-H. Dediu, C. Martn-Vide, B. Truthe, and M. Vega-Rodrguez, Eds., vol. 8273 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 68–79.

[14] Hendricks, J., Padilla, J., Patitz, M., and Rogers, T. Signal transmission across tile assemblies: 3d static tiles simulate active self-assembly by 2d signal-passing tiles. In *DNA Computing and Molecular Programming*, D. Soloveichik and B. Yurke, Eds., vol. 8141 of *Lecture Notes in Computer Science*. Springer International Publishing, 2013, pp. 90–104.

[15] Jang, B., Kim, Y.-B., and Lombardi, F. Error rate reduction in dna self-assembly by non-constant monomer concentrations and profiling. In *Design, Automation Test in Europe Conference Exhibition, 2007. DATE '07* (April 2007), pp. 1–6.

[16] JONOSKA, N., and Karpenko, D. Active tile self-assembly, part 1: universality at temperature 1. *International Journal of Foundations of Computer Science 25*, 02 (2014), 141–163.

[17] JONOSKA, N., and Karpenko, D. Active tile self-assembly, part 2: self-similar structures and structural recursion. *International Journal of Foundations of Computer Science 25*, 02 (2014), 165–194.

[18] Kamtekar, S., Berman, A. J., Wang, J., Lzaro, J. M., de Vega, M., Blanco, L., Salas, M., and Steitz, T. A. Insights into strand displacement and processivity from the crystal structure of the protein-primed dna polymerase of bacteriophage φ29. *Molecular Cell 16*, 4 (2004), 609 – 618.

[19] Keenan, A., Schweller, R., and Zhong, X. Exponential replication of patterns in the signal tile assembly model. In *DNA Computing and Molecular*

Programming, D. Soloveichik and B. Yurke, Eds., vol. 8141 of *Lecture Notes in Computer Science*. Springer International Publishing, 2013, pp. 118–132.

[20] LaBean, T., Yan, H., Kopatsch, J., Liu, F., Winfree, E., Reif, J., and Seeman, N. Construction, Analysis, Ligation, and Self-Assembly of DNA Triple Crossover Complexes. *Journal of the American Chemical Society 122*, 9 (2000), 1848–1860.

[21] Majumder, U., Sahu, S., LaBean, T., and Reif, J. Design and simulation of self-repairing dna lattices. In *DNA Computing*, C. Mao and T. Yokomori, Eds., vol. 4287 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 195–214.

[22] Mao, C., Labean, T., Reif, J., and Seeman, N. Logical Computation Using Algorithmic Self-assembly of DNA Triple-crossover Molecules. *Nature 407* (2000), 493–496.

[23] Murata, S. Self-assembling dna tiles - mechanisms of error suppression. In *SICE 2004 Annual Conference* (Aug 2004), vol. 3, pp. 2764–2767.

[24] Padilla, J., Liu, W., and Seeman, N. Hierarchical self assembly of patterns from the robinson tilings: Dna tile design in an enhanced tile assembly model. *Natural Computing 11*, 2 (2012), 323–338.

[25] Padilla, J., Patitz, M., Pena, R., Schweller, R., Seeman, N., Sheline, R., Summers, S., and Zhong, X. Asynchronous signal passing for tile self-assembly: Fuel efficient computation and efficient assembly of shapes. In *Unconventional Computation and Natural Computation*, G. Mauri, A. Dennunzio, L. Manzoni, and A. Porreca, Eds., vol. 7956 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 174–185.

[26] Reif, J., Sahu, S., and Yin, P. Compact Error-Resilient Computational DNA Tiling Assemblies. *DNA Computing* (2004), 293–307.

[27] Rosenbaum, D. M., and Liu, D. R. Efficient and sequence-specific dna-templated polymerization of peptide nucleic acid aldehydes. *Journal of the American Chemical Society 125*, 46 (2003), 13924–13925. PMID: 14611205.

[28] Rothemund, P. Folding DNA to Create Nanoscale Shapes and Patterns. *Nature 440* (2006), 297–302.

[29] Rothemund, P., Papadakis, N., and Winfree, E. Algorithmic Self-Assembly of DNA Sierpinski Triangles. *PLoS Biology 2* (2004), 424–436.

[30] Sahu, S., and Reif, J. Capabilities and limits of compact error resilience methods for algorithmic self-assembly in two and three dimensions. In *DNA Computing*, C. Mao and T. Yokomori, Eds., vol. 4287 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 223–238.

[31] Saturno, J., Blanco, L., Salas, M., and Esteban, J. A. A novel kinetic analysis to calculate nucleotide affinity of proofreading dna polymerases:: Application to 29 dna polymerase fidelity mutants. *Journal of Biological Chemistry 270*, 52 (1995), 31235–31243.

[32] Schulman, R., and Winfree, E. Programmable Control of Nucleation for Algorithmic Self-Assembly. *SIAM Journal on Computing 39*, 4 (2009), 1581–1616.

[33] Schulman, R., Yurke, B., and Winfree, E. Robust self-replication of combinatorial information via crystal growth and scission. *Proceedings of the National Academy of Sciences* (2012).

[34] Soloveichik, D., and Winfree, E. Complexity of compact proofreading for self-assembled patterns. In *DNA Computing*, A. Carbone and N. Pierce, Eds., vol. 3892 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 305–324.

[35] Thompson, B. J., Escarmis, C., Parker, B., Slater, W., Doniger, J., Tessman, I., and Warner, R. C. Figure-8 configuration of dimers of s13 and φ-x174 replicative form dna. *Journal of Molecular Biology 91*, 4 (1975), 409 – 419.

[36] Wang, H. Proving Theorems by Pattern Recognition II. *Bell Systems Technical Journal* (1961).

[37] Winfree, E. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, 1998.

[38] Winfree, E. Simulations of Computing by Self-Assembly. Tech. rep., California Institute of Technology, 1998.

[39] Winfree, E. Self-healing tile sets. In *Nanotechnology: Science and Computation*, J. Chen, N. Jonoska, and G. Rozenberg, Eds., Natural Computing Series. Springer Berlin Heidelberg, 2006, pp. 55–78.

[40] Winfree, E., and Bekbolatov, R. Proofreading Tile Sets: Error Correction for Algorithmic Self-Assembly. *DNA Computing* (2003), 126–144.

[41] Yan, H., Park, S. H., Finkelstein, G., Reif, J., and LaBean, T. DNA-Templated Self-Assembly of Protein Arrays and Highly Conductive Nanowires. *Science 301*, 5641 (2003), 1882–1884.

[42] Zhang, D. Y. Cooperative hybridization of oligonucleotides. *Journal of the American Chemical Society 133*, 4 (2011), 1077–1086.

[43] Zhang, D. Y., Hariadi, R. F., Choi, H. M., and Winfree, E. Integrating dna strand-displacement circuitry with dna tile self-assembly. *Nature Communications 4:1965* (2013).

[44] Zhang, D. Y., and Winfree, E. Control of DNA Strand Displacement Kinetics Using Toehold Exchange. *Journal of the American Chemical Society 131*, 48 (2009), 17303–17314.