PHILOSOPHICAL
TRANSACTIONS A

rsta.royalsocietypublishing.org

Intrinsic universality and the computational power of self-assembly

Damien Woods

California Institute of Technology, Pasadena, CA, USA

Review

Cite this article: Woods D. 2015 Intrinsic universality and the computational power of self-assembly. *Phil. Trans. R. Soc. A* **373**: 20140214.
http://dx.doi.org/10.1098/rsta.2014.0214

Accepted: 28 April 2015

One contribution of 13 to a Theo Murphy meeting issue 'Heterotic computing: exploiting hybrid computational devices'.

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

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 would 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 initialization 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 non-cooperative model, proving it weaker than the full Tile Assembly Model. In the two-handed model, where large structures can bind together in 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.

THE ROYAL SOCIETY
PUBLISHING

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 two-dimensional (2D) [1,2] lattices from small artificially synthesized 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 [3], as well as three-dimensional (3D) lattices [4] and shapes [5] including spheres and vases [6], and Latin, Arabic and Chinese characters [7]. These structures are hardcoded in the sense that for each position a unique DNA molecule is synthesized; the larger the structure the more unique the molecular components we need to design. Thinking of Seeman's periodic 2D DNA structures as lattices of tiles, Winfree [8] 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 [9], fractal structures [10,11], bit-copying systems [12–14] and binary counters [12,15]. 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 [8], is a kind of asynchronous non-deterministic cellular automaton that models crystal growth processes. Put another way, the abstract Tile Assembly Model restricts classical square Wang tiling [16] 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 [8] 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 $\tau \in \{1, 2, 3, \ldots\}$, as shown in figure 1*a*. Each side of a square tile has a glue (or colour) g which in turn has a strength $s \in \{0, 1, 2, \ldots\}$. 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 non-deterministically. 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 [8], and indeed computation via Turing machine simulation can be used to guide the process of self-assembly [17].

Of course individual tiles need not be square: models with triangles [18,19], hexagons [18–21] and arbitrary polyominos and polygons [19,21–23] 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 [24–27] which allows large structures to stick together if enough of their tiles' edge colours match. Another is the Nubot model [20] 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 effectivizations 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

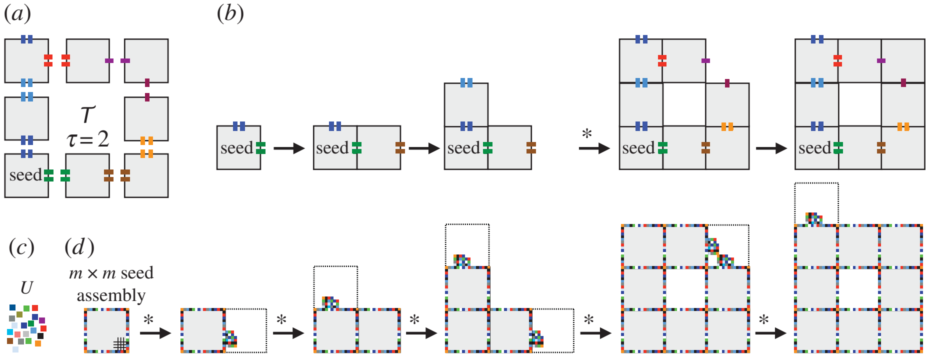

Figure 1. An instance of the abstract Tile Assembly Model, and an example showing simulation and intrinsic universality. (*a*) A tile assembly system $\mathcal{T}$ consists of a tile set, seed tile and a temperature $\tau \in \mathbb{N}$. Coloured glues on the tiles' sides have a natural number strength (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 initialized 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. (Online version in colour.)

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 summarized in figure 2 which the reader should refer to throughout the text.

Tile assemblers have borrowed and generalized the powerful idea of intrinsic universality from the cellular automata community where it has given rise to a rich theory [28–33].[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 non-deterministic. 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 [36,37]. 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

[1] This notion has also been studied in the context of Wang tiling [34,35].

[2] In particular, our definition allows for simulator assemblies α', which 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.

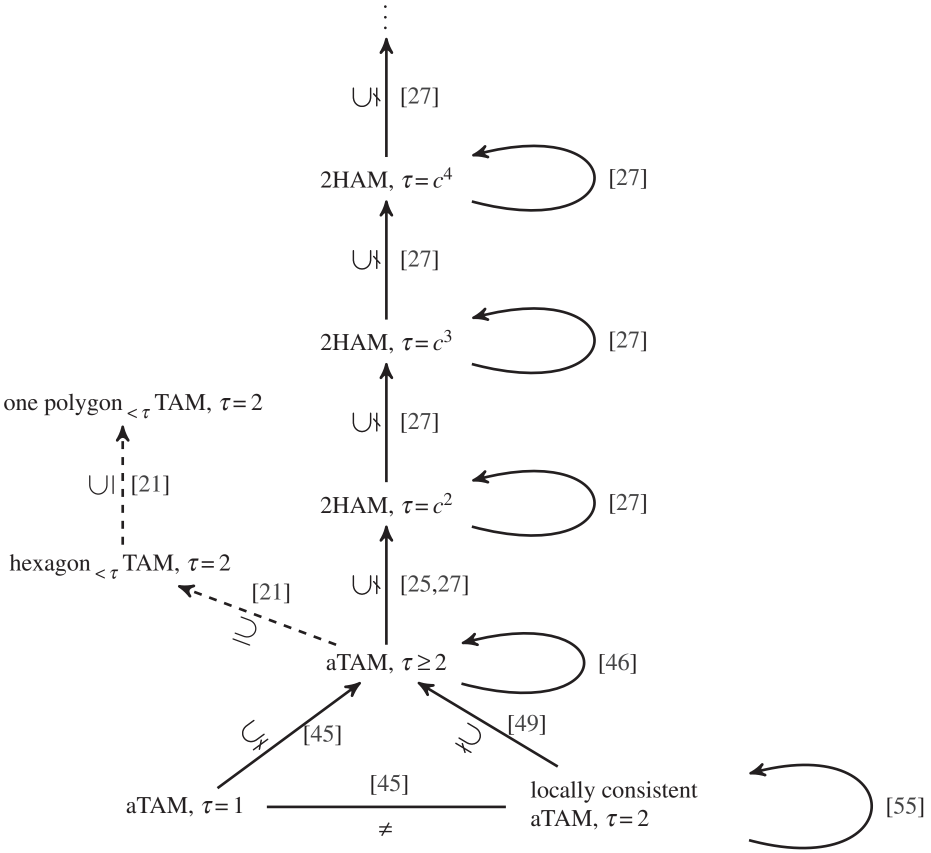

Figure 2. Classes of tile assembly systems and their relationship with respect to simulation. There is an arrow from B to A if A contains B with respect to simulation: that is, for each tile assembly system $\mathcal{B} \in B$ there is a tile assembly system $\mathcal{A}_{\mathcal{B}} \in A$ that simulates $\mathcal{B}$. 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 $c \in \{2, 3, 4, \ldots\}$ and, in fact, for each such c the set of temperatures $\{c^i | i \in \{2, 3, \ldots\}\}$ 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.

the study of intrinsic universality in self-assembly, some previous self-assembly papers [17,38] 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 [8,39], efficiency in terms of the number of tile types needed to build shapes or patterns [40,41] and computational complexity of verification of properties of tile assembly systems [42]. See other surveys for more details [43,44].

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: (i) each tile of $\mathcal{T}$ is represented by one or more $m \times m$ blocks of tiles in $\mathcal{S}$ called *supertiles*, (ii) the seed assembly of $\mathcal{T}$ is represented by the *seed assembly* of $\mathcal{S}$ (one or more connected $m \times m$ supertiles), and (iii) 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 non-deterministic distributed process with many supertiles growing independently and in parallel in the simulator system. See, for example, [45] 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 C is said to be *intrinsically universal* if there exists a single set of tiles U that simulates any instance of C. For each such simulation, U should be appropriately initialized as an instance (i.e. a tile assembly system) of C itself. For example, the abstract Tile Assembly Model has been shown to be intrinsically universal [46]. Specifically, this means that there is a single set of tiles U that, when appropriately initialized, 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 $\mathcal{U}$. Then if we observe 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 [46] the simulating system is always (merely) at temperature $\tau = 2$ no matter how large the temperature ($\tau \geq 1$) of the simulated system is.

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 have 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

[3] It 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 in [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.

Tile Assembly Model is intrinsically universal, and that this holds with a fairly strict notion of simulation, we can attempt to use 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 non-cooperative 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 [40], efficient production of $n \times n$ squares and other simple size-n shapes using $\Theta(\log n/\log\log n)$ tile types [48], efficient production of arbitrary finite connected shapes using a number of tile types within a logarithmic factor of the Kolmogorov complexity of the shape [17] and even intrinsic universality [46] (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 cannot 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 non-cooperative binding. A reasonable analogy is to think of cooperative binding as context sensitive, and non-cooperative 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. non-cooperative binding) is provably weaker than the full model [45]: 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 cannot be simulated by any non-cooperative 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 non-cooperative simulation of cooperative binding. This lemma has since found use elsewhere and indeed has been generalized in various ways [49–53]. An interesting aspect of the negative result is that it holds for 3D non-cooperative systems; they too cannot simulate arbitrary tile assembly systems. This seems quite shocking, given that 3D non-cooperative systems are Turing-universal [39]! So in particular, 3D non-cooperative 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 [45] 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 non-cooperative systems can be programmed to grow assemblies of size larger than the number of tile types [54], it remains open whether 2D non-cooperative systems are intrinsically universal for themselves, or capable of Turing machine simulation in an error-free way.

[4] Or as Paul W.K. Rothemund has put it, 3D non-cooperative systems can dream about tile assembly, but cannot actually *do* tile assembly.

It was also shown that in 3D there is a (cube) tile set that non-cooperatively simulates all 2D non-cooperative systems [45].

(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 [55]. 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 laboratory to date with DNA [10–15]. This begs the question: are locally consistent systems capable of simulating the full model? Recent work by Becker & Meunier [49] shows that the answer is 'no'. In particular, their results show that any class of tile assembly systems that has no mismatches, or disallows excess binding strength, cannot 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 [46] 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 cannot simulate those with mismatches [49] tells us that this aspect of the simulation is required. One of the key innovations in [49] is to generalize the window movie lemma (a pumping lemma) from [45] so that it can be applied in significantly more complicated settings. It will be interesting to see if this generalized 'bisimulation lemma' finds use elsewhere. Finally, Becker & Meunier [49] show 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 characterize 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 [27]. 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 $\tau \in \{2, 3, 4, \ldots\}$ there is a tile set 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 formalization 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 [25,46], 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 [46] (which

[5] Moreover, Becker & Meunier [49] show that mismatch-free systems and systems without excess binding strength cannot simulate each other.

runs at temperature 2) which in turn is simulated at temperature 2 using the two-handed construction in [25].

(b) The one polygon

Demaine *et al.* [21] take the existence of an intrinsically universal tile set for the abstract Tile Assembly Model [46] as merely the first in a sequence of simulations that routes from square 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 [21] 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 one-dimensional (1D) cellular automaton). In the simpler setting of Wang plane tiling it is shown [21] 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 complied to a single regular polygonal tile, all of who's 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 [56] Signal-Passing [57], and Active [58], 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 [59] beautiful toehold-mediated DNA strand displacement mechanism. Recently, Hendricks *et al.* [60] have shown that in the two-handed Tile Assembly 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 Karpenko [58], including simulation of the temperature 2 abstract Tile Assembly Model with a spatial scale factor of 2, and simulation of cellular automata.

Recent work [50] shows that the abstract Tile Assembly Model and a non-cooperative version of it with both square or domino (or duple) tiles have incomparable power with respect to simulation. Both models exhibit systems that cannot be simulated by the other! Another paper [61] 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 *et al.* [22] 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 analogue 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 cannot prove it.

As mentioned above, the abstract Tile Assembly Model can be thought of as an asynchronous and non-deterministic cellular automaton (CA) that has the notion of a crystal growth frontier. Hendricks & Patitz [62] 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 non-deterministic 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 non-deterministic) and so their constructions need to handle this. Jonoska & Karpenko [63] show that 1D CA 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 CA 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 [58,64]. Together these pieces of work open the possibility of further comparing and contrasting the CA and tile assembly models.

5. Conclusion 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 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 [21], various polygonal tile assembly models [19,21], the Nubot model [20] and Signal-Passing Tile Assembly Model [60,63] 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 *et al.* [19] investigate the computational power of various kinds of polygonal tile assembly systems, showing that regular polygon tiles with more than six 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 [46,55] 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 [29], 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 [12–14,65]. Can the negative result of [45] 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 [45] 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 *et al.* [22] show that a certain, rather general, class of 'geometric bit-reading gadgets' cannot 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 [27] 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 cannot simulate temperature $\tau+1$ systems and that temperature τ systems can simulate temperature τ' systems where $\tau/\tau' \in \mathbb{N}$ [27]. Also, it is conjectured that temperature τ systems cannot simulate all temperature $\tau-1$ systems [27], even though they seem, at least naively, to have sufficient cooperative capabilities. Hendricks *et al.* [66] show progress on this question by proving that for certain simulators the answer depends on the exact notion of simulation used.) 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 [43,44]. Researchers have looked at shape and pattern building, tile complexity, time complexity, determinism versus non-determinism and randomized (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 thank *all* my co-authors on the topic of intrinsic universality in tile assembly, I have enjoyed working with you all immensely. It has 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. Rothemund 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. TJ Fu, Seeman NC. 1993 DNA double-crossover molecules. *Biochemistry* **32**, 3211–3220. (doi:10.1021/bi00064a003)
2. Seeman NC. 1997 DNA components for molecular architecture. *Acc. Chem. Res.* **30**, 357–363. (doi:10.1021/ar9601407)
3. Rothemund PWK. 2006 Folding DNA to create nanoscale shapes and patterns. *Nature* **440**, 297–302. (doi:10.1038/nature04586)
4. Zheng J, Birktoft JJ, Chen Y, Wang T, Sha R, Constantinou PE, Ginell SL, Mao C, Seeman NC. 2009 From molecular to macroscopic via the rational design of a self-assembled 3D DNA crystal. *Nature* **461**, 74–77. (doi:10.1038/nature08274)
5. Dietz H, Douglas SM, Shih WM. 2009 Folding DNA into twisted and curved nanoscale shapes. *Science* **325**, 725–730. (doi:10.1126/science.1174251)
6. Han D, Pal S, Nangreave J, Deng Z, Liu Y, Yan H. 2011 DNA origami with complex curvatures in three-dimensional space. *Science* **332**, 342–346. (doi:10.1126/science.1202998)
7. Wei B, Dai M, Yin P. 2012 Complex shapes self-assembled from single-stranded DNA tiles. *Nature* **485**, 623–626. (doi:10.1038/nature11075)
8. Winfree E. 1998 Algorithmic self-assembly of DNA. PhD thesis, California Institute of Technology, Pasadena, CA, USA.
9. Winfree E, Liu F, Wenzler LA, Seeman NC. 1998 Design and self-assembly of two-dimensional DNA crystals. *Nature* **394**, 539–544. (doi:10.1038/28998)
10. Rothemund PWK, Papadakis N, Winfree E. 2004 Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biol.* **2**, 2041–2053. (doi:10.1371/journal.pbio.0020424)

11. Fujibayashi K, Hariadi R, Park SH, Winfree E, Murata S. 2007 Toward reliable algorithmic self-assembly of DNA tiles: a fixed-width cellular automaton pattern. *Nano Lett.* **8**, 1791–1797. (doi:10.1021/nl0722830)
12. Barish RD, Rothemund PWK, Winfree E. 2005 Two computational primitives for algorithmic self-assembly: copying and counting. *Nano Lett.* **5**, 2586–2592. (doi:10.1021/nl052038l)
13. Barish RD, Schulman R, Rothemund PWK, Winfree E. 2009 An information-bearing seed for nucleating algorithmic self-assembly. *Proc. Natl Acad. Sci. USA* **106**, 6054–6059. (doi:10.1073/pnas.0808736106)
14. Schulman R, Yurke B, Winfree E. 2012 Robust self-replication of combinatorial information via crystal growth and scission. *Proc. Natl Acad. Sci. USA* **109**, 6405–6410. (doi:10.1073/pnas.1117813109)
15. Evans CG. 2014 Crystals that count! Physical principles and experimental investigations of DNA tile self-assembly. PhD thesis, California Institute of Technology, Pasadena, CA, USA.
16. Wang H. 1961 Proving theorems by pattern recognition—II. *Bell Syst. Tech. J.* **XL**, 1–41. (doi:10.1002/j.1538-7305.1961.tb03975.x)
17. Soloveichik D, Winfree E. 2007 Complexity of self-assembled shapes. *SIAM J. Comput.* **36**, 1544–1569. (doi:10.1137/S0097539704446712)
18. Kari L, Seki S, Xu Z. 2012 Triangular and hexagonal tile self-assembly systems. In *Computation, physics and beyond* (eds MJ Dinneen, B Khoussainov, A Nies). Lecture Notes in Computer Science, vol. 7160, pp. 357–375. New York, NY: Springer. (doi:10.1007/978-3-642-27654-5_28)
19. Gilbert O, Hendricks J, Patitz MJ, Rogers TA. 2015 Computing in continuous space with self-assembling polygonal tiles. (http://arxiv.org/abs/1503.00327)
20. 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 *Proc. 4th Conf. on Innovations in Theoretical Computer Science*, pp. 353–354. New York, NY: ACM. (doi:10.1145/2422436.2422476)
21. Demaine ED, Demaine ML, Fekete SP, Patitz MJ, Schweller RT, Winslow A, Woods D. 2014 One tile to rule them all: simulating any tile assembly system with a single universal tile. In *Automata, languages, and programming* (eds J Esparza, P Fraigniaud, T. Husfeldt, E Koutsoupias). Lecture Notes in Computer Science, vol. 8572, pp. 368–379. Berlin, Germany: Springer. (doi:10.1007/978-3-662-43948-7_31)
22. Fekete SP, Hendricks J, Patitz MJ, Rogers TA, Schweller RT. 2015 Universal computation with arbitrary polyomino tiles in non-cooperative self-assembly. In *Proc. 26th Annual ACM-SIAM Symp. on Discrete Algorithms, San Diego, CA, 4–6 January 2015*, pp. 148–167.
23. Fu B, Patitz MJ, Schweller RT, Sheline R. 2012 Self-assembly with geometric tiles. In *Automata, languages, and programming* (eds A Czumaj, K Mehlhorn, A pitts, R Wattenhofer). Lecture Notes in Computer Science, vol. 7391, pp. 714–725. Berlin, Germany: Springer. (doi:10.1007/978-3-642-31594-7_60)
24. Aggarwal G, Cheng Q, Goldwasser MH, Kao M-Y, deEspanés PM, Schweller RT. 2005 Complexities for generalized models of self-assembly. *SIAM J. Comput.* **34**, 1493–1515. (doi:10.1137/S0097539704445202)
25. Cannon S, Demaine ED, Demaine ML, Eisenstat S, Patitz MJ, Schweller R, Summers SM, Winslow A. 2013 Two hands are better than one (up to constant factors): self-assembly in the 2HAM vs. aTAM. In *STACS: Proc. 30th Int. Symp. on Theoretical Aspects of Computer Science*. LIPIcs, vol. 20, pp. 172–184.
26. Doty D, Patitz MJ, Reishus D, Schweller RT, Summers SM. 2010 Strong fault-tolerance for self-assembly with fuzzy temperature. In *Proc. 51st Annual IEEE Symp. on Foundations of Computer Science*, pp. 417–426. (doi:10.1109/FOCS.2010.47)
27. Demaine ED, Patitz MJ, Rogers TA, Schweller RT, Summers SM, Woods D. 2013 The two-handed tile assembly model is not intrinsically universal. In *Automata, languages, and programming, 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8–12, 2013, Proceedings, Part I* (eds FV Formin, R Freivalds, M Kwiatkowska, D Peleg). Lecture Notes in Computer Science, vol. 7965, pp. 400–412. Berlin, Germany: Springer. (doi:10.1007/978-3-642-39206-1_34) (http://arxiv.org/abs/1306.6710)
28. Delorme M, Mazoyer J, Ollinger N, Theyssier G. 2011 Bulking I: an abstract theory of bulking. *Theor. Comput. Sci.* **412**, 3866–3880. (doi:10.1016/j.tcs.2011.02.023)
29. Delorme M, Mazoyer J, Ollinger N, Theyssier G. 2011 Bulking II: classifications of cellular automata. *Theor. Comput. Sci.* **412**, 3881–3905. (doi:10.1016/j.tcs.2011.02.024)
30. Ollinger N. 2008 Universalities in cellular automata: a (short) survey. In *Proc. Symp. on Cellular Automata Journées Automates Cellulaires, Uzès, France, 21–25 April 2008*, pp. 102–118.

31. Ollinger N, Richard G. 2011 Four states are enough! *Theor. Comput. Sci.* **412**, 22–32. (doi:10.1016/j.tcs.2010.08.018)
32. Goles E, Meunier P-É, Rapaport I, Theyssier G. 2011 Communication complexity and intrinsic universality in cellular automata. *Theor. Comput. Sci.* **412**, 2–21. (doi:10.1016/j.tcs.2010.10.005)
33. Arrighi P, Schabanel N, Theyssier G. 2012 Intrinsic simulations between stochastic cellular automata. In *JAC 2012: 3rd Int. Symp. Journées Automates Cellulaires. Electron. Proc. Theor. Comput. Sci.* **90**, 208–224. (doi:10.4204/EPTCS.90.17)
34. Lafitte G, Weiss M. 2007 Universal tilings. In *STACS 2007* (eds W Thomas, P Weil). Lecture Notes in Computer Science, vol. 4393, pp. 367–380. Berlin, Germany: Springer. (doi:10.1007/978-3-540-70918-3_32)
35. Lafitte G, Weiss M. 2009 An almost totally universal tile set. In *Theory and applications of models of computation* (eds J Chen, S Cooper). Lecture Notes in Computer Science, vol. 5532, pp. 271–280. Berlin, Germany: Springer. (doi:10.1007/978-3-642-02017-9_30)
36. Milner R. 1991 Operational and algebraic semantics of concurrent processes. In *Handbook of theoretical computer science*, vol. B, pp. 1201–1242. Cambridge, MA: MIT Press.
37. Milner R, Sangiorgi D. 1992 Barbed bisimulation. In *Automata, languages and programming. Int. Colloquium on Automata, Languages and Programming* (ed. W Kuich). Lecture Notes in Computer Science, vol. 623, pp. 685–695. Berlin, Germany: Springer. (doi:10.1007/3-540-55719-9_114)
38. 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**, 347–370. (doi:10.1007/s11047-008-9073-0)
39. Cook M, Fu Y, Schweller RT. 2011 Temperature 1 self-assembly: deterministic assembly in 3D and probabilistic assembly in 2D. In *SODA 2011: Proc. 22nd Annual ACM-SIAM Symp. on Discrete Algorithms*, pp. 570–589. Philadelphia, PA: SIAM.
40. Rothemund PWK, Winfree E. 2000 The program-size complexity of self-assembled squares (extended abstract). In *Proc. 32nd Annual ACM Symp. on Theory of Computing*, pp. 459–468. New York, NY: ACM. (doi:10.1145/335305.335358)
41. Malchik C, Winslow A. 2014 Tight bounds for active self-assembly using an insertion primitive. In *Algorithms—ESA 2014. Proc. European Symposium on Algorithms* (eds AS Schulz, D Wagner). Lecture Notes in Computer Science, vol. 8737, pp. 677–688. Berlin, Germany: Springer. (doi:10.1007/978-3-662-44777-2_56)
42. Adleman LM, Cheng Q, Goel A, Huang M-DA, Kempe D, de Espanés PM, Rothemund PWK. 2002 Combinatorial optimization problems in self-assembly. In *Proc. 34th Annual ACM Symp. on Theory of Computing*, pp. 23–32. New York, NY: ACM. (doi:10.1145/509907.509913)
43. Patitz MJ. 2013 An introduction to tile-based self-assembly and a survey of recent results. *Nat. Comput.* **13**, 195–224. (doi:10.1007/s11047-013-9379-4)
44. Doty D. 2012 Theory of algorithmic self-assembly. *Commun. ACM* **55**, 78–88. (doi:10.1145/2380656.2380675)
45. Meunier P-É, Patitz MJ, Summers SM, Theyssier G, Winslow A, Woods D. 2014 Intrinsic universality in tile self-assembly requires cooperation. In *SODA: Proc. ACM-SIAM Symp. on Discrete Algorithms*, pp. 752–771. (http://arxiv.org/abs/1304.1679)
46. Doty D, Lutz JH, Patitz MJ, Schweller RT, Summers SM, Woods D. 2012 The tile assembly model is intrinsically universal. In *FOCS: Proc. 53rd Annual IEEE Symp. on Foundations of Computer Science*, pp. 302–310. (doi:10.1109/FOCS.2012.76)
47. Neary T, Woods D. 2006 Small fast universal Turing machines. *Theor. Comput. Sci.* **362**, 171–195. (doi:10.1016/j.tcs.2006.06.002)
48. Adleman L, Cheng Q, Goel A, Huang M-D. 2001 Running time and program size for self-assembled squares. In *Proc. 33rd Annual ACM Symp. on Theory of Computing*, pp. 740–748. New York, NY: ACM. (doi:10.1145/380752.380881)
49. Becker F, Meunier P-É. 2015 It's a tough nanoworld: in tile assembly, cooperation is not (strictly) more powerful than competition. (http://arxiv.org/abs/1502.05558)
50. Hendricks J, Patitz MJ, Rogers TA, Summers SM. 2014 The power of duples (in self-assembly): it's not so hip to be square. In *COCOON: Proc. 20th Int. Computing and Combinatorics Conference*. (http://arxiv.org/abs/1402.4515)
51. Furcy D, Summers SM. 2014 Scaled pier fractals do not strictly self-assemble. (http://arxiv.org/abs/1406.4197)

52. Barth K, Furcy D, Summers SM, Totzke P. 2014 Scaled tree fractals do not strictly self-assemble. In *Unconventional computation and natural computation* (eds OH Ibarra, L Kari, S Kopecki). Lecture Notes in Computer Science, vol. 8553, pp. 27–39. Berlin, Germany: Springer. (doi:10.1007/978-3-319-08123-6_3)
53. Chalk CT, Fernandez DA, Huerta A, Maldonado MA, Schweller RT, Sweet L. 2014 Strict self-assembly of fractals using multiple hands. (http://arxiv.org/abs/1407.7900)
54. Meunier P-É. 2015 Noncooperative algorithms in self-assembly. *Unconventional Computing and Natural Computing* (*UCNC*). (http://arxiv.org/ abs/1406.6889)
55. Doty D, Lutz JH, Patitz MJ, Summers SM, Woods D. 2009 Intrinsic universality in self-assembly. In *STACS: Proc. 27th Int. Symp. on Theoretical Aspects of Computer Science*, pp. 275–286.
56. Padilla JE, Sha R, Kristiansen M, Chen J, Jonoska N, Seeman NC. 2015 A signal-passing DNA-strand-exchange mechanism for active self-assembly of DNA nanostructures. *Angew. Chem. Int. Ed.* **54**, 5939–5942. (doi:10.1002/anie.201500252)
57. 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**, 323–338. (doi:10.1007/s11047-011-9268-7)
58. Karpenko D. 2015 Active tile self-assembly and simulations of computational systems. PhD thesis, University of South Florida, Tampa, FL, USA.
59. Yurke B, Turberfield AJ, Mills AP, Simmel FC, Neumann JL. 2000 A DNA-fuelled molecular machine made of DNA. *Nature* **406**, 605–608. (doi:10.1038/35020524)
60. Hendricks J, Padilla JE, Patitz MJ, Rogers TA. 2013 Signal transmission across tile assemblies: 3D static tiles simulate active self-assembly by 2D signal-passing tiles. (http://arxiv.org/abs/1306.5005)
61. Hendricks J, Patitz MJ, Rogers TA. 2014 Doubles and negatives are positive (in self-assembly). In *Unconventional computation and natural computation* (eds OH Ibarra, L Kari, S Kopecki). Lecture Notes in Computer Science, vol. 8553, pp. 190–202. Berlin, Germany: Springer. (doi:10.1007/978-3-319-08123-6_16)
62. Hendricks J, Patitz MJ. 2013 On the equivalence of cellular automata and the tile assembly model. In *MCU: Machines, Computations and Universality, Zürich, Switzerland*, vol. 128 (eds T Neary, M Cook). *Electron. Proc. Theor. Comput. Sci.* **128**, 167–189. (doi:10.4204/EPTCS.128.21)
63. Jonoska N, Karpenko D. 2014 Active tile self-assembly, part 1: universality at temperature 1. *Int. J. Found. Comput. Sci.* **25**, 141–163. (doi:10.1142/S0129054114500087)
64. Jonoska N, Karpenko D, Seki S. In press. Dynamic simulation of 1D cellular automata in the Active aTAM.
65. Schulman R, Winfree E. 2007 Synthesis of crystals with a programmable kinetic barrier to nucleation. *Proc. Natl Acad. Sci. USA* **104**, 15 236–15 241. (doi:10.1073/pnas.0701467104)
66. Hendricks J, Patitz MJ, Rogers TA. 2015 The simulation powers and limitations of higher temperature hierarchical self-assembly systems. (http://arxiv.org/abs/1503.04502)