



Review

Article submitted to journal

Subject Areas:

Theory of computing, Molecular computing

Keywords:

Intrinsic universality, simulation, self-assembly, abstract Tile Assembly Model

Author for correspondence:

Damien Woods

e-mail: woods@caltech.edu

Intrinsic universality and the computational power of self-assembly

Damien Woods

California Institute of Technology

Molecular self-assembly, the formation of large structures by small pieces of matter sticking together according to simple local interactions, is a ubiquitous phenomenon. A challenging engineering goal is to design a few molecules so that large numbers of them can self-assemble into desired complicated target objects. Indeed we'd like to understand the ultimate capabilities and limitations of this bottom-up fabrication process. We look to theoretical models of algorithmic self-assembly, where small square tiles stick together according to simple local rules in order to carry out a crystal growth process. In this survey we focus on the use of simulation between such models to classify and separate their computational and expressive powers. Roughly speaking, one model simulates another if they grow the same structures, via the same dynamical growth processes.

Our journey begins with the result that there is a single intrinsically universal tile set that, with appropriate initialisation and spatial scaling, simulates any instance of Winfree's abstract Tile Assembly Model. This universal tile set exhibits something stronger than Turing universality: it captures the geometry and dynamics of any simulated system in a very direct way. From there we find that there is no such tile set in the more restrictive noncooperative model, proving it weaker than the full Tile Assembly Model. In the two-handed model, where large structures can bind together on one step, we encounter an infinite set of infinite hierarchies of strictly increasing simulation power. Towards the end of our trip, we find one tile to rule them all: a single rotatable flipable polygonal tile that simulates any tile assembly system. We find another tile that aperiodically tiles the plane (but with small gaps). These and other recent results show that simulation is giving rise to a kind of computational complexity theory for self-assembly. It seems this could be the beginning of a much longer journey, so directions for future work are suggested.

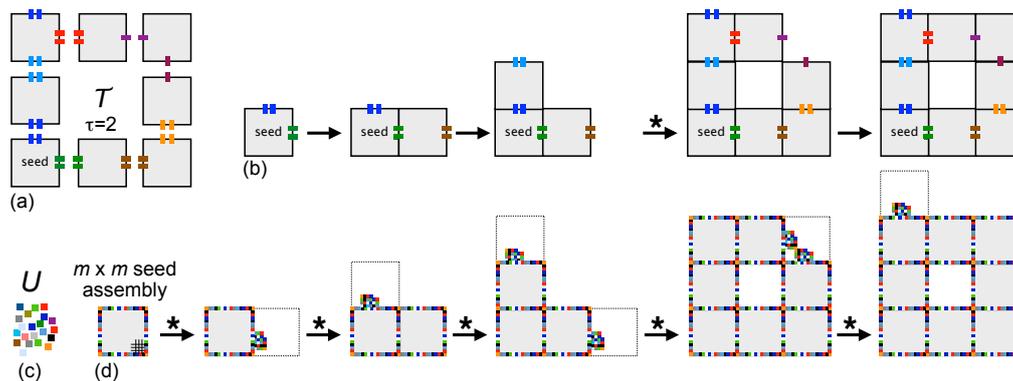


Figure 1. An instance of the abstract Tile Assembly Model, and an example showing simulation and intrinsic universality. (a) A tile assembly system \mathcal{T} consists of a tile set, seed tile and a temperature $\tau \in \mathbb{N}$. Coloured glues on the tiles' sides have a natural number strength (shown here as 0, 1 or 2 coloured tabs). (b) Growth begins from the seed with tiles sticking to the growing assembly if the sum of the strengths of the matching glues is at least τ . (c) An intrinsically universal tile set U . (d) When initialised with a seed assembly (that encodes \mathcal{T}) and at temperature 2, the intrinsically universal tile set simulates the dynamics of \mathcal{T} with each tile placement in \mathcal{T} being simulated by the growth of an $m \times m$ block of tiles. Single tile attachment is denoted by \rightarrow and $\xrightarrow{*}$ denotes multiple tile attachments. Note that both systems have many other growth dynamics that are not shown.

1. Introduction

Molecular self-assembly, the formation of large structures by small pieces of matter sticking together according to simple local interactions, is a ubiquitous phenomenon in nature. It is also a phenomenon that we are beginning to learn how to control, or program, by designing molecules and their local interaction rules. Since Seeman's pioneering work on assembling 2D [25,58] lattices from small artificially-synthesised DNA molecules, we have learned how to self-assemble various kinds of shapes and patterns, such as Rothemund's nanoscale squares, tiny maps and smiley faces [54], as well as 3D lattices [66] and shapes [17] including spheres and vases [30], and Latin, Arabic and Chinese characters [61]. These structures are hardcoded in the sense that for each position a unique DNA molecule is synthesised; the larger the structure the more unique molecular components we need to design. Thinking of Seeman's periodic 2D DNA structures as lattices of tiles, Winfree [62] asked if one could use smarter molecules, that exploit computation to come together in such a way that individual tile types end up appearing in many different locations in the target shape or pattern. Since then a number of examples of such algorithmic tile-based structures have been self-assembled, including regular arrays of tiles [63], fractal structures [26,55], bit-copying systems [5,6,57] and binary counters [5,22]. Given these experimental successes, it is imperative to have a theory of self-assembly to guide this rapidly developing field of nanoscale engineering. Models of algorithmic self-assembly are capable of Turing-universality, and so of an infinite variety of computational behaviours, but yet distinct enough from existing computational models to present interesting theoretical challenges.

The abstract Tile Assembly Model, put forward by Winfree [62], is a kind of asynchronous nondeterministic cellular automaton that models crystal growth processes. Put another way, the abstract Tile Assembly Model restricts classical square Wang tiling [60] to use a mechanism for crystal-like growth of a tiling, one tile at a time, starting from a special seed tile. Formally, an instance of the abstract Tile Assembly Model [62] is called a tile assembly system and is a triple $\mathcal{T} = (T, \sigma, \tau)$ consisting of a finite set T of square tiles, a seed assembly σ (one or more tiles stuck together), and a temperature $\tau \in \{1, 2, 3, \dots\}$, as shown in Figure 1(a). Each side of a square tile has a glue (or colour) g which in turn has a strength $s \in \{0, 1, 2, \dots\}$. Growth occurs on the integer

plane and begins from a seed assembly (or a seed tile) placed at the origin, as shown in Figure 1(b). A tile sticks to a partially-formed assembly if it can be placed next to the assembly in such a way that enough of its glues match the glues of the adjacent tiles on the assembly and the sum of the matching glue strengths is at least the temperature. Intuitively, the tile sticks if its binding is sticky enough to overcome thermal fluctuations in the environment. Growth proceeds one tile at a time, asynchronously and nondeterministically. In this model tiles may not overlap nor rotate, and unlike Wang tiling, adjacent glues (colours) may mismatch in an assembly. The model is capable of Turing machine simulation [62], and indeed computation via Turing machine simulation can be used to guide the process of self-assembly [59].

Of course individual tiles need not be square, models with triangles [28,38], hexagons [15, 28,38,64] and arbitrary polyominoes and polygons [15,23,24,28] have been considered. Yet other models of self-assembly allow for non-local rules and large-scale interactions. One such model is the two-handed (or hierarchical) model [3,9,16,21] which allows large structures to stick together if enough of their tiles' edge colours match. Another is the Nubot model [64] of molecular robotics where large assemblies of molecules can grow and then move relative to each other in a rigid-body fashion. Although some self-assembly models can be thought of as generalizations of cellular automata, or effectivisations of Wang tiling, these models are all quite distinct from each other in terms of both questions that can be sensibly asked and results that can be obtained.

(a) Introduction to the use of simulation in self-assembly

In this survey we discuss the relative computational and expressive power of self-assembly models using *simulation* as a method to compare these models. Our notion of simulation is specifically designed to capture, in a formal and natural sense, both production (assemblies) and dynamics (how those assemblies are built) of self-assembly systems. In the past few years quite a number of results have come to fruition showing that simulation and the related notion of intrinsic universality can be used to classify the power of self-assembly systems. Simulation provides one clean theory to unify a wide range of self-assembly models including models that use different modes of assembly (single-tile versus multi-tile assembly, with or without rotations and/or flips) and various geometries (square tiles versus polyomino or polygonal tiles with information-encoding geometries). Some of these results formally show that intrinsic universality is a distinct notion from computational (Turing) universality while others show infinite hierarchies of tile assembly systems with increasing simulation power as we move up the hierarchies. Some of the results mentioned in this survey are summarised in Figure 2 which the reader should refer to throughout the text.

Tile assemblers have borrowed and generalised the powerful idea of intrinsic universality from the cellular automata community where it has given rise to a rich theory [4,12,13,29,48,49].¹ This short survey attempts to show that we are beginning to see this in self-assembly too. Tile self-assembly systems are distributed, asynchronous and nondeterministic. Hence it should not be surprising that our definition of simulation, and even some techniques used in proofs, use concepts similar to those seen in concurrency control, database theory and the theory of asynchronous state transition systems. For example, the definition of simulation is related, although with essential differences, to some notions of weak bisimulation used in the study of asynchronous distributed systems [45,46]. In particular, in the simulator system we interpret certain parts of assemblies as representing empty space in the simulated system, and for a time many tiles can be added to such regions before that region commits to encoding some specific tile in the simulated system. Hence the simulator makes "hidden", or "internal", actions as is seen in weak bisimulation. However, our notion of simulation is not an instance of weak bisimulation.² Also, it should be noted that before the study of intrinsic universality in self-assembly, some previous self-assembly papers [14,59]

¹This notion has also been studied in the context of Wang tiling [40,41].

²In particular, our definition allows for simulator assemblies α' , that represent α in the simulated system, to represent merely a *proper subset* of the dynamics of α ; as long as α' can be reached by an assembly from which the full set of dynamics is indeed possible. Weak bisimulation forbids this as there is no way to consistently label the states, or assemblies, of such a simulator by states in the simulated system.

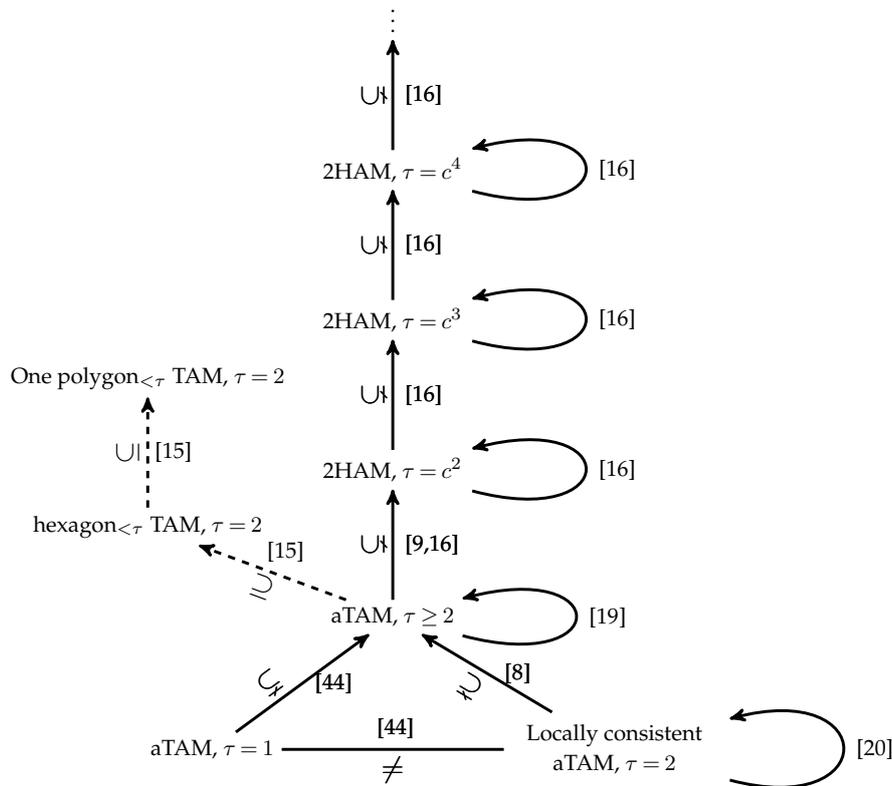


Figure 2. Classes of tile assembly systems, and their relationship with respect to simulation. There is an arrow from B to A if A contains B with respect to simulation: that is for each tile assembly system $B \in B$ there is a tile assembly system $A_B \in A$ that simulates B . Dashed arrows denote containment, solid arrows denote strict containment and a self-loop denotes the existence of an intrinsically universal tile set for a class and its omission implies that the existence of such a tile set is an open problem. aTAM: abstract Tile Assembly Model (growth from a seed assembly by single tile addition in 2D), τ denotes “temperature”. 2HAM: Two-Handed Tile Assembly Model (assemblies of tiles stick together in 2D). A 2HAM temperature hierarchy is shown for some $c \in \{2, 3, 4, \dots\}$ and, in fact, for each such c the set of temperatures $\{c^i | i \in \{2, 3, \dots\}\}$ gives an infinite hierarchy of classes of strictly increasing simulation power in the 2HAM. Citations proving the results are given in square brackets. Simulation results for a number of other models are described in the main text.

made use of notions that can be seen as precursors to the definitions and results discussed here which perhaps lends weight to the intuition that spatially-scaled dynamics-preserving simulation is a natural way to compare self-assembly systems.

It is important to point out that besides simulation there are a number of other methods to compare self-assembly models that also lead to interesting results and proof techniques. Examples include computational power [11,62], efficiency in terms of the number of tile types needed to build shapes or patterns [42,53], and computational complexity of verification of properties of tile assembly systems [2]. See other surveys for more details [18,52].

In this informal survey we forgo formal definitions and proofs, which can be found in the cited literature.

2. Simulation and a result: The abstract Tile Assembly Model is intrinsically universal

Intuitively, one self-assembly model simulates another if they grow the same structures, via the same dynamical growth processes, possibly with some spatial scaling. In order to be a little more precise, let \mathcal{S} and \mathcal{T} be tile assembly systems of the abstract Tile Assembly Model described above. \mathcal{S} is said to *simulate* \mathcal{T} if the following conditions hold: (1) each tile of \mathcal{T} is represented by one or more $m \times m$ blocks of tiles in \mathcal{S} called *supertiles*, (2) the seed assembly of \mathcal{T} is represented by the *seed assembly* of \mathcal{S} (one or more connected $m \times m$ supertiles), and (3) via supertile representation every sequence of tile placements in the simulated system \mathcal{T} has a corresponding sequence of supertile placements in the simulator system \mathcal{S} , and vice versa. It is worth pointing out that although the intuitive idea of one assembly system simulating another is fairly simple, the formal definition gets a little technical as the filling out of supertiles in the simulator is an asynchronous and nondeterministic distributed process with many supertiles growing independently and in parallel in the simulator system. See, for example, [44] for a formal definition.

With our notion of simulation in hand we are ready to describe intrinsic universality. Figure 1(d) illustrates the concept. A class of tile assembly systems \mathcal{C} is said to be *intrinsically universal* if there exists a single set of tiles U that simulates any instance of \mathcal{C} . For each such simulation, U should be appropriately initialised as an instance (i.e. a tile assembly system) of \mathcal{C} itself. For example, the abstract Tile Assembly Model has been shown to be intrinsically universal [19]. Specifically, this means that there is a single set of tiles U that when appropriately initialised, is capable of simulating an arbitrary tile assembly system \mathcal{T} . To program such a simulation, tiles from \mathcal{T} are represented as $m \times m$ supertiles (built from tiles in U) and the seed assembly of \mathcal{T} is represented as a connected assembly $\sigma_{\mathcal{T}}$ of such supertiles. Furthermore, the entire tile assembly system \mathcal{T} (a finite object) is itself encoded in the supertiles of $\sigma_{\mathcal{T}}$ of U . Then if we watch all possible growth dynamics in both $\mathcal{T} = (T, \sigma, \tau)$ and $\mathcal{U} = (U, \sigma_{\mathcal{T}}, 2)$ we get that both systems produce the same set of assemblies via the same dynamics where we use a supertile representation function to map from supertiles over U to tiles from T . It is worth pointing out that in this particular construction [19] the simulating system is always (merely) at temperature $\tau = 2$ no matter how large the temperature ($\tau \geq 1$) of the simulated system.

This intrinsically universal tile set has the ability to simulate both the geometry and growth order of any tile assembly system. Modulo spatial rescaling, this universal tile set U represents the full power and expressivity of the entire abstract Tile Assembly Model.

3. A complexity theory for self-assembly: the abstract Tile Assembly Model

We've seen that the abstract Tile Assembly Model is intrinsically universal: a kind of completeness for the model with respect to our notion of simulation. The class of all Turing machines also exhibits a kind of completeness shown via the existence of a universal Turing machine, although typically using a much weaker notion of simulation than ours that cares less about capturing the dynamics of the simulated machine.³ Now that we know the abstract Tile Assembly Model is intrinsically universal, and that this holds with a fairly strict notion of simulation, we can attempt to use

³It turns out that one can have a much tighter notion of simulation for Turing machines inspired by the constant spatial scale factor simulations discussed in this survey, and furthermore it turns out that Turing machines are intrinsically universal under this notion. As described on the final page of [47], it is possible to have a universal Turing machine that simulates any Turing machine M with only a constant factor time overhead and a constant factor "spatial rescaling" of tape contents. This universal machine stores the entire simulated program M (which is of constant size: i.e. independent of the input length) at the simulated tape head position. Simulating a transition rule involves reading and copying information from the simulated program, and simulating a move left or right involves moving the entire simulated program one step to the left or right. Each step is simulated in time independent of the time and space of the simulated machine, and quadratic in the constant-length simulated program. Intuition compels belief in this being the morally-correct notion of intrinsic universality for Turing machines. The construction is particularly straightforward as the Turing machine model is both sequential and local.

simulation to tease apart the power of different tile assembly models. Specifically we ask if natural subclasses of the model can achieve the full expressive power of the model via spatially-scaled dynamics-preserving simulation and/or if such subclasses can even simulate each other.

(a) Separating the power of cooperative and noncooperative tile assembly systems

It has been known for some time that the abstract Tile Assembly Model at temperature 2, where at least some of the tiles are required to match on two or more sides for correct binding, i.e. *cooperative binding*, is capable of highly non-trivial behaviour. Turing machine simulation [53], efficient production of $n \times n$ squares and other simple size- n shapes using $\Theta(\log n / \log \log n)$ tile types [1], efficient production of arbitrary finite connected shapes using a number of tile types within a logarithmic factor of the Kolmogorov complexity of the shape [59], and even intrinsic universality [19] (as already discussed) can all be achieved with cooperative, or temperature 2, growth.

The fact that the (full) abstract Tile Assembly Model is intrinsically universal means that there is a subclass of the model, namely the class of systems that use the intrinsically universal tile set U , that is capable of simulating the full model. This suggests an obvious question: can we show that some subclasses of the model are provably weaker than the full model, by showing that systems from these subclasses can not simulate the full model?

The most notorious such subclass is called temperature 1. Despite its esoteric name, it models a fundamental and ubiquitous form of growth: asynchronous growing and branching tips in Euclidian space where each new tile is added if it matches on at *least one side*. Since temperature 1 binding does *not* require matching glues on multiple sides, it is called noncooperative binding. A reasonable analogy is to think of cooperative binding as context sensitive, and noncooperative binding as context free. In 2D, it's like snakes on a plane.

Recently it has been shown that that the temperature 1 abstract Tile Assembly Model (i.e. noncooperative binding) is provably weaker than the full model [44]: in particular, temperature 1 tile assembly is not capable of *simulating* arbitrary tile assembly systems. In fact, there is a very simple cooperative tile assembly system, that uses cooperative binding on two sides in merely one location, that can not be simulated by any noncooperative tile assembly system. This is the first fully general negative result about temperature 1 that does not assume restrictions on the model nor unproven hypotheses. The proof uses a simple pumping lemma (called the window movie lemma) for self-assembly that gives a sufficient condition to modify assembly sequences and swap parts of assemblies. It is used to fool any claimed noncooperative simulation of cooperative binding. This lemma has since found use elsewhere and indeed has been generalised in various ways [7,8,10,27,35]. An interesting aspect of the negative result is that it holds for 3D noncooperative systems, they too can not simulate arbitrary tile assembly systems. This seems quite shocking, given that 3D noncooperative systems are Turing-universal [11]! So in particular, 3D noncooperative systems can simulate 2D (or 3D) cooperative systems by simulating a Turing machine that in turn simulates the cooperative system, but this loose style of simulation ends up destroying the geometry and dynamics of tile assembly by encoding everything as “geometry-less” strings.⁴ Hence, the negative result in [44] can be interpreted to mean that Turing-universal algorithmic behaviour in self-assembly does not imply the ability to simulate, in a direct geometric fashion, arbitrary algorithmic self-assembly processes. Despite this negative result, and a recent positive result showing that noncooperative systems can be programmed to grow assemblies of size larger than the number of tile types [43], it remains open whether 2D noncooperative systems are intrinsically universal for themselves, or capable of Turing machine simulation in an error-free way.

⁴Or as Paul W.K. Rothmund has put it, 3D noncooperative systems can dream about tile assembly, but can not actually *do* tile assembly.

It was also shown that in 3D there is a (cube) tile set that noncooperatively simulates all 2D noncooperative systems [44].

(b) Separating the power of cooperative tile assembly systems

Besides the abstract Tile Assembly Model being intrinsically universal, it is also known that a restricted sub-model, called the locally consistent Tile Assembly Model, is intrinsically universal [20]. A locally consistent tile assembly system is one where tiles bind without mismatches, and with binding strength of exactly 2. This sub-model is quite expressive: the standard methods to simulate Turing machines with tile assembly systems are locally consistent, as are many systems that have been implemented in the lab to date with DNA [5,6,22,26,55,57]. This begs the question: are locally consistent systems capable of simulating the full model? Recent work by Becker and Meunier shows that the answer is “no” [8]. In particular, their results show that any class of tile assembly systems that has no mismatches, or disallows excess binding strength, can not simulate the abstract Tile Assembly Model.⁵

The result tells us that at least some of the tricky aspects of the intrinsic universality simulation in [19] are required. In particular, in that simulation binding mismatches occur in numerous places (often as a mechanism to decide which of the competing parts of an $m \times m$ simulator supertile will “win” a competition to decide which simulated tile the supertile encodes). The fact that systems without mismatches can not simulate those with mismatches [8], tells us that this aspect of the simulation is required. One of the key innovations in [8] is to generalise the window movie lemma (a pumping lemma) from [44] so that it can be applied in significantly more complicated settings. It will be interesting to see if this generalised “bisimulation lemma” finds use elsewhere. Finally, Becker and Meunier show [8] shows that 3D mismatch-free tile assembly systems are intrinsically universal and leaves open the question for 2D mismatch-free systems.

It remains as future work to further characterise the power of interesting subclasses of the abstract Tile Assembly Model, and in particular, to separate such subclasses. Work in this direction will enable us to understand exactly which of the model’s features are required for specific kinds of global behaviour.

4. A complexity theory for self-assembly: comparing models of self-assembly

What about other models of self-assembly besides the abstract Tile Assembly Model?

(a) Two-hands

It has been shown that the two-handed, or hierarchical, model of self-assembly (where large assemblies of tiles may come together in a single step) is not intrinsically universal [16]. Specifically there is no tile set that, in the two-handed model, can simulate all two-handed systems for all temperatures. However, the same paper shows that for each temperature $\tau \in \{2, 3, 4, \dots\}$ there is a tileset U_τ that is intrinsically universal for the class of two-handed systems that work at temperature τ . Also, there is an infinite hierarchy of classes of such systems with each level strictly more powerful (can assemble more complicated shapes) than the one below. In fact there are an infinite set of such hierarchies, as described in the caption of Figure 2. These results give a formalisation of the intuition that multiple long range interactions are more powerful than fewer long range interactions in the two-handed model.

By combining results from [9] and [19], we get that there is a tile set for the two-handed model that (at temperature 2) simulates any tile assembly system \mathcal{T} of the abstract Tile Assembly Model. Specifically, \mathcal{T} is simulated using the intrinsically universal tile set U from [19] (which runs at

⁵Moreover, Becker and Meunier show [8] that mismatch-free and systems without excess binding strength can not simulate each other.

temperature 2) which in turn is simulated at temperature 2 using the two-handed construction in [9].

(b) The one polygon

Demaine, Demaine, Fekete, Patitz, Schweller, Winslow, and Woods [15] take the existence of an intrinsically universal tile set for the abstract Tile Assembly Model [19] as merely the first in a sequence of simulations that routes from squares tiles, to the intrinsically universal tile set, to hexagons (with strength $< \tau$, or weak, glues) to a *single* polygon that is translatable, rotatable and flipable. Their fixed-sized polygon, when appropriately seeded, simulates any tile assembly system from the abstract Tile Assembly Model. This polygon, The One, captures the power of the entire abstract Tile Assembly Model: to simulate a tile assembly system \mathcal{T} one simply puts together a seed assembly of polygons that encodes \mathcal{T} and just lets it go! Likewise, Turing machines can be simulated with this single tile. It is also shown [15] that with translation only (i.e. no rotation), such results are not possible with a small (size ≤ 3) seed (although with larger seeds a single translation-only polyomino simulates the space-time diagram of a 1D cellular automaton). In the simpler setting of Wang plane tiling, it is shown [15] how to take any tile set T (on the square or hexagonal lattice) and “compile” it using a very simple procedure to get a single regular polygon that simulates exactly the tilings of T , except with tiny gaps between the polygons. In particular, if one starts with any aperiodic square or hexagon tile set, that tile set can be compiled to a single regular polygonal tile, all of whose tilings are aperiodic, with tiny gaps between the polygons.

(c) Signal tiles, negative glues, polyominos

Of course, one can imagine reasonable self-assembly models that are quite different from those already discussed. For a number of such models simulation has been used as a method to compare their power. These include the experimentally motivated [51] Signal-Passing [50], and Active [39], Tile Assembly Models with tiles that have molecular (DNA) wires on their surface. Essentially there is a use-once circuit sitting on the assembly itself! The circuit’s wires make use of Yurke et al’s [65] beautiful toehold mediated DNA strand displacement mechanism. Recently, Hendricks, Padilla, Patitz and Rogers [31] have shown that in the two-handed Tile Assembly Model model there is a single 3D tile set, that can simulate any 2D signal-passing tile assembly system (that does not have tile detachment). Their result shows that a constant number of planes (in the third dimension) is sufficient to handle wire crossings and asynchronous signal passing. A number of results on simulation using the (closely-related) Active Tile Assembly Model can be found in [39], including simulation of the temperature 2 abstract Tile Assembly model with a spatial scale factor of 2, and simulation of cellular automata.

Recent work [35] shows that the abstract Tile Assembly Model, and a noncooperative version of it with both square or domino (or duple) tiles have incomparable power with respect to simulation. Both models exhibit systems that can not be simulated by the other! Another paper [33] shows that negative glues (glues that repel each other) on square and domino tiles can be used, at temperature 1, to simulate the temperature 2 abstract Tile Assembly Model, and that the former model is actually more powerful than the latter. The same paper points out that the landscape of self-assembly simulation power versus computational simulation power is rather subtle in the sense that there are a number of classes of computationally universal systems that are unable to simulate the (algorithmic and geometric) process of self-assembly.

Fekete, Hendricks, Patitz, Rogers, and Schweller [23] define scale-free (or scale factor 1) simulation and show that polyominoes which are “3-position limited”, meaning they have only three perimeter locations where glues can be placed, can be simulated by the temperature 1 abstract Tile Assembly Model. (Same for 4-position limited polyominoes where those positions have a restriction called “uniquely paired”.) Appropriate future negative results on the temperature 1 abstract Tile Assembly Model will apply to these models. This result can be thought of as a self-assembly analog to the computational complexity notion that a problem is likely to have an

efficient algorithm if we can place it in a class of problems that are all strongly conjectured to have efficient algorithms, but for which we can not prove it.

As mentioned above, the abstract Tile Assembly Model can be thought of as an asynchronous and nondeterministic cellular automaton (CA) that has the notion of a crystal growth frontier. Hendricks and Patitz [32] formally relate the abstract Tile Assembly Model and CA: they give a single CA that simulates any tile assembly system, as well as a single tile set that simulates any nondeterministic CA with a finite initial configuration. The methods of updating configurations in both models are quite different (CA are infinite, synchronous and deterministic, while tile assembly is finite, asynchronous and nondeterministic) and so their constructions need to handle this. Jonoska and Karpenko [36] show that 1D cellular automata can be simulated by the Active abstract Tile Assembly Model (mentioned above) at temperature 1 by storing the time-space history in a large assembly. Further work shows that 1D and 2D cellular automata are simulated in this active model, but where (by using the feature of tile detachment) the entire space-time history does not need to be stored [37,39]. Together these pieces of work open the possibility of further comparing and contrasting the CA and tile assembly models.

5. Conclusions and future work

The results cited in this survey show that simulation between self-assembly models can be used as a method to classify their relative power. It is worth pointing out that our notion of simulation seems to strike a nice balance between being loose enough so that we can find intrinsically universal systems of various kinds but restricted enough that negative results separating the power of a various systems can actually be shown. As Figure 2 shows, we are beginning to see a kind of complexity theory for self-assembly. Indeed gaps in the figure (i.e. missing solid arrows and missing models) suggest a variety of open questions.

It is an open question whether or not the hexagonal Tile Assembly Model [15], various polygonal Tile Assembly Models [15,28], the Nubot model [64] and Signal-Passing Tile Assembly Model [31, 36] are intrinsically universal. And independent of whether or not these models turn out to be intrinsically universal, we suggest that simulation can be used to tease apart their computational and expressive power, as well as the power of subclasses of these models. For example, Gilbert, Hendricks, Patitz, and Rogers [28] investigate the computational power of various kinds of polygonal tile assembly systems, showing that regular polygon tiles with > 6 sides simulate Turing machines. What is the relationship between tile geometry and simulation power? Do more sides give strictly more simulation power?

A desirable feature of a simulator is not only that it simulates all possible dynamics of some simulated system, but that this is done in a probabilistically fair manner. Probabilities come into play when we think about the assembly process as a continuous time Markov process where tile placement is a random event (chosen uniformly from the set of possible placements) that occurs after an amount of time that is also a random variable, and in particular when we consider the number of tiles and order of their placement to assemble a given structure. Is there an intrinsically universal tile set that simulates a wide class of systems in a probabilistically fair manner? Here, the probability of seeing a given dynamics or assembly in a simulator should be close to that of the simulated system, where “close” means, say, within a factor proportional to the spatial scaling. In particular, can the intrinsically universal simulations in [19,20] be improved to have this property?

Does there exist a tile set U for the abstract Tile Assembly Model, such that for any (adversarially chosen) seed assembly σ , at temperature 2, this tile assembly system simulates some tile assembly system \mathcal{T} ? Moreover, U should be able to simulate all such members \mathcal{T} of some non-trivial class S . U is a tile set that can do one thing and nothing else: simulate tile assembly systems from the class S . This question about U is inspired by the factor simulation question in CA [13], although it differs in the details.

Many algorithmic tile assembly systems use cooperative self-assembly to simulate Turing machines in a “zig-zag” fashion, as do a number of experimentally implemented systems [5,6,56,57]. Can the negative result of [44] be extended to show 2D temperature 1 abstract Tile Assembly Model

systems do not simulate zig-zag tile assembly systems? This would be a non-trivial extension as the negative result in [44] holds for the 3D temperature 1 model, which can indeed simulate zig-zag systems, and would show that no deterministic 2D temperature 1 system can simulate Turing machines in the “usual way”. Furthermore, Fekete, Hendricks, Patitz, Rogers, Schweller [23] show that a certain, rather general, class of “geometric bit-reading gadgets” can not be built in the 2D temperature 1 model, which gives some evidence that the standard method of simulating Turing machines in the abstract Tile Assembly Model is impossible in 2D at temperature 1.

There are a number of future research directions for the two-handed model. One open question [16] asks whether or not temperature τ two-handed systems can simulate temperature $\tau - 1$ two-handed systems. (We know that in this model temperature τ systems can not simulate temperature $\tau + 1$ systems and that temperature τ systems can simulate temperature τ' systems where $\tau/\tau' \in \mathbb{N}$ [16]). Also, it is conjectured that temperature τ systems can not simulate all temperature $\tau - 1$ systems [16], even though they seem, at least naïvely, to have sufficient cooperative capabilities. Hendricks, Patitz and Rogers show progress on this question by proving that for certain simulators the answer depends on the exact notion of simulation used [34]). Another direction involves finding which aspects of the model (e.g. mismatches, excess binding strength, geometric blocking) are required for intrinsic universality at a given temperature, in order to tease apart and better understand the intricacies of this very powerful, but natural, model.

Of course, there are many other ways to compare the power of self-assembly models, for details see for example two other surveys on the theory of algorithmic tile assembly [18,52]. Researchers have looked at shape and pattern building, tile complexity, time complexity, determinism versus nondeterminism, and randomised (coin-flipping) algorithms in self-assembly. It remains as important future work to find relationships between these notions on the one hand, and intrinsic universality and simulation on the other hand. Can ideas from intrinsic universality be used to answer questions about these notions?

Acknowledgements

I would like to thank *all* of my co-authors on the topic of intrinsic universality in tile assembly, I've enjoyed working with you all immensely. It's been a fun ride so far, yet it seems there is much to learn. I thank Matt Patitz, Pierre-Étienne Meunier, Trent Rogers and Florent Becker for insights related to this article and Erik Winfree and Paul W.K. Rothmund for discussions on intrinsic universality over the years. Many thanks to Viv Kendon, Susan Stepney and Angelika Sebald for the invitation to present this work.

References

1. Leonard Adleman, Qi Cheng, Ashish Goel, and Ming-Deh Huang.
Running time and program size for self-assembled squares.
In *STOC: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 740–748, 2001.
2. Leonard M. Adleman, Qi Cheng, Ashish Goel, Ming-Deh A. Huang, David Kempe, Pablo Moisset de Espanés, and Paul W. K. Rothmund.
Combinatorial optimization problems in self-assembly.
In *STOC: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 23–32, 2002.
3. Gagan Aggarwal, Qi Cheng, Michael H. Goldwasser, Ming-Yang Kao, Pablo Moisset de Espanés, and Robert T. Schweller.
Complexities for generalized models of self-assembly.
SIAM Journal on Computing, 34:1493–1515, 2005.
4. Pablo Arrighi, Nicolas Schabanel, and Guillaume Theyssier.
Intrinsic simulations between stochastic cellular automata.
In *JAC 2012: 3rd international symposium Journées Automates Cellulaires*, volume 90 of *Electronic Proceedings in Theoretical Computer Science*, pages 208–224. Open Publishing Association, 2012.

5. Robert D Barish, Paul WK Rothmund, and Erik Winfree.
Two computational primitives for algorithmic self-assembly: Copying and counting.
Nano letters, 5(12):2586–2592, 2005.
6. Robert D Barish, Rebecca Schulman, Paul WK Rothmund, and Erik Winfree.
An information-bearing seed for nucleating algorithmic self-assembly.
Proceedings of the National Academy of Sciences, 106(15):6054–6059, 2009.
7. Kimberly Barth, David Furcy, Scott M Summers, and Paul Totzke.
Scaled tree fractals do not strictly self-assemble.
In *Unconventional Computation and Natural Computation*, pages 27–39. Springer, 2014.
Arxiv preprint [arXiv:1411.3044](https://arxiv.org/abs/1411.3044) [cs.CG].
8. Florent Becker and Pierre-Étienne Meunier.
It's a tough nanoworld: in tile assembly, cooperation is not (strictly) more powerful than competition.
Technical report, 2015.
Arxiv preprint [arXiv:1502.05558](https://arxiv.org/abs/1502.05558) [cs.CG].
9. Sarah Cannon, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Matthew J. Patitz, Robert Schweller, Scott M. Summers, and Andrew Winslow.
Two hands are better than one (up to constant factors): Self-assembly in the 2HAM vs. aTAM.
In *STACS: Proceedings of the Thirtieth International Symposium on Theoretical Aspects of Computer Science*, volume 20 of *LIPICs*, pages 172–184, 2013.
10. Cameron T Chalk, Dominic A Fernandez, Alejandro Huerta, Mario A Maldonado, Robert T Schweller, and Leslie Sweet.
Strict self-assembly of fractals using multiple hands.
Technical report, 2014.
Arxiv preprint [arXiv:1407.7900](https://arxiv.org/abs/1407.7900) [cs.CG].
11. Matthew Cook, Yunhui Fu, and Robert T. Schweller.
Temperature 1 self-assembly: Deterministic assembly in 3D and probabilistic assembly in 2D.
In *SODA 2011: Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 570–589. SIAM, 2011.
12. Marianne Delorme, Jacques Mazoyer, Nicolas Ollinger, and Guillaume Theyssier.
Bulking I: an abstract theory of bulking.
Theoretical Computer Science, 412(30):3866–3880, 2011.
13. Marianne Delorme, Jacques Mazoyer, Nicolas Ollinger, and Guillaume Theyssier.
Bulking II: Classifications of cellular automata.
Theoretical Computer Science, 412(30):3881–3905, 2011.
14. Erik D. Demaine, Martin L. Demaine, Sándor P. Fekete, Mashhood Ishaque, Eynat Rafalin, Robert T. Schweller, and Diane L. Souvaine.
Staged self-assembly: nanomanufacture of arbitrary shapes with $O(1)$ glues.
Natural Computing, 7(3):347–370, 2008.
15. Erik D. Demaine, Martin L. Demaine, Sándor P. Fekete, Matthew J. Patitz, Robert T. Schweller, Andrew Winslow, and Damien Woods.
One tile to rule them all: Simulating any tile assembly system with a single universal tile.
In *ICALP: Proceedings of the 41st International Colloquium on Automata, Languages, and Programming*, volume 8572 of *LNCS*, pages 368–379. Springer Berlin Heidelberg, 2014.
Arxiv preprint [arXiv:1212.4756](https://arxiv.org/abs/1212.4756) [cs.DS].
16. Erik D. Demaine, Matthew J. Patitz, Trent A. Rogers, Robert T. Schweller, Scott M. Summers, and Damien Woods.
The two-handed tile assembly model is not intrinsically universal.
In *ICALP: 40th International Colloquium on Automata, Languages and Programming. Proceedings, part 1*, volume 7965 of *LNCS*, pages 400–412, Riga, Latvia, July 2013.
Arxiv preprint [arXiv:1306.6710](https://arxiv.org/abs/1306.6710) [cs.CG].
17. Hendrik Dietz, Shawn M Douglas, and William M Shih.
Folding DNA into twisted and curved nanoscale shapes.
Science, 325(5941):725–730, 2009.
18. David Doty.
Theory of algorithmic self-assembly.
Communications of the ACM, 55(12):78–88, December 2012.
19. David Doty, Jack H. Lutz, Matthew J. Patitz, Robert T. Schweller, Scott M. Summers, and Damien Woods.

- The tile assembly model is intrinsically universal.
 In *FOCS: Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, pages 302–310, October 2012.
20. David Doty, Jack H. Lutz, Matthew J. Patitz, Scott M. Summers, and Damien Woods.
 Intrinsic universality in self-assembly.
 In *STACS: Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science*, pages 275–286, 2009.
 21. David Doty, Matthew J. Patitz, Dustin Reishus, Robert T. Schweller, and Scott M. Summers.
 Strong fault-tolerance for self-assembly with fuzzy temperature.
 In *FOCS: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 417–426, 2010.
 22. Constantine Glen Evans.
Crystals that count! Physical principles and experimental investigations of DNA tile self-assembly.
 PhD thesis, California Institute of Technology, 2014.
 23. Sándor P. Fekete, Jacob Hendricks, Matthew J. Patitz, Trent A. Rogers, and Robert T. Schweller.
 Universal computation with arbitrary polyomino tiles in non-cooperative self-assembly.
 In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 2015.
 24. Bin Fu, Matthew J Patitz, Robert T Schweller, and Robert Sheline.
 Self-assembly with geometric tiles.
 In *ICALP: The 39th International Colloquium on Automata, Languages and Programming*, volume 7391 of LNCS, pages 714–725. Springer, 2012.
 25. Tsu Ju Fu and Nadrian C Seeman.
 DNA double-crossover molecules.
Biochemistry, 32(13):3211–3220, 1993.
 26. Kenichi Fujibayashi, Rizal Hariadi, Sung Ha Park, Erik Winfree, and Satoshi Murata.
 Toward reliable algorithmic self-assembly of DNA tiles: A fixed-width cellular automaton pattern.
Nano Letters, 8(7):1791–1797, 2007.
 27. David Furcy and Scott M Summers.
 Scaled pier fractals do not strictly self-assemble.
 Technical report, 2014.
 Arxiv preprint [arXiv:1406.4197](https://arxiv.org/abs/1406.4197) [cs.CG].
 28. Oscar Gilbert, Jacob Hendricks, Matthew J Patitz, and Trent A Rogers.
 Computing in continuous space with self-assembling polygonal tiles.
 Technical report, 2015.
 Arxiv preprint [arXiv:1503.00327](https://arxiv.org/abs/1503.00327) [cs.CG].
 29. Eric Goles Ch., Pierre-Etienne Meunier, Ivan Rapaport, and Guillaume Theyssier.
 Communication complexity and intrinsic universality in cellular automata.
Theoretical Computer Science, 412(1-2):2–21, 2011.
 30. Dongran Han, Suchetan Pal, Jeanette Nangreave, Zhengtao Deng, Yan Liu, and Hao Yan.
 DNA origami with complex curvatures in three-dimensional space.
Science, 332(6027):342–346, 2011.
 31. Jacob Hendricks, Jennifer E Padilla, Matthew J Patitz, and Trent A Rogers.
 Signal transmission across tile assemblies: 3D static tiles simulate active self-assembly by 2D signal-passing tiles.
 Technical report, 2013.
 Arxiv preprint [arXiv:1306.5005](https://arxiv.org/abs/1306.5005) [cs.ET].
 32. Jacob Hendricks and Matthew J. Patitz.
 On the equivalence of cellular automata and the tile assembly model.
 In Turlough Neary and Matthew Cook, editors, *MCU: Machines, Computations and Universality*, Zürich, Switzerland, volume 128 of *Electronic Proceedings in Theoretical Computer Science*, pages 167–189. Open Publishing Association, 2013.
 33. Jacob Hendricks, Matthew J. Patitz, and Trent A. Rogers.
 Doubles and negatives are positive (in self-assembly).
 In *UCNC: Proceeding of Unconventional Computation and Natural Computation*, volume 8553 of LNCS, pages 190–202. Springer, 2014.
 34. Jacob Hendricks, Matthew J Patitz, and Trent A Rogers.
 The simulation powers and limitations of higher temperature hierarchical self-assembly systems.

- Technical report, 2015.
Arxiv preprint [arXiv:1503.04502](https://arxiv.org/abs/1503.04502) [cs.CG].
35. Jacob Hendricks, Matthew J. Patitz, Trent A. Rogers, and Scott M. Summers.
The power of duples (in self-assembly): It's not so hip to be square.
In *COCOON: Proceedings of 20th International Computing and Combinatorics Conference*, 2014.
 36. Nataša Jonoska and Daria Karpenko.
Active tile self-assembly, part 1: Universality at temperature 1.
International Journal of Foundation of Computer Science, 25:141–163, 2014.
doi:10.1142/S0129054114500087.
 37. Nataša Jonoska, Daria Karpenko, and Shinnosuke Seki.
Dynamic simulation of 1D cellular automata in the Active aTAM.
In submission.
 38. Lila Kari, Shinnosuke Seki, and Zhi Xu.
Triangular and hexagonal tile self-assembly systems.
In *International Workshop on Theoretical Computer Science*, volume 7160 of LNCS, pages 357–375.
Springer, 2012.
 39. Daria Karpenko.
Active Tile Self-assembly and Simulations of Computational Systems.
PhD thesis, University of South Florida, 2015.
 40. Grégory Lafitte and Michael Weiss.
Universal tilings.
In *STACS: 24th Annual Symposium on Theoretical Aspects of Computer Science*, volume 4393 of LNCS, pages 367–380. Springer, 2007.
 41. Grégory Lafitte and Michael Weiss.
An almost totally universal tile set.
In Jianer Chen and S. Barry Cooper, editors, *TAMC: Theory and Applications of Models of Computation, Changsha, China*, volume 5532 of LNCS, pages 271–280. Springer, 2009.
 42. Caleb Malchik and Andrew Winslow.
Tight bounds for active self-assembly using an insertion primitive.
In *ESA: Proceedings of European Symposium on Algorithms*, LNCS, pages 677–688. Springer, 2014.
 43. Pierre-Étienne Meunier.
Noncooperative algorithms in self-assembly.
Technical report, 2014.
Arxiv preprint [arXiv:1406.6889](https://arxiv.org/abs/1406.6889) [cs.CG].
 44. Pierre-Étienne Meunier, Matthew J. Patitz, Scott M. Summers, Guillaume Theyssier, Andrew Winslow, and Damien Woods.
Intrinsic universality in tile self-assembly requires cooperation.
In *SODA: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 752–771, 2014.
Arxiv preprint [arXiv:1304.1679](https://arxiv.org/abs/1304.1679) [cs.CG].
 45. Robin Milner.
Operational and algebraic semantics of concurrent processes.
In *Handbook of Theoretical Computer Science (vol. B)*, pages 1201–1242. MIT Press, 1991.
 46. Robin Milner and Davide Sangiorgi.
Barbed bisimulation.
In *ICALP: International Colloquium on Automata, Languages and Programming*, volume 623, pages 685–695. Springer, 1992.
 47. Turlough Neary and Damien Woods.
Small fast universal Turing machines.
Theoretical Computer Science, 362(1):171–195, 2006.
 48. Nicolas Ollinger.
Universality in cellular automata a (short) survey.
In *JAC: Symposium on Cellular Automata Journées Automates Cellulaires*, pages 102–118, 2008.
[hal-00271840](https://hal.archives-ouvertes.fr/hal-00271840).
 49. Nicolas Ollinger and Gaétan Richard.
Four states are enough!
Theoretical Computer Science, 412(1-2):22–32, 2011.
 50. Jennifer E. Padilla, Wenyan Liu, and Nadrian C. Seeman.
Hierarchical self assembly of patterns from the robinson tilings: DNA tile design in an enhanced tile assembly model.

- Natural Computing*, 11(2):323–338, June 2012.
51. Jennifer E. Padilla, Ruojie Sha, Martin Kristiansen, Junghuei Chen, Nataša Jonoska, and Nadrian C. Seeman.
A signal-passing DNA-strand-exchange mechanism for active self-assembly of DNA nanostructures.
Angewandte Chemie Int. Ed., 54, 2015.
 52. Matthew J Patitz.
An introduction to tile-based self-assembly and a survey of recent results.
Natural Computing, pages 1–30, 2013.
 53. Paul W. K. Rothmund and Erik Winfree.
The program-size complexity of self-assembled squares (extended abstract).
In *STOC: Proceedings of the thirty-second annual ACM Symposium on Theory of Computing*, pages 459–468. ACM, 2000.
 54. Paul W.K. Rothmund.
Folding DNA to create nanoscale shapes and patterns.
Nature, 440(7082):297–302, 2006.
 55. Paul W.K. Rothmund, Nick Papadakis, and Erik Winfree.
Algorithmic self-assembly of DNA Sierpinski triangles.
PLoS Biology, 2(12):2041–2053, 2004.
 56. Rebecca Schulman and Erik Winfree.
Synthesis of crystals with a programmable kinetic barrier to nucleation.
Proceedings of the National Academy of Sciences, 104(39):15236–15241, 2007.
 57. Rebecca Schulman, Bernard Yurke, and Erik Winfree.
Robust self-replication of combinatorial information via crystal growth and scission.
Proceedings of the National Academy of Sciences, 109(17):6405–6410, 2012.
 58. Nadrian C Seeman.
DNA components for molecular architecture.
Accounts of chemical research, 30(9):357–363, 1997.
 59. David Soloveichik and Erik Winfree.
Complexity of self-assembled shapes.
SIAM Journal on Computing, 36(6):1544–1569, 2007.
 60. Hao Wang.
Proving theorems by pattern recognition – II.
The Bell System Technical Journal, XL(1):1–41, 1961.
 61. Bryan Wei, Mingjie Dai, and Peng Yin.
Complex shapes self-assembled from single-stranded DNA tiles.
Nature, 485(7400):623–626, 2012.
 62. Erik Winfree.
Algorithmic Self-Assembly of DNA.
PhD thesis, California Institute of Technology, June 1998.
 63. Erik Winfree, Furong Liu, Lisa A. Wenzler, and Nadrian C. Seeman.
Design and self-assembly of two-dimensional DNA crystals.
Nature, 394(6693):539–44, 1998.
 64. Damien Woods, Ho-Lin Chen, Scott Goodfriend, Nadine Dabby, Erik Winfree, and Peng Yin.
Active self-assembly of algorithmic shapes and patterns in polylogarithmic time.
In *ITCS: Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 353–354. ACM, 2013.
Arxiv preprint [arXiv:1301.2626](https://arxiv.org/abs/1301.2626) [cs.DS].
 65. Bernard Yurke, Andrew J Turberfield, Allen P Mills, Friedrich C Simmel, and Jennifer L Neumann.
A DNA-fuelled molecular machine made of DNA.
Nature, 406(6796):605–608, 2000.
 66. Jianping Zheng, Jens J Birktoft, Yi Chen, Tong Wang, Ruojie Sha, Pamela E Constantinou, Stephan L Ginell, Chengde Mao, and Nadrian C Seeman.
From molecular to macroscopic via the rational design of a self-assembled 3D DNA crystal.
Nature, 461(7260):74–77, 2009.