Invadable Self-Assembly: Combining Robustness with Efficiency

Ho-Lin Chen [*]
Stanford University

Qi Cheng [†]
University of Oklahoma

Ashish Goel [‡]
Stanford University

Ming-Deh Huang [§]
University of Southern California

Pablo Moisset de Espanés [¶]
University of Southern California

Abstract

DNA self-assembly is emerging as a key paradigm for nanotechnology, nano-computation, and several related disciplines. In nature, DNA self-assembly is often equipped with explicit mechanisms for both error prevention and error correction. For artificial self-assembly, these problems are even more important since we are interested in assembling large systems with great precision. So far, theoretical studies of DNA self-assembly have primarily focused on the efficiency of the assembly process in terms of the program size and the running time. In this paper, we perform a preliminary study of algorithms for DNA self-assembly that are both robust and efficient.

Strand invasion is an important error-correction mechanism observed in several natural self-assembling systems. We first define *invadable self-assemblies* as self-assembling systems which can effectively use the strand invasion mechanism for error-correction. We then show that $O(\log^2 n/\log\log n)$ tiles are sufficient to assemble an $n \times n$ square in this model. The running time of our system is $\tilde{O}(n)$. We obtain our result by growing a counter which simulates Chinese remaindering. The running time and the program size of our invadable system are within polylogarithmic factors of known lower bounds for general systems, i.e. the efficiency penalty for obtaining robustness is small in our model. We also show how to simulate an arbitrary Turing machine using an invadable self-assembly system.

1 Introduction

Self-assembly is the ubiquitous process by which objects autonomously assemble into complexes. Nature provides many examples: Atoms react to form molecules. Molecules react to form crystals and supramolecules. Cells sometimes coalesce to form organisms. It has been suggested that self-assembly will ultimately become an important technology, enabling the fabrication of great quantities of small complex objects such as computer circuits. DNA has emerged as an important component to use in *artificial* self-assembly of nanoscale systems due to its small size, its incredible versatility, and the precedent set by the abundant use of DNA self-assembly in nature. Accordingly, DNA self-assembly has received significant attention over the last few years, both by practitioners [13, 14, 11], and by theoreticians [5, 6, 12, 1, 7, 8, 2, 3]. The theoretical results have focused on efficiently assembling structures of a controlled size (the canonical example being assembly of $n \times n$ squares). In this paper, we perform a preliminary study of algorithms for DNA self-assembly that are both robust and efficient.

The Tile Assembly Model, originally proposed by Rothemund and Winfree [8], and later extended by Adleman *et al.* [2], provides a useful framework to study the *efficiency* (as opposed to robustness) of DNA self-assembly. In this model, a square tile is the basic unit of an assembly. Each tile has a glue on each side; each glue has a label and a strength (typically 1 or 2). A tile can add to a position in an existing assembly if at all the edges where this tile "abuts" the assembly, the glues on the tile and the assembly are the same, and the total strength of these glues is at least equal to a system parameter called the temperature (typically 2). Assembly starts from a single seed crystal and proceeds by repeated accretion of single tiles. The speed of an addition (and hence the time for the entire process

[*]Ho-Lin Chen is at the Department of Computer Science, Stanford University. Email: holin@stanford.edu. Research supported in part by NSF Award 0323766.

[†]Qi Cheng is at the Department of Computer Science, University of Oklahoma. Research supported by an NSF CAREER award. Email: qcheng@ou.edu.

[‡]Ashish Goel is at the Department of Management Science and Engineering and (by courtesy) Computer Science, Stanford University, Terman 311, Stanford CA 94305. Email: ashishg@stanford.edu. Research supported by NSF CAREER Award 0133968 and a grant from NASA.

[§]Ming-Deh Huang is at the Department of Computer Science, University of Southern California. Email: huang@usc.edu. Research supported in part by a grant from NASA and by NSF Grant CR-9820778.

[¶]Pablo Moisset de Espanés is at the Department of Computer Science, University of Southern California. Email: pmoisset@usc.edu. Research supported in part by a grant from NASA/JPL.

to complete) is determined by the concentrations of different tiles in the system. Details are in Section 2.

Rothemund and Winfree [8] gave an elegant self-assembling system for constructing squares by self-assembly in this model. Their construction of $n \times n$ squares requires time $\Theta(n \log n)$ and program size $\Theta(\log n)$. Adleman *et al.* [2] presented a new construction for assembling $n \times n$ squares which uses optimal time $\Theta(n)$ and optimal program size $\Theta(\frac{\log n}{\log \log n})$. Both constructions first assemble a roughly $\log n \times n$ rectangle (at temperature 2) by simulating a binary counter, and then complete the rectangle into a square. Later, Adleman *et al.* [3] studied several combinatorial optimization problems related to self-assembly. Together, the above results are a comprehensive treatment of the efficiency of self-assembly, but they do not address robustness.

Robust Self-Assembly and Strand Invasion: In nature, DNA self-assembly is often equipped with explicit mechanisms for both error prevention and error correction. For artificial self-assembly, these problems are even more important since we are interested in assembling large systems with great precision. In reality, several effects are observed which lead to a loss of robustness compared to the above model. The assembly tends to be reversible, i.e., tiles can fall away from an existing assembly. Also, incorrect tiles sometimes get incorporated and locked into a growing assembly, much like defects in a crystal. However, for sophisticated combinatorial assemblies like counters, which form the basis for controlling the size of a structure, a single error can lead to assemblies drastically larger or smaller (or different in other ways) than the intended structure. Finally, the temperature of the system can be controlled only imperfectly.

To address these issues, we need greater understanding of natural tools used to correct DNA self-assemblies, model them algorithmically, design robust self-assembling systems which incorporate these tools, and analyze the performance of these new system. This is an exciting direction. In this paper, we present one such paradigm based on the notion of *strand invasion* observed in natural systems [14].

Consider DNA strands 1, 2, and 3 in Figure 1(A). DNA strands 3 and 1 are complementary to each other, and we would like them to attach to each other. DNA strand 2 is complementary to a part of strand 1, but not to the entire strand 1. Imagine that DNA strands 2 and 1 are attached to each other. Strand 3 can still form a weak bond with the remaining unpaired portion of strand 1. Now, strands 3 and 2 compete for that part of strand 1 which is paired with strand 2. Since strand 3 is anchored, it ultimately wins, and "invades" strand 2 off. This can be modeled using a random walk on a line;

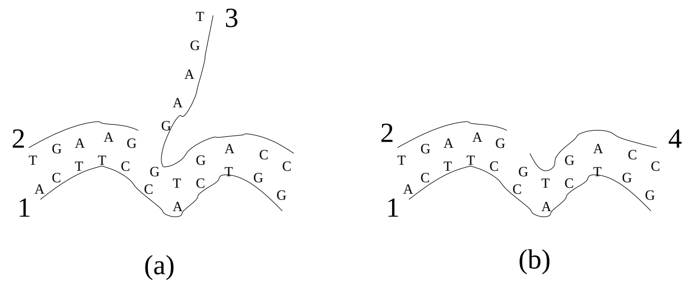


Figure 1: Illustrating strand invasion

we omit the details. In Figure 1(b), each of the incorrect strands 2 and 4 has attached to one half of strand 1, leaving no "foothold" for strand 3. Our basic idea is to design tile systems such that every correct tile that must attach to a growing assembly has an opportunity to get a foothold and invade incorrect tiles away. We call such tile systems *invadable*; a more formal definition is presented in Section 2. Invadable assemblies appear to be a good first step towards obtaining robustness in the self-assembly process.

Quite apart from their connection to strand invasion, invadable systems may well be of interest as subroutines for other mechanisms to obtain robustness – since a correct tile always has a foothold, an error can only happen from one direction at any given site which might make it easier to detect and repair.

Our Results: We observe in Section 3.1 that the constructions by Adleman *et al.* [2], as well as the one by Rothemund and Winfree [8] to assemble $n \times n$ squares are non-invadable. Winfree's tile system [11] to assemble a Sierpinski triangle, a useful fractal shape, is also non-invadable. Hence, the problem of designing invadable systems to build structures is non-trivial and interesting. We then show (section 3.2) that a constant number of tiles can be used to extend an $n \times 1$ rectangle into an $n \times n$ square using invadable assemblies at temperature 2; this can not be accomplished at temperature 1 in the non-invadable case and hence gives us hope that invadable self-assemblies can be used for additional non-trivial constructions. Our main results are the following:

1. We show (Section 4) how to construct a $K \times n$ rectangle, where $K = O(\log n / \log \log n)$, in the invadable self-assembly model at temperature 2. We use $O(\log^2 n / \log \log n)$ tiles in our construction. Our construction involves growing a counter whose height is controlled using the Chinese remainder theorem. As an intermediate step, we obtain a system that works at temperature 3, and

then convert it into a temperature 2 system using twice the number of tiles. Using results from Section 3.2, this rectangle can be converted into a square. The running time[1] in our construction is $\tilde{O}(n)$. Both the number of tiles and the assembly time are only polylogarithmic factors away from the lower bounds even without the invadability restriction. Closing the gap between the lower and upper bounds remains an important open problem.

2. We then prove (Section 5) that invadable tile systems can simulate a Turing machine, matching the computational power of unrestricted tile systems [11]. To achieve this, we first define *rectilinear* tile systems, where the assembly can grow only in one horizontal direction (either east to west or west to east) and in only one vertical direction. We show how any rectilinear tile system that has K tiles can be made invadable with only an $O(\log K)$-factor increase in the number of tiles. Since Turning machines can be simulated using a rectilinear tile system [11], invadable tile systems are also universal. As an illustrative example, we show that the Sierpinski tile system can be made invadable. All our constructions in Section 5 leave holes in the assembled structure. Avoiding the use of holes (or proving that they are necessary) remains an interesting open problem.

So far, the ideas presented in this paper have not been investigated in a laboratory setting. An alternate approach to error-correction would be to simulate an error-correction code. Winfree's lab [10] has recently used this approach to reduce the probability of misinsertions from p to p^2 when assembling a structure that resembles a Sierpinski triangle. Another approach would be to use the idea of step-wise self-assembly proposed by Reif [6]. We have made some preliminary progress in using the step-wise model; details are omitted from this version. We are optimistic that the results from our work as well as these alternate approaches would serve as useful rules of thumb for achieving efficient and robust DNA self-assembly in practice.

2 Definitions

The Tile Assembly Model: The tile assembly model [8, 2] extends the theoretical model of tiling by Wang [9] to include a mechanism for growth based on the physics of molecular self-assembly. We will present a succinct definition, with minor modifications for ease of explanation.

[1] The $\tilde{O}(n)$ notation hides polynomial factors of $\log n$.

A tile is an oriented unit square with the north, east, south and west edges labeled from some alphabet Σ of glues. For each tile t, the labels of its four edges are denoted $\sigma_N(t)$, $\sigma_E(t)$, $\sigma_S(t)$, and $\sigma_W(t)$. Sometimes we will describe a tile t as the quadruple $(\sigma_N(t), \sigma_E(t), \sigma_S(t), \sigma_W(t))$. Consider the triple $< T, g, \tau >$ where T is a finite set of tiles, $\tau \in \mathbf{Z}_{>0}$ is the *temperature*, and g is the *glue strength* function from $\Sigma \times \Sigma$ to $\mathbf{Z}_{\geq \mathbf{0}}$, where Σ is the set of glues. It is assumed that for all $x, y \in \Sigma$, $(x \neq y) \Rightarrow g(x, y) = 0$. A *configuration* is a partial function from $\mathbf{Z}^2$ to T.

Let C and D be two configurations. Suppose there exist some $t \in T$ and some $(x, y) \in \mathbf{Z}^2$ such that $(x, y) \notin Dom(C)$, $D(x, y) = t$ and $D = C$ except at (x, y). Let $f_{N,C,t}(x, y) = g(\sigma_N(t), \sigma_S(C(x, y+1))$ if $(x, y+1) \in Dom(C)$ and $f_{N,C,t}(x, y) = 0$ otherwise. Informally $f_{N,C,t}(x, y)$ is the strength of the bond between C and the north side of t. Define $f_{S,C,t}(x, y), f_{E,C,t}(x, y)$ and $f_{W,C,t}(x, y)$ similarly. Then we say that tile t is *attachable* to C at position (x, y) iff $f_{N,C,t}(x, y) + f_{S,C,t}(x, y) + f_{E,C,t}(x, y) + f_{W,C,t}(x, y) \geq \tau$, and we write $C \rightarrow_{\mathbf{T}} D$ to denote the transition from C to D in attaching a tile to C at position (x, y). Informally, $C \rightarrow_{\mathbf{T}} D$ iff D can be obtained from C by adding a tile t such that the total strength of interaction between t and C is at least τ.

A *tile system* is a quadruple $\mathbf{T} =< T, s, g, \tau >$, where T, g, τ are as above and $s \in T$ is a special tile called the "seed". We define the notion of a *derived supertile* of a tile system $\mathbf{T} =< T, s, g, \tau >$ recursively as follows:

1. The configuration Γ such that $Dom(\Gamma) = (0, 0)$ and $\Gamma(0, 0) = s$ is a derived supertile of $\mathbf{T}$, and
2. if $C \rightarrow_{\mathbf{T}} D$ and C is a supertile of $\mathbf{T}$, then D is also a derived supertile of $\mathbf{T}$.

Informally, a derived supertile is either just the seed (condition 1 above), or obtained by legal addition of a single tile to another derived supertile (condition 2). We will often omit the word "derived" in the rest of the paper, and use the terms "seed supertile" or just "seed" or s to denote the special supertile in condition 1.

A *terminal supertile* of the tile system $\mathbf{T}$ is a derived supertile A such that there is no supertile B for which $A \rightarrow_{\mathbf{T}} B$. If there is a terminal supertile A such that for any derived supertile B, $B \rightarrow^*_{\mathbf{T}} A$, we say that the tile system *uniquely produces* A. Given a tile system $\mathbf{T}$ which uniquely produces a supertile, we say that the program size complexity of the system is $|T|$ i.e. the number of tile types.

A *shape* is a finite connected subset of $\mathbf{Z}^2$. The *shape of a supertile* Γ is $Dom(\Gamma)$. A tile system $\mathbf{T}$ is said to *uniquely produce a shape* W iff it uniquely produces some supertile Γ and the shape of Γ is identical (upto translation) to W.

We will now add the notion of running time to this model. We associate with each tile $t \in T$ a nonnegative probability $P(t)$, such that $\sum_{t \in T} P(t) = 1$. We assume that the tile system has an infinite supply of each tile, and $P(t)$ models the concentration of tile t in the system. Now self-assembly of the tile system corresponds to a continuous time Markov process where the states are in a one-one correspondence with derived supertiles, and the initial state corresponds to the seed s. Suppose a single tile t can be added to supertile B to produce supertile C. Then there is a transition from state B to C in the Markov chain, and the rate of the transition is $P(t)$. Suppose the tile system produces a unique terminal supertile A_T. In the Markov chain, the time for reaching A_T from s is a random variable. The "running time" of the self-assembly process is defined as the expected value of this random variable.

Note that the Markov process modeling the self assembly process is inherently parallel. For details see [2].

Invadable Self-Assembly: Consider a tile system **T**, a supertile Γ of **T** and a tile $t \in T$ that is attachable to Γ at some position p. We say t has a *north-foothold* in Γ at p iff $f_{N,\Gamma,t}(p) > 0$ and no tile in T but t has glue $\sigma_N(t)$ on its north edge. We define *(south,west,east)-foothold* similarly.

Definition 2.1. *We say tile t has a foothold in Γ at p iff t has a north, south, east or west-foothold in Γ at p.*

Definition 2.2. *We say the attachment of t to Γ at p is safe iff t has a foothold in Γ at p and no tile in T other than t is attachable to Γ at p.*

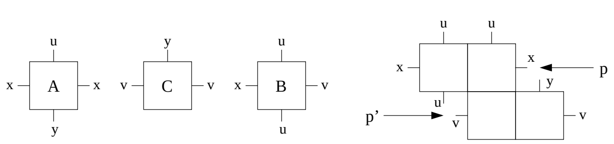


Figure 2: The attachment of A at p is safe if $\tau = 2$ since A has a south-foothold, while the attachment of B at p' is non-safe.

Figure 2 shows an example of a safe and of a non-safe attachment at temperature 2. The attachment of A to the supertile displayed in the figure at p is safe because no tile other than A has glue y on its south side, i.e. A has a south-foothold, and neither B nor C can attach at p. The attachment of B at p' is not safe because B has no foothold in the supertile at p' even though B fits perfectly there. In a lab experiment with tiles made out of DNA, it could be the case that B attaches at p though the bond will be weak because the strength of x is 1. A has a chance of forming a bond using its south glue and invade B off. For the attachment of B at p' the former reasoning is not true anymore. An "incorrect" A tile could form a weak bond on its north side while an "incorrect" C tile forms a weak bond on its east side. Then, B would not be able to invade A and C off. Hence, it seems plausible to posit that safe attachments reduce the probability of tile mis-insertions.

Definition 2.3. *A tile system **T** is invadable if and only if all attachments to all supertiles that can be assembled from a seed supertile are safe.*

Intuitively, we are saying that every possible attachment to every supertile that can be derived from the seed has to be safe. Note that in this paper we are not extending the self assembly models in [8] and [2]. Invadable tile systems form a proper subset of all the tile systems allowed in [8] and [2]. Invadable systems are interesting because they may lead to less error-prone self assembly processes.

3 Preliminaries

We first show that existing square constructions are non-invadable, and then show that invadable systems at $\tau = 2$ are strictly more powerful than regular tile systems at $\tau = 1$.

3.1 Non-invadability of existing tile systems for assembling counters

Adleman *et al.* [2] described a $\tau = 3$ tile system of size $O(1)$ that uniquely produces a rectangle on top of a seed row. The self assembly process resembles a binary counter that takes its initial value from the bottom row. A simpler $\tau = 2$ tile system that represents a binary counter was described later in [4]. These constructions are interesting because they prove that $\tau > 1$ systems can be smaller and run faster than $\tau = 1$ systems.

Figure 3(a) shows the tile system described in [4] as well as a derived supertile from a seed row representing the binary number 0001. Observe that tile labeled as "$0'$" is the only one attachable to the left of the only tile in the top row. That attachment is not safe because the tile with label "$1'$" has glue c on its east side and tile "O^*" has glue n on its south side. Hence, the tile system is not invadable. The earlier constructions [8, 2] of counters are also non-invadable, and it is not obvious whether or not an invadable counter can be created. Other interesting constructions such as the Sierpinski tile system described in [11] are also non-invadable.

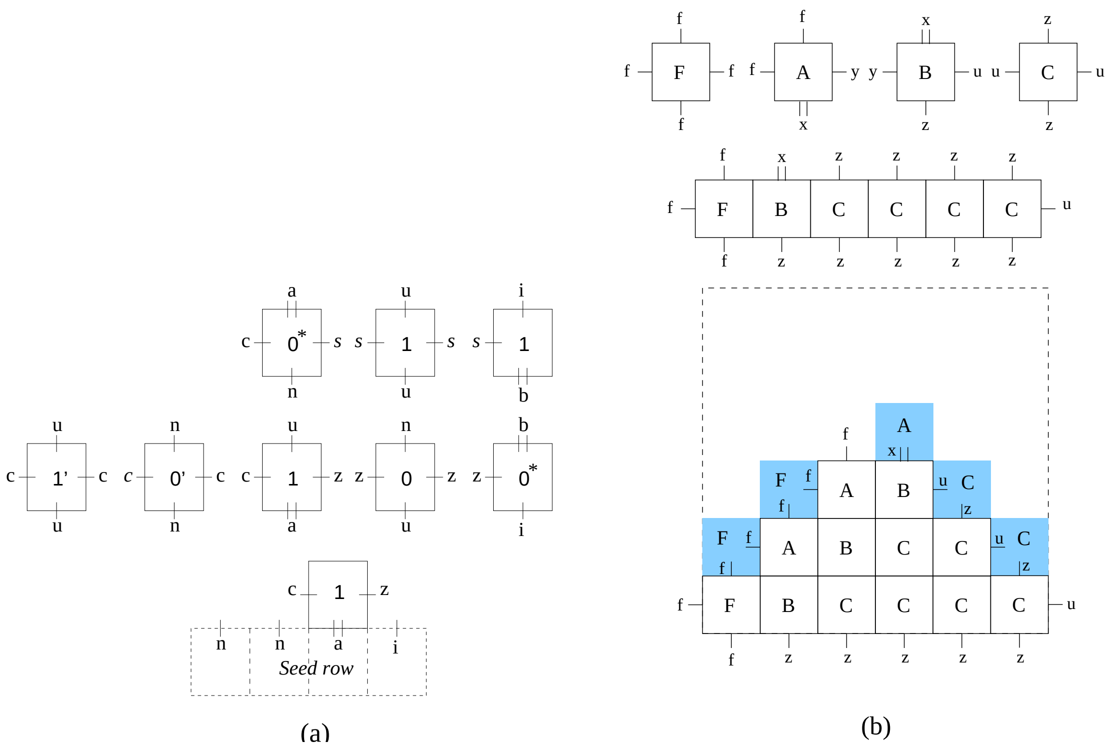


Figure 3: (a) A tile system at $\tau = 2$ that behaves as a binary counter and a derived, non-terminal supertile. (b) A tile system that solves $SQC(6)$ and a derived, non-terminal supertile. The shaded areas indicate where attachments can occur.

3.2 On the power of invadable systems First, observe that if a $\tau = 1$ system $\mathbf{T}$ uniquely produces some shape S, then $\mathbf{T}$ is invadable, because a non-safe attachment would immediately contradict the unique production of S. We will now present a problem that can be solved more efficiently in terms of both program size and time complexity with $\tau > 1$ invadable tile systems than with regular tile systems at $\tau = 1$.

The Square completion problem SQC(n): Given a positive integer n find a tile system $\mathbf{T}$, such that $\mathbf{T}$ uniquely produces an $n \times n$ square from a seed supertile whose shape is a horizontal line of length n. We are allowed to assume arbitrary glues on the north sides of tiles on the line. If $\mathbf{T}$ satisfies the above definition, then it is said to solve SQC(n).

We observe now that if a tile system $\mathbf{T}$ solves $SQC(n)$ at $\tau = 1$, then $|\mathbf{T}| = \Omega(n)$, because of the arguments presented by Rothemund [7]. Based on the results in [2], we also observe that the time complexity for producing the square from the seed in $\mathbf{T}$ is $\Omega(n^2)$ if $\tau = 1$. The next theorem gives upper bounds for time and space complexity of $\tau = 2$ invadable solutions to SQC.

THEOREM 3.1. *For all positive integers n, there exists a $\tau = 2$ invadable system $\mathbf{T}$ and a concentrations function P for $\mathbf{T}$ such that $\mathbf{T}$ solves $SQC(n)$, $|\mathbf{T}| = O(1)$ and the time to complete the square is $O(n)$.*

PROOF OUTLINE: The proof is constructive. For all n we use the same set of tiles depicted in Figure 3(b). We will refer to the tiles by the labels on their centers, so $T = \{A, B, C, F\}$. Make the set of glues equal to $\{f, u, v, x, y, z\}$. For a given n, we define the seed row s in such a way that the west-most tile is F, the tile immediately to the east of F is B and all the remaining tiles in the seed are C tiles. We can prove by induction that $\mathbf{T}$ uniquely produces the $n \times n$ square and all attachments are safe.

To prove the $O(n)$ time bound we make $P(A) = P(B) = P(C) = P(F) = 1/4$ and we analyze the time complexity using the techniques described in [2]. □

Theorem 3.1 shows that there are invadable solutions to SQC which are asymptotically faster and smaller than the fastest or smallest solutions at temperature 1. This gives us hope that additional non-trivial assemblies are possible using invadable systems without sacrificing efficiency too much. Also, the square completion problem is a useful subroutine in Section 4.

4 Invadable Assembly of $n \times n$ squares using Chinese Remaindering

We describe a tile system which assembles into an $n \times n$ square and uses only a polylogarithmic number of different tiles. The system works at temperature 3, and results in invadable self-assembly.

Our tile system first assembles a $k \times (n-k)$ rectangle (that we call the counter) where k is roughly $\log n / \log \log n$, and then completes this into an $n \times n$ square. Completing a line to obtain a square using invadable self-assembly has already been described in Section 3.2. The process for converting an $a \times b$ rectangle into an $(a+b) \times (a+b)$ square is very similar, and we will not repeat our earlier arguments.

First, we pick k distinct primes $p_1, p_2, \ldots, p_k$ such that $\Pi_{i=1}^{k} p_i \geq n$. The tile system contains a seed tile s, and $k-1$ "base" tiles $B_2, B_3, \ldots, B_k$; we also use the term B_1 to denote s. For each prime p_i, the tile system contains p_i "counter" tiles labeled $C_{i,0}, C_{i,1}, \ldots, C_{i,p_i-1}$.

The north side of the tile $C_{i,j}$ has glue $g_{i,j}$; all the $g_{i,j}$'s are distinct. The south side of tile $C_{i,j+1}$ also has glue $g_{i,j}$. Here additions are modulo p_i, so C_{i,p_i} is the same as $C_{i,0}$. The glue $g_{i,j}$ has strength 3 if $j \neq p_i - 1$ and strength 2 otherwise. All tiles $C_{i,j}$ have the same glue α of strength 1 on their west and east sides.

There is no glue on the south sides of the B_i tiles. There is no glue on the east side of tile B_1 or the west side of tile B_k. There is a glue β_i of strength 3 on the west side of tile B_i for $1 \leq i < k$. All the β_i's are distinct. The same glue β_i occurs on the east side of tile B_{i+1}. The north glue of tile B_i is the same as the north glue on tile $C_{i,0}$.

We will give an informal description of the assembly process. The final counter will be composed of k columns. The base row will have the tiles $B_1, B_2, \ldots, B_k$. For each prime p_i, there is going to be a "counter column" which will repeatedly count from 0 to $p_i - 1$ and then roll over to 0 again. Notice that the glues are defined such that in the i-th counter column, the tiles have sufficient strength to count from 0 to $p_i - 1$. But then to roll over to 0, one of the the adjacent columns needs to provide a glue of strength one. If even one tile in a row is of type $C_{i,j}$ for $j \neq p_i - 1$ then that tile can attach a tile of type $C_{i,j+1}$ to its north using the glue $g_{i,j}$ of strength 3. This provides a glue of strength 1 to the east and west, and inductively, allows an entire new row to assemble. When all tiles in a row are of type C_{i,p_i-1} then the assembly process stops. Figure 4(a) illustrates the assembly process for $k = 2, p_1 = 2, p_2 = 3$.

Since the primes are all distinct and the base tiles are designed to look like tiles $C_{i,0}$ on the north, the Chinese remainder theorem tells us that the counter will grow to have $\Pi_{i=1}^{k} p_i$ rows. The number of base tiles in the systems is k and the number of counter tiles is $\sum_{i=1}^{k} p_i$. Hence the total number of tiles is $\sum_{i=1}^{k} (p_i + 1)$.

Two points are worth noting:

1. *Invadability:* Consider the assembly process. Tile B_i can not attach in the base row till the tile to its east (i.e. tile B_{i-1}) is in place, except of course for the case $i = 1$ since B_1 is the seed. For tiles not in the base row, a tile can only attach when the tile to its south is in place. But the glue on the west sides of the base tiles is unique, and the glue on the south side of each counter tile is also unique. Hence, whenever a tile X is attachable at position P, there is at least one glue at that position where only tile X can attach. Further, it is easy to see that the remaining glues at that position can not have a total strength greater than 2.

2. *Height control:* By the Chinese remainder theorem, each vector $\langle x_1, x_2, \ldots, x_k \rangle$ occurs exactly once as the vector of labels in a row in the above construction, where $0 \leq x_i < p_i$. Let $V^{(r)} = \langle x_1^{(r)}, x_2^{(r)}, \ldots, x_k^{(r)} \rangle$ denote the vector which occurs in the r-th row, starting from the base. Thus, $V^{(1)} = \langle 0, 0, \ldots, 0 \rangle$ and $V^{(\Pi p_i)} = \langle p_1 - 1, p_2 - 1, \ldots, p_k - 1 \rangle$. In order to make a $k \times H$ square for $H \leq \Pi_{i=1}^{k} p_i$, we can make the north glue of tile B_i resemble the north glue of tile $C_{i,x_i^{(r)}}$ where $r = \Pi_{i=1}^{k} p_i - H + 1$. This would be like starting the counting from the vector $V^{(r)}$ and going to $V^{(\Pi p_i)}$.

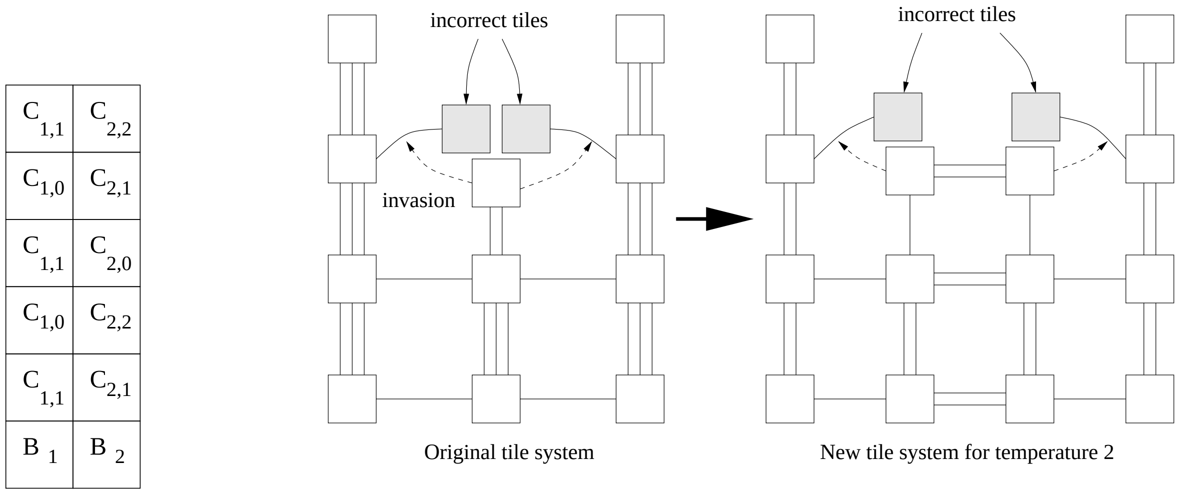
Now consider the size of the tile system. We want to choose a number k as well as distinct primes $p_1, p_2, \ldots, p_k$ such that $\Pi_{i=1}^{k} p_i \geq n$ but the number of tiles (i.e., $\sum_{i=1}^{k} (p_i + 1)$) is small.

We can choose $\log n / \log \log n$ primes between $\log n$ and $3 \log n$. Their product is at least n and their sum is at most $3 \log^2 n / \log \log n$.

Now assign the same concentration to each tile. We will use the $\tilde{O}$ notation to hide factors which are polynomial in $\log n$. Each time a tile is attachable, it takes an expected time $\tilde{O}(1)$ for it to attach. Since there are $\tilde{O}(n)$ positions in the counter, it takes $\tilde{O}(n)$ expected time to assemble the counter. The time to complete the rectangle into a square can easily be made $O(n)$ using arguments from an earlier work by Adleman *et al.* [2].

Given the tile system and the discussion above, the following theorem can be derived easily. We omit a formal proof.

Theorem 4.1. *There exists an invadable tile system with $O(\log^2 n / \log \log n)$ tiles which uniquely produces an $n \times n$ square in expected time $\tilde{O}(n)$.*



(a) (b)

Figure 4: (a) Assembly process of a 2×6 rectangle. (b) In the $\tau = 3$ invadable system, a correct tile may have to invade off two incorrect tiles whereas in the $\tau = 2$ invadable system, the correct tile has to invade off at most one incorrect tile.

In the above construction, a newly attaching tile can have competitors on both the east and the west, but only one foothold on the south. Thus, the correct tile may have to invade off two incorrect tiles (see Figure 4(b)). We now describe how to modify our construction to reduce the temperature to 2 and also eliminate the above problem in the process. The basic intuition is to split the column corresponding to each prime into two columns. Since each column only interacts with one other prime's column, each newly attaching tile can only have competitors on the east or the west, but not at both places.

Theorem 4.2. *There exists an invadable tile system with $O(\log^2 n/\log\log n)$ tiles which uniquely produces an $n \times n$ square in expected time $\tilde{O}(n)$ at temperature 2.*

Proof. Since the "square completion problem" described in Section 3.2 already uses temperature 2, we just have to demonstrate construction of a counter at temperature 2. Our proof is constructive: we use a simple variation of Chinese Remainder system described above.

For each prime p_i, we construct two columns C_{i1} and C_{i2} with the same period p_i. Namely, column C_{i1} has tiles $D_{i1,1}$, $D_{i1,2}$, ..., D_{i1,p_i}; column C_{i2} has tiles $D_{i2,1}$, $D_{i2,2}$, ..., D_{i2,p_i}. For all i and k, $D_{i1,k} = (G_{i1,k}, G_1, G_{i1,k-1}, G_2)$, $D_{i2,k} = (G_{i2,k}, G_2, G_{i2,k-1}, G_1)$. The glues $G_{i1,k}$ and $G_{i2,k}$ have strength 2 for $k = 1, 2, \ldots, p_{i-1}$. The glues $G_{i1,p_i} = G_{i1,0}$ and $G_{i2,p_i} = G_{i2,0}$ have strength 1. Glue G_2 has strength 2 and Glue G_1 has strength 1.

It is easy to see that this tile system is invadable and uniquely produces a rectangle of height $p_1 * p_2 * \ldots * p_n$ at temperature 2. We can use $O(\log n/\log\log n)$ different primes between $\log n$ and $3\log n$ as before.

5 Universality of invadable systems

In this section, we show that we can use invadable tile systems to perform universal computation. We will do this by constructing an invadable system to simulate the Sierpinski tile system described below, and then generalize this method to a large class of tile systems.

The Sierpinski tile system by Winfree [12] assembles a useful and well-known fractal shape. It consists of seven tiles. In this system, the seed tile is at the southwest corner of all derived supertiles. Another tile builds a vertical line on top of the seed and a third tile builds a horizontal line to the east of the seed. These tiles are called boundary tiles. The core computation is performed by four rule-tiles $S_{0,0}, S_{0,1}, S_{1,0}, S_{1,1}$. Each tile $S_{x,y}$ can be described by the quadruple $(x \textit{ xor } y, x \textit{ xor } y, y, x)$, where all glues are of strength 1. At temperature 2, the system grows from south and west to north and east. In each step, a rule-tile that matches with both glues on the south and west will attach and provide glues on its north and east sides. Conceptually, each rule-tile reads two inputs (i.e. the glues on its west and south sides), computes the *xor* of the inputs and outputs the result on its north and east sides. We observe that the system is not invadable since every rule-tile can be blocked by two other partially matched rule-tiles.

Lemma 5.1. *The Sierpinski tile system described above can be simulated by an invadable system.*

Proof Outline: The proof is constructive. We will use a 3 by 3 square of tiles (called a block) to simulate one individual tile in the original system. We will use coordinates to refer to positions within a block. Position $(1, 1)$ is the south-west corner, $(1, 3)$ is the south-east corner of a block and so forth. Each block uses the glues provided by two other blocks on its south and west to grow and provides glues on its east and north sides for future growth. There are four blocks $B_{00}, \ldots, B_{11}$ corresponding to the four original rule-tiles. See Figure 5 for a description of the four blocks, the glues on their boundaries, and the tiles that compose them. Note that some edges do not have any glue. In the same figure we also describe the strength of each bond. The details of the glue assignment to edges that are not part of the block boundary are omitted.

The tiles $T_{00A}, T_{01A}, T_{10A}, T_{11A}$ have the function of the original rule-tiles, i.e. computing *xor* of the inputs, but there are two locations where these tiles can attach. T_{00A} and T_{01A} can only attach at position $(1, 1)$ in a block while T_{10A} and T_{11A} can only attach at position $(2, 2)$. Our tile system is designed in such a way that in the final assembly, only one of two these locations will have a tile while the other position will be empty. Once one of these four tiles is added to the assembly, its north and east sides of will provide strength 2 glues that allow tiles to attach at positions $(1, 3)$,$(2, 3)$,$(3, 3)$,$(3, 2)$ and $(3, 1)$. These tiles provide glues that allow new blocks to start growing.

We will now describe how B_{01} grows. B_{01} can only grow on the north side of either B_{01} or B_{10} and to the east of either B_{00} or B_{11}. The first tile to be attached is either T_{01A} at $(1, 1)$ or T_{1S} at $(2, 1)$. After T_{10A} attaches, T_{01B}... T_{01G} can be attached in that order. The analysis for B_{10} is similar and hence we omit it.

The growth of B_{11} is as follows: B_{11} can only grow on the north side of either B_{01} or B_{10} and to the east of either B_{10} or B_{10}. The first tile to be attached is either T_{1W} at $(1, 2)$ or T_{1S} at $(2, 1)$. After both T_{1W} and T_{1S} are in place, T_{11A} becomes attach able. After T_{11A} attaches, T_{11B}...T_{11G} can be attached completing the block. The analysis for B_{00} is similar.

Note that in the final assembly, all B_{11} and B_{00} blocks will have their respective $(1, 1)$ positions empty. All B_{01} and B_{10} blocks will have their respective $(2, 2)$ positions empty.

In this construction, we can show that each block can actually perform the function of one tile in the original Sierpinski tile system. It is easy to show that we can use 7 tiles (one at the corner, three tiles on each side) to replace the boundary tiles to get the same pattern.

In our system, T_{0S}, T_{1S} and T_{1W} are uniquely matched with one of the other blocks; $T_{00A}, T_{01A}, T_{10A}, T_{11A}$ are uniquely matched with the glue on its south side. All the output tiles are uniquely determined by the tile attached at $(1, 1)$ or $(2, 2)$, so the tile system is invadable. □

We will now define a class of tile systems that can be simulated with invadable systems. A temperature-2 tile system is an *SW-system* iff all of the following are true:

1. The strength of all glues is 1.
2. Each tile can be uniquely identified by its south and west glues, i.e., any two different tiles must either have different south glues, or different west glues, or both.
3. For all positions p for all derivable supertiles of the system, if a tile t is attachable at p then positions $p + (-1, 0)$ and $p + (0, -1)$ are occupied (i.e. there are already tiles attached to the immediate south and the immediate west of the newly attaching tile) while $p + (1, 0)$ and $p + (0, 1)$ are empty.

Note that since in an SW-system a tile t can be identified by the ordered pair $(\sigma_S(t), \sigma_W(t))$, there is a function $F : \Sigma^2 \to T$ such that $t = F(\sigma_S(t), \sigma_W(t))$ for all tiles t in the tile set. For all (g_1, g_2) in $Dom(F)$, we define $N(g_1, g_2) = \sigma_N(F(g_1, g_2))$, and $E(g_1, g_2) = \sigma_E(F(g_1, g_2))$. Conceptually, the attachment of a tile to a supertile can be viewed as evaluating N and E.

We can similarly define SE, NW, and NE systems. A tile system is said to be *rectilinear* if it is either a SW, a SE, a NW, or a NE system.

Theorem 5.1. *A rectilinear tile system with n tiles can be simulated by an invadable system using $O(n \log n)$ tiles.*

Proof Outline: Without loss of generality, we will assume that the rectilinear system is a SW system. Let G_S be the set of all glues used on south sides of tiles, and let G_W be the set of all glues used on west sides of tiles. Call $g_1, g_2, \ldots, g_r$ the glues in G_S and call $1, 2, \ldots, k$ the glues in G_W. Note that $k \leq n$ because there are n tiles and, therefore, there are at most n glues in G_W.

In this case, we can build our system in a way similar to the Sierpinski tile system. The construction is shown in Figure 6. We call *macro-blocks* the structures we will use to simulate tiles in the SW-system . In each macro-block, there are $\lfloor \log k \rfloor + 1$ blocks. Each block is a 3×3 square that has the same structure as the blocks described in the proof of Lemma 5.1. Consider a tile t in the SW-system with glue g_i on its south side and

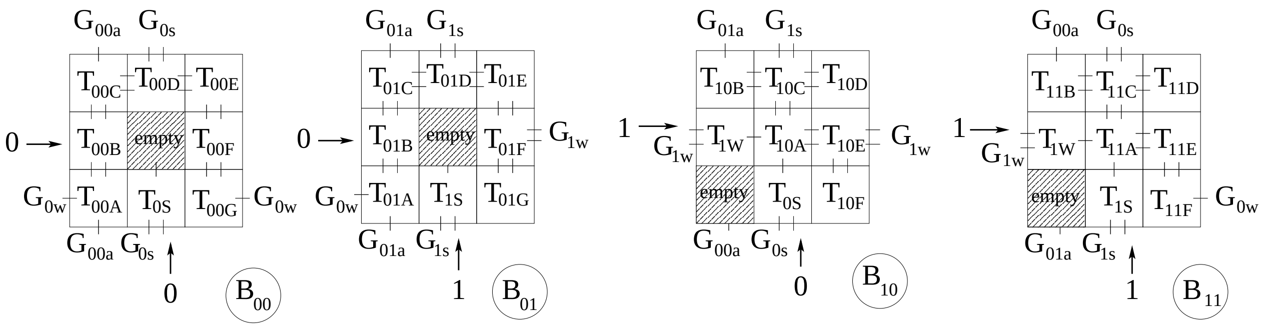


Figure 5: Blocks to simulate the Sierpinski tile system.

glue j on its west side. Conceptually, the corresponding macro-block will read j *in binary* from the west and g_i from its south side. The macro block will output $N(g_i, j)$ on its north side and $E(g_i, j)$ on its east side, also in binary. The east-most column of the macro-block will be the last to be assembled.

Neglecting the east-most column, the blocks are built from south to north. Consider the assembly of the m^{th} block counting from the south. Assume $j = (j_1 j_2 \ldots j_{\lfloor \log k \rfloor + 1})_2$. As in the Sierpinski triangle invadable system, position $(1, 1)$ or $(2, 2)$ will be filled with a tile that will trigger the assembly of the boundaries of the block. Call $T_{i,j_1 j_2 \ldots j_m}$ the tile that attaches at position $(1, 1)$ or $(2, 2)$ in the block we are considering. $T_{i,j_1 j_2 \ldots j_m}$ attaches at $(1, 1)$ iff j_m is zero, while it attaches at $(2, 2)$ iff j_m is 1. The blocks will contain a hole, as before. Conceptually, $T_{i,j_1 j_2 \ldots j_m}$ represents a partial computation of $N(g_i, j)$ and $E(g_i, j)$, after reading m bits of j. In the north-most block, the tile $T_{i,j_1 j_2 \ldots j_{\lfloor \log k \rfloor + 1}}$ will be attached and force the output to be $N(g_i, j)$ and trigger the construction of the east-most column in the macro-block representing $E(g_i, j)$.

There are n possible macro-blocks, and each macro-block is of size $O(\log k) = O(\log n)$. So, the system we constructed uses $O(n \log n)$ tiles. □

THEOREM 5.2. *There exists a $\tau = 2$ invadable system* **T** *that can perform universal computation.*

PROOF OUTLINE: Winfree[12] defined a self assembly tile system that simulates a block cellular automata (and hence a Turing machine). His construction uses a rectilinear tile system. Invoking theorem 5.1, it is immediate that a Turing machine can be simulated by an invadable tile system. □

Acknowledgement

We would like to thank Len Adleman for useful discussions.

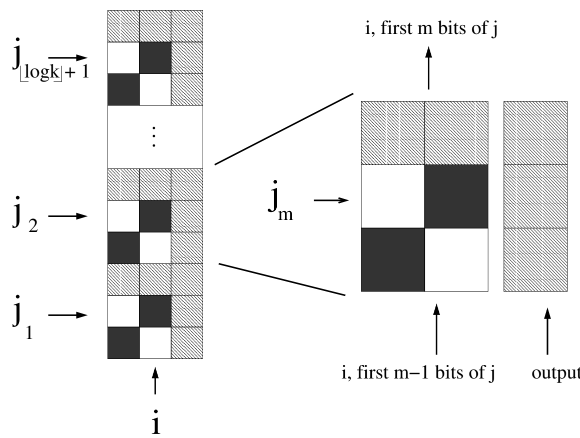


Figure 6: Making a general rectilinear system invadable.

References

[1] L. Adleman. Towards a mathematical theory of self-assembly. Technical Report 00-722, Department of Computer Science, University of Southern California, 2000.

[2] L. Adleman, Q. Cheng, A. Goel, and M.-D. Huang. Running time and program size for self-assembled squares. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 740–748. ACM Press, 2001.

[3] L. Adleman, Q. Cheng, A. Goel, M.-D. Huang, D. Kempe, P. Moisset de Espans, and P. Rothemund. Combinatorial optimization problems in self-assembly. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 23–32. ACM Press, 2002.

[4] Q. Cheng and P. Moisset de Espanes. Resolving two open problems in the self-assembly of squares. Technical Report 03-793, University of Southern California, 2003.

[5] M. Lagoudakis and T. LaBean. 2d dna self-assembly for satisfiability. In *Proceedings of the 5th DIMACS Workshop on DNA Based Computers in DIMACS Se-*

ries in Discrete Mathematics and Theoretical Computer Science, volume 54. MIT: Cambridge, 1999.

[6] J. Reif. Local parallel biomolecular computation. In H. Rubin, editor, *Third Annual DIMACS Workshop on DNA Based Computers, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 1998.

[7] P. Rothemund. *Theory and Experiments in Algorithmic Self-Assembly*. PhD thesis, University of Southern California, 2001.

[8] P. Rothemund and E. Winfree. The program-size complexity of self-assembled squares (extended abstract). In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 459–468. ACM Press, 2000.

[9] H. Wang. Proving theorems by pattern recognition ii. Bell Systems Technical Journal, 1961. 40:1-42.

[10] E. Winfree. Personal communication.

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

[12] E. Winfree, F. Liu, L. Wenzler, and N. Seeman. Design and self-assembly of two-dimensional dna crystals, 6 pages. *Nature*, (394):539–544, Aug 1998.

[13] E. Winfree, X. Yang, and N. Seeman. Universal computation via self-assembly of dna: Some theory and experiments. In *Proceedings of the Second Annual Meeting on DNA Based Computers*. Princeton University, June 1996.

[14] B. Yurke, A. Turberfield, A. Mills Jr, F. Simmel, and J. Neumann. A dna-fuelled molecular machine made of dna. *Nature*, (406):605–608, Aug 2000.