The Program-Size Complexity of Self-Assembled Squares

[Extended Abstract, Feb. 23, 2000]

Paul W. K. Rothemund
Dept. of Computer Science
University of Southern California
pwkr@cs.usc.edu

Erik Winfree
Dept. of Computer Science and CNS
California Institute of Technology
winfree@caltech.edu

ABSTRACT

Molecular self-assembly gives rise to a great diversity of complex forms, from crystals and DNA helices to microtubules and holoenzymes. We study a formal model of pseudocrystalline self-assembly, called the Tile Assembly Model, in which a tile may be added to the growing object when the total interaction strength with its neighbors exceeds a parameter $\mathcal{T}$. This model has been shown to be Turing-universal. Thus, self-assembled objects can be studied from the point of view of computational complexity. Here, we define the program size complexity of an $N\times N$ square to be the minimum number of distinct tiles required to self-assemble the square and no other objects. We study this complexity under the Tile Assembly Model and find a dramatic decrease in complexity, from N^2 tiles to $O(\log N)$ tiles, as $\mathcal{T}$ is increased from 1 (where bonding is noncooperative) to 2 (allowing cooperative bonding). Further, we find that the size of the largest square uniquely produced by a set of n tiles grows faster than any computable function.

1. INTRODUCTION

The spontaneous self-organization of complicated structures in natural systems has long fascinated physical scientists. They ask, "How should order be defined for such structures?" and, "How are such structures generated?" It is now clear that computational mechanisms play an important role in understanding natural self-organization, at least in biological systems: algorithms control the generation of order. Research in DNA and molecular computation [Adleman, 1994] has established that universal computation can be performed in biochemical systems, such as enzymatic (ribosome-like) modification or translation of a heteropolymer [Bennett, 1982; Kurtz et al., 1997], signal-transduction cascades [Hjelmfelt and Ross, 1995; Magnasco, 1997], and the self-assembly of protein or DNA into supramolecular structures [Radin, 1991; Winfree, 1996]. How widespread is influence of computational mechanisms in the generation of order – does it spread beyond the biological domain?

STOC '00 Portland, Oregon USA

For most of this century, order in self-assembled chemical systems was thought to be well understood. Order was synonymous with periodic order – the order of crystals. The term crystal was reserved for materials characterized by one of the 230 space groups; everything else was described as disordered, amorphous, or glassy. The discovery of quasicrystalline materials[Schectman et al., 1984], with their "forbidden" five-fold symmetry shattered this monopoly but left a vacuum–what is order if not periodic? One answer is to define crystal as "a structure with an essentially discrete diffraction pattern"[Senechal, 1995]. This patch for the existing framework includes quasicrystals, but leaves little room for still more exotic structures that may lurk undiscovered and excludes altogether biological materials that have complex order. These concerns have led the crystallographer Alan Mackay to propose that a "generalized crystallography" might define order[Mackay, 1995] using computer programs and cellular automata.

Such an algorithmic framework for studying self-assembly is attractive for two reasons. First, because of Church's thesis, we expect that computer programs will be able to capture all of the complex behaviour of self-assembly – no more complicated theory will be required. Second, such a framework will allow principles of computer science to be translated into statements about the physical world. For example, the self-assembly of DNA structures may be mapped naturally onto the languages of the Chomsky Hierarchy [Winfree et al., 1998b].

Here, we are interested in studying the self-assembly of objects from the point of view of computational complexity. Standard complexity measures in computer science are based on time, space, program size, and decidability. To study the time complexity of self-assembly, Leonard Adleman has proposed a model that emphasizes counting time steps during the self-assembly of a single copy of each of a finite number of tiles into the final structure. He has used this model to analyze the self-assembly of N-long linear polymers [Adleman, 2000]. Adleman has also asked, "What is the complexity of generating an $N\times N$ square by self-assembly?" Here, we answer this question for program-size complexity under the Tile Assembly Model, where self-assembly occurs in the presence of an infinite supply of a finite number of tile types.

The Tile Assembly Model is a formal model for the self-assembly of molecules, such as protein or DNA, constrained to self-assemble on a square lattice; *i.e.* it is a model of pseudo-crystalline growth. The model is an extension of the theory of tiling by Wang tiles [Wang, 1961] to include a specific mechanism for growth based on the physics of molecular self-assembly. A "program" consists of a finite set of unit square tiles with colored sides (each available in an unlimited number of copies). Each color represents a type of molecular binding domain, and thus each color has an associated "binding strength," which in our model must be an integer. Starting from a chosen seed tile, growth occurs by addition of single tiles. (The growth of crystals by monomer addition, as opposed to merging of crystal fragments, is a common assumption in studies of crystal growth [Markov, 1995]; large defect-free crystals are not observed under physical conditions where growth occurs by aggregation of small fragments.) Tiles bind a growing assembly only if their binding interactions are of sufficient strength, as determined by the "temperature" parameter $\mathcal{T}$.

$\mathcal{T}$ measures the "cooperativity" of the binding interactions. At $\mathcal{T} = 1$, any binding interaction of strength 1 or greater is strong enough, by itself, to hold a tile in place. This lack of cooperativity appears to go hand-in-hand with a lack of computational power. At $\mathcal{T} = 2$, however, single strength-1 interactions are too weak to hold a new tile in place; at least two strength-1 bonds must cooperate for a tile to be added to an assembly. Under $\mathcal{T} = 2$ conditions it has been shown that one-dimensional cellular automata can be simulated; hence $\mathcal{T} = 2$ self-assembly is universal [Winfree, 1996]. It is interesting to observe that cooperative effects play a major role in gene regulation [Ptashne, 1992] and many other biological systems.

Branched DNA molecules [Seeman, 1998] provide a direct physical motivation for the Tile Assembly Model. DNA double-crossover molecules, each bearing four "sticky ends" analogous to the four sides of a Wang tile, have been designed to self-assemble into a periodic two dimensional lattice [Winfree et al., 1998a]. The binding interactions between double-crossover molecules may be redesigned by changing the base sequence oof their sticky ends, thus allowing arbitrary sets of molecular Wang tiles to be investigated in the laboratory. From a physically-based stochastic model of such a system, the Tile Assembly Model is obtained in the limit of strong binding domains and low monomer concentrations [Winfree, 1998].

Macroscopic systems for 2D self-assembly based on lateral capillary forces [Hosokawa et al., 1996; Bowden et al., 1997; Bowden et al., 1999; Rothemund, 2000] provide additional motivation for the Tile Assembly Model. In these systems millimeter-scale plastic tiles float at an interface between hydrophobic and hydrophilic liquids (*e.g.*, oil and water) and self-assemble into lattices as the system is agitated on a shaker. Binding interactions between tiles are specified by sequences of hydrophilic and hydrophobic patches applied to the edges of tiles; when sequences match, capillary forces mediate bonds between tiles. Tile sets with up to four distinct Wang tiles have been created by this method [Rothemund, 2000]. Analogies between such systems and molecular self-assembly are not yet quantitative, but it has been observed that the frequency of shaking acts similarly to temperature and that dimers bind cooperatively to lattices. Thus cooperative $\mathcal{T} = 2$ assembly may be possible in a capillary force-based system.

It is straightforward to restrict the Tile Assembly Model to 1D, or to extend it to 3D. However, the 1D case allows no interesting computation to be performed; it is easy to see that to produce a 1D line of N tiles requires N tiles for all $\mathcal{T} > 0$. This result exactly parallels the decidability the 1D tiling problem; the 2D tiling problem, inc contrast, is undecidable [Berger, 1966]. At the other extreme, it seems unlikely that 3D allows for phenomena fundamentally different from 2D, since universal computation is already possible in 2D.

2. A MODEL OF SELF-ASSEMBLY

Our discussion of the Tile Assembly Model will make use of the following definitions. $\mathbb{N}$ is the set of natural numbers $\{0, 1, 2, \ldots\}$, $\mathbb{Z} = \mathbb{N} \cup -\mathbb{N}$ is the set of integers, and $\mathbb{R}$ is the set of real numbers. We will be working the two-dimensional grid of integer positions, $\mathbb{Z} \times \mathbb{Z}$. The directions, $\mathcal{D} = \{N, E, S, W\}$, will be used as functions from $\mathbb{Z}\times\mathbb{Z}$ to $\mathbb{Z}\times\mathbb{Z}$: $N(x, y) = (x, y+1)$, $E(x, y) = (x+1, y)$, $S(x, y) = (x, y-1)$, and $W(x, y) = (x-1, y)$. We say that (x, y) and (x', y') are neighbors if $(x', y') \in \{N(x, y), E(x, y), S(x, y), W(x, y)\}$. Note that $E^{-1} = W$, and $N^{-1} = S$.

A **partially ordered set** (poset) $(S, \leq)$ is a set S and a reflexive, transitive, antisymmetric relation $\leq$. If $m \leq a$ and $m \leq b$ and $\forall c \in S, [c \leq a \text{ and } c \leq b] \implies c \leq m$, then m is called the **meet** of a and b. If $a \leq j$ and $b \leq j$ and $\forall c \in S, [a \leq c \text{ and } b \leq c] \implies j \leq c$, then j is called the **join** of a and b. If all pairs a, b have both a meet and a join, then $(S, \leq)$ is called a **lattice**.

A (Wang) **tile** over Σ is a unit square where each side is colored from the set Σ of **binding domains**; formally, a tile t is a 4-tuple $(\sigma_N, \sigma_E, \sigma_S, \sigma_W) \in \Sigma^4$ indicating the binding domains on the north, east, south, and west sides. For $D \in \mathcal{D}$, we write $bd_D(t)$ to refer to the binding domain of the respective side of tile t. According to this definition, tiles may not be rotated; $(\sigma_N, \sigma_E, \sigma_S, \sigma_W) \neq (\sigma_W, \sigma_N, \sigma_E, \sigma_S)$. A special binding domain *null* represents a non-interaction, and the special tile $empty = (null, null, null, null)$ is used to represent the absense of any other tile.

The binding domains determine the interaction between tiles; that is, when two tiles may be placed next to each other. A function $g : \Sigma \times \Sigma \to \mathbb{R}$, where $null \in \Sigma$, is a **strength function** if $\forall \sigma, \sigma' \in \Sigma$, $g(\sigma, \sigma') = g(\sigma', \sigma)$ and $g(null, \sigma) = 0$. Two tiles that abut on sides labelled σ and σ' bind with strength $g(\sigma, \sigma')$, as discussed below. Here, we will only consider g such that mismatched sides have no interaction strength and matching sides have positive strengths given in integral units, in which case the strength of a side labeled by σ is $\hat{g}(\sigma) \in \mathbb{N}$ and $g(\sigma, \sigma') = \begin{cases} \hat{g}(\sigma) & \text{if } \sigma = \sigma' \\ 0 & \text{otherwise.} \end{cases}$

Let T be a set of tiles containing the special tile *empty*. A **configuration** of T is a function $A : \mathbb{Z} \times \mathbb{Z} \to T$. We write $(x, y) \in A$ iff $A(x, y) \neq empty$. For $D \in \mathcal{D}$, we say the tiles at (x, y) and $D(x, y)$ **bind** to each other with strength

$$g_D^A(x, y) = g(bd_D(A(x, y)), bd_{D^{-1}}(A(D(x, y)))).$$

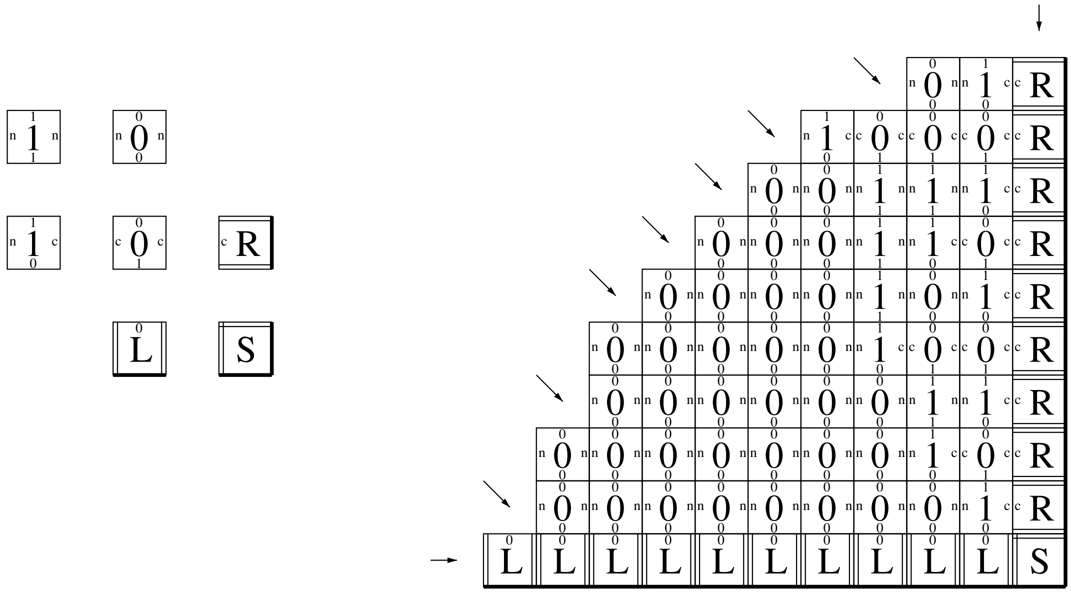

Figure 1: Simulating a binary counter with self-assembly. At left, a set T of seven tiles is depicted. In this figure and all figures that follow thick sides have strength 0, thin sides have strength 1, and double-lined sides have strength 2. At right, an assembly produced by $\mathbf{T} = \langle T, \{S\}, g, 2\rangle$ is shown. The assembly is not terminal and arrows indicate positions at which it may grow.

If $g_D^A(x,y) > 0$, then the tiles **make a bond**. If t is a tile, $A_t^{(x,y)}$ is the configuration such that $A_t^{(x,y)}(x,y) = t$ and all other sites are empty. $A_{empty}^{(0,0)}$ is called the **empty** configuration.

Addition of configurations A and B is defined by $C = A + B$ where

$$C(x,y) = \begin{cases} A(x,y) & \text{if } B(x,y) = empty \\ B(x,y) & \text{if } A(x,y) = empty \\ \infty & \text{otherwise.} \end{cases}$$

Note that C is not necessarily a configuration, because C might contain ∞ values.

Union of configurations A and B is defined by $C = A \cup B$ where

$$C(x,y) = \begin{cases} A(x,y) & \text{if } A(x,y) = B(x,y) \text{ or } B(x,y) = empty \\ B(x,y) & \text{if } A(x,y) = B(x,y) \text{ or } A(x,y) = empty \\ \infty & \text{otherwise.} \end{cases}$$

Note that C is not a configuration iff there is a site (x,y) s.t $A(x,y)$ and $B(x,y)$ are distinct non-empty tiles.

Intersection of configurations A and B is defined by $C = A \cap B$ where

$$C(x,y) = \begin{cases} A(x,y) & \text{if } A(x,y) = B(x,y) \\ empty & \text{if } A(x,y) = empty \text{ or } B(x,y) = empty \\ \infty & \text{otherwise.} \end{cases}$$

Note that C is not a configuration iff there is a site (x,y) s.t $A(x,y)$ and $B(x,y)$ are distinct non-empty tiles.

The **free energy** of a configuration C is the sum of all interaction strengths between tiles (in contrast to standard usage in chemistry, favorable interactions are given by positive numbers):

$$G(C) = \frac{1}{2} \sum_{x,y\in\mathbb{Z}} \sum_{D\in\mathcal{D}} g_D^C(x,y).$$

The **temperature** $\mathcal{T}$ gives the minimal interaction strength required to overcome thermal disruption. A configuration C is a **$\mathcal{T}$-stable assembly** if for all non-empty configurations A and B such that $C = A + B$, $G(C) \geq G(A) + G(B) + \mathcal{T}$. That is, a $\mathcal{T}$-stable assembly cannot fall apart into two pieces without decreasing the total G by $\mathcal{T}$ or more. Note that for $\mathcal{T} > 0$, a $\mathcal{T}$-stable assembly must contain a single connected component. When $\mathcal{T}$ is understood, we simply say that C is an assembly.

A **tile system T** is specified by the quadruple $\langle T, S, g, \mathcal{T}\rangle$, where T is a finite set of tiles containing *empty*, S is a set of $\mathcal{T}$-stable **seed assemblies**, g is a strength function, and $\mathcal{T} \geq 0$ is the temperature. In this paper, we consider only $|S| = 1$, where $S = A_s^{(0,0)}$ for some **seed tile** s.

Self-assembly is defined by a relation between configurations: $A \rightarrow_{\mathbf{T}} B$ if there exists a tile $t \in T$ and a site (x,y) such that $B = A + A_t^{(x,y)}$ and B is $\mathcal{T}$-stable. Since $G(A_t^{(x,y)}) = 0$, $G(B) \geq G(A) + \mathcal{T}$; *i.e.*, a tile may be added to an assembly if the summed strength of its interactions with its neighbors exceeds a threshold set by the temperature. In particular, at $\mathcal{T} = 1$, a tile may be added if it makes *any* bond to a neighbor, whereas at $\mathcal{T} = 2$, to be added the tile must either make two weak bonds or a single strong bond. $\rightarrow_{\mathbf{T}}^*$ is the reflexive transitive closure of $\rightarrow_{\mathbf{T}}$.

The tile system defines a partially ordered set, the **produced** assemblies $Prod(\mathbf{T})$, where:

$$Prod(\mathbf{T}) = \{A \text{ s.t. } \exists s \in S \text{ s.t. } s \rightarrow^*_{\mathbf{T}} A\}$$

and

$$A \leq B \text{ iff } A \rightarrow^*_{\mathbf{T}} B.$$

Another set, the **terminal** assemblies $Term(\mathbf{T})$, is defined as the maximal elements of $Prod(\mathbf{T})$:

$$Term(\mathbf{T}) = \{A \in Prod(\mathbf{T}) \text{ s.t. } \nexists B \text{ s.t. } A < B\}.$$

The produced assemblies include intermediate products of the self-assembly process, whereas the terminal assemblies are just the end products, and may be considered the "output." If

$$A \in Prod(\mathbf{T}) \implies \exists B \in Term(\mathbf{T}) \text{ s.t. } A \rightarrow^*_{\mathbf{T}} B$$

then $\mathbf{T}$ is said to be **haltable**, in the sense that every path of self-assembly *can* eventually terminate. If $\mathbf{T}$ is haltable and $Term(\mathbf{T})$ is finite, $\mathbf{T}$ is said to be **halting** in the sense that every path of self-assembly *does* eventually terminate. A halting tile system **uniquely produces** C if $Term(\mathbf{T}) = \{C\}$. Note that if a tile system uniquely produces C then $Prod(\mathbf{T})$ is a lattice: the join of A and B is $A \cup B$, and the meet of A and B is $\max\{C' \in Prod(\mathbf{T}) \text{ s.t. } C' \leq (A \cap B)\}$. In general, if $Prod(\mathbf{T})$ is a lattice, we say that $\mathbf{T}$ produces a **unique pattern** – $\mathbf{T}$ need not be halting nor even haltable.

The universality of the Tile Assembly Model follows from an elaboration of the ideas used to prove the undecidability of the origin- and diagonal-constrained tiling problems [Wang, 1963; Winfree, 1998]. In this construction, the perimeter of produced assemblies encodes the state of the Turing machine. Tile additions change the information exposed on the perimeter, effecting the state transitions. Thus, information computed as by a Turing machine can direct the growth of the assembly, and thus direct complex pattern formation.

As an example, consider the tile system of Figure 1, consisting of four rule tiles with strength-1 binding domains, two border tiles with strength-1 and 2 binding domains, and one seed tile with strength-2 binding domains. At $\mathcal{T} = 2$, these tiles count in binary; the n^{th} row above the origin represents the binary integer n. This self-assembly "program" is analogous to an infinite loop – there are no terminal assemblies. The reader is encouraged to start with the seed tile S and to verify that a unique pattern is produced: *i.e.* $Prod(\mathbf{T})$ is a lattice. Rule tiles may be added only if both their eastern and southern neighbors are already in place, and there is a unique rule tile for each possible pair of binding domains the neighbors could present; furthermore, the property that only northern and western sides are exposed in the assembly is preserved from step to step. For the same tile set at $\mathcal{T} = 1$, the order of self-assembly is not similarly constrained; tiles may be added even when one of two neighbors is a mismatch, and thus many disordered assemblies are produced.

3. COMPLEXITY OF SELF-ASSEMBLY

In this section we will be measuring program-size complexity using asymptotic notation. All functions will be from $\mathbb{N} \rightarrow \mathbb{N}$. A function $f(n)$ is **non-decreasing** iff $\forall n,\ f(n) \leq f(n+1)$. A function $f(n)$ is **unbounded** iff $\forall c,\ \exists n$ s.t. $f(n) \geq c$. We say $f(n) = O(g(n))$ iff $\exists c, n_0$ s.t. $\forall n > n_0,\ f(n) \leq cg(n)$. We say $f(n) = \Omega(g(n))$ iff $\exists c, n_0$ s.t. $\forall n > n_0,\ f(n) \geq cg(n)$. We assert proposition $P(n)$ **infinitely often** iff $\forall n_0 > 0, \exists n > n_0$ s.t. $P(n)$. Define $O_{i.o.}$ ("big-O infinitely often") such that $f(n) = O_{i.o.}(g(n))$ iff $\exists c$ s.t. $f(n) \leq cg(n)$ infinitely often. We assert proposition $P(n)$ **for almost all** n iff $lim_{n_0 \rightarrow \infty} \frac{|\{1 \leq n \leq n_0 \text{ s.t. } P(n)\}|}{n_0} = 1$. Define $\Omega_{a.a.}$ ("big-Ω almost always") such that $f(n) = \Omega_{a.a.}(g(n))$ iff $\exists c$ s.t. $f(n) \geq cg(n)$ for almost all n.

We can now formally describe the program-size complexity of an $N{\times}N$ square. An assembly A is an $N{\times}N$ **square** if there exists a site (x_0, y_0) such that $(x, y) \in A$ iff $x \geq x_0$ and $x < x_0 + N$ and $y \geq y_0$ and $y < y_0 + N$. In other words, the choice of tiles may be arbitrary, so long as they're there. Square A is a **full square** if for all $(x, y), (x', y') \in A$ such that (x, y) and (x', y') are neighbors, (x, y) and (x', y') bind with non-zero strength. In other words, every adjacent pair of tiles must have non-zero interaction strength. We are interested in which squares can be self-assembled by tile systems:

$$\begin{aligned} Sq^{\mathcal{T}} = \{(N, n) \in \mathbb{N} \times \mathbb{N} \text{ s.t. } & \text{there exists a tile system} \\ & \mathbf{T} = \langle T, \{s\}, g, \mathcal{T}\rangle, |T| = n + 1, \\ & \text{and } \mathbf{T} \text{ uniquely produces an } N{\times}N \text{ full square } \}. \end{aligned}$$

We define the program size complexity $K^{\mathcal{T}}_{SA}(N)$ of a square to be the minimum number of distinct non-empty tiles required to uniquely produce the square – physically the number of distinct types of molecules that must be prepared.

$$K^{\mathcal{T}}_{SA}(N) = \min\{n \text{ s.t. } (N, n) \in Sq^{\mathcal{T}}\}$$

Our investigations rely on several constructions. We need an easy way to verify that these constructions do indeed *uniquely* produce the target structure. For each construction, the argument is an elaboration of the argument given for the binary counter tiles, only now an assembly may have more than one diagonal growth front. Specifically, the property that is preserved from step to step is that the assembly is "stop-sign"-shaped: the orientations of the exposed sides along the (clockwise) perimeter are of the form $N^*\{N, E\}^*E^*\{E, S\}^*S^*\{S, W\}^*W^*\{W, N\}^*$. These arguments rely on showing that there is exactly one strength-2 bond joining each row and each column.

We begin by studying $K^{\mathcal{T}}_{SA}(N)$ for $\mathcal{T} = 1$ and obtain the following theorem:

THEOREM 1. $K^1_{SA}(N) = N^2$.

PROOF. To show $K^1_{SA}(N) \leq N^2$, we construct N^2 tiles, one for each position in the square, with a unique strength-1 binding domain for each adjacent pair of tiles as in Figure 2. To show $K^1_{SA}(N) \geq N^2$, suppose a tile set T with $|T| < N^2$ produces an $N{\times}N$ full square A (Figure 3). Then some tile i is present at two sites in A, say (x_1, y_1) and (x_2, y_2). Let L be the "L"-shaped (or possibly linear) assembly consisting only of the tiles at $(x_1, y_1) \ldots (x_2, y_1) \ldots (x_2, y_2)$; let L^1 be

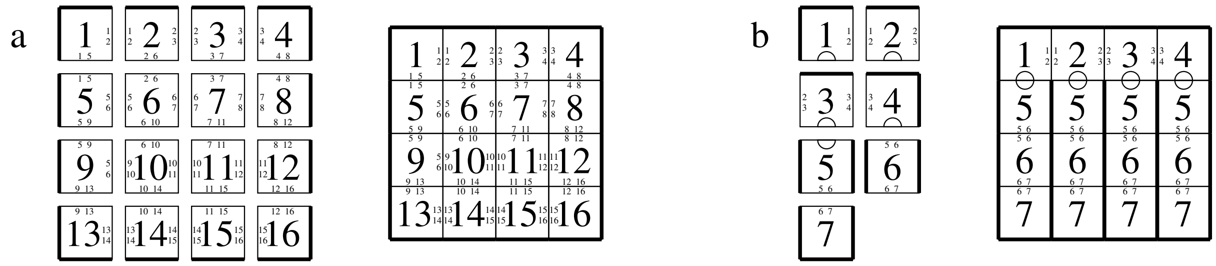

Figure 2: Formation of squares at $\mathcal{T} = 1$. (a) $N^2 = 16$ tiles with unique side labels uniquely produce a terminal 4×4 full square at $\mathcal{T} = 1$. (b) $2N - 1 = 7$ tiles uniquely produce a 4×4 square (but this is not a full square since thick sides have strength 0). Except for the sides labeled with a circle, each interacting pair of tiles share a unique side label. This comb-like construction is conjectured to be minimal for $N \times N$ squares assembled at $\mathcal{T} = 1$.

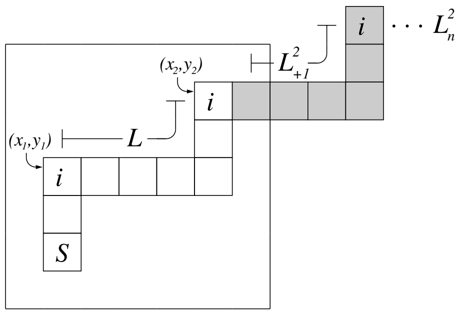

Figure 3: No $\mathcal{T} = 1$ tile system with fewer than N^2 tiles can uniquely produce an $N \times N$ square. A full $N \times N$ square with fewer than N^2 tiles must have some tile i present at two sites. Consider the assembly R (the white tiles) which includes an assembly L (bounded by the tiles i), the seed tile S, and a tile that connects the seed tile to L. R can be extended indefinitely with the addition of translated segments of L (*e.g.* L^2_{+1} shown in gray).

the assembly such that $L^1 + (x_2, y_2) = L$; let L^2 be the assembly such that $(x_1, y_1) + L^2 = L$; let $L^k_n(x, y) = L^k(x + n * (x_2 - x_1), y + n * (y_2 - y_1))$ be a translated version of L^k for $k = 1, 2$; and let R consist of L, S, and the fewest tiles in A required to connect S to L. Because R is contained in A and A is a full square, all adjacent pairs of tiles interact on a strength-(at least)-1 side, and therefore $S \rightarrow^*_{\mathbf{T}} R$. At least one of $\{L^1_{-1}, L^1_{+1}, L^2_{-1}, L^2_{+1}\}$, say L^r_s, can be added to R, resulting in a larger assembly also produced by $\mathbf{T}$. This can be continued indefinitely: if $s = +1$ then for all n, $R + \sum_{i=+1}^{n} L^r_i$ is in $Prod(\mathbf{T})$; if $s = -1$ then for all n, $R + \sum_{i=-n}^{-1} L^r_i$ is in $Prod(\mathbf{T})$. This contradicts the assumption that $\mathbf{T}$ is halting and terminates in $N \times N$ full squares. ∎

At $\mathcal{T} = 2$ the situation is markedly different.

Theorem 2. $K^2_{SA}(N) = O(N)$.

Proof. Figure 4 shows two constructions for an $N \times N$ full square using $2N$ (Figure 4a) and $N + 3$ (Figure 4b) tiles respectively. Self-assembly from the seed tile 1 proceeds initially by single strength-2 interactions creating the borders with the numbered tiles. As the border grows, two cooperative strength-1 interactions allow the blank tile to fill in and complete the square. For the tiles at the right, the **A** and **B** tiles enter a new column by their strength-2 side, thus allowing the rest of the column to be filled with blanks. The $N \times N$ full square can be easily verified to be a terminal assembly. ∎

This is only the beginning. The construction in Figure 4b can be combined with a fixed-width version of the binary counter of Figure 1 to obtain a set of tiles that produce the $N \times N$ full square by counting in binary instead of by counting in unary.

Theorem 3. $K^2_{SA}(N) = O(\log N)$.

Proof. Figure 5 constructs an $N \times N$ full square using $n + 22$ tiles, where $n = \lceil \log N \rceil$. $n + 2$ tiles, including the seed tile, produce an $(n - 1) \times (n - 1)$ square as in the previous construct. Additionally, the $n - 1$ tiles in the seed row have upper sides encoding the bits of the integer $c = 1 + 2^{n-1} - (N - n)/2$, the initial value of the counter. We must use a fixed-width version of the counter tiles of Figure 1; this requires a special set of tiles for the leftmost and rightmost columns of bits. The counter counts from c to 2^{n-1}, using two rows for each integer. In order to detect

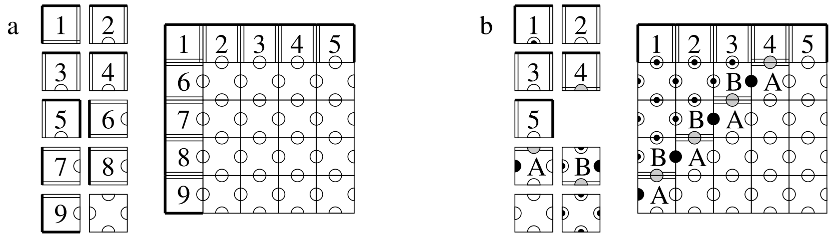

Figure 4: Formation of full squares at $\mathcal{T} = 2$. (a) $2N = 10$ tiles uniquely produce 5×5 full square. Except for the sides labeled with a circle, each interacting pair of tiles share a unique side label (but we do not label them explicitly as in Figure 2.) (b) $N + 4 = 9$ tiles are used.

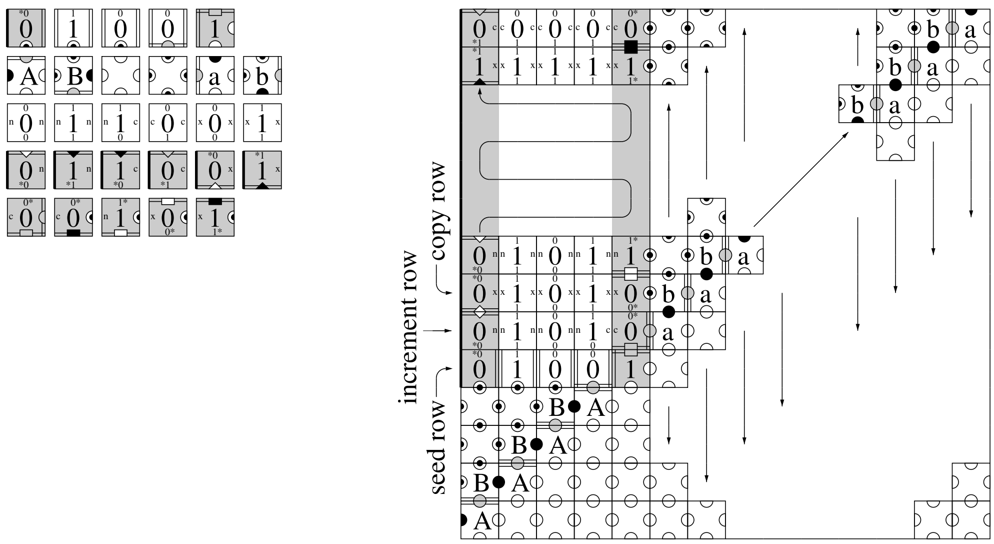

Figure 5: Formation of $N{\times}N$ square using $O(\log N)$ tiles. Construction starts with an $n-1 \times n-1$ square as in Figure 4b. Here $N = 52, n = 6$ and 28 tiles are used. The construction illustrates the case for even $N - n$; the first row above the seed row is a copy row for odd $N - n$.

when the counter has finished, we use alternating rows to increment the counter from right to left, then to copy the the bits from left to right *unless* the leftmost bit just rolled over from 1 to 0. In the latter case, the tile presents a strength-2 side with a label not found on any other tiles, thus halting the counter. (The strength-2 side will be used in our next construction; here, any strength would suffice.) There is a special tile for the rightmost bit in the first increment row above the seed row. This tile contains a strength-2 side to initiate the **a-b** diagonal, thus filling in the rest of the square. Overall, the counter requires 18 tiles; the seed row requires $n - 1$ tiles; the two diagonals require 4 tiles; and there are two blank tiles. ∎

But we can do much better: by recursively iterating the above construction one can produce $N{\times}N$ squares with

$$N \geq \underbrace{2^{2^{2^{\cdots^{2}}}}}_{n\ times} \stackrel{def}{=} 2**n$$

using only $O(n)$ tiles. Define $\log^* N$ as the least n such that $2**n \geq N$.

Theorem 4. $K^2_{SA}(N) = O_{i.o.}(\log^* N)$.

Proof. Our proof is by induction. Let S^n refer to a tile system containing fewer than $22n$ tiles (including the **a**, **b**, and *blank* tiles) that uniquely produces an $N{\times}N$ full square such that

- $N > 2**n$.
- All binding domains on the left and bottom are of strength 1 or 0.
- All binding domains on the right have the strength-1 blank label.
- The binding domains on the upper side conform to the pattern xy^*zb^*a where x is a strength-2 binding domain that occurs nowhere else, and y, z, b, and a are distinct strength-1 binding domains.

We show that S^n exists for all n. The base case $n = 1$ is trivial. The inductive step is illustrated in Figure 6. First,

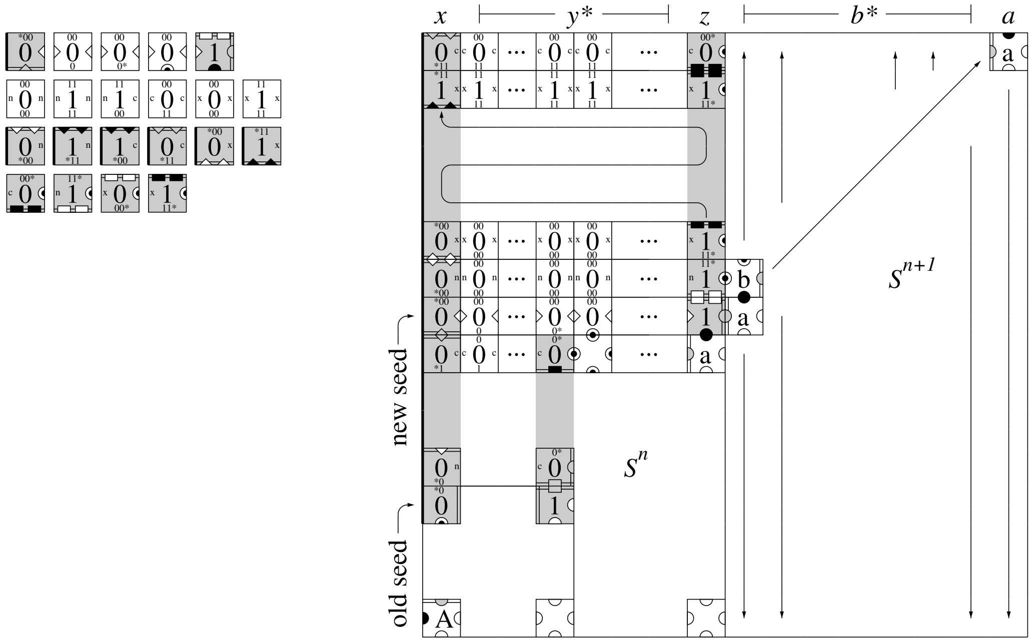

Figure 6: Formation of $N{\times}N$ square using $O_{i.o.}(\log^* N)$ tiles. Given a set of tiles S^n that produce an $N{\times}N$ full square that satisfies the recurrence, the addition of 22 new tiles results in S^{n+1} and produces a $(N + 2 \times 2^N) \times (N + 2 \times 2^N)$ full square. New side labels (with doubled symbols) prevent counter tiles from S^n from incorporating in the S^{n+1} counter.

there are 5 tiles that, initiated by x, produce an initial string of 0's for a new fixed-width counter, and provide a strength-2 side for a new **a-b** diagonal. Then there are 16 tiles equivalent to the counter tiles in Theorem 3 but using new side labels; the counter counts to 2^N. The diagonal fills in the rest of the square, now with sides of length $N + 2 \times 2^N > 2^N > 2 ** (n+1)$. Therefore S^n exists for all n, and for those n,

$$22 \log^* N \geq 22n \geq K^2_{SA}(N).$$

∎

$\log^* N$ is an exceedingly slowly growing function; the above construction shows that very large squares can be assembled with a very small number of tiles. But we can do much better yet! By embedding the simulation of a Turing machine in the growth of a square we show that:

Theorem 5. *$K^2_{SA}(N) = O_{i.o.}(f(N))$ for $f(N)$ any non-decreasing unbounded computable function.*

Proof. Our proof relies on a self-assembly version of the Busy Beaver problem [Rado, 1962]. Define:

$$B^{\mathcal{T}}_{SA}(n) = \max\{N \text{ s.t. } (N, n) \in Sq^{\mathcal{T}}\}.$$

To show Theorem 5, we first show

$$B^2_{SA}(n) = \Omega(F(n)) \text{ for any computable function } F(n). \tag{1}$$

Theorem 5 follows from (1) by contradiction: if false, then there exists a computable, non-decreasing, unbounded function $f(N)$ such that $\exists N_0$ s.t. $\forall N > N_0$, $K^2_{SA}(N) \geq f(N)$. Let $F(n) = \max\{N \text{ s.t. } N = 0 \text{ or } f(N) \leq n\}$; this is a computable function. Note that $B^2_{SA}(n) \geq F(n)$ requires that $\exists (N, n) \in Sq^2$ s.t. $N \geq F(n)$ and therefore $f(N) > n$ and $K^2_{SA}(N) \leq n$. For $N > N_0$ this contradicts $K^2_{SA}(N) \geq f(N)$. Therefore, for all $n > f(N_0)$, $B^2_{SA}(n) < F(n)$, contradicting (1) and establishing Theorem 5.

Recall that $B_t(m) = \Omega(F'(m))$ for any computable function $F'(m)$ where:

$$B_t(m) = \max\{t \text{ s.t. } m = qs \text{ and there exists a } q\text{-state, } s\text{-symbol Turing machine that halts on a blank tape in } t \text{ steps}\}$$

Let M be a q-state, s-symbol Turing machine that halts on a blank tape in $B_t(m)$ steps, where $m = qs$. We will construct a square of size $N = 2B_t(m){+}3$ using $n = 12qs{+}4s{+}9$ tiles by simulating M with tiles, similar to the construction of Robinson [Robinson, 1971]. Given any $n > 41$, we will use $s_n = 2$, $q_n = \lfloor \frac{n-17}{24} \rfloor$, and $m_n = q_n s_n$; our construction will need only $12 q_n s_n + 4 s_n + 9 < n$ tiles. Then $B^2_{SA}(n) \geq 2B_t(m_n) + 3 = \Omega(F'(m_n))$. For any computable function $F(n)$, we can find another computable function $F'(m)$ s.t. $\forall n$, $F'(m_n) > F(n)$. Therefore, we arrive at (1).

We construct the square by growing four identical simulations of the Turing machine M, one from each side of a seed tile. Each simulation stays within one of the four

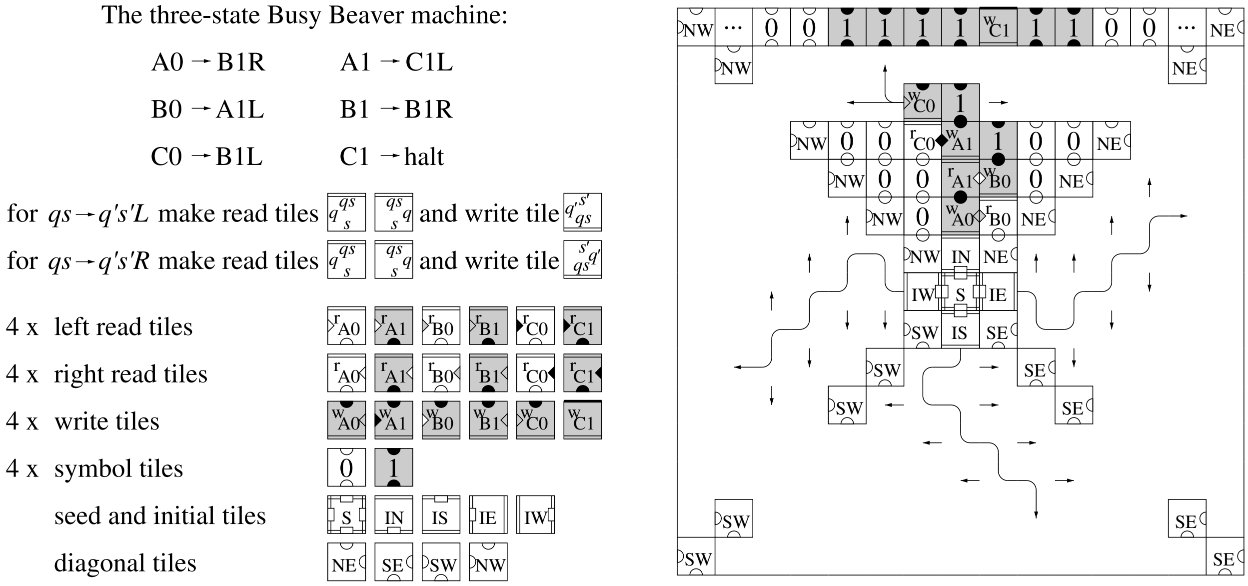

Figure 7: Formation of an $N{\times}N$ square by growing four identical simulations of a given Turing machine. The Busy Beaver machine simulated here has three states ($q_0 = A, q_1 = B, q_2 = C$) and two symbols ($s_0 = 0, s_1 = 1$). Note that R denotes right, L denotes left, and "4x" indicates that four variations of a tile are used, one for each compass direction.

regions bounded by the diagonals of the square; when M halts, the square is complete. We require 4 tiles to create the four "half-diagonals" defining these boundaries between simulations. For each simulation we require 1 "initial state" that matches the seed tile, s symbol tiles, qs write tiles, and $2qs$ read tiles, giving a total of $3qs + s + 1$ tiles per simulation. We describe these tiles for the TM simulation to the north of the seed tile. Recall that a tile is a 4-tuple $(\sigma_N, \sigma_E, \sigma_S, \sigma_W)$ representing the north, east, south, and west binding domains. Binding domain strengths are 1 unless noted. Each of the four simulations has its own version of the side labels described, distinguished by superscripts (we omit the superscript ^{N} from the description of north-facing simulation below). The symbol tile for symbol s is $(\sigma_s, \sigma_e, \sigma_s, \sigma_e)$, where σ_s is a binding domain representing the symbol s and σ_e is a binding domain indicating that the TM head is not present. For each state-symbol pair (q, s), the left read tile $(\sigma_{q,s}, \sigma_e, \sigma_s, \sigma_q)$ and the right read tile $(\sigma_{q,s}, \sigma_q, \sigma_s, \sigma_e)$ represent the TM head in state q entering a tape cell (from the left or from the right) and reading the symbol s. The binding domains $\sigma_{q,s}$ have strength 2; this is necessary for the TM head to enter the next row of the simulation. The write tiles, representing the action the TM head takes, depend on the form of the state transition table entry. For each entry of the form $(q, s) \rightarrow (q', s', L)$ there is a write tile $(\sigma_{s'}, \sigma_e, \sigma_{q,s}, \sigma_{q'})$; for each entry of the form $(q, s) \rightarrow (q', s', R)$ there is a write tile $(\sigma_{s'}, \sigma_{q'}, \sigma_{q,s}, \sigma_e)$; for each entry of the form $(q, s) \rightarrow halt$ there is a write tile $(\sigma_{halt}, \sigma_e, \sigma_{q,s}, \sigma_e)$.

To start the Turing Machine in state q_0 reading the blank symbol s_0, the initial tile for the northern simulation is $IN = (\sigma_{q_0,s_0}, \sigma_e, \sigma_S, \sigma_e)$, where σ_S is a strength-2 binding domain. The initial tiles for all four simulations bind to the seed tile $S = (\sigma_S^N, \sigma_S^E, \sigma_S^S, \sigma_S^W)$. The four diagonal tiles, $NW = (\sigma_{s_0}^N, \sigma_e^N, \sigma_e^W, \sigma_{s_0}^W)$, $NE = (\sigma_{s_0}^N, \sigma_{s0}^E \sigma_e^E, \sigma_e^N,)$, $SE = (\sigma_e^E, \sigma_{s_0}^E, \sigma_{s_0}^S, \sigma_e^S)$, and $SW = (\sigma_e^W, \sigma_e^S, \sigma_{s_0}^S, \sigma_{s_0}^W)$ pad the tapes with extra cells containing the blank symbol s_0 and delimit the four simulations. ∎

Theorem 4 gives the construction of infinite number of very large squares made from a very small number of tile types. Theorem 5 implies that, for an infinite set of N, the number of tiles required to assemble an $N{\times}N$ square can be made "arbitrarily small". How well can one do in general? Unfortunately, extremely concise self-assembly programs are not common. Then we show:

THEOREM 6. $K^2_{SA}(N) = \Omega_{a.a.}(\frac{\log N}{\log \log N})$.

PROOF. The Kolmogorov complexity of an integer N with respect to a universal Turing machine U is

$$K_U(N) = \min |p| \text{ s.t. } U(p) = \#N$$

where $\#N$ is the binary string representing N. (See [Li and Vitanyi, 1997] for results on Kolmogorov complexity.) Recall that $K_U(N) < \lceil \log N \rceil - \Delta$ for at most $\frac{1}{2}^\Delta$ of all N, by the pidgeonhole principle. Therefore, for any $\epsilon > 0$, $K_U(N) > (1 - \epsilon) \log N$ for almost all N.

There exists a Turing machine $SA2$ (with program p_{SA2}) that, given a binary description of a $\mathcal{T} = 2$ tile system, simulates their self-assembly, making an arbitrary choice when multiple tile additions are possible, and returns the maximal dimension of the resulting assembly if self-assembly terminates. A tile system $\mathbf{T}$ with n tiles can be described by $d_{\mathbf{T}}$

containing $f(n) = 4n\lceil \log 3n \rceil$ bits; each of $4n$ sides may have a strength in $\{0, 1, 2\}$, and non-zero-strength labels may be defined by the first tile (in some arbitrary order) with the same tile on the opposite side. Thus if tile system $\mathbf{T}$ uniquely produces an $N \times N$ full square, then $p = p_{SA2} d_{\mathbf{T}}$ will return $\#N$ when input to U. Therefore, for almost all N,

$$(1-\epsilon)\log N < K_U(N) \leq |p_{SA2}| + f(K^2_{SA}(N))$$
$$< C_1 + C_2 K^2_{SA}(N)[C_3 + \log\log N],$$

where we used $K^2_{SA}(N) = O(\log N)$ from Theorem 3. The final result follows from simple algebra. ∎

In other words, a Kolmogorov-random integer N cannot be compressed by the self-assembly model.

4. DISCUSSION

Tiles or labels. This paper discussed the program-size complexity of self-assembled squares, where complexity was measure by the number of distinct tile types involved. An alternative complexity measure is the minimum number of distinct side labels required to uniquely produce the object. The number of labels will be relevant in a physical system where the number of distinct binding interactions is limited due to imperfect specificity of binding. Do both measures give asymptotically similar results?

Kolmogorov complexity. A main conclusion of this paper is that the program-size complexity of self-assembled objects (at $\mathcal{T} = 2$) looks remarkably similar to the usual program-size complexity with respect to Turing machines. This is hardly surprising, since self-assembly at $\mathcal{T} = 2$ can simulate Turing machines. However, Figure 8 makes K^2_{SA} look perhaps more similar to K_U than it ought to: K^2_{SA} is computable due to the monotonic nature of self-assembly growth (each tile set can be simulated in turn until it halts or exceeds an $N \times N$ region), whereas of course K_U is not computable.

Disintegration. The difference between K^2_{SA} and K_U comes from the monotonic nature of self-assembly: the growing object always gets bigger, so "temporary results" cannot be larger than the object itself. A simple device circumvents this difficulty: select a subset $D \subset T$; after self-assembly is complete, the tiles in D are destroyed in all assemblies in $Term(\mathbf{T})$, and the resulting assemblies are considered the output of the computation. A molecular implementation might make the tiles in D out of RNA, while the tiles in $T \setminus D$ are DNA; then, an RNase enzyme can be used to destroy all tiles in D.

Let $\hat{K}^2_{SA}$ be the full square complexity for the model with disintegration. The ability of $\mathcal{T} = 2$ self-assembly to simulate Turing machines can now be used to make squares: given a Turing machine program p such that $U(p) = \#N$, we generate $C + |p|$ tiles (all in D) that simulate p and expand the result into a template of length N. Now a few final tiles (not in D) complete the square; only this square will survive disintegration. Thus $\hat{K}^2_{SA} < K_U(N) + C$; we already know $K_U(N) = O(\hat{K}^2_{SA}(N) \log\log N)$. That is, the only difference is that $\hat{K}^2_{SA}$ measures the number of tiles instead of the number of bits required to specify the tiles.

5. ACKNOWLEDGEMENTS

We thank Len Adleman for discussions and for raising the questions that motivate this paper. We thank Richard Lipton, John Reif, Ming-Deh Huang, Ashish Goel, Qi Cheng, Lior Pachter and Lisa O'Rourke for comments and discussions.

6. REFERENCES

Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024.

Adleman, L. M. (unpublished manuscript, 2000). Toward a mathematical theory of self-assembly.

Bennett, C. H. (1982). The thermodynamics of computation – a review. *International Journal of Theoretical Physics*, 21(12):905–940.

Berger, R. (1966). The undecidability of the domino problem. *Memiors of the AMS*, 66:1–72.

Bowden, N., Choi, I., Gryzbowski, B., and Whitesides, G. (1999). Mesoscale self-assembly of hexagonal plates using lateral capillary forces: Synthesis using the "capillary bond". *Journal of the American Chemical Society*, 121:5373–5391.

Bowden, N., Terfort, A., Carbeck, J., and Whitesides, G. (1997). Self-assembly of mesoscale objects into ordered two-dimensional arrays. *Science*, 276:233–235.

Hjelmfelt, A. and Ross, J. (1995). Implementation of logic functions and computations by chemical kinetics. *Physica D*, 84:180–193.

Hosokawa, K., Shimoyama, I., and Miura, H. (1996). Two-dimensional micro-self-assembly using the surface tension of water. *Sensors and Actuators A*, 57:117–125.

Kurtz, S. A., Mahaney, S. R., Royer, J. S., and Simon, J. (1997). Biological computing. In Hemaspaandra, L. A. and Selman, A. L., editors, *Complexity Theory Retrospective II*, pages 179–195. Springer.

Li, M. and Vitanyi, P. (1997). *An Introduction to Kolmogorov Complexity and Its Applications (Second Edition)*. Springer Verlag, New York.

Mackay, A. (1995). Generalised crystallography. *Journal of Molecular Structure (Theochem)*, 336:293–303.

Magnasco, M. O. (1997). Chemical kinetics is Turing universal. *Physical Review Letters*, 78(6):1190–1193.

Markov, I. V. (1995). *Crystal Growth for Beginners: fundamentals of nucleation, crystal growth, and epitaxy.* World Scientific, Singapore.

Ptashne, M. (1992). *A Genetic Switch, 2nd ed.* Cell Press & Blackwell.

Radin, C. (1991). Global order from local sources. *Bulletin of the AMS*, 25(2):335–364.

Rado, T. (1962). On non-computable functions. *Bell System Technical Journal*, 41(3):877–884.

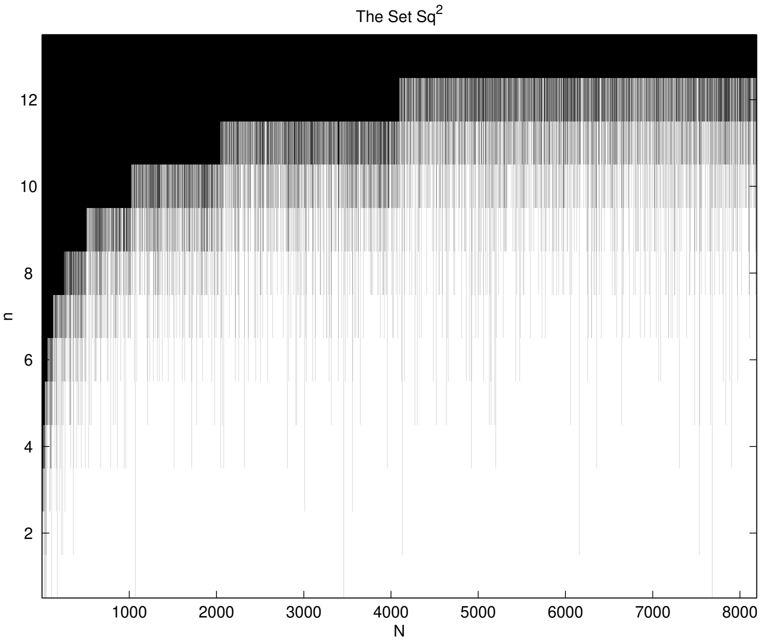

Figure 8: Artist's impression of the set Sq^2 of pairs (N,n) where n tiles produce an $N{\times}N$ full square(black). $K^2_{SA}(N)$ is the lowest point in a column; the vertical lines are due to the fact that $(N,n) \in Sq^2 \implies (N,n+1) \in Sq^2$. $B^2_{SA}(n)$ is the rightmost point in a row.

Robinson, R. M. (1971). Undecidability and nonperiodicity of tilings of the plane. *Inventiones Math.*, 12:177–209.

Rothemund, P. W. K. (2000). Using lateral capillary forces to compute by self-assembly. *Proceedings of the National Academy of Sciences*, 97:984–989.

Schectman, D., Blech, I., Gratias, D., and Cahn, J. (1984). Metallic phase with long-range orientational order and no translational symmetry. *Phys. Rev. Lett.*, 53:1951–1953.

Seeman, N. C. (1998). DNA nanotechnology: novel DNA constructions. *Annual Review of Biophysics and Biomolecular Structure*, 27:225–248.

Senechal, M. (1995). *Quasicrystals and geometry.* Cambridge University Press, Cambridge.

Wang, H. (1961). Proving theorems by pattern recognition. II. *Bell System Technical Journal*, 40:1–42.

Wang, H. (1963). Dominoes and the AEA case of the decision problem. In Fox, J., editor, *Mathematical Theory of Automata*, pages 23–55, Brooklyn, New York. Polytechnic Press.

Winfree, E. (1996). On the computational power of DNA annealing and ligation. In Lipton, R. J. and Baum, E. B., editors, *DNA Based Computers: DIMACS Workshop, April 4, 1995*, volume 27, pages 199–221, Providence, RI. American Mathematical Society.

Winfree, E. (preliminary, 1998). Simulations of computing by self-assembly. In Kari, L., Rubin, H., and Wood, D. H., editors, *Proceedings of the* 4^{th} *DIMACS Meeting on DNA Based Computers, held at the University of Pennsylvania, June 16-19, 1998.*

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

Winfree, E., Yang, X., and Seeman, N. C. (1998b). Universal computation via self-assembly of DNA: Some theory and experiments. In Landweber, L. F. and Baum, E. B., editors, *DNA Based Computers II: DIMACS Workshop, June 10-12, 1996*, volume 44, Providence, RI. American Mathematical Society.