

An introduction to tile-based self-assembly and a survey of recent results

Matthew J. Patitz

Published online: 21 June 2013
© Springer Science+Business Media Dordrecht 2013

Abstract We first give an introduction to the field of tile-based self-assembly, focusing primarily on theoretical models and their algorithmic nature. We start with a description of Winfree’s abstract Tile Assembly Model (aTAM) and survey a series of results in that model, discussing topics such as the shapes which can be built and the computations which can be performed, among many others. Next, we introduce the more experimentally realistic kinetic Tile Assembly Model (kTAM) and provide an overview of kTAM results, focusing especially on the kTAM’s ability to model errors and several results targeted at preventing and correcting errors. We then describe the 2-Handed Assembly Model (2HAM), which allows entire assemblies to combine with each other in pairs (as opposed to the restriction of single-tile addition in the aTAM and kTAM) and doesn’t require a specified seed. We give overviews of a series of 2HAM results, which tend to make use of geometric techniques not applicable in the aTAM. Finally, we discuss and define a wide array of more recently developed models and discuss their various tradeoffs in comparison to the previous models and to each other.

Keywords Tile-based self-assembly · Algorithmic self-assembly · Tile Assembly Model · Wang tiles

1 Introduction

Self-assembly is the process by which a collection of relatively simple components, beginning in a disorganized

state, spontaneously and without external guidance coalesce to form more complex structures. The process is guided by only local interactions between the components, which typically follow a basic set of rules. Despite the seemingly simplistic nature of self-assembly, its power can be harnessed to form structures of incredible complexity and intricacy. In fact, self-assembling systems abound in nature, resulting in everything from the delicate crystalline structure of snowflakes to many of the structurally and functionally varied components of biological systems.

Beyond the purely mathematically interesting properties of self-assembling systems, such systems have been recognized as an excellent template for the fabrication of artificial structures on the nanoscale. In order to precisely manipulate and organize matter on the scale of individual atoms and molecules, several artificial self-assembling systems have been designed. In order to model such systems, theoretical models have been developed, and one of the most popular among these is the Tile Assembly Model introduced by Winfree in his 1998 Ph.D. thesis (Winfree 1998). Formulated in two basic versions, the abstract Tile Assembly Model (aTAM) and the kinetic Tile Assembly Model (kTAM), it was based on a cross between the theoretical study of Wang tiles (1963) (flat squares with colored markings on their edges and matching rules for the ways those edges can be placed next to each other) and novel DNA complexes being synthesized within Ned Seeman’s laboratory (1982). The aTAM provides a more high-level abstraction which ignores the possibility of errors and provides a framework for theoretical studies of the mathematical boundaries to the powers of such systems. The kTAM, on the other hand, injects more of the physical reality of the chemical kinetics into the model and allows for the study of the causes of errors and potential mechanisms for detecting, preventing, and/or correcting

M. J. Patitz (✉)
Department of Computer Science and Computer Engineering,
University of Arkansas, Fayetteville, AR, USA
e-mail: mpatitz@self-assembly.net

them. In fact, the kTAM serves as such a realistic model that it has helped to accurately predict and shape the experimental direction of several laboratory experiments in which actual tile-based assemblies form. Just a few examples of laboratory implementations include Barish (2009), Chen (2007), Mao (2000), Rothmund (2004), and Winfree (1998) where designs from binary counters, to the fractal pattern known as the Sierpinski triangle, to implementations of sophisticated error prevention techniques have been realized.

Tile-based self-assembly has proven to be a very rich area of research, and the early proof of its computational universality by Winfree (1998) showed that it can be algorithmically directed (putting it into the general field of *algorithmic self-assembly*). The theoretical results to be discussed here represent a wide variety of fundamental insights into the power of self-assembling systems which are likely to pave the way for even deeper theoretical results (which impact other areas of theoretical Computer Science, Mathematics, etc.). They also provide an increasingly firm foundation for the physical development of artificial self-assembling systems, continuing in research laboratories but eventually migrating to large scale fabrication facilities. This paper is an extension of the tutorial (Matthew 2012) presented at the 11th International Conference on Unconventional Computation and Natural Computation, and is meant to serve as an introduction to tile-based self-assembly via the aTAM, kTAM, and several other related models, as well as a survey of a wide variety of results related to those models and to the theoretical study of tile-based self-assembly in general. [For another excellent survey of this area, the reader is encouraged to refer to Doty (2012)].

We will first introduce the aTAM, giving a high-level overview and then the technical definition of the model, providing comparisons and contrasts between it and Wang tiling. Next we present a complete example of an aTAM system to clearly show how the model works and how to design a basic system in it. After this we will present a survey of results based on the aTAM, broadly categorizing much of the work in the field into a series of categories related to goals such as: what types of shapes can be built, what computations can be performed, how efficiently (as measured in a variety of ways) can assemblies be built, etc.

In the second main portion of the paper, we will introduce the kTAM and provide an explanation of relevant definitions and formulas. We will then provide an example of how to design a kTAM system to provide basic error prevention. Next we will survey a series of results based on the kTAM to provide a picture of the progress that has been made in terms of algorithmic approaches to error prevention and correction, as well as modifications made to tile designs toward those ends. We will then introduce the

2-Handed Assembly Model (2HAM), in which, rather than requiring seeded assemblies which can grow only one tile at a time, assemblies can spontaneously nucleate and arbitrarily large assemblies are allowed to combine with each other two at a time. We will also provide a complete example of a 2HAM system before discussing a variety of 2HAM results, emphasizing especially those which provide comparisons and contrasts with the aTAM.

The last main portion of the tutorial will be comprised of high-level introductions to a wide array of newer, derivative models. Such models have been introduced for a variety of reasons: to provide greater resilience to errors, to potentially provide more feasible laboratory implementations, to overcome theoretical limitations of the base models, to more faithfully mimic the behavior of given natural (especially biological) self-assembling systems, or simply to more fully explore the vast landscape of alternatives. Examples of such models include: temperature and concentration programming, the Staged Assembly Model, the Geometric Tile Assembly Model, and the Signal passing Tile Assembly Model. The results presented for these models and the discussions provided will attempt to paint a clear picture of the salient differences between models and the powers imbued by those differences.

The results surveyed in this paper cannot cover the full set of work in tile-based self-assembly as it is quite extensive, and mention of several results is unfortunately omitted. We hope that the high-level descriptions and simple examples presented here can provide a solid introduction to the area and perhaps serve as an aid for a course on the topic. The reader is encouraged to use this as a broad roadmap covering a large but incomplete collection of results which attempts to sketch the main lines of research that have been pursued, but to refer to the full papers referenced here for much more detail and also for references to works missing in this survey.

2 Preliminaries and notation

In this section we provide a set of definitions and conventions that are used throughout this paper.

We work in the 2-dimensional discrete space \mathbb{Z}^2 . Define the set $U_2 = \{(0, 1), (1, 0), (0, -1), (-1, 0)\}$ to be the set of all *unit vectors* in \mathbb{Z}^2 . We also sometimes refer to these vectors by their cardinal directions N, E, S, W , respectively. All *graphs* in this paper are undirected. A *grid graph* is a graph $G = (V, E)$ in which $V \subseteq \mathbb{Z}^2$ and every edge $\{\vec{a}, \vec{b}\} \in E$ has the property that $\vec{a} - \vec{b} \in U_2$.

Intuitively, a tile type t is a unit square that can be translated, but not rotated, having a well-defined “side \vec{u} ” for each $\vec{u} \in U_2$. Each side \vec{u} of t has a “glue” with “label”

label_{*t*}(\vec{u})—a string over some fixed alphabet—and “strength” $\text{str}_t(\vec{u})$ —a nonnegative integer—specified by its type t . Two tiles t and t' that are placed at the points \vec{a} and $\vec{a} + \vec{u}$ respectively, *bind* with *strength* $\text{str}_t(\vec{u})$ if and only if $(\text{label}_t(\vec{u}), \text{str}_t(\vec{u})) = (\text{label}_{t'}(-\vec{u}), \text{str}_{t'}(-\vec{u}))$.

In the subsequent definitions, given two partial functions f, g , we write $f(x) = g(x)$ if f and g are both defined and equal on x , or if f and g are both undefined on x .

Fix a finite set T of tile types. A T -assembly, sometimes denoted simply as an *assembly* when T is clear from the context, is a partial function $\alpha : \mathbb{Z}^2 \rightarrow T$ defined on at least one input, with points $\vec{x} \in \mathbb{Z}^2$ at which $\alpha(\vec{x})$ is undefined interpreted to be empty space, so that $\text{dom } \alpha$ is the set of points with tiles.

We write $|\alpha|$ to denote $|\text{dom } \alpha|$, and we say α is *finite* if $|\alpha|$ is finite. For assemblies α and α' , we say that α is a *sub-assembly* of α' , and write $\alpha \sqsubseteq \alpha'$, if $\text{dom } \alpha \subseteq \text{dom } \alpha'$ and $\alpha(\vec{x}) = \alpha'(\vec{x})$ for all $x \in \text{dom } \alpha$.

3 The abstract Tile Assembly Model (aTAM)

As the aTAM is based on, and similar to in some general aspects, an older model known as Wang tiling (1961), we first give a brief introduction to Wang tiling with the goal being to eventually show how the two models are similar, but more importantly how they differ as well.

3.1 Wang tiling

Introduced by Hao Wang in (Wang 1961), Wang tiles are defined as equally sized, two dimensional unit squares which have colors on each edge. They can be arranged side by side, with edges aligned, on a regular square grid as long as abutting edges have matching colors, and each tile has a fixed orientation so that it cannot be rotated or flipped. The key problem considered in Wang tiling is, given a set of Wang tiles, with an infinite number of each type, can they be placed so that they tile the plane? More specifically, the question is whether there exists an arrangement of tiles from a given set such that they completely cover the infinite plane \mathbb{Z}^2 leaving no holes and with all adjacent tile edges having matching colors.

Wang initially conjectured that any set of tiles which could tile the plane would be able to do so in a periodic way. The implication was that there must exist an algorithm which can decide whether or not a given set of Wang tiles can tile the plane. However, Berger disproved this conjecture (1965) by showing how to convert an arbitrary Turing machine definition into a set of Wang tiles which “simulate” the Turing machine in such a way that they admit a tiling of the plane if and only if the Turing machine

does not halt. Thus, since the halting problem is undecidable, so must be the problem of determining whether or not a given set of Wang tiles can tile the plane. This further implied the existence of a (finite) set of Wang tiles which can tile the plane but only aperiodically. Berger’s original tile set had over 20,000 tile types, but since then several additional aperiodic tile sets have been discovered with increasingly smaller and smaller numbers of tile types, with a sequence of improvements by Berger, Knuth, Läuchli, Robinson, Penrose, Ammann, and Culik finally resulting in an aperiodic set of just 13 Wang tiles (Culik II 1996). [Note that, using different tiling systems, aperiodic tilings of the plane have been achieved with tiles set as small as 2, as in the Penrose tilings (Penrose 1979).]

3.2 aTAM definition

The aTAM was developed to, in some sense, be an effective-ization of Wang tiling. (See Sect. 3.3 for more about this relationship.) Namely, it provides a defined process by which an initial (called the *seed*) assembly can grow into a resultant structure. This is essentially accomplished by assigning a positive integer *strength* value to each edge color in a set of Wang tiles and stipulating that when two tile edges are adjacent, if their colors match then the edges bind with force equivalent to the strength of the edge color. Then, starting with a preformed seed assembly (usually taken to be a single tile of a specified type), additional tiles can attach one at a time as long as the sum of the strengths of the bonds that each makes with tiles already in the assembly meets a system wide threshold value called the *temperature*.

We now give a brief formal definition of the aTAM. See Lathrop (2009), Rothmund (2001), Rothmund and Winfree (2000), Winfree (1998) for other developments of the model. Our notation is that of Lathrop (2009), which also contains a more complete definition.

Given a set T of tile types, an *assembly* is a partial function $\alpha : \mathbb{Z}^2 \rightarrow T$. An assembly is τ -stable if it cannot be broken up into smaller assemblies without breaking bonds of total strength at least τ , for some $\tau \in \mathbb{N}$.

Self-assembly begins with a *seed assembly* σ and proceeds asynchronously and nondeterministically, with tiles adsorbing one at a time to the existing assembly in any manner that preserves τ -stability at all times. A *tile assembly system* (TAS) is an ordered triple $\mathcal{T} = (T, \sigma, \tau)$, where T is a finite set of tile types, σ is a seed assembly with finite domain, and $\tau \in \mathbb{N}$. A *generalized tile assembly system* (GTAS) is defined similarly, but without the finiteness requirements. We write $\mathcal{A}[T]$ for the set of all assemblies that can arise (in finitely many steps or in the limit) from \mathcal{T} . An assembly $\alpha \in \mathcal{A}[T]$ is *terminal*, and we write $\alpha \in \mathcal{A}_{\square}[T]$, if no tile can be τ -stably added to it. It is clear that $\mathcal{A}_{\square}[T] \subseteq \mathcal{A}[T]$.

An assembly sequence in a TAS \mathcal{T} is a (finite or infinite) sequence $\vec{\alpha} = (\alpha_0, \alpha_1, \dots)$ of assemblies in which each α_{i+1} is obtained from α_i by the addition of a single tile. The *result* $\text{res}(\vec{\alpha})$ of such an assembly sequence is its unique limiting assembly. (This is the last assembly in the sequence if the sequence is finite.) The set $\mathcal{A}[\mathcal{T}]$ is partially ordered by the relation \longrightarrow defined by

$$\alpha \longrightarrow \alpha' \text{ iff there is an assembly sequence } \vec{\alpha} = (\alpha_0, \alpha_1, \dots) \text{ such that } \alpha_0 = \alpha \text{ and } \alpha' = \text{res}(\vec{\alpha}).$$

We say that \mathcal{T} is *directed* (a.k.a. *deterministic*, *confluent*, *produces a unique assembly*) if the relation \longrightarrow is directed, i.e., if for all $\alpha, \alpha' \in \mathcal{A}[\mathcal{T}]$, there exists $\alpha'' \in \mathcal{A}[\mathcal{T}]$ such that $\alpha \longrightarrow \alpha''$ and $\alpha' \longrightarrow \alpha''$. It is easy to show that \mathcal{T} is directed if and only if there is a unique terminal assembly $\alpha \in \mathcal{A}[\mathcal{T}]$ such that $\sigma \longrightarrow \alpha$.

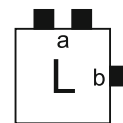
In general, even a directed TAS may have a very large (perhaps uncountably infinite) number of different assembly sequences leading to its terminal assembly. This seems to make it very difficult to prove that a TAS is directed. Fortunately, Soloveichik and Winfree (2007) have defined a property, *local determinism*, of assembly sequences and proven the remarkable fact that, if a TAS \mathcal{T} has *any* assembly sequence that is locally deterministic, then \mathcal{T} is directed. Intuitively, an assembly sequence $\vec{\alpha}$ is locally deterministic if (1) each tile added in $\vec{\alpha}$ “just barely” binds to the existing assembly (meaning that it does so with a summed strength of bonds equal to exactly τ); (2) if a tile of type t_0 at a location \vec{m} and its immediate “output-neighbors” (i.e. those adjacent tiles which bound after the tile at \vec{m}) are deleted from the *result* of $\vec{\alpha}$, then no tile of type $t \neq t_0$ can attach itself to the thus-obtained configuration at location \vec{m} ; and (3) the result of $\vec{\alpha}$ is terminal.

A set $X \subseteq \mathbb{Z}^2$ *weakly self-assembles* if there exists a TAS $\mathcal{T} = (T, \sigma, \tau)$ and a set $B \subseteq T$ such that $\alpha^{-1}(B) = X$ holds for every terminal assembly $\alpha \in \mathcal{A}_{\square}[\mathcal{T}]$. Essentially, weak self-assembly can be thought of as the creation (or “painting”) of a pattern of tiles from B (usually taken to be a unique “color”) on a possibly larger “canvas” of uncolored tiles.

A set X *strictly self-assembles* if there is a TAS \mathcal{T} for which every assembly $\alpha \in \mathcal{A}_{\square}[\mathcal{T}]$ satisfies $\text{dom } \alpha = X$. Essentially, strict self-assembly means that tiles are only placed in positions defined by the shape. Note that if X strictly self-assembles, then X weakly self-assembles. (Let all tiles be in B .)

Tiles are often depicted as squares whose various sides contain 0, 1, or 2 attached black squares, indicating whether the glue strengths on these sides are 0, 1, or 2, respectively. Thus, for example, a tile of the type shown in Fig. 1 has glue of strength 0 on the left (W) and bottom (S), glue of color ‘a’ and strength 2 on the top (N), and glue of

Fig. 1 An example tile type



color ‘b’ and strength 1 on the right (E). This tile also has a label ‘L’, which plays no formal role but may aid our understanding and discussion of the construction.

3.3 Wang tiling vs. the aTAM

Despite some superficial similarities between Wang tiling and the aTAM, there are several important differences between the two which make results in one area not necessarily applicable in the other. In general, the problem being considered in Wang tiling is whether or not there exists *at least one* configuration of tiles such that all adjacent edges match (i.e. there are *no mismatching sides*) and the entire plane is covered. Any partial configuration which does not completely cover the plane but which cannot be extended while following the matching rules is ignored. Also, Wang tiling has no notion of time, or the growth of a tiling, but instead allows for the instantaneous appearance of an infinite pattern.

The aTAM differs in several ways. First, finite assemblies are often (and in fact usually) the desired goal, and the question typically being asked is whether *all* assembly sequences for a given system result in the desired output. Thus, if any assembly sequence which is possible in the system represents a valid growth path from its seed structure into an undesired assembly, including situations where the system “gets stuck” in a partial assembly which no longer allows tile attachments, the entire system (not just that assembly sequence) is considered to be incorrect. This differs from Wang tiling, where a partial (finite) assembly to which no tile additions can legally be made is completely discounted and left out of the set of producible tilings, and such a system is still considered “correct” as long as there exists *some* correct arrangement. Next, in the aTAM, tile attachments are allowed as long as edges with sufficient summed strength bind regardless of whether or not any remaining edges may be mismatched with adjacent tiles. The temperature parameter specifies a threshold for binding which every tile must meet before being able to attach to the assembly, making it crucial for at least one growth path to exist where each incomplete assembly has sufficiently many exposed glues in the necessary locations to allow further tiles to bind one by one. This differs from Wang tiling in which it is sufficient simply for there to exist some arrangement of tiles which could be perhaps simultaneously placed without mismatches. Furthermore, in the aTAM a seed tile or assembly is allowed to be defined as the starting point for an

assembly, thus guaranteeing its inclusion in any producible assembly, while in (traditional) Wang tiling no such seed is defined and any guarantee of inclusion of a particular tile in a tiling can only be enforced by careful design of the tile set.

Both models allow for a huge diversity of complex constructions and are capable of universal computation and the production of aperiodic structures. However, the differences cited above mean that the techniques used to design systems for each tend to be non-trivially different. In fact, the exact differences between the models, their similarities, and methods to transform results between each are still being explored.

Perhaps a model with more similarity to the aTAM is that of asynchronous and nondeterministic cellular automata (ACA). (See Chandesris (2011); Ingerson and Buvel (1984) as a few of the many references to ACA and what they are capable of.) A d -dimensional ACA consists of an infinite d -dimensional array of cells, where each cell maintains its own state but shares the same transition function (which causes the state of a cell to transition based on the state of that cell and the states of its neighbors). ACA allow the transition of one cell at a time, in arbitrary order. An aTAM system can be thought of as an ACA where there is exactly one state for each tile in the aTAM system plus one state representing an empty location, and all cells other than those representing the seed begin in the state which represents the empty position, while the states of the locations corresponding to the seed tiles begin in the states corresponding to the tiles of the seed. An empty cell adjacent to cells representing tiles can transition to a state which represents a particular tile type if the necessary glue bindings would occur for the corresponding tiles. The main difference between an ACA system simulating an aTAM system and a general ACA system is that each cell is allowed to transition from an empty location to a state representing a tile exactly one time, and then can never transition to another state again (reflecting the static nature of tiles that have joined an assembly). This one-time transition, representing the static nature of tiles and their permanent occupation of space once placed, plays a very important role in the design of aTAM systems and what they are capable of, as will be seen in later sections.

3.4 aTAM example: a binary counter

The aTAM is capable of Turing universal computation, so our first example will consist of a system which self-assembles a simple computation, namely an infinite binary counter. Figure 2a shows three tile types which will be used to form the boundary of the counter on its bottom and right sides. Figure 2b shows the additional 4 tile types needed to perform the actual counting and to display, via their labels, the bits of each number. We will define our binary counter TAS as $\mathcal{T} = \{T, (S, (0, 0)), 2\}$, that is, it

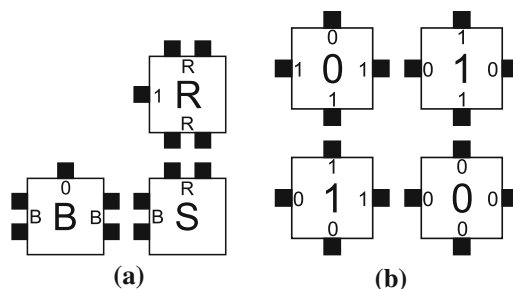


Fig. 2 This tile set, seeded with the S tile at $\tau = 2$, self-assembles into an infinite binary counter. **a** The tile types which form the border of the counter. **b** The “rule” tile types which compute and represent the values of the counter

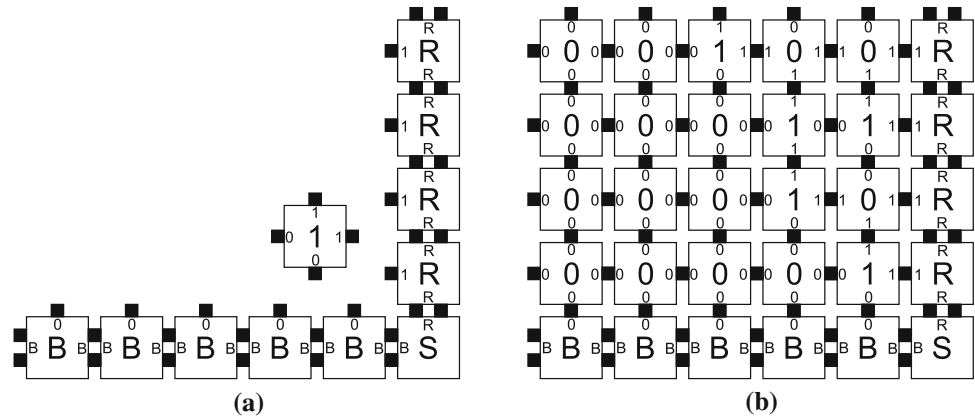
will consist of tile set T which will contain all 7 of the tile types defined in Fig. 2, it will have a seed consisting of a single copy of a tile of type S placed at position $(0, 0)$, and it will be a temperature 2 system (meaning that free tiles need to bind with at least a single strength-2 glue or two individual strength-1 glues on tiles within an existing assembly in order to attach to that assembly).

Figure 5 shows a small portion of the infinite assembly produced by \mathcal{T} . In Fig. 3a, the beginning of the formation of the border is shown. Starting from S , border tiles R can attach and form an infinite column upward using their strength-2 glues, and B tiles can do the same to the left. No rule tiles can attach until there are 2 strength-1 bonds correctly positioned for them to bind to. Figure 3a also shows the first rule tile which is about to attach into the corner. In Fig. 3b the bottom-right square of width and height 6 of the infinite square assembly is shown. Each horizontal row represents a single binary number in the counter, read from left to right (but which will have an infinite number of leading 0's to the left), and each row represents a binary number exactly one greater than the row immediately beneath it. The computation is performed by the rule tiles which, essentially, receive as “input” a bit from beneath (representing the current value of that column) and a bit from the right (representing the carry bit being brought in from the bit position which is immediately less significant). The labels and the northern glues of the rule tiles simply represent the (possibly new) “output” bit to be represented by that column (based on the two inputs), and the western glue represents the “output” carry bit which results. The computation is possible because of the “cooperation” between two tiles providing input, enforced by the system parameter temperature = 2 and the single-strength glues of the rule tiles.

3.5 Survey of aTAM results

Results in the aTAM can often be mapped into two groups: (1) What can or can't self-assemble?, and (2) How hard is

Fig. 3 Portions of the assembly formed by the binary counter. **a** Border tiles can attach to the seed and form arbitrarily long *bottom* and *right* borders. Rule tiles can bind only once two “inputs” are available. **b** A view of the 6×6 square of tiles at the *bottom right corner* of the assembly produced by the binary counter. Note that the terminal assembly would actually continue infinitely far up and to the *left*



it to self-assemble a particular object? Thus, sometimes the interest lies strictly in showing that something is possible or impossible, but often, even though we may know that something is possible, it turns out to be interesting to determine how efficiently it can be done. The most common measure of efficiency is the number of unique tile types required, which can be thought of as the size of the “program” being used to direct the assembly. Finding optimally small tile sets which self-assemble into targeted shapes is of great interest, both theoretically and also for the sake of making potential laboratory implementations more feasible. Another common measure is the scale factor. Oftentimes it is, perhaps counterintuitively, possible to design tile sets with many fewer tile types which can self-assemble a target shape at a blown up scaling factor than it is to self-assemble the same shape without scaling it up. Yet another measure may be assembly time. We now provide an overview of a series of results in the aTAM which seek to answer these and other questions.

3.5.1 Building $n \times n$ squares

Since Winfree (1998) showed in his thesis that the aTAM is computationally universal, we know that we can algorithmically direct the growth of assemblies. This ability allows for not only the creation of complicated and precise shapes, but also often for them to be created very tile type efficiently (i.e. they require small tile sets—those with few numbers of unique tile types). A benchmark problem for tile-based self-assembly is that of assembling an $n \times n$ square since this requires that the tiles somehow compute the value of n and thus “know when to stop” at the boundaries. In Rothemund and Winfree (2000) showed that binary counters can be used to guide the growth of squares and that thereby it is possible to self-assemble an $n \times n$ square using $O(\log n)$ tile types.

Figure 4 shows a high-level overview of the construction. Essentially, $\log n$ tile types are required so that each

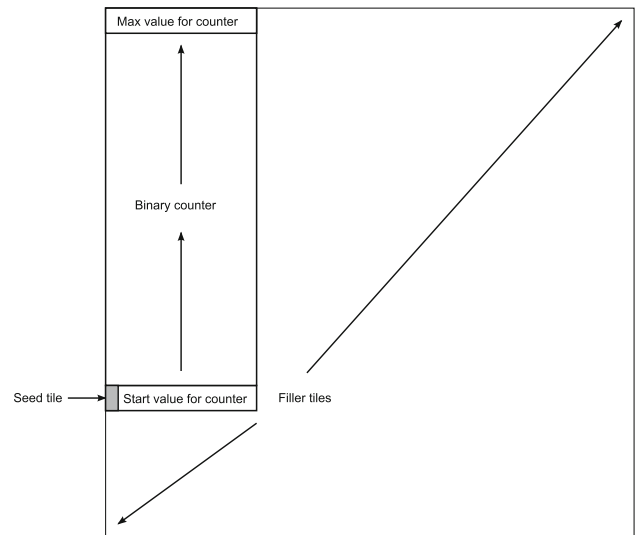


Fig. 4 The high level schematic for building an $n \times n$ square using $O(\log n)$ tile types

bit of (a number related to) the dimension n can be encoded with a unique tile type. The seed is taken to be one of those tile types so that the row of them forms attached to the seed. Above that, a fixed-width binary counter (which is composed of the same constant set of tile types regardless of n) begins counting upward from that value until it reaches its maximum possible value (i.e. all 1’s), at which point it terminates upward growth. With the vertical bar representing the counter in place, a very basic constant (for all n) set of tiles can be used to “pass a signal” along a diagonal path which is limited by the height (and width) of the counter, and to finally fill in below the diagonal to finish the formation of the square.

Adleman et al. (2001) improved the previous construction for squares to require the slightly fewer $O\left(\frac{\log n}{\log \log n}\right)$ tile types, which was also proven to be a matching lower bound (for almost all n) by using an information theoretic argument.

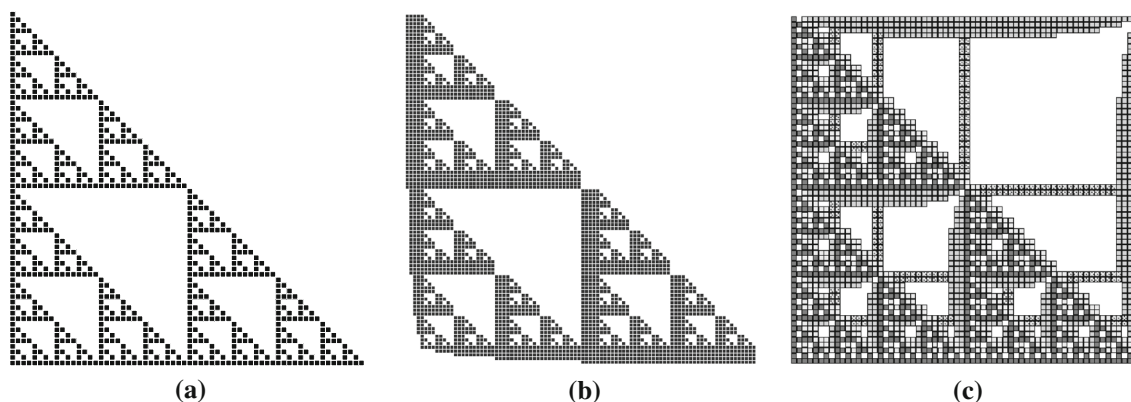


Fig. 5 Various patterns corresponding to the Sierpinski triangle. **a** A portion of the discrete Sierpinski triangle. **b** A portion of the approximate Sierpinski triangle of Lathrop et al. (2009). **c** A portion of the approximate Sierpinski triangle of Lutz and Shutters (2012)

Note that while squares can be quite efficiently self-assembled, the tile complexity of lines and rectangles differs. For instance, the tile complexity lower bounds for $1 \times n$ lines is n , and for $k \times n$ rectangles (where $k \leq n$) is $\frac{n^{1/k}}{k}$ [which was shown by Cheng et al. (2006)].

As another measure of efficiency, Becker et al. (2008) considered the time required for squares (and cubes) to self-assemble. Of course, for this they had to consider that growth of the assembly need not be constrained to single tile additions at each step (since in that case an $n \times n$ square would clearly take $n^2 - 1$ time steps to grow from a single seed tile), but instead used a model equivalent to one in which every tile which can individually attach at any given step simultaneously attaches (rather than having just one of them nondeterministically chosen as in the regular aTAM). They were able to produce constructions for time optimal assembly of $n \times n$ squares in $2n - 2$ assembly steps.

3.5.2 Building finite shapes

In order to build any given finite shape, it is trivial to define a TAS which will self-assemble it: simply create a unique tile type for every point in the shape so that the glue between each tile and each neighbor is unique to that pair in that location. Obviously, however, this is the worst possible construction in terms of tile type complexity. Soloveichik and Winfree (2007), showed that as long as the shape can be scaled up (meaning that every point in the original shape is replaced by a square block of tiles of some fixed dimension) the tile type complexity for a finite shape S is bounded above and below by the Kolmogorov complexity of the shape. The Kolmogorov complexity of S , denoted $K(S)$, is the length in bits of the shortest program which, when run by a universal Turing machine, outputs exactly the points of S and then halts. They showed that the tile complexity of S is $\Theta\left(\frac{K(S)}{\log K(S)}\right)$ by showing that the

lower bound holds because otherwise it would contradict the Kolmogorov complexity of the shape, and for the upper bound they provided a construction in which a Turing machine is simulated inside of each scaled up block to read a compressed definition of S and determine which neighboring locations should have blocks filled in and then passing the program into those blocks and simulating the Turing machine within them, etc. Therefore, the scaling factor c is proportional to the running time of the Turing machine (and thus can be very large), and the tile complexity arises from the compressed definition of S .

Another interesting aspect to the tile complexity of finite shapes was demonstrated by Bryans et al. (2011) where they showed that there exist finite shapes which can self-assemble more tile type efficiently by nondeterministic systems than by deterministic, or directed, systems. Both types of systems always create the exact same shape, but where a directed system does so by ensuring that no matter which assembly sequence is followed, a given location always receives a tile of the same type, a nondeterministic system may allow tiles of differing types to occupy a particular position based on the assembly sequence followed. They also showed that the problem of determining the minimum number of tile types which are required to uniquely assemble a given finite shape, if the system isn't constrained to being directed, is complete for the complexity class $\Sigma_2^P = NP^{NP}$, while it was shown by Adleman et al. (2002) to be NP-complete for directed systems. These results suggest that such nondeterminism adds power and complexity to the aTAM.

3.5.3 Building infinite shapes

As it has been shown that any finite shape can self-assemble in the aTAM, in order to test the limits of the model and find shapes which are impossible to self-assemble, it is

necessary to look at infinite shapes. While the self-assembly of infinite shapes may not have typical practical (i.e. physical, laboratory) applications, the study provides insights into fundamental limitations of self-assembling systems, in particular regarding their ability to propagate information through the growth fronts of assemblies.

Due to their complex, aperiodic nature, discrete self-similar fractals have provided an interesting set of infinite shapes to explore. Lathrop et al. (2009) showed that it is impossible for the discrete Sierpinski triangle (see Fig. 5a) to strictly self-assemble in the aTAM (at any temperature). Note that this is in contrast to the fact that it can weakly self-assemble, with a very simple tile set of 7 tile types. The proof relies on the fact that at each successive stage, as the stages of the fractal structure grow larger, each is connected to the rest of the assembly by a single tile. Since there are an infinite number of stages, all of different sizes, it is impossible for the single tiles connecting each of them to the assembly to transmit the information about how large the newly forming stage should be, and thus it is impossible for the fractal to self-assemble. Patitz and Summers (2010) extended this proof technique to cover a class of similar fractals. It is conjectured by the author of this paper that no discrete self-similar fractal strictly self-assembles in the aTAM, but that remains an open question.

Despite the impossibility of strictly self-assembling the discrete Sierpinski triangle, in 2009 it was shown that an approximation of that fractal, which the authors called the *fibred* Sierpinski triangle, does in fact strictly self-assemble. The fibred version is simply a rough visual approximation of the original but with one additional row and column of tiles added to each subsequent stage of the fractal during assembly (see Fig. 5b). Not only does the approximation look similar to the original, it was shown to have the same fractal (or zeta) dimension. In Patitz and Summers (2010), the fibering construction was extended to an entire class of fractals. Along a similar line, Shutters and Lutz (2012) showed that a different type of approximation of the Sierpinski triangle strictly self-assembles. This approximation also retains the same approximate appearance and fractal dimension, but instead of “spreading” out successive stages of the fractal with fibering, it utilizes a small portion of each hole in the definition of the shape (see Fig. 5c). Kautz and Shutters (2011) further extended this construction to an entire class of fractals.

Similar to their result about finite shapes mentioned in Sect. 3.5.2, Bryans et al. (2011) also showed a result about the power of nondeterminism in forming infinite structures, proving that there exist infinite shapes which can *only* self-assemble in non-deterministic systems. This means that no deterministic system is able to self-assemble such shapes, and is a further testament to the fact that nondeterminism is a source of increased power in the aTAM.

3.5.4 Performing computations

Early work in DNA computing by Adleman (1994) investigated the feasibility of using custom designed DNA molecules to solve NP-complete problems by performing massively parallel computations. The general concept is to have huge numbers of individual molecular complexes which nondeterministically each select a potential solution to a given instance of an NP-complete problem and then each perform the necessary computation to determine if the selected solution is correct. As long as there is a way to easily select the correct answers from the sea of failures, the hope was to provide a method to quickly solve such problems by harnessing the massive numbers of molecules which can compute in parallel. Adleman was able to solve a version of the Hamiltonian path problem for a graph of 7 vertices, proving the concept. Since then, a series of results by Brun (2008), Cheng et al. (2010), Cheng and Xiao (2012), and Wang et al. (2011) have continued to demonstrate the theoretical ability of the aTAM to solve such problems. Unfortunately, however, as the size of a problem instance approaches useful sizes (e.g. even a few hundred nodes for a graph problem), the exponential number of possible solutions inevitably destroys the utility of this approach, for reasonably-sized inputs requiring the number of assemblies to be at least equivalent to the number of particles in the universe.

While tile-based self-assembly may not be practically useful for solving computationally intractable NP-complete problems, there are still many other interesting problems to ask about its computational power. While the previously mentioned methods for solving such problems was to use many assemblies in parallel, it is interesting to consider what is possible for any individual assembly in terms of computation. Since the aTAM has been shown to be computationally universal, a single seeded assembly can simulate an arbitrary Turing machine. However, there are even more complicated computations which can be considered, and in doing so one of the fundamental characteristics of tile-based self-assembly is confronted: computational space, which is consumed by tiles attaching to an assembly, is analogous to write-once memory. Once a tile is placed, having performed its part of the computation (by converting the information encoded by its input glues into information encoded by its output glues), it can never change or be removed. This causes difficulties related to performing computations which are unique to such a physical model, and the following results have helped to uncover the complex ways in which geometry can be related to computation.

Patitz and Summers (2011) showed that a set of natural numbers $D \subseteq \mathbb{N}$ is decidable if and only if $D \times \{0\}$ and $D^c \times \{0\}$ weakly self-assemble. That is, the canonical

representations of D and the complement of D weakly self-assemble. For $D \times \{0\}$ to weakly self-assemble, at every point along the x -axis such that the value of the x coordinate is contained in D , the tile placed at that location is colored black. All other locations remain either untiled or receive a tile which is not black. The construction for Patitz and Summers (2011) is a relatively straightforward “stacking” of Turing machine simulations, so that a given Turing machine M which decides the language in question is first simulated on input 0, then immediately above that $M(1)$ is simulated, etc. As each simulation completes, the “answer” of whether or not that input is in the language is propagated via a one-tile-wide path down the side of the previous computations to the x -axis where the appropriately colored tile attaches.

Lathrop et al. (2011) answered the more complicated question of whether a similar result applied to computably enumerable (a.k.a. recursively enumerable) languages. They showed that a set of natural numbers $D \subseteq \mathbb{N}$ is computably enumerable if and only if the set $X_A = \{(f(n), 0) \mid n \in D\}$ weakly self-assembles (where f is a roughly quadratic function). For that construction, since any Turing machine M used to determine membership in D cannot be guaranteed to halt for non-members, the simple “stacking” construction cannot work. Instead, the construction performs the infinite series of computations side-by-side, spread out along the x -axis (hence the need for f), providing a potentially infinite amount of tape space for each computation while ensuring that none of them collide and a path to the relevant point on the x -axis always remains available for cases in which a black tile must be placed. The space reserved for each computation is achieved by a scheme in which each computation proceeds with each row simply copying the row beneath it for most rows, and then with a frequency half that of the computation to its immediate left, a row performs a new step of the computation. This, combined with a unique and well-defined slope for the assembly representing each computation ensures that the potentially infinite space requirements for every computation can be assured.

On the other hand, showing a limitation to the power of computation by self-assembly in the aTAM, in 2011 they showed there exist decidable sets of pairs of integers, or points (i.e. $D \subseteq \mathbb{Z}^2$), which do not weakly self-assemble in the aTAM. Their proof leverages the fact that space is not reusable in aTAM assembly, and that new space must therefore constantly be used to perform each subsequent step of a computation. They designed a pattern consisting of an infinite sequence of concentric diamonds which were centered on the origin and whose diameters were specified by a decidable set of natural numbers. By employing the time hierarchy theorem (Hartmanis and Stearns 1965), they

were able to show that there exist sets of diameters whose time complexity is so great (i.e. the amount of time required to compute whether a value is in the set) that if the pattern of diamonds with those diameters could self-assemble it would contradict the time complexity of the set. Essentially, the computation to determine whether or not the diamond at some particular diameter should be included in the pattern could not be performed by tiles from within that diamond and must therefore use space that may be required to mark subsequent diamonds. This result shows a limitation to the computational power of the aTAM, and the strong correlation between geometry and computation within it.

3.5.5 Speed of assembly

An important efficiency measure which we’ve discussed for several of the previous results is tile complexity. However, another interesting and important measure which has been investigated is the speed at which an assembly can form, or the number of assembly steps required by a system to reach the final, desired target structure. Of course, in the basic aTAM where each step of assembly consists of a single tile addition, the assembly time for a shape consisting of n points cannot vary, and is fixed at $n - s$ steps (where s is the number of tiles in the seed, usually 1). However, by considering slight variants of the model, such as a version where at each time step all tiles which are able to individually attach do so, the assembly time becomes variable and an interesting metric.

Adleman et al. (2001) proved that the deterministic assembly of a shape of diameter d requires time $\Omega(d)$. Doty and Chen (2012) showed that this bound also holds for nondeterministic systems. In 2001 they provided a matching upper bound for a construction which was able to self-assemble an $n \times n$ square and also used the optimal $O\left(\frac{\log n}{\log \log n}\right)$ tile types. See Sect. 3.5.1 for more discussion of assembly time related to $n \times n$ squares.

3.5.6 The influence of temperature

To this point, the example and results discussed have been largely based upon aTAM systems where the temperature parameter is 2. At temperature 2 and above, it is possible to design systems which make use of a feature commonly referred to as *cooperation* in which the prior placement of two tiles in specific relative positions is required before the attachment of a third tile is possible. This cooperative behavior is what is commonly attributed with providing the aTAM with its ability to perform computations, and disappears at temperature =1. Thus, for aTAM systems whose temperature is 1, it is conjectured that both: (1) Turing

universal computation by a deterministic aTAM system is impossible, and (2) any aTAM system which deterministically produces an $n \times n$ square requires a minimum of $2n - 1$ tile types. Partial progress toward the proof of these conjectures was achieved by Doty et al. (2011). Mañuch et al. (2010) also studied a variant of the problem, focusing on finite assemblies in which mismatches are not allowed, and proved that in such cases $\Omega(n)$ tile types are required to assemble a shape whose diameter is n . Nonetheless, the general problem remains open.

Despite the previous conjectures about the aTAM at temperature 1, it was shown by Cook et al. (2011) that, by slightly relaxing the requirements, Turing universal computation is in fact possible. Namely, if the assembly is allowed to proceed into the third-dimension, utilizing only 2 planes, or if the computation is allowed to prematurely terminate with some arbitrarily low probability, then a universal Turing machine can be simulated at temperature 1. (See Sect. 3.5.7 for related results.)

Moving in the other direction, Chen et al. (2011) showed that there exist TASs which require temperatures exponential in the number of tile types they contain. They showed that for every n , there exists a TAS with n tile types whose “behavior” cannot be preserved while using a temperature less than $2^{n/4}$, which means that it is not possible to modify the system to use a lower temperature while ensuring that all tiles are still only able to bind using the same subsets of sides, and produce the same result. For this result they utilize cooperative binding on 3 sides, which they call *3-cooperative*, as opposed to the *2-cooperative* systems previously discussed. It turns out that while 3-cooperativity results in systems which require temperatures exponential in the number of tile types they contain, 2-cooperative system only require temperatures linear in the number of tile types. (Note that these results are based on the assumption of integer strength glues, which is in fact how the aTAM is defined.) They also gave an algorithm which is able to find the minimal tile system to build an $n \times n$ square at any temperature in polynomial time. Further, Seki and Okuno (2012) show that given a temperature $\tau > 4$ and a shape, it is NP-hard to find the minimum TAS which assembles the shape at or below temperature τ , and that it is also NP-hard to find the optimal (lowest) temperature for a system for which the glue strengths and temperature are not specified, but the cooperative behaviors of the tiles are (i.e. how they can cooperate to form sufficient bonds).

3.5.7 Intrinsic universality

An intrinsically universal model is one which contains some system U , such that for any arbitrary system T within that model, U can be given a starting condition based on

T such that U will then completely simulate the behavior of T . That is, U will mimic all behaviors of T , but at a re-scaling in which each $n \times n$ block within U , for some $n \in \mathbb{N}$, can be mapped to a single element of T . Cellular automata and Turing machines are both examples of models which are intrinsically universal. While an aTAM system can be designed to simulate an arbitrary Turing machine, which could computationally simulate an arbitrary aTAM system, another interesting question was whether or not the aTAM is intrinsically universal, or: Is there a single tile set which can be used to simulate the behavior of any arbitrary aTAM system? Essentially, if the tiles of this “universal” tile set could be arranged to form a seed structure such that that structure contains an encoding of some other aTAM system, say \mathcal{T} , could additional copies of tiles from the universal tile set attach to grow into an assembly which simulates the system \mathcal{T} ? Of course, the simulation would be a scaled up version of the original system, but it must be the case that every behavior that \mathcal{T} is capable of, the simulating system is also capable of. Preliminary work by Doty et al. (2009) showed that for a constrained set of aTAM systems, namely those in which all tiles bind with exactly strength τ and there are no glue mismatches between adjacent tile edges, that class is intrinsically universal. Furthermore, it was later shown by Doty et al. (2012) that the entire, unconstrained class of aTAM systems is intrinsically universal. In fact, they demonstrated a tile set U and a method for using the definition of an arbitrary aTAM system \mathcal{T} of any temperature to form a seed structure for U so that the system with that seed, the tiles from U , and at temperature 2, can simulate \mathcal{T} . Thus, a single tile set in a properly seeded system at temperature 2 can simulate the behavior of any aTAM system at any temperature.

The previous result shows a powerful symmetry to the aTAM, since there is a system within it that can behave exactly like any other system within it. Meunier et al. (2013) showed that the temperature 2 parameter for systems using the intrinsically universal tile set is in fact a lower bound. They showed that no aTAM tile set exists which can simulate arbitrary aTAM systems of temperature >1 , while operating in a system of temperature 1, proving that the cooperative behavior provided by temperature 2 self-assembly can not be simulated at temperature 1. Further, their impossibility result extends to 3D, showing that even 3D temperature 1 aTAM systems cannot simulate 2D temperature 2 aTAM systems, which is contrasted with the facts that 3D temperature 1 systems are capable of universal computation (see Sect. 3.5.6, and the second result of (Meunier et al. 2013) shows that 3D temperature 1 systems can simulate arbitrary 2D temperature 1 systems. These results especially emphasize the fact that the power to perform universal computation does not imply the

power to simulate arbitrary behaviors of algorithmic self-assembly.

3.5.8 Verification of aTAM systems

Several “verification problems” (answering the question of whether or not a given system has a specific property) have been studied in relation to the aTAM, and characterized by their complexity. Among them are:

1. Does aTAM system \mathcal{T} uniquely produce a given assembly? This was shown to require time polynomial in the size of the assembly and tile set by Adleman et al. (2002).
2. Does aTAM system \mathcal{T} uniquely produce a given shape? This was shown to be in co-NP-complete for temperature 1 by Cannon et al. (2012) and co-NP-complete for temperature 2 (Cheng et al. 2005).
3. Is a given assembly terminal in aTAM system \mathcal{T} ? This was shown to require time linear in the size of the assembly and tile set in Adleman et al. (2002).
4. Given an aTAM system \mathcal{T} , does it produce a finite terminal assembly? An infinite terminal assembly? These were both shown to be uncomputable in Cannon et al. (2012).

3.5.9 PATS problem and tile set generation

In order to produce a surface with a complex template for potentially guiding the attachment of functional materials, an interesting problem in tile-based self-assembly is the Patterned self-Assembly Tile set Synthesis (PATS) problem. The PATS problem is concerned with finding the minimal tile set which will self-assemble into a given 2-D pattern of colors (where tile types are assumed to be assigned colors) and was introduced by Ma and Lombardi (2008). Göös and Orponen (2010) presented an exhaustive branch-and-bound algorithm which works well for finding exact solutions to patterns of sizes up to 6×6 , and approximate solutions for larger patterns. Lempiäinen et al. (2011) modified the previous algorithm to be more efficient (but still require exponential time). Czeizler and Popa (2012) proved that the PATS problem is NP-hard, and Seki (2013) examined the parameterized version of the problem, *c*-PATS, in which any given pattern is guaranteed to contain at most *c* colors, and showed that 59-PATS is NP-hard by using a 3-SAT reduction.

3.5.10 Simulators and programming tools

In order to visualize complex constructions and to help verify their correctness, several simulators have been

developed and released to the research community. Included among them are Winfree’s xgrow, which simulates the aTAM as well as the kTAM (see Sect. 4), and Patitz’s (2009) ISU TAS, which simulates the aTAM (in 2-D and 3-D), kTAM, and 2HAM (see Sect. 5) as well as providing a graphical tile type editor. The xgrow simulator is specifically designed to accommodate a wide variety of options for experimentally accurate kTAM simulations, while ISU TAS is designed with more of an emphasis on aTAM simulations and ease of use for beginners, while also allowing for larger tile sets and simulated assemblies.

Since the generation of large tile sets can be tedious, difficult, and error-prone, work has been done to abstract some of the high-level notions utilized by researchers developing tile sets and to turn those into tools which can be used to algorithmically generate tile sets. In particular, the idea of “signals” propagating through an assembly, as a series of glue bindings which propagate a particular piece of information, has been studied. Becker (2009) showed how to design systems of signals for given sets of shapes and then how to transform the defined signals into tile sets which self-assemble into those shapes. Doty and Patitz (2009) exploited a similar notion of signal propagation, combined with the notion of tiles performing computations based on input signals and providing the output to the computations in the form of output signals. They developed a *domain specific* programming language which could be used to programmatically generate tile sets and also created a graphical editor for designing systems using their language.

4 The kinetic Tile Assembly Model (kTAM)

In reality, DNA tile self-assembly is a more complicated process than that modeled by the aTAM, and therefore a different model is required for a realistic simulation of the physical process of self-assembling DNA tiles. Whereas the aTAM is a great model for studying the capabilities and limitations of tile assembly, and for programming tile sets to understand issues related to computation and geometry, the *kinetic Tile Assembly Model* (kTAM) (Winfree 1998) was developed as a more physically realistic model for laboratory settings, and considers the reversible nature of self-assembly, factoring in the rates of association and dissociation of basic molecular elements (so-called monomers, or tiles) within the original framework provided by the aTAM. The kTAM describes the dynamics of assembly according to a set of reversible chemical reactions: A tile can attach to an assembly anywhere that it makes even a weak bond, and any tile can dissociate from the assembly at a rate dependent on the total strength with which it adheres to the assembly. In this section, we first give a more formal definition of the kTAM, then describe the

types of errors that it captures, and then discuss several results which have successfully demonstrated methods for reducing those errors. Techniques such as those discussed below have been responsible for a rapid and steady decline in the frequency of errors seen in laboratory experiments, plummeting from error rates of 10 % per tile in 2004 to only 0.13 % by 2009, and continuing to shrink.

4.1 Model definition

In the kTAM (Fujibayashi et al. 2009; Winfree 1998; Winfree and Bekbolatov 2003), a monomer tile can be added to the assembly with some association (forward) rate, or removed from the assembly with some dissociation (reverse) rate. Similar to the aTAM, only the singleton tiles are allowed to attach to, and in this case detach from, a seeded assembly. These rates are denoted by r_f and $r_{r,b}$, respectively. At every available site on the perimeter of an assembly (i.e. the *frontier*), every possible monomer tile can associate to the assembly, regardless of whether the monomer is correct or not (i.e. whether or not the glues match). The forward rate depends only on the monomer tile concentration, $[monomer]$:

$$r_f = k_f [monomer] = k_f e^{-G_{mc}} \quad (1)$$

where $G_{mc} > 0$ is the non-dimensional entropic cost of associating to an assembly. In the kTAM, for simplicity it is assumed that tile concentrations remain constant at $[monomer] = e^{-G_{mc}}$. Therefore, since the forward rate constant k_f is a constant, the entire forward rate r_f is also constant.

The reverse rate is dependent upon the binding strength b of the tile to the assembly, and in fact the relationship is exponential:

$$r_{r,b} = k_{r,b} = k_f e^{-bG_{se}} \quad (2)$$

where G_{se} is the non-dimensional free energy cost of breaking a single bond and b is the number of “single-strength” bonds the tile has made.

The kTAM’s equivalent to the aTAM’s temperature τ parameter is the ratio of the concentration of the tiles to the strength of their individual bonds, or G_{mc}/G_{se} . As a simplifying assumption, the tile concentrations are considered to remain constant during assembly (despite the fact that singleton tiles will be transitioning from freely floating individual tiles to being attached to growing assemblies), which in turn causes the temperature parameter to remain constant. (It should be noted that despite this and other simplifying assumptions, the kTAM does in fact provide a quite accurate model of the systems observed in laboratory settings.) Because the kTAM allows for the binding of tiles whether or not their glues correctly match those on the boundary of a growing assembly, bindings which would be

considered errors in the aTAM are possible. By lowering the ratio of G_{mc}/G_{se} , which is intuitively similar to lowering the temperature τ threshold in the aTAM, assembly happens more quickly but is more error prone. If the number of correct bonds that a tile has with an assembly, b , is less than τ , then a tile is more likely to detach than to attach.

Because the kTAM accurately models the behavior of DNA based tile self-assembly in the laboratory, most especially the common types of errors observed, it has provided an excellent foundation for work in error prevention and correction.

4.2 Error types

In order to discuss the types of errors that can occur during self-assembly in the kTAM, we will refer to an example system which is designed to weakly self-assemble the Sierpinski triangle. See Fig. 6 for details.

The errors that occur during assembly can be divided into three general types: (1) *growth errors* (or *mismatch errors*), (2) *facet errors*, and (3) *nucleation errors* (Fujibayashi et al. 2009). A growth error, an example of which can be seen in Fig. 7, occurs when one or more sides of a tile which binds to an assembly have glues which do not match the adjacent glues (called *glue mismatches*). Such a tile may bind with insufficient strength to remain permanently bound, but before it has an opportunity to dissociate, a previously unoccupied neighboring position may be filled by a tile which binds without mismatches, thus resulting in an assembly where every tile has sufficient strength to remain permanently attached despite the mismatch. This essentially “locks” the incorrect tile into place and potentially allows assembly to proceed with an incorrectly placed tile which may cause further deviations from the desired shape or pattern. Somewhat similarly, a facet error also occurs on the edge of a growing assembly. A facet error (see Fig. 8 for an example) again occurs when a tile binds with insufficient strength for permanent attachment (but this time with no mismatches), and again is locked into place by a subsequent tile addition. The third type of errors, nucleation errors, occur when tiles aggregate with each other without any attachment to the seed structure, and thus “seed” a new type of assembly.

4.3 Survey of kTAM results

The ability of the kTAM to accurately model the errors seen in laboratory settings coupled with its clean theoretical definition make it an ideal model in which to study mechanisms of error prevention and correction. Additionally, the algorithmic nature of self-assembly in the kTAM provides the opportunity to effectively apply a variety of

Fig. 6 Details of the Sierpinski triangle example. **a** The tile types for weakly self-assembling the Sierpinski triangle. **b** A view of the 9×9 square of tiles at the *bottom right corner* of the weakly self-assembled Sierpinski triangle. Note that the terminal assembly would actually continue infinitely far up and to the *left*

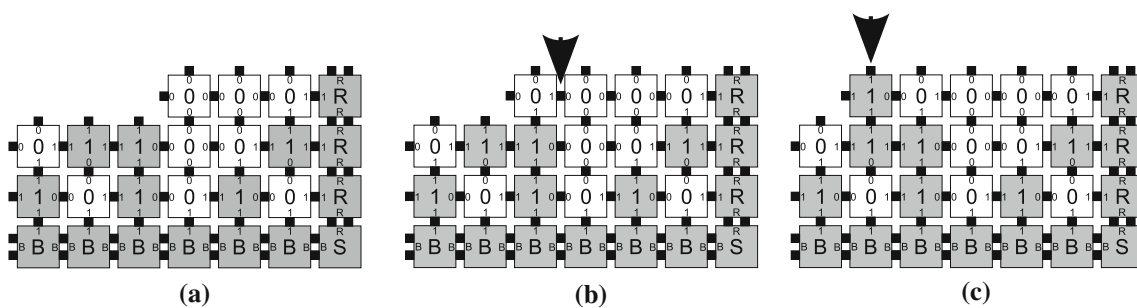
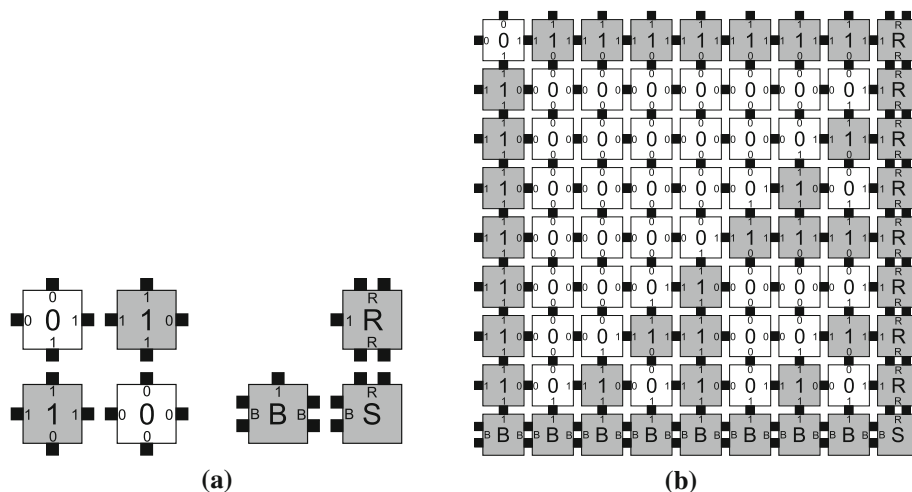


Fig. 7 Example growth error in the kTAM: a tile initially binds with insufficient strength due to a mismatch, but the error is then “locked in” by a tile which arrives later. **a** A partial assembly which is error-free. **b** The binding of a tile with one glue match and one mismatch

(shown by *arrow*). **c** Before the erroneously attached tile can detach, another tile (shown by *arrow*) attaches with 2 matching bonds so that all tiles are now connected by two correctly formed bonds

algorithms from seemingly unrelated fields such as data transmission to make kTAM systems more robust.

While simply adjusting the ratio of G_{mc} to G_{se} is sufficient to drive error rates arbitrarily low, that comes at the cost of a huge slow-down to the overall assembly process. We now provide a brief overview of some results in the kTAM which are focused on one or both of the dual goals of decreasing the rate of errors during assembly and minimizing assembly time. Note that there are several laboratory experiments which utilize novel techniques aimed at meeting these and other goals which are omitted from this discussion.

4.3.1 Error suppression via block replacement

Kinetic proofreading, which was independently discovered by Hopfield (1974) and Ninio (1975), is an error correcting mechanism employed by a variety of biological processes (e.g. RNA to protein translation) where a sequence of steps are utilized such that the process must progress through each, with step each “testing”, or helping to ensure, the

correctness of the last step. Winfree and Bekbolatov (2003) demonstrated such a technique (which they simply called *proofreading*) to reduce growth errors in the kTAM. In proofreading, individual tile types are replaced by $n \times n$ blocks of unique tile types such that the perimeter of the $n \times n$ block formed by them represents the same glues as the original single tile. (New glues are created for the interior of the block which are specific to the tile types composing each particular block.) However, those original glues are now split into n separate glues. The goal is to force multiple errors to occur before an incorrect $n \times n$ block can fully form, as opposed to the single error which would allow the analogous incorrect tile from the original tile set to bind. They found that by increasing n , it is possible to reduce the growth errors—or alternatively to increase the speed of assembly while maintaining the same error rate.

For this example, we construct two of the substitutions for the 2×2 proofreading tile set for the Sierpinski triangle (shown in its original form in Fig. 6a). In Fig. 9, two of the tiles from the original set are replaced by 4 tiles

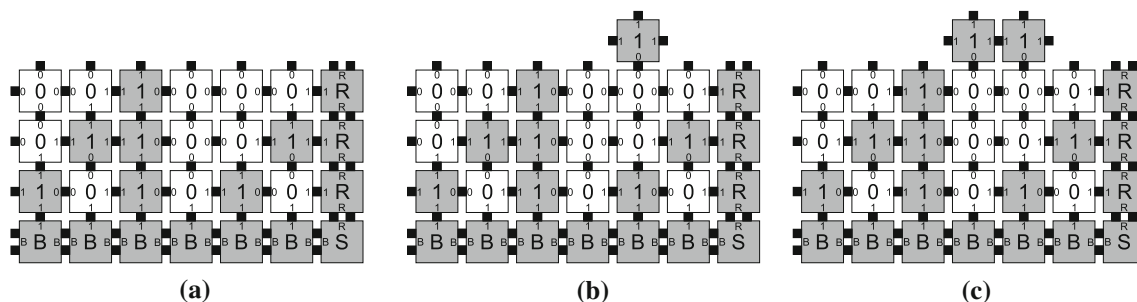


Fig. 8 Example facet error in the kTAM. **a** A partial assembly which is error-free. **b** The binding of a tile via a single glue. **c** Before the erroneously attached tile can detach, another tile attaches with 2 matching bonds so that all tiles are now connected by two correctly formed bonds

each. Note how each group of 4 tiles forms a 2×2 block whose perimeter has versions of the glues from the original tile in corresponding locations. For example, the 0 glue on the south side of each original tile is now represented by two separate glues, 0_L and 0_R , which correspond to the left 0 glue of each 2×2 block, and the right 0 glue. The glues on adjacent edges of tiles in the interior of blocks are replaced by new glues which are specific to each such location.

The tile set resulting from such a transformation reduces errors in the following way. In order for an incorrect block to assemble in a given location (i.e. a block which doesn't match the input from one of the two input directions), it must have more than one tile originally bind with only strength 1. Each of those tiles must then get “locked in” by subsequent tile attachments before falling off. Since it is much more unlikely for multiple instances of such errors to be locked in before detachment than it is for one, the overall likelihood of error is smaller for the proofreading system.

The block replacement scheme necessarily imposes a scaling factor on the transformed, more error resistant system, making a trade off of resolution for correctness. Reif, Sahu, and Yin (Majumder et al. 2007b) introduced a scheme of *compact proofreading* in which no scaling factor is required. However, the tradeoff imposed by their transformation is an increase in tile complexity, in fact an exponential increase. Unfortunately, it turns out that any compact proofreading scheme would, for the general case (ignoring a relatively small set of special cases), require such an exponential explosion, and this was proven by Soloveichik and Winfree (2005).

4.3.2 Facet error handling

Winfree and Bekbolatov (2003), the proofreading technique previously discussed was sufficient to reduce growth errors, but was ineffective for handling facet errors. These types of errors were more common in systems “whose

growth process[es] intrinsically involve facets”, meaning that they frequently require growth to be initiated by extending from a flat surface. In order to reduce these errors, Winfree and Bekbolatov were able to redesign a system used to build an $n \times n$ square by changing the pattern of growth to one which avoids large facets. Specifically, the design used to build the square in Fig. 4 was redesigned so that, instead of using a single binary counter growing along one side and then filler tiles which are dependent upon facet growth, two binary counters were used to form two sides of the square and then filler tiles which use cooperative attachments between those walls. These modifications (along with a few other small changes) were able to greatly reduce the incidence of errors in the growth of squares.

4.3.3 Snaked proofreading

Chen and Goel (2004) demonstrated a tile set transformation which provided improvements over the previous proofreading technique. In fact, their *snaked proofreading* technique not only provides substantial improvements in error correction, it also provides for “provably good” assembly time, or specifically that it allows for close to linear assembly time (within a logarithmic factor of irreversible error-free growth). Snaked proofreading relies on a block replacement scheme similar to the proofreading of Winfree and Bekbolatov (2003), but with a different internal bond structure. An example of the difference can be seen in Fig. 10. The general technique is to force multiple insufficient attachments to occur and be locked into place before an error can persist.

Especially notable is the fact that snaked proofreading does not only provide benefits in simulations of the kTAM, but Chen et al. (2007b) actually created a tile set which utilized the technique and experimented with it in a wet-lab setting. They created tile sets which self-assembled into long ribbons, some which were designed to implement snaked proofreading and some which were not, and were

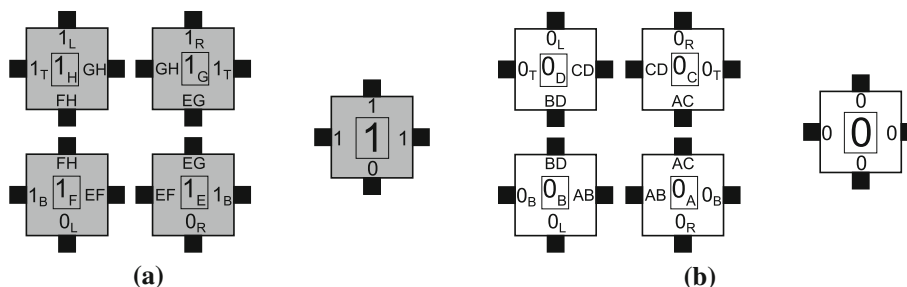


Fig. 9 Example tile substitution by 2×2 blocks of tiles for proofreading. Each image shows a single rule tile for the Sierpinski triangle tile set on the *right* and the 4 tiles which replace it in the proofreading tile set on the *left*. **a** Replacing a “0 – 1” rule tile. **b** Replacing a “0 – 0” rule tile

able to verify via atomic force microscopy that the snaked proofreading tile sets experienced a 4-fold reduction in facet nucleation errors.

4.3.4 Self-healing

First studied by Winfree in 2006, the notion of self-healing is that in which a growing assembly is damaged (perhaps by the removal of a group of tiles somewhere in its interior) but then it can correctly re-grow to “heal” the damage without allowing internal errors. The major problem is that many computations are not reversible (meaning that if you’re given the output from the computation, you can’t know for sure what the inputs were), but when an assembly whose normal forward growth is determined by such a computation (e.g. the system forming the Sierpinski triangle pattern, which performs the *xor* operation) receives such damage, it is likely to re-grow on all edges of the hole. Thus, it will attempt to grow “backwards” in some areas, with tiles attaching to the assembly using their “output” sides, causing nondeterministic choices for the inputs to the computational steps represented by those tiles, frequently resulting in mistakes.

Soloveichik et al. (2008) showed that both proofreading and self-healing properties can be incorporated into tile set transformations which make them robust to both problems simultaneously. In a remarkable example of self-healing, Chen et al. (2007a) demonstrated a method to allow an entire $n \times n$ square to regrow from *any subassembly* which has at least one dimension which is $2 \log n$ or greater. With scaling, they can apply their technique to general shapes.

4.3.5 Enhanced tile design

While the above (and other) work has successfully demonstrated several techniques for reducing errors that occur during DNA tile-based self-assembly, they have all done so without allowing for the modification of the basic structures of the tiles themselves. However, the simple and

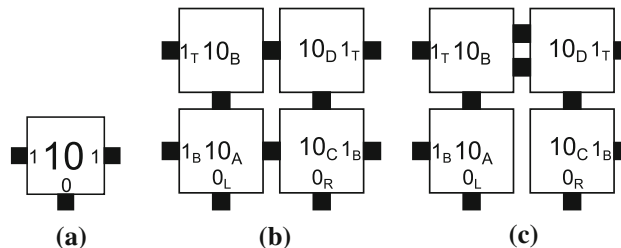


Fig. 10 A comparison of the block replacement transformations used in standard proofreading and snaked proofreading. **a** A tile type from the original, unaltered tile set. **b** The block used as a replacement in standard proofreading. **c** The block used as a replacement in snaked proofreading

static nature of DNA tiles lends itself to the possibility of extension.

Majumder et al. (2007a) proposed such an extension. Namely, the authors defined a model in which the “input” glues of tiles are “active” (that is, free to bind to complementary glue strands) when the tiles are freely floating in solution, but their “output” glues are “inactive” (this is, prevented from forming bonds). Only once a tile has associated to an assembly and bound with its input sides are its output sides activated. They presented a theoretical model of such systems and showed that they provide instances of compact (i.e. not requiring scaling factors over the original tile set), error-resilient, and self-healing assembly systems. Furthermore, they provided a possible physical implementation for such systems using DNA polymerase enzymes and strand displacement.

Fujibayashi et al. (2009) introduced a similar approach in order to provide for both error-resilience and fast speed of assembly. The *Protected Tile Mechanism* and the *Layered Tile Mechanism*, which utilize stand displacement, were presented. These mechanisms make use of additional DNA strands which “protect”, or cover, glues either partially or fully. By balancing the length of the glue strands available for binding on input and output sides at various stages of tile binding, they were able to demonstrate—via simulation—that these mechanisms can in fact improve error rates while maintaining fast assembly.

4.3.6 Controlling nucleation

Another major source of potential errors is caused by spurious nucleation, or the formation of an assembly separate from and not containing the designated seed structure. Typically, systems are designed so that growth begins from a seed which essentially provides the initial input to the algorithm that directs the growth of the assembly. Thus, when an assembly forms in the absence of the seed, it has the possibility of “running the algorithm” starting at a random point and as though from a random input (and perhaps even running it in reverse as well from that point).

In order to combat the problem of spurious nucleation, Schulman and Winfree (2009) designed systems which were able to quickly grow from seeded assemblies, but which were highly unlikely to form large unseeded assemblies due to high kinetic barriers. The “zig-zag” systems they introduced grow as fixed-width ribbons (i.e. long and thin rectangles) such that to increase the length of the strip, the assembly must grow a row first in one direction across the growing end, and then the next row grows back in the opposite direction. They were able to provide simulations to prove the effectiveness of their systems at resisting spurious nucleation and yet growing relatively quickly and correctly from seeds.

To further improve the ability of experimentalists to create seed structures, especially those which contain a reasonable amount of information used to direct a growing assembly, Barish et al. (2009) demonstrated, in a wet-lab, the use of DNA origami (introduced by Rothmund in 2006) to serve as a seed structure. They were able to design DNA origami seeds which displayed up to 32 glues and binding sites to which tiles could attach, and these seeds provided the relatively easy production of high-yield, low error-rate TASs. They were able to successfully demonstrate the use of DNA origami seeds to nucleate the growth of three different algorithmically directed systems. (See Sect. 6.3 for another example of tiles created using DNA origami.)

5 The 2-Handed Assembly Model (2HAM)

5.1 Informal model description

The 2HAM (Cheng et al. 2005; Demaine et al. 2008) is a generalization of the aTAM meant to model systems where self-assembly of multiple sub-assemblies can occur separately and in parallel, and then those sub-assemblies can combine with each other. The “2-handed” portion of the name comes from the fact that each combination is of exactly two assemblies at a time. Note that variations of this model have appeared in several papers and by several

different names (e.g. hierarchical self-assembly, polyominoes, etc.) (Adleman 2000; Adleman et al. 2001; Cheng et al. 2005; Demaine et al. 2012; Luhrs 2008; Winfree 2006). We now give a brief, informal, sketch of the 2HAM.

The 2HAM is formulated without a seed structure, so that all individual tiles have equal status in the initial solution, and assembly begins as separate assemblies nucleate in parallel. Each step of assembly occurs as any two existing assemblies (which at first are just the singleton tiles) which are able to bind to each other, with strength at least equal to the temperature parameter and without any overlaps, combine to form a new assembly. Since it is experimentally challenging to enforce the seeded nature of growth in the aTAM (see Sect. 4.3.6), the 2HAM provides a perhaps more experimentally feasible model in that respect, by removing the seed constraint. However, since the 2HAM allows for pairs of arbitrarily large assemblies to combine with each other as long as there are no overlaps of any portions of those assemblies in the final configuration, two new difficulties arise in terms of experimental viability. First, the rate of diffusion of assemblies will decrease as their sizes increase, making it less and less likely for combinations of larger assemblies to occur. Second, in order to enforce the requirement that pairs of assemblies can only join in configurations in which they don’t contain overlaps, it would need to be the case that assemblies are completely rigid (which is certainly not the case with DNA implementations of tiles) so that portions of the assemblies couldn’t bend to avoid the overlaps. The fact that the 2HAM allows for the combination of arbitrarily large assemblies gives rise to the phenomenon that, although all interactions are local in the context of being between exactly two assemblies which are immediately adjacent to each other, there is also a notion of instantaneous long range interactions on the scale of individual tiles. This is because the existence of a tile at a location arbitrarily far from another can dictate whether or not that tile will be able to bind to a tile in another assembly by perhaps providing enough cooperative binding, or instead perhaps by blocking the assemblies from achieving a binding configuration. This long range interaction provides for a great amount of difference in the power of the 2HAM versus the aTAM, and is also the reason that the 2HAM isn’t immediately similar to ACA systems (see Sect. 3.3).

A *supertile* (a.k.a., *assembly*) is a positioning of tiles on the integer lattice \mathbb{Z}^2 . Two adjacent tiles in a supertile *interact* if the glues on their abutting sides are equal and have positive strength. Each supertile induces a *binding graph*, a grid graph whose vertices are tiles, with an edge between two tiles if they interact. The supertile is τ -*stable* if every cut of its binding graph has strength at least τ , where the weight of an edge is the strength of the glue it

represents. That is, the supertile is stable if at least energy τ is required to separate the supertile into two parts. A 2HAM TAS is a pair $\mathcal{T} = (T, \tau)$, where T is a finite tile set and τ is the *temperature*, usually 1 or 2. Given a TAS $\mathcal{T} = (T, \tau)$, a supertile is *producible*, written as $\alpha \in \mathcal{A}[\mathcal{T}]$ if either it is a single tile from T , or it is the τ -stable result of translating two producible assemblies without overlap.¹ A supertile α is *terminal*, written as $\alpha \in \mathcal{A}_{\square}[\mathcal{T}]$ if for every producible supertile β , α and β cannot be τ -stably attached. A TAS is *directed* if it has only one terminal, producible supertile. Given a connected shape $X \subseteq \mathbb{Z}^2$, we say a TAS \mathcal{T} *self-assembles* X if every producible, terminal supertile places tiles exactly on those positions in X (appropriately translated if necessary).

5.2 Formal model definition

We now give a more formal definition of the 2HAM. For most readers, the informal description of Sect. 1 should be sufficient and the more technical description in this section can be skipped.

Two assemblies α and β are *disjoint* if $\text{dom}\alpha \cap \text{dom}\beta = \emptyset$. For two assemblies α and β , define the *union* $\alpha \cup \beta$ to be the assembly defined for all $\vec{x} \in \mathbb{Z}^2$ by $(\alpha \cup \beta)(\vec{x}) = \alpha(\vec{x})$ if $\alpha(\vec{x})$ is defined, and $(\alpha \cup \beta)(\vec{x}) = \beta(\vec{x})$ otherwise. Say that this union is *disjoint* if α and β are disjoint.

The *binding graph* of an assembly α is the grid graph $G_{\alpha} = (V, E)$, where $V = \text{dom } \alpha$, and $\{\vec{m}, \vec{n}\} \in E$ if and only if (1) $\vec{m} - \vec{n} \in U_2$, (2) $\text{label}_{\alpha(\vec{m})}(\vec{n} - \vec{m}) = \text{label}_{\alpha(\vec{n})}(\vec{m} - \vec{n})$, and (3) $\text{str}_{\alpha(\vec{m})}(\vec{n} - \vec{m}) > 0$. Given $\tau \in \mathbb{N}$, an assembly is τ -*stable* (or simply *stable* if τ is understood from context), if it cannot be broken up into smaller assemblies without breaking bonds of total strength at least τ ; i.e., if every cut of G_{α} has weight at least τ , where the weight of an edge is the strength of the glue it represents. In contrast to the model of Wang tiling, the nonnegativity of the strength function implies that glue mismatches between adjacent tiles do not prevent a tile from binding to an assembly, so long as sufficient binding strength is received from the (other) sides of the tile at which the glues match.

For assemblies $\alpha, \beta : \mathbb{Z}^2 \rightarrow T$ and $\vec{u} \in \mathbb{Z}^2$, we write $\alpha + \vec{u}$ to denote the assembly defined for all $\vec{x} \in \mathbb{Z}^2$ by $(\alpha + \vec{u})(\vec{x}) = \alpha(\vec{x} - \vec{u})$, and write $\alpha \simeq \beta$ if there exists \vec{u} such that $\alpha + \vec{u} = \beta$; i.e., if α is a translation of β . Define the *supertile* of α to be the set $\tilde{\alpha} = \{\beta | \alpha \simeq \beta\}$. A supertile $\tilde{\alpha}$ is τ -*stable* (or simply *stable*) if all of the assemblies it contains are τ -stable; equivalently, $\tilde{\alpha}$ is stable if it contains a stable assembly, since translation preserves the property of stability. Note also that the notation $|\tilde{\alpha}| \equiv |\alpha|$ is the size of

the super tile (i.e., number of tiles in the supertile) is well-defined, since translation preserves cardinality (and note in particular that even though we define $\tilde{\alpha}$ as a set, $|\tilde{\alpha}|$ does not denote the cardinality of this set, which is always \aleph_0).

For two supertiles $\tilde{\alpha}$ and $\tilde{\beta}$, and temperature $\tau \in \mathbb{N}$, define the *combination* set $C_{\tilde{\alpha}, \tilde{\beta}}^{\tau}$ to be the set of all supertiles $\tilde{\gamma}$ such that there exist $\alpha \in \tilde{\alpha}$ and $\beta \in \tilde{\beta}$ such that (1) α and β are disjoint (steric protection), (2) $\gamma \equiv \alpha \cup \beta$ is τ -stable, and (3) $\gamma \in \tilde{\gamma}$. That is, $C_{\tilde{\alpha}, \tilde{\beta}}^{\tau}$ is the set of all τ -stable supertiles that can be obtained by attaching $\tilde{\alpha}$ to $\tilde{\beta}$ stably, with $|C_{\tilde{\alpha}, \tilde{\beta}}^{\tau}| > 1$ if there is more than one position at which β could attach stably to α .

It is common with seeded assembly to stipulate an infinite number of copies of each tile, but our definition allows for a finite number of tiles as well. Our definition also allows for the growth of infinite assemblies and finite assemblies to be captured by a single definition, similar to the definitions of Lathrop et al. (2009) for seeded assembly.

Given a set of tiles T , define a *state* S of T to be a multiset of supertiles, or equivalently, S is a function mapping supertiles of T to $\mathbb{N} \cup \{\infty\}$, indicating the multiplicity of each supertile in the state. We therefore write $\tilde{\alpha} \in S$ if and only if $S(\tilde{\alpha}) > 0$.

A (*two-handed*) *tile assembly system* (TAS) is an ordered triple $\mathcal{T} = (T, S, \tau)$, where T is a finite set of tile types, S is the *initial state*, and $\tau \in \mathbb{N}$ is the temperature. If not stated otherwise, assume that the initial state S is defined $S(\tilde{\alpha}) = \infty$ for all supertiles $\tilde{\alpha}$ such that $|\tilde{\alpha}| = 1$, and $S(\tilde{\beta}) = 0$ for all other supertiles $\tilde{\beta}$. That is, S is the state consisting of a countably infinite number of copies of each individual tile type from T , and no other supertiles. In such a case we write $\mathcal{T} = (T, \tau)$ to indicate that \mathcal{T} uses the default initial state.

Given a TAS $\mathcal{T} = (T, S, \tau)$, define an *assembly sequence* of \mathcal{T} to be a sequence of states $\vec{S} = (S_i | 0 \leq i < k)$ (where $k = \infty$ if \vec{S} is an infinite assembly sequence), and S_{i+1} is constrained based on S_i in the following way: There exist supertiles $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ such that (1) $\tilde{\gamma} \in C_{\tilde{\alpha}, \tilde{\beta}}^{\tau}$, (2) $S_{i+1}(\tilde{\gamma}) = S_i(\tilde{\gamma}) + 1$,² (3) if $\tilde{\alpha} \neq \tilde{\beta}$, then $S_{i+1}(\tilde{\alpha}) = S_i(\tilde{\alpha}) - 1, S_{i+1}(\tilde{\beta}) = S_i(\tilde{\beta}) - 1$, otherwise if $\tilde{\alpha} = \tilde{\beta}$, then $S_{i+1}(\tilde{\alpha}) = S_i(\tilde{\alpha}) - 2$, and (4) $S_{i+1}(\tilde{\omega}) = S_i(\tilde{\omega})$ for all $\tilde{\omega} \notin \{\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}\}$. That is, S_{i+1} is obtained from S_i by picking two supertiles from S_i that can attach to each other, and attaching them, thereby decreasing the count of the two reactant supertiles and increasing the count of the product supertile. If $S_0 = S$, we say that \vec{S} is *nascent*.

¹ The restriction on overlap is a formalization of the physical mechanism of steric protection.

² with the convention that $\infty = \infty + 1 = \infty - 1$

Given an assembly sequence $\vec{S} = (S_i \mid 0 \leq i < k)$ of $\mathcal{T} = (T, S, \tau)$ and a supertile $\tilde{\gamma} \in S_i$ for some i , define the *predecessors* of $\tilde{\gamma}$ in \vec{S} to be the multiset $\text{pred}_{\vec{S}}(\tilde{\gamma}) = \{\tilde{\alpha}, \tilde{\beta}\}$ if $\tilde{\alpha}, \tilde{\beta} \in S_{i-1}$ and $\tilde{\alpha}$ and $\tilde{\beta}$ attached to create $\tilde{\gamma}$ at step i of the assembly sequence, and define $\text{pred}_{\vec{S}}(\tilde{\gamma}) = \{\tilde{\gamma}\}$ otherwise. Define the *successor* of $\tilde{\gamma}$ in \vec{S} to be $\text{succ}_{\vec{S}}(\tilde{\gamma}) = \tilde{\alpha}$ if $\tilde{\gamma}$ is a predecessor of $\tilde{\alpha}$ in \vec{S} , and define $\text{succ}_{\vec{S}}(\tilde{\gamma}) = \tilde{\gamma}$ otherwise. A sequence of supertiles $\vec{\alpha} = (\tilde{\alpha}_i \mid 0 \leq i < k)$ is a *supertile assembly sequence* of \mathcal{T} if there is an assembly sequence $\vec{S} = (S_i \mid 0 \leq i < k)$ of \mathcal{T} such that, for all $1 \leq i < k$, $\text{succ}_{\vec{S}}(\tilde{\alpha}_{i-1}) = \tilde{\alpha}_i$, and $\vec{\alpha}$ is *nascent* if \vec{S} is nascent.

The *result* of a supertile assembly sequence $\vec{\alpha}$ is the unique supertile $\text{res}(\vec{\alpha})$ such that there exist an assembly $\alpha \in \text{res}(\vec{\alpha})$ and, for each $0 \leq i < k$, assemblies $\alpha_i \in \tilde{\alpha}_i$ such that $\text{dom} \alpha = \bigcup_{0 \leq i < k} \text{dom} \alpha_i$ and, for each $0 \leq i < k$, $\alpha_i \sqsubseteq \alpha$. For all supertiles $\tilde{\alpha}, \tilde{\beta}$, we write $\tilde{\alpha} \rightarrow_{\mathcal{T}} \tilde{\beta}$ (or $\tilde{\alpha} \rightarrow \tilde{\beta}$ when \mathcal{T} is clear from context) to denote that there is a supertile assembly sequence $\vec{\alpha} = (\tilde{\alpha}_i \mid 0 \leq i < k)$ such that $\tilde{\alpha}_0 = \tilde{\alpha}$ and $\text{res}(\vec{\alpha}) = \tilde{\beta}$. It can be shown using the techniques of Rothemund (2001) for seeded systems that for all two-handed TASs \mathcal{T} supplying an infinite number of each tile type, $\rightarrow_{\mathcal{T}}$ is a transitive, reflexive relation on supertiles of \mathcal{T} . We write $\tilde{\alpha} \rightarrow_{\mathcal{T}}^1 \tilde{\beta}$ ($\tilde{\alpha} \rightarrow^1 \tilde{\beta}$) to denote an assembly sequence of length 1 from $\tilde{\alpha}$ to $\tilde{\beta}$ and $\tilde{\alpha} \rightarrow_{\mathcal{T}}^{\leq 1} \tilde{\beta}$ ($\tilde{\alpha} \rightarrow^{\leq 1} \tilde{\beta}$) to denote an assembly sequence of length 1 from $\tilde{\alpha}$ to $\tilde{\beta}$ if $\tilde{\alpha} \neq \tilde{\beta}$, and otherwise (i.e. $\tilde{\alpha} = \tilde{\beta}$) an assembly sequence of length 0.

A supertile $\tilde{\alpha}$ is *producible*, and we write $\tilde{\alpha} \in \mathcal{A}[\mathcal{T}]$, if it is the result of a nascent supertile assembly sequence. A supertile $\tilde{\alpha}$ is *terminal* if, for all producible supertiles $\tilde{\beta}$, $C_{\tilde{\alpha}, \tilde{\beta}}^{\tau} = \emptyset$.³ Define $\mathcal{A}_{\square}[\mathcal{T}] \subseteq \mathcal{A}[\mathcal{T}]$ to be the set of terminal and producible supertiles of \mathcal{T} . \mathcal{T} is *directed* (a.k.a., *deterministic, confluent*) if $|\mathcal{A}_{\square}[\mathcal{T}]| = 1$.

Let $X \subseteq \mathbb{Z}^2$ be a shape. We say X *self-assembles* in \mathcal{T} if, for each $\tilde{\alpha} \in \mathcal{A}_{\square}[\mathcal{T}]$, there exists $\alpha \in \tilde{\alpha}$ such that $\text{dom} \alpha = X$; i.e., \mathcal{T} uniquely assembles into the shape X .

5.2.1 An example 2HAM system

In this section we provide an example of a simple 2HAM system and show exactly what assemblies are producible within it in order to help clarify the ways in which assemblies are produced within the model.

³ Note that a supertile $\tilde{\alpha}$ could be non-terminal in the sense that there is a producible supertile $\tilde{\beta}$ such that $C_{\tilde{\alpha}, \tilde{\beta}}^{\tau} \neq \emptyset$, yet it may not be possible to produce $\tilde{\alpha}$ and $\tilde{\beta}$ simultaneously if some tile types are given finite initial counts, implying that $\tilde{\alpha}$ cannot be “grown” despite being non-terminal. If the count of each tile type in the initial state is ∞ , then all producible supertiles are producible from any state, and the concept of terminal becomes synonymous with “not able to grow”, since it would always be possible to use the abundant supply of tiles to assemble $\tilde{\beta}$ alongside $\tilde{\alpha}$ and then attach them.

Let $\mathcal{T} = (T, 2)$ be a 2HAM system where T is defined as the tile types in Fig. 11a. Figures 11a–12c show the complete set of 29 supertiles which make up $\mathcal{A}[\mathcal{T}]$, and Fig. 12c shows the single member of $\mathcal{A}_{\square}[\mathcal{T}]$. The producible supertiles are broken into groups to show the earliest step of combinations during which they can appear, although for some there are multiple paths of combinations which can form them. (We don’t show duplicate copies.) Furthermore, recall from the definition of the model that all producible supertiles are available at every step, so for example a supertile produced in step 2 may combine with one produced in step 1 to create a new supertile in step 3. Also note that the use of “steps” is merely a convenience for discussing this example, but typically the sets $\mathcal{A}[\mathcal{T}]$ and $\mathcal{A}_{\square}[\mathcal{T}]$ are simply defined as those supertiles producible in the limit.

5.3 Survey of 2HAM results

We now provide a brief, incomplete review of some results in the 2HAM.

5.3.1 Simulation of the aTAM

The aTAM assumes a controlled, well-defined origin for the initiation of all assemblies, while the 2HAM allows for “spontaneous” nucleation caused by any two producible assemblies (including singleton tiles) which can bind with sufficient strength. Given this much greater level of freedom, the question of whether or not that could be constrained and forced to behave in a way similar to the aTAM was asked by Cannon et al. (2012). The answer was “yes”, and in fact in Cannon et al. (2012) a construction was presented which, given an arbitrary aTAM system \mathcal{T} , provides for a way to construct a 2HAM system \mathcal{S} which can faithfully simulate \mathcal{T} ; the cost is a constant scaling factor of 5. The general technique is to allow \mathcal{S} to form 5×5 blocks which represent the tiles in \mathcal{T} but in a very constrained way so that the blocks can only fully form and present their output glues once they’ve attached to a growing assembly which contains a seed block (and therefore they can’t spontaneously combine away from the “seeded” assembly). This result is especially notable since, as long as the constant scaling factor is allowed, it shows that any seeded growth of the aTAM can be simulated by a system in the unseeded 2HAM, making it unnecessary for the model itself to enforce a particular starting point for growth, but instead each system can be designed to enforce a well-defined starting point of growth, if desired.

5.3.2 Intrinsic universality in the 2HAM

The existence of a single tile set which can be configured to simulate any aTAM system was discussed in Sect. 3.5.7. In contrast, Demaine et al. (2013) proved that no such tile set

Fig. 11 An example 2HAM system and some producible assemblies. **a** The tile set (a.k.a. singleton tiles) for the 2HAM example system. **b** The new supertiles producible after one step of combinations

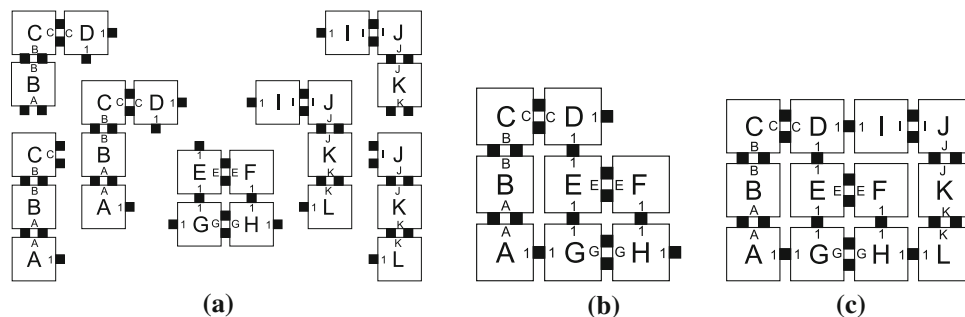
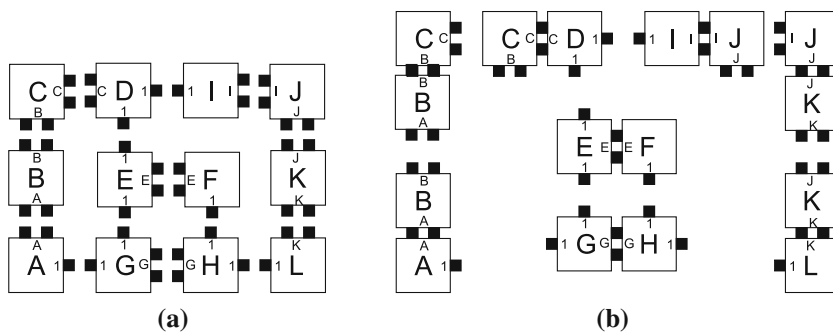


Fig. 12 Continuation of the example 2HAM system’s producible assemblies. **a** The new supertiles producible after the second step of combinations. **b** The only new supertile producible after the third step

of combinations. **c** The only new supertile producible after the fourth step of combinations, and which is the unique terminal assembly of the system

exists for the 2HAM. More precisely, they showed that for every 2HAM system at temperature τ , there exists some system at temperature $\tau + 1$ which cannot be simulated by it. Their proof is based upon the ability of the 2HAM to simultaneously utilize the binding of multiple glues positioned on tiles arbitrarily far apart. They describe a system which produces assemblies which look like ladders that each have $\tau + 1$ rungs and form in such a way that each half of a ladder with all $\tau + 1$ half-rungs must fully form before binding to a complementary half-ladder to form a full ladder. They then prove that any system whose temperature is $< \tau + 1$ can’t simulate such a system, since it would have to “fake” the binding of one or more rungs and must therefore also be able to form ladders which have $< \tau + 1$ rungs. Since the original system couldn’t form these, the simulator can’t correctly simulate it.

While the entire 2HAM is not intrinsically universal, in Demaine et al. (2013) they went on to show how, for each individual $\tau > 1$, the class of 2HAM systems at τ is intrinsically universal. This means that for each temperature τ , there exists a single tile set which can simulate all 2HAM systems at temperature τ . Their constructions showed a variety of tradeoffs in the number of unique input supertiles required for each simulation, their sizes, and the scale factor of the simulations. For their final result, they exhibited a construction which provides a single tile set for each τ

which requires no input supertiles and which simultaneously and in parallel simulates every 2HAM system at temperature τ .

As a corollary to their results related to intrinsic universality in the 2HAM, the authors of Demaine et al. (2013) show that within the 2HAM there is an infinite set of infinite hierarchies of 2HAM systems with strictly increasing power within each hierarchy, which creates a much more complex landscape than the fully unifying result of Doty et al. (2012) for the aTAM!

5.3.3 Verification of 2HAM systems

Given that the 2HAM allows for a greater variety of behaviors than the aTAM, and in fact in some sense for the transmission of information over arbitrary distances (by the placements of glues and general geometric shapes of arbitrarily large supertiles which are combining), it shouldn’t be surprising that many verification problems are more difficult for the 2HAM than for the aTAM (see Sect. 3.5.8).

Several verification problems have been characterized in terms of their complexity, some of which include:

1. Does 2HAM system \mathcal{T} uniquely produce a given assembly? This was shown to be co-NP-complete for 3D temperature 2 systems in the 2HAM Cannon et al.

- (2012). The complexity of this verification problem is still open in 2D. (But note that it is solvable in polynomial time in the aTAM in both 2D and 3D.)
2. Does 2HAM system \mathcal{T} uniquely produce a given shape? This was shown to be in co-NP for temperature 1 and co-NP-complete for temperature 2 by Cheng et al. (2005).
 3. Is a given assembly terminal in 2HAM system \mathcal{T} ? In Cannon et al. (2012) this was shown to be uncomputable for temperature 2 systems in the 2HAM [while it is computable in polynomial time in the aTAM (Adleman et al. 2012), and also for the 2HAM at temperature 1 (Cannon et al. 2012).]
 4. Given a 2HAM system \mathcal{T} , does it produce a finite terminal assembly? This was shown to be uncomputable in Cannon et al. (2012).
 5. Given a 2HAM system \mathcal{T} , does it produce an infinite terminal assembly? This was shown to be uncomputable for temperature 2 2HAM systems in Cannon et al. (2012).

5.3.4 Impossibility and efficiency comparisons with the aTAM

Given that the 2HAM can simulate the aTAM (and that the converse is not true), it seems that the 2HAM is more powerful. Thus, it may be somewhat surprising that in Cannon et al. (2012) it was shown that there is a simple class of shapes (so-called *loops*) which can be assembled with slightly greater tile type efficiency in the aTAM at temperature 1 than in the 2HAM at temperature 1. (However, this separation disappears at temperature 2.) Nonetheless, in Cannon et al. (2012) it was also shown that there are shapes called *staircases* which can self-assemble in the 2HAM using roughly n tile types, while the aTAM requires a number exponential in n (and this can in fact be extended to the busy beaver function, $BB(n)$). In terms of impossibility, it was shown that there is a class of infinite shapes which self-assembles in the aTAM but not the 2HAM, and also a class of shapes which can self-assemble (in a weaker sense) in the 2HAM but not in the aTAM.

5.3.5 Speed of assembly

Since the 2HAM allows for assemblies to begin forming in parallel and then to combine in pairs, it would seem that perhaps this would allow for sublinear assembly times. However, Chen and Doty (2012) developed a physically inspired timing model for the 2HAM (referred to there as the *Hierarchical aTAM*) and showed that it is impossible to build shapes of diameter n in time less than $\Omega(n)$ in deterministic systems under that timing model. Nonetheless,

they then exhibited a nondeterministic system which can assemble an $n \times n'$ rectangle (where $n > n'$) in time $O(n^{4/5} \log n)$, breaking the linear-time lower bound (which applies not only to deterministic 2HAM systems, but also to seeded aTAM systems as mentioned in Sect. 3.5.5).

5.3.6 Fuzzy temperature fault tolerance

Recall that the 2HAM allows for the nucleation of an assembly by any pair of tiles with binding strength equal to the temperature. It therefore seems that self-assembly in the 2HAM at temperature 1, where every pair of matching glues on any pair of tile edges is sufficient to initiate the growth of an assembly, is doomed to either (1) make nothing but the most simple of periodic structures, or (2) require tile complexity equivalent to the number of points in the desired shape. However, temperature 2 assembly in the 2HAM is computationally universal, so, as in the aTAM, the question becomes: is temperature 1 provably strictly weaker? While that remains an open question, Doty et al. (2010) introduced a variation to the model where the temperature parameter isn't fixed, but instead can drift between 1 and 2, staying at one or the other for arbitrarily long. However, there is a guarantee that the temperature will eventually at some point return to 2 and stay there for arbitrarily long. They called this model *fuzzy temperature*, and showed that they could develop systems which exhibited strong fault-tolerance in such conditions (meaning that they were guaranteed to always produce the correct assembly) while building $n \times n$ squares using only $O(\log n)$ tile types. To obtain this fault tolerance, the construction had to ensure that any unintended growth that occurred during a phase of temperature 1 could not become stably "locked in" at temperature 2, meaning that they would always have to dissolve when the temperature raised.

6 Newer Models

The wide variety of previously discussed results in the aTAM and 2HAM have helped researchers to develop a much stronger understanding of the fundamental powers and limitations of systems in which:

1. The entire growth process is guided solely by local interactions of the constituent components, with no global source of information input.
2. The glues of tiles are simple in that they each bind with a positive-valued integer strength bond to only other copies of the same glue and have no interaction with other glues.
3. The properties of individual tiles (i.e. the glues they possess, their shapes, etc.) are fixed and unchanging.

4. Once a tile attaches to an assembly it never detaches.

While much power has been demonstrated for systems within these restrictive models, they are still bounded by several fundamental limitations (some of which have been shown). In order to both determine if and how such theoretical limitations may be surpassed, as well as to help guide the design of artificial self-assembling systems within the laboratory, a large array of derivative models have been defined and developed. The goal for many of these models is to find (at least theoretically) plausible methods of removing one or more of the above restrictions. Thus, these models tend to be more powerful, and an important aspect of studying such models is to carefully characterize the differences between them and the original models, along with the provably different powers of the models. In this way, we gain a much better understanding of which powers are afforded by which properties of self-assembling systems. Such understanding is theoretically very interesting, but also can help with the design of laboratory systems by providing insight into which properties are most valuable and the tradeoffs between them.

In this section, we provide high-level descriptions of several models which have been developed to extend the aTAM and/or 2HAM, along with a few of the results in those models. We attempt to characterize the fundamental differences between these models and to show how their powers differ.

6.1 Probabilistic assembly and concentration programming

It was previously discussed in Sect. 3.5.2 that nondeterminism provides additional power over deterministic systems in the aTAM. Extending work along that front, Chandran et al. (2009), introduced the Probabilistic Tile Assembly Model (PTAM) in which a tile set is defined as a multi-set of tiles (meaning that more than one tile of each type can be included in the tile set), and at each step of assembly a tile is chosen with uniform probability from the tiles in that multi-set. They were able to effectively harness nondeterminism by designing PTAM systems where tiles of more than one type can bind at many points during the assembly, and demonstrated a construction which produces lines of expected length n using a tile set with a mere $\Theta(\log n)$ tiles, along with a matching lower bound of $\Omega(\log n)$. This is an enormous improvement over the lower bound of n in the aTAM, at the cost of a bit of imprecision due to variance in the length of the lines actually produced. Furthermore, by introducing a variant of the PTAM in which each edge of a tile can have multiple glues and allowing binding to occur between tiles as long as a single glue matches, they were able to lower both of those bounds by a factor of $\log \log n$, i.e. to $\Theta(\frac{\log n}{\log \log n})$ and $\Omega(\frac{\log n}{\log \log n})$.

In the standard aTAM, the “program” that is being executed during self-assembly can be thought of as being specified by the specific tile types of the system. It is the information encoded in the glues that direct the behavior of the system and guide assembly. Also, it is assumed that not only do the concentrations of free tiles in solution remain constant during assembly (clearly a simplifying assumption as long as new tiles are not added to the solution during assembly), it is also assumed that tiles of all types have the same concentration. *Tile concentration programming*, introduced by Becker et al. (2006), allows for the manipulation of tile concentrations and thus the inclusion of additional information as input to a TAS as the relative concentrations of the various tile types (somewhat similar to the multiplicity of tile types within PTAM tile sets). This can be thought of as a global source of information as the concentrations are set as global ratios between tile types. This tool has been used for reducing both assembly time and the frequency of errors in the kTAM. It has also been used in a variant of the aTAM to provide nondeterministic “competitions” between tiles of different types for binding at specified locations. The results of these competitions can be used by the system to sample the relative concentrations of the tile types and thus “read” the input information thus provided.

In a series of results by Becker et al. (2006) to Kao and Schweller (2008) to Doty (2010), it was shown how to use this information to efficiently build shapes such as squares with increasing precision. Most recently, Doty (2010) showed how to combine tile concentration programming with a constant tile set to form any $n \times n$ square with high probability (for sufficiently large n), and also how to self-assemble arbitrary scaled shapes using a constant tile set and tile type concentrations dependent upon the definition of the shape.

Adleman et al. (2002) examined the effects of varying the relative concentrations of tile types in order to optimize assembly time and provided an algorithm to find the tile type concentrations which approximate the minimum expected assembly time within a $O(\log n)$ factor. Jang et al. (2006), and Chen and Kao (2011), studied the effects of varying concentrations on both error prevention and assembly time and found that it is possible to improve both. Chen and Kao (2011), they showed that the rate of growth errors is minimized by setting the concentration of tiles of type T_i proportional to the square root of the number of times that tiles of type T_i appear in the final assembly (outside of the seed structure). Further, by using those concentrations the expected assembly time is also minimized for constrained systems where the size of the growth frontier (i.e. the number of locations where a tile can attach correctly and with sufficient strength) is limited to 1 at all times. (Note that such systems, although constrained, have been shown to be computationally universal.)

6.2 Staged self-assembly

Self-assembly in the aTAM is considered a “one pot” reaction, meaning that all assembly for a given system occurs in one test tube. Furthermore, during the entire assembly process all tile types are present. Demaine et al. (2008) defined the *Staged Tile Assembly Model* in which arbitrary subsets of tile types and previously produced assemblies can be placed into distinct test tubes, or bins, for portions of the assembly process. Once the assemblies in each bin have reached terminal states, it is possible to combine or separate the contents of bins and individual tile types into new bins, and perform the next *stage* of assembly. (During each stage, assemblies are allowed to combine as in the 2HAM.) This increases the resources required for a self-assembling system, but provides additional input in the form of the staging algorithm (the definition of the series of stages) and dramatically increases in the power of such systems. For instance, in Demaine et al. (2008) they were able to demonstrate that a constant tile set can be used to self-assemble arbitrary shapes—with no scaling! This construction requires a number of bins and stages dependent on the particular shape, and they presented a variety of constructions which exhibited tradeoffs between the number of tile types, number of bins, number of stages, and scaling factor.

In a more recent paper, Demaine et al. (2012) studied the problem of assembling labeled $1 \times n$ lines, i.e. lines where each position is assigned a character from a given alphabet and thus each assembly represents a string over that alphabet. Tile labels were used to represent the characters. They considered the original formulation of the staged model in which the output of each mixing operation is restricted to being a single terminal assembly, and also a version in which multiple terminal assemblies can be produced by each. They were able to show that in both versions, the minimum number of stages required is within a constant factor of the size of the smallest context-free grammar which generates exactly the string.

A simplification of staged self-assembly (and introduced before it), *step assembly* was introduced by Reif (1999) as a model where only one bin is used, but with the additional constraint that assembly growth can only happen one tile at a time, similar to the aTAM. Mañuch et al. (2009) showed that in step assembly 24 tile types are sufficient to assemble any shape at a scale factor of 2, but with a number of steps proportional to the number of points in the shape. Also, Behsaz et al. (2012) showed that both staged and step assembly models are Turing universal at temperature 1.

6.3 Geometrically complex tiles

Work in the aTAM is generally done with the assumption of a “diagonal” glue function, which means that the

function that maps the strength of interaction between pairs of glues returns a 0 for all pairs of glues where both are not the same glue type, and a positive number for pairs of glues of matching type. Given such a glue function, which is the standard, as previously mentioned the lower bound on the unique number of tile types which can self-assemble an $n \times n$ square is $O\left(\frac{\log n}{\log \log n}\right)$. However, for a non-diagonal (or flexible) glue function, which is one that allows interactions between each glue type and any subset of other glue types, that lower bound falls to $\sqrt{\log n}$ as shown by Cheng et al. (2005). In order to provide a potentially realistic means of implementing non-diagonal glue functions, the Geometric Tile Assembly Model (GTAM) was introduced by Fu et al. (2012), and a series of constructions in the GTAM were presented which:

1. Self-assemble an $n \times n$ square in the optimal $O(\sqrt{\log n})$ tile types and at temperature 1
2. Simulate a computationally universal class of temperature 2 aTAM constructions at temperature 1
3. In a 2-handed version of the GTAM (and allowing 4 planes to be used in the third dimension), self-assemble an $n \times n$ square using only $O(\log \log n)$ tile types

In experimental work, Woo and Rothmund in 2011 fabricated tiles from DNA origami, but rather than relying on Watson–Crick base pairing as the bonding mechanism between tile edges, they instead relied solely on non-specific blunt-end stacking interactions and instead enforced specificity by making use of geometrically diverse edges which enforced shape complementarity, using a methodology quite similar to that of Fu et al. (2012).

6.4 Tiles which are not square

Kari et al. (2012) studied systems with tiles of various shapes, including triangular and hexagonal. They showed that by some definition of simulate, systems of square tiles and systems of triangular tiles are not able to simulate each other. Beyond comparing the abilities of the variously shaped tiles to simulate each other, they also provided some constructions showing what triangular and hexagonal tiles are capable of (for instance, triangular tiles can be used to tile the Sierpinski triangle).

In a further departure from the standard square, and rigid, tile type utilized in the aTAM, Jonoska et al. (1999) introduced *flexible* tiles, which (unlike the *rigid* tiles of the aTAM) have their glues on the ends of bendable arms. This allows for much more complex patterns of binding across tiles by removing the standard geometric restrictions imposed by planarity in the aTAM, allowing each glue of one tile to bend into different positions to bind with the

glues of neighboring tiles in a variety of locations. Using flexible tiles, Jonoska and McColm (2006) were able to show that computations performed by rigid tiles can (within certain polynomial restrictions) be simulated by systems of flexible tiles. Furthermore, in Jonoska and McColm (2009) they proved the power of various flexible tile systems in terms of corresponding nondeterministic complexity classes (while restricting them to bounded numbers of tiles).

6.4.1 Tile which are not square and are rotatable

Most tile-based models stipulate that tiles do not rotate, and for standard square tiles the additional freedom of rotation offers no increase in power. However, the question of whether or not additional power comes from allowing non-square tiles to rotate was first addressed by Demaine et al. (2012). In Demaine et al. (2012) they showed how to create a *single* rotatable tile type which can simulate *any* aTAM system \mathcal{T} as long as a seed encoding information about \mathcal{T} and consisting of copies of that single tile type is provided. Part of this construction relies on the intrinsic universality result of Doty et al. (2012), and their rotatable tile type is nearly circular with a number of sides proportional to the number of tiles in the tile set of Doty et al. (2012), with each side having a small set of geometric bumps and dents. Alternatively, they showed how to take any aTAM system \mathcal{T}' and create a single rotatable tile type so that a system consisting of *no seed structure* and only copies of the rotatable tile type will simulate \mathcal{T}' . They were able to extend their construction to apply to Wang tile systems so that any Wang tile system can be simulated by a single rotatable tile type as long as they are placed on a hexagonal grid and small gaps between tiles are allowed. These results provide the first single-tile systems which are capable of universal computation and aperiodic tilings, while using only local binding rules. As a tool to achieve their constructions, they also proved that any aTAM system of square tiles can be converted into a system of hexagonal tiles which simulate it but without containing any τ -strength glues (and requiring a seed of size 3).

In contrast to their positive results, Demaine et al. (2012) proved that any system composed of a single tile type which can only translate but not rotate, and with no seed structure, must either form infinite structures or not grow at all. However, if given a seed structure such a system is capable of simulating 1D cellular automata for a limited number of steps.

6.5 Dynamic models

The previously discussed models, despite having many differences, have a fundamental similarity. They are all

“static” in terms of the behaviors of tiles once they attach to an assembly, meaning that all bonds, once formed, remain permanently, and no properties of the tiles (other than potential un-bound edges becoming bound later) ever change. The next several models provide mechanisms where either bonds can be broken, allowing assemblies to break apart, or some properties of tiles themselves can change. This additional power opens the door to several new construction techniques and for many of the limitations of static models to be overcome.

6.5.1 Temperature programming

The multiple temperature model, or *temperature programming*, introduced by Cheng et al. (2005), is a variant of the seeded aTAM which allows for the temperature of the system to be changed (raised or lowered) during the assembly process and at specified points. More specifically, a series of temperature transitions, along with the tile set, seed, and initial temperature, are specified to define a temperature programming system. Assembly progresses from the seed until the assembly is terminal. At that point, the first temperature transition is made and assembly continues until it is terminal. If another temperature transition has been specified it is made and assembly once again continues, and so on until assembly is terminal and no additional temperature transitions have been specified.

Somewhat akin to tile concentration programming, temperature programming provides a way to supply information globally to the system. The addition of a series of temperature transitions as input turns out to be a powerful tool, and, in Cheng et al. (2005) they used it to demonstrate how to build $n \times n$ squares, for any given n , using a constant tile set and $O(\log n)$ temperature changes. Summers (2012) extended those results to show that there exist systems using one of two constant tile sets that can self-assemble scaled-up versions of arbitrary shapes. One system uses a larger scaling factor dependent upon the shape but a “Kolmogorov-optimum” temperature sequence, while the other uses a small, constant scaling factor but a temperature sequence proportional to the number of points in the shape. Summers also proved that there exists no single tile set which can self-assemble an arbitrary shape in this model without scaling.

Using the power of the model to split assemblies apart, in Cheng et al. (2005) they showed that with one temperature change it is possible to self-assemble *thin* rectangles (i.e. $n \times k$ where $k < \frac{\log n}{\log \log n - \log \log \log n}$) using only $O(\frac{\log n}{\log \log n})$ tile types, which beats the lower bound of $\Omega(\frac{n^{1/k}}{k})$ for the aTAM.

6.5.2 Repulsive glues

In the aTAM, all pairs of glues interact with either a positive (i.e. attractive) force when the glues match, or no force at all when the glues do not match. However, in natural systems there is also another option: a negative (i.e. repulsive) force. For instance, two objects with the same electric charge (or two magnets whose same poles are brought together) will repel each other. Several variations of models allowing so-called *negative glues* have been defined, along with a series of related results.

Calling it the self-destructive graph assembly model, Reif et al. (2006) studied systems with repulsive glues and the problem of the sequential construction of a target assembly's binding graph. They showed that the complexity of this problem is PSPACE-complete. Doty et al. (2013) studied a slightly different version of a model allowing repulsive glues [and a nice description of a set of variations can be found in the appendix of Doty et al. (2013)], and were able to show that repulsive glues do not allow for unlimited reuse of tiles and the growth of an assembly must necessarily “lock in” a number of tiles proportional to the number of tile addition steps. They also showed how to simulate a s -space bounded and t -time bounded Turing machine while keeping the size of all assemblies bounded by $O(s)$ rather than the $O(st)$ bound required by aTAM constructions. Patitz et al. (2011), defined a model (that they called the *restricted glue* Tile Assembly Model, or rgTAM) in which 1. only diagonal glue functions are allowed, 2. the absolute value of every glue strength is 1, and 3. only one single glue type with a repulsive force is allowed in any tile set. They then provided a construction which efficiently self-assembles an $n \times n$ square using $O(\log n)$ tile types [improved in Patitz et al. (2012)] to $O(\frac{\log n}{\log \log n})$, and one which simulates an arbitrary Turing machine, showing that the model is computationally universal. Most recently, Schweller and Sherman (2013) presented a construction using a diagonal glue function which can simulate an arbitrary Turing machine in a fuel-efficient manner (see Sect. 6.5.4 for a description of fuel-efficient Turing machines).

6.5.3 Staged assembly with RNase

As an extension to staged self-assembly as discussed in Sect. 6.2, Abel et al. (2010) defined a model in which tile types are created out of two different materials (e.g. DNA and RNA), and then it is possible to dissolve one type (e.g. RNA tiles) at specified points during the assembly (e.g. by the addition of an RNase enzyme).

Using the additional power of this model, they were able to develop systems in which the input was not only a set of

tiles, but also assemblies of unspecified hole-free shapes. Their systems then replicate the shapes of the input assemblies, so that the outputs of their systems are assemblies whose shapes are the same as the shapes of the input assemblies. They were able to develop constructions which are capable of producing an exact number of copies of the input shape, as well as constructions which produce infinitely many copies of the shapes. They also showed how to vary the constructions to use either a constant number of tile types and $\log(n)$ stages (where n is the number of copies to make in the finite case, or the number of boundary corners of the shape in the case of infinite replication), or $\log n$ tile types and a constant number of stages.

In later work, Demaine et al. (2011) showed how to self-assemble arbitrary shapes using an asymptotically optimal number of tile types (proportional to the Kolmogorov complexity of the shapes), a scaling factor related to the log of the shape's size, and a constant number of stages.

Making further use of this model, Patitz and Summers (2012) defined a new problem and showed how to use tile-based self-assembly to solve it. They asked: Is it possible to start with a collection of objects of an unspecified variety of shapes, and to design TASs which will uniquely identify exactly those objects of a predetermined shape? They called this problem the shape identification problem, and provided a series of results for the 2-dimensional version of the problem (which requires that shapes be hole-free). In it, the input objects were defined to have glues of a single type completely surrounding their perimeters, and the goal was for terminal systems to have all instances of the target shape completely surrounded by a one-tile-wide perimeter, while all input objects not matching that shape have absolutely no tiles attached. The general technique used was to design systems in which tiles begin to attach to the perimeters of all input objects, but then combine information about the shape as the assemblies grow (attached to the input shapes) in a way that prevents complete growth of the perimeter if the object is not of the correct target shape. They then rely on the addition of the RNase enzyme to dissolve away all tiles other than those forming the immediate perimeters of input objects, and for those whose shape was not correct, since the perimeter is not complete, it will not be τ -stable and will thus unravel. They showed matching lower and upper bounds of $O(\frac{\log n}{\log \log n})$ for the tile complexity of identifying $n \times n$ squares, for given n . They then gave a constant tile set which is capable of identifying all squares (of any dimension). They also demonstrated constructions for identifying a wider class of shapes and were able to show that those constructions were optimal in terms of tile complexity, as they were proportional to the Kolmogorov complexities of the shapes.

6.5.4 Signal passing tiles

In the previously discussed models (other than those in Sect. 4.3.5), the tiles are static objects which do not change in structure or function upon binding. To study a more “active” model, Padilla et al. (2012) introduced the Signal passing Tile Assembly Model (STAM), which was based on previous work by Padilla et al. (2012). In the STAM, which is based on the 2HAM, tiles are allowed to have possibly multiple glues on each side. At any point in time each glue can be in one of three states: (1) “latent” (inactive and has never been active), (2) “on” (active, available to bind), and (3) “off” (has been deactivated). A tile’s glues can initially begin as either latent or on. Only glues which are on are able to bind, and when a glue binds it is possible for it to signal any subset of glues on the same tile to perform one of the following transitions: (1) latent \rightarrow on, (2) latent \rightarrow off, or (3) on \rightarrow off. Multiple adjacent tiles in an assembly activating glues in sequence, with each depending on the one before it, can be thought of as passing a signal through the assembly, and this signal can be used to further modify the tiles within the assembly. Signals are thus able to allow tiles on the perimeter to activate glues which provide new binding domains for additional tiles, or to deactivate glues which may cause portions of an assembly to dissociate. The STAM is highly asynchronous, so there is no guarantee about when a signal will be acted upon, only that it will happen at some point in the future, and no guarantees can be made about the relative timing of tile attachments and signal propagation. Furthermore, it is important to note that each tile has a constant number of glues, and thus a constant number of signals that it can initiate and react to.

Complexity measures of STAM systems include the maximum number of glues that appear on the face of any tile in a given system (called the *signal complexity*), and in Padilla et al. (2012), the authors demonstrated constructions which are able to self-assemble $1 \times n$ lines with: (1) a constant number of tile types and signal complexity $O(\log n)$ without using glue deactivation and tile detachments, and (2) a constant number of tile types and $O(\frac{\log n}{\log \log n})$ signal complexity by using glue deactivation. Next they also presented a construction which is able to simulate a Turing machine without making a new copy of the entire row representing the tape at each step, but which instead uses only a constant number of new tiles per computational step, which they called “fuel-efficient”. Their final construction is the first known of any model which can strictly self-assemble a discrete self-similar fractal, namely the Sierpinski triangle (which is provably impossible in models such as the aTAM and 2HAM).

In a similar direction, Jonoska and Karpenko (2012) introduced the Active Tile Assembly Model which also

allows tiles to have multiple glues on each edge and to pass signals, although using a slightly different mechanism. Furthermore, their model is synchronous in that signals are activated immediately upon tile bindings and signal propagation is guaranteed to complete as far as allowed by the tiles in the assembly before any other tile attachments may occur. They also defined a framework for describing recursive self-assembly and self-similarity which can be applied to constructions such as the one they present which self-assembles the aperiodic tiling known as the L-shape tiling.

6.6 Active self-assembly with nubots

Taking inspiration from biological processes such as mitosis and embryonic development, Woods et al. (2013) introduced the *nubot* model in which the fundamental components, called *nubot monomers*, are able to combine in ways similar to the tiles of other models, but are also able to change internal states, move relative to each other, and detach. This mixture of abilities combines aspects of passive tile assembly with the behaviors of systems which include molecular motors, molecular circuits, and reaction-diffusion systems. Additionally, the rules which allow relative movement between individual monomers are able to propagate motion through nubot systems in non-local ways, moving arbitrarily large sub-assemblies relative to each other.

The nubot model uses a two-dimensional triangular grid where at most one monomer can be positioned on each vertex and each monomer has six neighbors. Each monomer can be in exactly one of a finite set of states. Neighboring monomers can have no bond between them, a *flexible bond* (which allows relative motion between the monomers), or a *rigid bond*. A configuration of a nubot system is a description of the entire grid with the locations of monomer types, their states, and the bonds between them. Each step of the evolution of a system consists of either the application of an *interaction rule* or an *agitation*. Interaction rules consist of the following: adjacent monomers change state or bond type, one or both disappear, one monomer appears in an empty space, or one monomer moves relative to another by one unit space. Agitation allows connected components to move.

After defining the nubot model, they showed that, despite the fact that interaction rules can result in arbitrarily large amounts of non-local motion, the model can be efficiently simulated. (They also developed a simulator following the algorithms provided.) Then, using this model of “active self-assembly” they showed that the self-assembly of shapes and patterns can be done very efficiently in terms of the number of unique monomer

states and the time required. Their first constructions consist of the self-assembly of lines and squares in time logarithmic in their sizes. They then present constructions for the efficient, in terms of unique monomer states and assembly time, self-assembly of computable shapes and patterns. Namely, any computable shape or pattern can self-assemble in an amount of time equal to the worst-case running time for a Turing machine to compute a pixel in the shape/pattern plus an additional factor which is polylogarithmic in its size, and with a number of monomer states which is equal to the Kolmogorov complexity of the shape/pattern plus an additional factor which is logarithmic in its size.

6.6.1 Replication of assemblies and evolution of complexity

Perhaps one of the most fundamental questions in science is “how did life originate?” We are much more familiar with the process by which evolution through natural selection has given rise to the profound complexity of living systems, but that process must have had a beginning, some original replicator that was able to produce copies of itself. Furthermore, those copies must have in turn been able to replicate, and potentially with differences in the resulting copies, giving natural selection a toehold for favoring some copies over others and thus providing the driving evolutionary pressure that eventually led to the organisms we see (and are!) today. In order to provide insights into this question, Schulman and Winfree (2011) built on an initial proposal by Cairns-Smith (1998, 1996) that clay crystals may have been the first replicators. Using a slightly restricted version of the kTAM, they sought to determine if they could create an environment in which the dynamics of the growth of DNA crystals (instead of clay crystals) could give rise to a process of evolution that resulted in crystals of increasing complexity. The goal was to restrict the availability of subsets of tile types and see if they could use such a “resource restriction” to influence the evolution of the crystals (a.k.a. tile assemblies). Namely, could more complex crystal structure be selected for simply because its growth required tiles which were more abundant than crystals of simpler structure? Impressively, using computer simulations they were able to answer this in the affirmative, and for systems using only 12 tile types. The systems which they used grew into assemblies forming long, thin “ribbons”, and the operation equivalent to replication was the shearing of a crystal, or cutting a ribbon into two portions, such that each half represented an offspring and growth could occur on the newly exposed edges. The measure of complexity used was the width of the ribbons, so ribbons which grew wider were considered to be more complex.

Schulman et al. (2012) further developed these ideas in wet-lab experiments to demonstrate how the growth and

shearing, or scission, of ribbon assemblies could be used to propagate information encoded in the ribbons. They studied the fidelity of information copying over two generations by encoding information in DNA origami seeds to which tiles attached to form ribbons, and then caused the ribbons to break so that new growth fronts would be exposed, to which tiles would attach and continue growth of the ribbons. Using 4-bit sequences in their laboratory experiments, after two generations 99.98 % of bits were copied correctly, and 78 % of 4-bit sequences remained correct. Theoretical extrapolation suggests that 1,000-fold replication of such sequences could provide 50 % yield. This provides some evidence that such processes alone, without the necessity of enzymes or covalent bond formation, could account for the replication of sequence information in the ribbons as well as for the evolution of increased complexity.

7 Conclusion

In the preceding sections, we introduced the basic concepts of tile-based self-assembly, the original and also newer theoretical models used to describe it, and presented a large spectrum of results in the area. The work surveyed largely focused on the algorithmic nature of the models and demonstrates a rapidly progressing field in which continued progress is being made in understanding the fundamental attributes of such self-assembling systems and the powers that they provide. It is our feeling that while great strides have been made in understanding these fundamentals, there are still some key issues to be resolved. Most notably, is temperature 1, i.e. non-cooperative, self-assembly (in two dimensions) capable of universal computation or even basic algorithmic self-assembly? If not, what are its full limitations? Although a great amount has been learned about the lower bounds associated with varying other aspects of self-assembling systems, the difference between temperature 1 and 2 self-assembly isn't yet fully understood. If self-assembly is to become a practical and robust means of creating complex objects, will cooperative behavior between assembling components be a fundamental requirement? What other necessary characteristics exist, and which aspects of the molecular components can be traded for others and at what costs (e.g. scale for tile complexity, speed for robustness, etc.)?

Another important direction for continued research will be pursuing the development of new, more complex models whose components have capabilities not yet captured by any of the current models. These can be inspired by close interactions between theoreticians and experimentalists, as the tools available to each group become better understood by the other. These can also be inspired

by continued observation of natural systems. We feel that so-called “active” self-assembly systems, in which the components are able to change properties of themselves or their relationships with adjacent components during binding and after incorporation into structures, provide a great deal of promise toward the realization of truly powerful artificial self-assembling systems. However, the continued work to study the more basic and fundamental aspects of the simpler models is what will provide the solid bedrock of understanding on which the more complex models can be effectively and efficiently built. Care must be taken to fully understand the most basic systems, gaining a comprehensive understanding of the effects of each variable, before the most intelligent choices can be made for the new variables to be introduced.

It is our hope and expectation that such research will continue to advance both the theoretical and experimental understanding of self-assembling systems, and lead to important results which (1) provide valuable mathematical tools that contribute to a wide spectrum of related as well as seemingly unrelated scientific pursuits, and (2) meaningfully contribute to the continued experimental work aimed at eventually developing robust and scalable artificial self-assembling systems. Research into algorithmic self-assembly has the potential to dramatically impact technological advancement and to also provide deeper insight into the fundamental workings, and original emergence, of life. It is our hope that this survey helps to combine the work of so many great scientists into an accessible and easy to understand roadmap of what has been done in such a way that it excites and interests more into joining the effort to pave the way forward.

Acknowledgments The author would like to thank Scott Summers, Damien Woods, and Dave Doty for much helpful guidance and advice in putting together this survey. He would also like thank Nataša Jonoska and Jérôme Durand-Lose for the invitation to write it. Finally, he is greatly indebted to two anonymous reviewers whose extremely thorough reviews and comments greatly improved this manuscript. This author’s research was supported in part by National Science Foundation Grant CCF-1117672.

References

- Abel Z, Benbernou N, Damian M, Demaine E, Demaine M, Flatland R, Kominers S, Schweller R (2010) Shape replication through self-assembly and RNase enzymes. In: Proceedings of the twenty-first annual ACM-SIAM symposium on discrete algorithms, SODA 2010, Austin, Texas, Society for Industrial and Applied Mathematics
- Adleman LM (1994) Molecular computation of solutions to combinatorial problems. *Science* 266(11):1021–1024
- Adleman L (2000) Toward a mathematical theory of self-assembly (extended abstract), Tech. Report 00-722, University of Southern California
- Adleman L, Cheng Q, Goel A, Huang M-D (2001) Running time and program size for self-assembled squares. In: Proceedings of the 33rd annual ACM symposium on theory of computing, Herionissos, Greece, pp 740–748
- Adleman L, Cheng Q, Goel A, Huang M-D, Wasserman H (2001) Linear self-assemblies: equilibria, entropy and convergence rates. In: Sixth international conference on difference equations and applications, Taylor and Francis
- Adleman LM, Cheng Q, Goel A, Huang M-D, Kempe D, de Espanés PM, Rothmund PWK (2002) Combinatorial optimization problems in self-assembly. In: Proceedings of the thirty-fourth annual ACM symposium on theory of computing, pp 23–32
- Barish RD, Schulman R, Rothmund PWK, Winfree E (2009) An information-bearing seed for nucleating algorithmic self-assembly. *Proc Natl Acad Sci USA* 106(15):6054–6059
- Becker F (2009) Pictures worth a thousand tiles, a geometrical programming language for self-assembly. *Theor Comput Sci* 410(16):1495–1515
- Becker F, Rapaport I, Rémila E (2006) Self-assembling classes of shapes with a minimum number of tiles, and in optimal time. In: Arun-Kumar S, Garg N (eds) Foundations of software technology and theoretical computer science (FSTTCS). Lecture notes in computer science, vol 4337, Springer, Berlin, pp 45–56
- Becker F, Rémila É, Schabanel N (2008) Time optimal self-assembly for 2d and 3d shapes: the case of squares and cubes. In: Ashish G, Simmel FC, and Sosik P, (eds) DNA computing, 14th international meeting on DNA computing, DNA 14, Prague, Czech Republic, June 2–9 2008. Revised Selected Papers. Lecture notes in computer science, vol 5347, Springer, Berlin, pp 144–155
- Behsaz B, Mañuch J, Ladislav S (2012) Turing universality of stepwise and stage assembly at temperature 1. In: Stefanovic D, Turberfield A (eds) DNA computing and molecular programming. Lecture notes in computer science, vol 7433, Springer, Berlin, pp 1–11
- Berger R (1965) Undecidability of the domino problem. *Memoirs of the American Mathematical Society*, Providence, RI
- Brun Y (2008) Solving np-complete problems in the Tile Assembly Model. *Theor Comput Sci* 395(1):31–46
- Bryans N, Chiniforooshan E, Doty D, Kari L, Seki S (2011) The power of nondeterminism in self-assembly. In: Proceedings of the 22nd annual ACM-SIAM symposium on discrete algorithms, SODA 2011, SIAM, pp 590–602
- Cairns-Smith AG (1966) The origin of life and the nature of the primitive gene. *J Theor Biol* 10(1):53–88
- Cairns-Smith AG (1988) The chemistry of materials for artificial darwinian systems. *Int Rev Phys Chem* 7(3):209–250
- Cannon S, Demaine ED, Demaine ML, Eisenstat S, Patitz MJ, Schweller R, Summers SM, Winslow A (2012) Two hands are better than one (up to constant factors). Tech. Report 1201.1650, Computing Research Repository
- Chandesris J, Dennunzio A, Formenti E, Manzoni L (2011) Computational aspects of asynchronous cellular automata. In: Proceedings of the 15th international conference on developments in language theory, DLT’11, Springer, Berlin, pp 466–468
- Chandran H, Gopalkrishnan N, Reif JH (2009) The tile complexity of linear assemblies. In: Albers S, Marchetti-Spaccamela A, Matias Y, Nikolettseas SE, and Thomas W (eds) Automata, languages and programming, 36th international colloquium, ICALP 2009, Rhodes, Greece, July 5–12 2009. Proceedings, Part I. Lecture notes in computer science, vol 5555, Springer, Berlin, pp 235–253
- Chen H-L, Goel A (2004) Error free self-assembly using error prone tiles. In: Proceedings of the 10th international meeting on DNA based computers, pp 274–283

- Chen H-L, Kao M-Y (2011) Optimizing tile concentrations to minimize errors and time for dna tile self-assembly systems. In: Proceedings of the 16th international conference on DNA computing and molecular programming, DNA'10, Springer, Berlin, pp 13–24
- Chen H-L, Doty D (2012) Parallelism and time in hierarchical self-assembly. In: Proceedings of the 23rd annual ACM-SIAM symposium on discrete algorithms, SODA 2012, SIAM, pp 1163–1182
- Chen H-L, Doty D, Seki S (2011) Program size and temperature in self-assembly. In: Proceedings of the 22nd international symposium on algorithms and computation, ISAAC 2011. Lecture notes in computer science, vol 7074, Springer, Berlin, pp 445–453
- Chen H-L, Goel A, Winfree E, Luhrs C (2007a) Self-assembling tile systems that heal from small fragments. In: Preliminary Proceedings of DNA Computing, vol 30, pp 30–46
- Chen H-L, Schulman R, Goel A, Winfree E (2007b) Reducing facet nucleation during algorithmic self-assembly. *Nano Lett* 7(9): 2913–2919
- Cheng Z, Xiao J (2012) Algorithmic tile self-assembly model for the minimum set cover problem. *J Bionanosci* 6(2):69–77
- Cheng Z, Chen Z, Huang Y, Zhang X, Xu J (2010) Implementation of the timetable problem using self-assembly of DNA tiles. *Int J Comput Commun Control* V(4):490–505
- Cheng Q, Aggarwal G, Goldwasser MH, Kao M-Y, Schweller RT, de Espanés P Moisset (2005) Complexities for generalized models of self-assembly. *SIAM J Comput* 34:1493–1515
- Cook M, Fu Y, Schweller RT (2011) Temperature 1 self-assembly: Deterministic assembly in 3D and probabilistic assembly in 2D. In: Proceedings of the 22nd annual ACM-SIAM symposium on discrete algorithms, SODA 2011, SIAM
- Culik K II (1996) An aperiodic set of 13 Wang tiles. *Discret Math* 160(1–3):245–251
- Czeizler E, Popa A (2012) Synthesizing minimal tile sets for complex patterns in the framework of patterned dna self-assembly. In: Stefanovic D, Turberfield A (eds) DNA Computing and Molecular Programming. Lecture notes in computer science, vol 7433, Springer, Berlin, pp 58–72
- Demaine ED, Patitz MJ, Schweller RT, Summers SM (2011) Self-assembly of arbitrary shapes using RNase enzymes: meeting the Kolmogorov bound with small scale factor (extended abstract). Symposium on theoretical aspects of computer science, STACS 2011, pp 201–212
- Demaine ED, Eisenstat S, Ishaque M, Winslow A (2012) One-dimensional staged self-assembly. *Nat Comput* 12(2):247–258. English
- Demaine ED, Patitz MJ, Rogers TA, Schweller RT, Summers SM, Woods D (2013) The two-handed Tile Assembly Model is not intrinsically universal. In: Proceedings of the fortieth international colloquium on automata, languages and programming, ICALP 2013 (to appear)
- Demaine ED, Demaine ML, Fekete SP, Ishaque M, Rafalin E, Schweller RT, Souvaine DL (2008) Staged self-assembly: nanomanufacture of arbitrary shapes with $O(1)$ glues. *Nat Comput* 7(3): 347–370
- Demaine ED, Demaine ML, Fekete SP, Patitz MJ, Schweller RT, Winslow A, Woods D (2012) One tile to rule them all: Simulating any turing machine, tile assembly system, or tiling system with a single puzzle piece, Tech. Report 1212.4756, Computing Research Repository
- Doty D (2010) Randomized self-assembly for exact shapes. *SIAM J Comput* 39(8):3521–3552
- Doty D (2012) Theory of algorithmic self-assembly. *Commun ACM* 55(12): 78–88
- Doty D, Patitz MJ (2009) A domain specific language for programming in the Tile Assembly Model. In: Proceedings of the fifteenth international meeting on DNA computing and molecular programming, Fayetteville, Arkansas, USA, June 8–11 2009, pp 25–34
- Doty D, Patitz MJ, Summers SM (2011) Limitations of self-assembly at temperature 1. *Theor Comput Sci* 412:145–158
- Doty D, Kari L, Masson B (2013) Negative interactions in irreversible self-assembly. *Algorithmica* 66(1):153–172
- Doty D, Lutz JH, Patitz MJ, Summers SM, Woods D (2009) Intrinsic universality in self-assembly. In: Proceedings of the 27th international symposium on theoretical aspects of computer science, pp 275–286
- Doty D, Patitz MJ, Reishus D, Schweller RT, Summers SM (2010) Strong fault-tolerance for self-assembly with fuzzy temperature. In: Proceedings of the 51st annual IEEE symposium on foundations of computer science, FOCS 2010, pp 417–426
- Doty D, Lutz JH, Patitz MJ, Schweller RT, Summers SM, Woods D (2012) The Tile Assembly Model is intrinsically universal. In: Proceedings of the 53rd annual IEEE symposium on foundations of computer science, FOCS 2012 (to appear)
- Fu B, Patitz MJ, Schweller RT, Sheline R (2012) Self-assembly with geometric tiles. In: Proceedings of the 39th international colloquium on automata, languages and programming, ICALP (to appear)
- Fujibayashi K, Zhang DY, Winfree E, Murata S (2009) Error suppression mechanisms for DNA tile self-assembly and their simulation. *Nat Comput* 8(3):589–612
- Göös M, Orponen P (2010) Synthesizing minimal tile sets for patterned dna self-assembly. In: Sakakibara Y, Mi Y (eds) DNA. Lecture notes in computer science, vol 6518, Springer, Berlin, pp 71–82
- Hartmanis J, Stearns RE (1965) On the computational complexity of algorithms. *Trans Am Math Soc* 117:285–306
- Hopfield JJ (1974) Kinetic proofreading: a new mechanism for reducing errors in biosynthetic processes requiring high specificity. *Proc Natl Acad Sci USA* 71(10):4135–4139
- Ingerson TE, Buvel RL (1984) Structure in asynchronous cellular automata. *Phys D* 10(1–2):59–68
- Jang B, Kim Y-B, Lombardi F (2006) Error tolerance of DNA self-assembly by monomer concentration control. IEEE international symposium on defect and fault-tolerance in VLSI systems, Arlington, VA, October 4–6, 2006, pp 89–97
- Jonoska N, McColm GL (2006) Flexible versus rigid tile assembly. In: Calude CS, Dinneen MJ, Pun G, Rozenberg G, and Stepney S (eds) Unconventional computation. Lecture notes in computer science, vol 4135, Springer, Berlin, pp 139–151
- Jonoska N, McColm GL (2009) Complexity classes for self-assembling flexible tiles. *Theor Comput Sci* 410(4–5): 332–346
- Jonoska N, Karpenko D (2012) Active tile self-assembly, self-similar structures and recursion. Tech. Report 1211.3085, Computing Research Repository
- Jonoska N, Karl SA, Saito M (1999) Three dimensional DNA structures in computing. *BioSystems* 52(1–3):143–153.
- Kao M-Y, Schweller R (2008) Randomized self-assembly for approximate shapes. In: Proceedings of the 35th international colloquium on automata, languages and programming, Part I, ICALP '08, Springer, Berlin, pp 370–384
- Kari L, Seki S, Xu Z (2012) Triangular and hexagonal tile self-assembly systems. In: Proceedings of the 2012 international conference on theoretical computer science: computation, physics and beyond, WTCS'12, Springer, Berlin, pp 357–375
- Kautz SM, Shatters B (2011) Self-assembling rulers for approximating generalized sierpinski carpets. In: Fu B, Du D-Z (eds) COCOON. Lecture notes in computer science, vol 6842, Springer, Berlin, pp 284–296
- Lathrop JI, Lutz JH, Summers SM (2009) Strict self-assembly of discrete Sierpinski triangles. *Theor Comput Sci* 410:384–405

- Lathrop JJ, Lutz JH, Patitz MJ, Summers SM (2011) Computability and complexity in self-assembly. *Theory Comput Syst* 48(3): 617–647
- Lempiäinen T, Czeizler E, Orponen P (2011) Synthesizing small and reliable tile sets for patterned DNA self-assembly. In: *Proceedings of the 17th international conference on DNA computing and molecular programming, DNA'11*, Springer, Berlin, pp 145–159
- Luhurs C (2008) Polyomino-safe DNA self-assembly via block replacement. In: *DNA computing: DNA14. Lecture notes in computer science*, vol 5347, Springer, Berlin, pp 112–126
- Lutz JH, Shatters B (2012) Approximate self-assembly of the sierpinski triangle. *Theory Comput Syst* 51(3):372–400
- Ma X, Lombardi F (2008) Synthesis of tile sets for DNA self-assembly. *IEEE Trans CAD Integr Circuits Syst* 27(5):963–967
- Majumder U, Labean TH, Reif JH (2007a) Activatable tiles: compact, robust programmable assembly and other applications. In: Garzon M, Yan H (eds) *DNA computing: DNA13. Lecture notes for computer science*, Springer, Berlin, pp 15–25
- Majumder U, Labean TH, Reif JH (2007b) Activatable tiles for compact error-resilient directional assembly. In: *13th international meeting on DNA computing (DNA13)*, Memphis, Tennessee, June 4–8 2007
- Mao C, Labean TH, Reif JH, Seeman NC (2000) Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature* 407(6803):493–496
- Mañuch J, Stacho L, Stoll C (2009) Step-assembly with a constant number of tile types. In: *Proceedings of the 20th international symposium on algorithms and computation, ISAAC '09*, Springer, Berlin, pp 954–963
- Mañuch J, Stacho L, Stoll C (2010) Two lower bounds for self-assemblies at temperature 1. *J Comput Biol* 17(6):841–852
- Meunier P-E, Patitz MJ, Summers SM, Theyssier G, Winslow A, Woods D (2013) Intrinsic universality in tile self-assembly requires cooperation. Tech. Report 1304.1679, Computing Research Repository
- Ninio J (1975) Kinetic amplification of enzyme discrimination. *Biochimie* 57(5):587–595
- Padilla JE, Liu W, Seeman NC (2012) Hierarchical self assembly of patterns from the robinson tilings: DNA tile design in an enhanced Tile Assembly Model. *Nat Comput* 11(2):323–338
- Padilla JE, Patitz MJ, Pena R, Schweller RT, Seeman NC, Sheline R, Summers SM, Zhong X (2012) Asynchronous signal passing for tile self-assembly: Fuel efficient computation and efficient assembly of shapes. Tech. Report 1202.5012, Computing Research Repository
- Patitz MJ (2009) Simulation of self-assembly in the abstract Tile Assembly Model with ISU TAS. In: *6th annual conference on foundations of nanoscience: self-assembled architectures and devices*, Snowbird, Utah, USA, April 20–24 2009
- Patitz MJ (2012) An introduction to tile-based self-assembly. In: Durand-Lose J, Jonoska N (eds) *Unconventional computation and natural computation. Lecture notes in computer science*, vol 7445, Springer, Berlin, pp 34–62
- Patitz MJ, Summers SM (2010) Self-assembly of discrete self-similar fractals. *Nat Comput* 1:135–172
- Patitz MJ, Summers SM (2011) Self-assembly of decidable sets. *Nat Comput* 10(2): 853–877
- Patitz MJ, Summers SM (2012) Identifying shapes using self-assembly. *Algorithmica* 64(3):481–510
- Patitz MJ, Schweller RT, Summers SM (2011) Exact shapes and tiling universality at temperature 1 with a single negative glue. In: *Proceedings of the 17th international conference on DNA computing and molecular programming, DNA'11*, Springer, Berlin, pp 175–189
- Patitz MJ, Schweller RT, Summers SM (2012) Efficient squares and tiling universality at temperature 1 with a unique negative glue. Tech. Report 1105.1215v2, Computing Research Repository
- Penrose R (1979) Set of tiles for covering a surface. US Patent 4133152
- Reif JH (1999) Local parallel biomolecular computing. In: *Proceedings of DNA Based Computers III, DIMACS*, American Mathematical Society, Providence, RI, vol 48, pp 217–254
- Reif J, Sahu S, Yin P (2006) Complexity of graph self-assembly in accretive systems and self-destructible systems. In: Carbone A, Pierce N (eds) *DNA computing. Lecture notes in computer science*, vol 3892, Springer, Berlin, pp 257–274
- Rothmund PWK (2001) Theory and experiments in algorithmic self-assembly. Ph.D. thesis, University of Southern California
- Rothmund PWK (2006) Folding DNA to create nanoscale shapes and patterns. *Nature* 440(7082): 297–302
- Rothmund PWK, Winfree E (2000) The program-size complexity of self-assembled squares (extended abstract). In: *Proceedings of the thirty-second annual ACM symposium on theory of computing, STOC '00*, Portland, Oregon, United States, ACM, pp 459–468
- Rothmund PWK, Papadakis N, Winfree E (2004) Algorithmic self-assembly of DNA sierpinski triangles. *PLoS Biol* 2(12): e424
- Schulman R, Winfree E (2009) Programmable control of nucleation for algorithmic self-assembly. *SIAM J Comput* 39(4): 1581–1616
- Schulman R, Winfree E (2011) Simple evolution of complex crystal species. In: *Proceedings of the 16th international conference on DNA computing and molecular programming, DNA'10*, Springer, Berlin, pp 147–161
- Schulman R, Yurke B, Winfree E (2012) Robust self-replication of combinatorial information via crystal growth and scission. *Proc Natl Acad Sci USA* 109(17):6405–10
- Schweller RT, Sherman M (2013) Fuel efficient computation in passive self-assembly. In: *Proceedings of the 24th annual ACM-SIAM symposium on discrete algorithms, SODA 2013*, SIAM, pp 1513–1525
- Seeman NC (1982) Nucleic-acid junctions and lattices. *J Theor Biol* 99:237–247
- Seki S, Okuno Y (2012) On the behavior of tile assembly system at high temperatures. In: Barry Cooper S, Dawar A, and Löwe B (eds) *How the world computes. Proceedings of the Turing centenary conference, CiE 2012*, Cambridge, United Kingdom, June 18–23, 2012. *Lecture notes in computer science*, vol 7318, Springer, Berlin, pp 550–560 (Eng)
- Seki S (2013) Combinatorial optimization in pattern assembly. Tech. Report 1301.3771, Computing Research Repository
- Soloveichik D, Winfree E (2005) Complexity of compact proofreading for self-assembled patterns. In: Carbone A, Pierce NA (eds) *DNA computing. The eleventh international meeting on DNA computing. Lecture notes in computer science*, vol 3892, Springer, Berlin, pp 305–324
- Soloveichik D, Winfree E (2007) Complexity of self-assembled shapes. *SIAM J Comput* 36(6):1544–1569
- Soloveichik D, Cook M, Winfree E (2008) Combining self-healing and proofreading in self-assembly. *Nat Comput* 7(2):203–218
- Summers SM (2012) Reducing tile complexity for the self-assembly of scaled shapes through temperature programming. *Algorithmica* 63(1–2):117–136
- Wang H (1961) Proving theorems by pattern recognition—II. *Bell Syst Tech J* XL(1):1–41
- Wang H (1963) Dominoes and the AEA case of the decision problem. In: *Proceedings of the symposium on mathematical theory of automata*, New York, 1962, Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, pp 23–55
- Wang Y, Lu W, Bai X, Wei D, Cui G (2011) DNA tile assembly for degree-constrained minimum spanning tree. *J Bionosci* 5(1):41–46
- Winfree E The xgrow simulator. <http://www.dna.caltech.edu/Xgrow>. Accessed 17 June 2013

- Winfree E (1998) Algorithmic self-assembly of DNA. Ph.D. thesis, California Institute of Technology
- Winfree E (2006) Self-healing tile sets. In: Chen J, Jonoska N, and Rozenberg G (eds) *Nanotechnology: science and computation*. Natural computing series, Springer, Berlin, pp 55–78
- Winfree E, Bekbolatov R (2003) Proofreading tile sets: Error correction for algorithmic self-assembly. In: Chen J, Reif JH (eds) *DNA computing*. Lecture notes in computer science, vol 2943, Springer, Berlin, pp 126–144
- Winfree E, Liu F, Wenzler LA, Seeman NC (1998) Design and self-assembly of two-dimensional DNA crystals. *Nature* 394(6693): 539–44
- Woo S, Rothmund PWK (2011) Programmable molecular recognition based on the geometry of DNA nanostructures. *Nat Chem* 3(8): 620–7
- Woods D, Chen H-L, Goodfriend S, Dabby N, Winfree E, Yin P (2013) Active self-assembly of algorithmic shapes and patterns in polylogarithmic time. In: *Proceedings of the 4th conference on innovations in theoretical computer science, ITCS '13*, New York, NY, USA, ACM, pp 353–354