The Impacts of Dimensionality, Diffusion, and Directedness on Intrinsic Cross-Model Simulation in Tile-Based Self-Assembly

Daniel Hader ✉
Department of Computer Science and Computer Engineering,
University of Arkansas, Fayetteville, AR, USA

Matthew J. Patitz ✉
Department of Computer Science and Computer Engineering,
University of Arkansas, Fayetteville, AR, USA

Abstract

Algorithmic self-assembly occurs when components in a disorganized collection autonomously combine to form structures and, by their design and the dynamics of the system, are forced to intrinsically follow the execution of algorithms. Motivated by applications in DNA-nanotechnology, theoretical investigations in algorithmic tile-based self-assembly have blossomed into a mature theory with research strongly leveraging tools from computability theory, complexity theory, information theory, and graph theory to develop a wide range of models and to show that many are computationally universal, while also exposing a wide variety of powers and limitations of each. In addition to computational universality, the abstract Tile-Assembly Model (aTAM) was shown to be intrinsically universal (FOCS 2012), a strong notion of completeness where a single tile set is capable of simulating the full dynamics of all systems within the model; however, this result fundamentally required non-deterministic tile attachments. This was later confirmed necessary when it was shown that the class of directed aTAM systems, those in which all possible sequences of tile attachments eventually result in the same terminal assembly, is not intrinsically universal (FOCS 2016). Furthermore, it was shown that the non-cooperative aTAM, where tiles only need to match on 1 side to bind rather than 2 or more, is not intrinsically universal (SODA 2014) nor computationally universal (STOC 2017). Building on these results to further investigate the impacts of other dynamics, Hader et al. examined several tile-assembly models which varied across (1) the numbers of dimensions used, (2) restrictions imposed on the diffusion of tiles through space, and (3) whether each system is directed, and determined which models exhibited intrinsic universality (SODA 2020). Such results have shed much light on the roles of various aspects of the dynamics of tile-assembly and their effects on the universality of each model. In this paper we extend that previous work to provide direct comparisons of the various models against each other by considering intrinsic simulations between models. Our results show that in some cases, one model is strictly more powerful than another, and in others, pairs of models have mutually exclusive capabilities. This direct comparison of models helps expose the impacts of these three important aspects of self-assembling systems, and further helps to define a hierarchy of tile-assembly models analogous to the hierarchies studied in traditional models of computation.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Tile-Assembly, Tiles, aTAM, Intrinsic Simulation, Simulation

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.71

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2305.01877` [8]

Funding *Daniel Hader*: Supported in part by National Science Foundation Grant CAREER-1553166.
Matthew J. Patitz: Supported in part by National Science Foundation Grant CAREER-1553166.

© Daniel Hader and Matthew J. Patitz;
licensed under Creative Commons License CC-BY 4.0

1 Introduction

Self-assembling systems are those in which a disorganized collection of simple components spontaneously combine to form complex, organized structures through random motion and local interactions. From the pristine, periodic arrangements formed by crystallizing atoms to the robust coordination of dividing cells in developing organisms, such systems are the source of much complexity in nature and a topic of critical importance to many fields of research. Among them is the field of DNA nanotechnology, wherein artificial DNA strands are used as structural units that self-assemble according to the dynamics of DNA base pairing, which has seen immense success over the past several decades in harnessing the power of self-assembly to create microscopic structures with incredible precision [3, 11, 12, 18] and even perform algorithmic tasks at the nano-scale [4, 6, 13, 14, 17, 20, 24, 25]. Because it's difficult and expensive to accurately model the chemistry of DNA, a variety of simplifying models have been proposed to facilitate the design of DNA-based self-assembling systems. Among the more popular and effective ones are tile-assembly (TA) models where components, made of several bound DNA strands exposing small unbound portions with which other components can bind, are abstractly represented as geometric tiles whose labeled sides attach to one another according to predefined affinity rules [5, 16, 22]. The advantage of these models lies not only in their success as design tools, but in their similarity to existing models studied heavily in computer science such as Wang tiles and cellular automata. This similarity isn't a coincidence either; the first TA model proposed, the abstract Tile-Assembly Model (aTAM), was designed, at least in part, to show that the dynamics of DNA-based self-assembly are algorithmically universal [22]. Consequently, DNA nanotechnology shares a unique relationship with the theory of computation, with theorists frequently borrowing ideas from complexity, computability, and information theory to study questions regarding, among many other things, what kinds of structures can be self-assembled, the relative difficulty of assembling different shapes, and how variations in a model's dynamics affect its algorithmic power. This paper is particularly focused on that latter question. As with more conventional

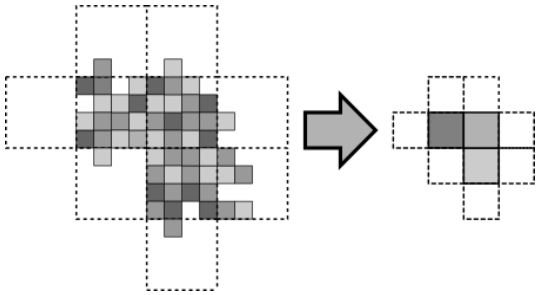

Figure 1 During an intrinsic simulation, the dynamics of individual tile attachments are simulated so that blocks of tiles in the simulating system "look like" individual tiles at scale.

models of computation, we generally study such questions by proving whether one model is capable of simulating all systems of another. We have to be careful about our definition of simulation however, as it's generally straightforward to show that many TA models are capable of universal computation. Consequently, most TA models are capable of "simulating" all others in that they can simulate a Turing machine which can in turn simulate the other model. To learn something useful about the relative power of two TA models therefore, we have to consider the geometry of the tile-assembly dynamics. We do this by adapting a tool from the theory of cellular automata, namely *intrinsic simulation*. For a simulation

to be *intrinsic*, we require that the simulation is not merely symbolic (i.e. how a Turing machine can simulate an aTAM system by storing an internal representation of the tiles as symbols on its tape), but rather geometric wherein blocks of tiles in the simulating system correspond to individual tiles in the simulated system and the order of tile attachments in these blocks follow those in the simulated system up to a fixed scale factor. In other words, such a simulation would appear identical to the system being simulated if we "zoomed out" sufficiently far. This approach is not novel to our results, in fact there is already a relatively mature theory of intrinsic simulations in tile-assembly which has resulted in a "kind of computational complexity theory for self-assembly" [23]. Such efforts have been instrumental in characterizing the relative power of TA models and has lead to a deeper understanding how different dynamics can be used for the same algorithmic purpose.

Our results

In an attempt to extend several previous results regarding intrinsic simulation, here we consider 3 specific variations of the aTAM: *dimensionality*, where both 2D and 3D systems are considered, *diffusion*, where tiles cannot attach in regions which have been surrounded by previously attached tiles, and *directedness*, where tile attachments in a system are required to result in exactly one terminal assembly. It's important to note that these variations aren't arbitrary either. The difference between directed and undirected systems is analogous to the difference between deterministic and probabilistic algorithms and, among other things, plays a role in the study of the complexity of shape assembly [21, 10]. The diffusion restriction on the other hand is often used to make 3D tile-assembly models more "realistic" by limiting tile attachments to those locations in which a tile could reasonably diffuse (i.e. not in a region completely surrounded by other tiles). These variations can be introduced into the aTAM in any combination to yield 8 different models and, considering all ordered pairs of these 8 models gives rise to a table consisting of 64 entries each representing one model's ability or inability to intrinsically simulate the dynamics of another. Generally speaking, results regarding these *cross-model simulations* are complex, involving intricate tile-assembly constructions and counterexamples; consequently, only a handful of these entries have been proved in past literature.

In this paper, we fill a considerable number of missing entries. Table 1 lays out our results along with past results denoted by an asterisk. In it, entries are labeled to indicate whether the model in the row's header can simulate the model in the column's header. There are of course a few entries for which the answer is obvious, which we state as observations with justification rather than full theorems, but many of our results are distinctly non-trivial and some were rather unexpected. For instance, while we initially suspected that the diffusion restricted version of the aTAM (i.e. the Planar aTAM or PaTAM) was, as it's name suggests, a weaker version of the aTAM, we found that both models exhibit dynamics which cannot be simulated by the other. While the table is still missing a few entries, our contributions have brought the number of known entries up to 52 from the 16 which previously existed in published literature (8 of which were technically not explicitly stated, but were trivial observations based on the tile sets and proofs presented in [7]).[1] The rest of our paper is laid out as follows. In Section 2, we provide definitions of the various models and concepts used.

[1] It should also be noted that most of the remaining unknown entries involve simulating directed, diffusion restricted systems. While we do hope to fill these entries in the future, we suspect that their proofs will be quite complicated since simulating diffusion restricted systems is tricky and counterexamples are often harder to find in directed systems.

Table 1 Table of our results, outlining whether the row's model can intrinsically simulate the column's model. PaTAM is the Planar aTAM, 3DaTAM the 3-dimensional aTAM, and SaTAM is the Spatial aTAM (see Section 2.2 for full definitions). *All* refers to the set of all systems in a model and *dir* refers to the subset of directed systems. Cells marked with an asterisk (*) are existing results and those marked with a dagger (†) are trivial observations using tile sets from existing results. All other results are novel.

		aTAM		**PaTAM**	
		all	**dir**	**all**	**dir**
aTAM	**all**	yes* [2]		no (thm. 11, obs. 5)	?
	dir	no (thm. 9)	no* [9]		?
PaTAM	**all**	no (thm. 10, obs. 5)		no* [7]	yes (thm. 13)
	dir			no (thm. 9)	no* [7]
3DaTAM	**all**	yes†(obs. 7)		?	?
	dir	no (thm. 9)	yes† (obs. 7)	no (thm. 9)	?
SaTAM	**all**	yes† (obs. 7)		?	?
	dir	no (thm. 9)	yes† (obs. 7)	no (thm. 9)	?

		3DaTAM		**SaTAM**	
		all	**dir**	**all**	**dir**
aTAM	**all**	no (obs. 6)			
	dir				
PaTAM	**all**				
	dir				
3DaTAM	**all**	yes* [7]		no (thm. 12, obs. 5)	?
	dir	no (thm. 9)	yes* [7]		?
SaTAM	**all**	yes† (obs. 8)		yes* [7]	?
	dir	no (thm. 9)	yes† (obs. 8)	no (thm. 9)	?

Then in Section 3 we state our results explicitly and sketch their proofs. That section is perhaps the most important since it is where we intuitively explain our results and describe how they follow from the dynamics of the model. Complete proofs and technical details can be found in a full version of the paper on arXiv [8].

2 Preliminary definitions

Throughout this paper we will use $\mathbb{Z}$, $\mathbb{Z}^+$, and $\mathbb{N}$ to denote the set of integers, positive integers, and non-negative integers respectively. We will also assume $\mathbb{Z}^d$ has the additional structure of a lattice graph so that each point is a vertex and two points are adjacent (i.e. share an edge) exactly when their Euclidean distance is 1.

2.1 Definition of the abstract Tile-Assembly Model

In this section, we define the abstract Tile-Assembly Model in 2 and 3 dimensions. We will use the abbreviation *aTAM* to refer to the 2D model and *3DaTAM* for the 3D model. These definitions are borrowed from [7] and we note that [19] is a good introduction to the model for unfamiliar readers.

Fix $d \in \{2, 3\}$ to be the number of dimensions and Σ to be some alphabet with Σ^* its finite strings. A *glue* $g \in \Sigma^* \times \mathbb{N}$ consists of a finite string *label* and non-negative integer *strength*. A *tile type* is a tuple $t \in (\Sigma^* \times \mathbb{N})^{2d}$, thought of as a unit square or cube with a glue on each side. A *tile set* is a finite set of tile types. We always assume a finite set of tile types, but allow an infinite number of copies of each tile type to occupy locations in the $\mathbb{Z}^d$ lattice, each called a *tile*.

Given a tile set T, a *configuration* is an arrangement (possibly empty) of tiles in the lattice $\mathbb{Z}^d$, i.e. a partial function $\alpha : \mathbb{Z}^d \dashrightarrow T$. Two adjacent tiles in a configuration *interact*, or are *bound* or *attached*, if the glues on their abutting sides are equal (in both label and strength) and have positive strength. Each configuration α induces a *binding graph* B_α whose vertices are those points occupied by tiles, with an edge of weight s between two vertices if the corresponding tiles interact with strength s. An *assembly* is a configuration whose domain (as a graph) is connected and non-empty. The *shape* $S_\alpha \subseteq \mathbb{Z}^d$ of assembly α is the domain of α. For some $\tau \in \mathbb{Z}^+$, an assembly α is *τ-stable* if every cut of B_α has weight at least τ, i.e. a τ-stable assembly cannot be split into two pieces without separating bound tiles whose shared glues have cumulative strength τ. Given two assemblies α, β, we say α is a *subassembly* of β (denoted $\alpha \sqsubseteq \beta$) if $S_\alpha \subseteq S_\beta$ and for all $p \in S_\alpha$, $\alpha(p) = \beta(p)$.

A *tile-assembly system* (TAS) is a triple $\mathcal{T} = (T, \sigma, \tau)$, where T is a tile set, σ is a finite τ-stable assembly called the *seed assembly*, and $\tau \in \mathbb{Z}^+$ is called the *binding threshold*. Given a TAS $\mathcal{T} = (T, \sigma, \tau)$ and two τ-stable assemblies α and β, we say that α *$\mathcal{T}$-produces β in one step* (written $\alpha \to_1^{\mathcal{T}} \beta$) if $\alpha \sqsubseteq \beta$ and $|S_\beta \setminus S_\alpha| = 1$. That is, $\alpha \to_1^{\mathcal{T}} \beta$ if β differs from α by the addition of a single tile. The *$\mathcal{T}$-frontier* is the set $\partial^{\mathcal{T}} \alpha = \bigcup_{\alpha \to_1^{\mathcal{T}} \beta} S_\beta \setminus S_\alpha$ of locations in which a tile could τ-stably attach to α.

We use $\mathcal{A}^T$ to denote the set of all assemblies of tiles in tile set T. Given a TAS $\mathcal{T} = (T, \sigma, \tau)$, a sequence of $k \in \mathbb{Z}^+ \cup \{\infty\}$ assemblies $\alpha_0, \alpha_1, \ldots$ over $\mathcal{A}^T$ is called a *$\mathcal{T}$-assembly sequence* if, for all $1 \leq i < k$, $\alpha_{i-1} \to_1^{\mathcal{T}} \alpha_i$. The *result* of an assembly sequence is the unique limiting assembly of the sequence. For finite assembly sequences, this is the final assembly; whereas for infinite assembly sequences, this is the assembly consisting of all tiles from any assembly in the sequence. We say that α *$\mathcal{T}$-produces* β (denoted $\alpha \to^{\mathcal{T}} \beta$) if there is a $\mathcal{T}$-assembly sequence starting with α whose result is β. We say α is *$\mathcal{T}$-producible* if $\sigma \to^{\mathcal{T}} \alpha$ and write $\mathcal{A}[\mathcal{T}]$ to denote the set of $\mathcal{T}$-producible assemblies. We say α is *$\mathcal{T}$-terminal* if α is τ-stable and there exists no assembly which is $\mathcal{T}$-producible from α. We denote the set of $\mathcal{T}$-producible and $\mathcal{T}$-terminal assemblies by $\mathcal{A}_\square[\mathcal{T}]$.

When $\mathcal{T}$ is clear from context, we may omit $\mathcal{T}$ from the notation above.

Cooperative Attachment

Given a TAS $\mathcal{T} = (T, \sigma, \tau)$, for a tile to attach to an assembly it must match glues whose cumulative strength is at least τ in order to result in a τ-stable assembly. This can happen if, for instance, one of the matched glues has strength at least τ, in which case any other matching glues are superfluous. Alternatively, a tile may still attach without any τ-strength glues though this requires multiple glues to match whose strengths sum to at least τ. We refer to such attachments as *cooperative*.

2.2 Model Variations

In this paper we consider 3 variations of the aTAM. Other than the 3D aTAM, these include *directed* and *diffusion restricted* versions of the models. We say that a TAS $\mathcal{T}$ is *directed* if $|\mathcal{A}_\square[\mathcal{T}]| = 1$, i.e. $\mathcal{T}$ admits only a single producible terminal assembly. When we refer to a *directed model* we simply mean the set of all directed systems in a model. Directed systems are desirable for self-assembly since we often want our tiles to grow into a single target shape.

For diffusion restricted models, we note that in the aTAM it's possible for tiles to attach within a region of space which has been completely surrounded by other tiles. In 2D, we can imagine that the tiles are able to navigate around the assembly through the 3rd dimension, but in 3D such attachments are difficult to justify. Consequently, we also consider models where such attachments are forbidden. In 2D, this restriction could model a self-assembly process on the surface of a droplet of water where surface tension prevents the components from taking advantage of the 3rd dimension. We call the 2D diffusion restricted aTAM the *Planar aTAM* or PaTAM, and we call the 3D diffusion restricted aTAM the *Spatial aTAM* or SaTAM. In these models, and their directed subsets, we refer to regions which have been completely surrounded (in which no tile attachments are allowed to occur) *constrained*. To formally model this restriction, we first note that given a finite d-dimensional assembly α, the graph $\mathbb{Z}^d \setminus S_\alpha$ consists of a finite number of connected components, exactly one of which will be infinite in size. We say that this component graph is the *outside* of α while the finite-sized components are *constrained*. In a diffusion restricted system we only allow tile attachments on the outside of an assembly.

2.3 Intrinsic Simulation

First we provide a high-level definition of the notion of *intrinsic simulation* which should be sufficient for understanding our results. A full technical definition follows afterward. For brevity, in this paper, unless explicitly stated, "simulation" will refer to intrinsic simulation.

High-Level Description of Simulation

Simulation of system $\mathcal{T}$ by system $\mathcal{S}$ occurs at a scale factor m, so that $m \times m$ (or $m \times m \times m$ in 3D) blocks of tiles from $\mathcal{S}$, which we refer to as *macrotiles*, correspond to individual tiles in $\mathcal{T}$. For a given simulation, we define a *macrotile representation function* R which describes this mapping of macrotiles to tiles. Additionally for convenience, using R we define an *assembly representation function* R^* which maps entire assemblies from $\mathcal{S}$ to assemblies in $\mathcal{T}$, essentially evaluating R on each macrotile location for a given assembly in $\mathcal{S}$. Note that we don't require all locations within a macrotile to contain a tile and macrotile blocks containing tiles can still be mapped to empty space under R. When a tile attachment causes the representation of a macrotile location to map to a tile for the first time, we say that the attachment has caused the macrotile to *resolve* and once a macrotile has resolved, any additional tile attachments within the macrotile cannot change its representation under R. While we do allow macrotile locations to map to empty space, for a simulation to be valid there must be restrictions on where tiles are allowed to attach in $\mathcal{S}$. For our notion of simulation to be useful as a metric of comparing the relative capabilities of models, we require that $\mathcal{S}$ only place tiles within the macrotile regions immediately adjacent (not diagonally) to those which have already resolved, and we call such locations *fuzz*. This allows tiles in $\mathcal{S}$ to attach only in macrotiles which could potentially resolve during a valid simulation, since only the locations in $\mathcal{T}$ mapped to by the fuzz locations could possibly receive tiles in $\mathcal{T}$. If a class of systems C can all be simulated by another class of systems C' sharing a single tile set (though each may have a different seed assembly), we say that class C' *intrinsically simulates* C with a *universal tile set*. We can also say that C' is *intrinsically universal* (IU) for C.

Formal Definition of Simulation

Now we provide formal definitions for *intrinsic simulation*. The definitions here are taken from [7] and specifically refer to 3D systems. Similar definitions for 2D intrinsic simulation are given in [9]. For simulation of a 2D system by a 3D system, we use the 3D definitions and assume that all systems in the 2D system are defined in 3D so that assemblies occupy only the $z = 0$ plane.

From this point on, let T be a tile set and let the scale factor be $m \in \mathbb{Z}^+$. An *m-block macrotile* over T is a partial function $\alpha : \mathbb{Z}^3_m \dashrightarrow T$, where $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$. Let B^T_m be the set of all m-block macrotiles over T. The m-block with no domain is said to be *empty*. For a general assembly $\alpha : \mathbb{Z}^3 \dashrightarrow T$ and $(x', y', z') \in \mathbb{Z}^3$, define $\alpha^m_{(x',y',z')}$ to be the m-block macrotile defined by $\alpha^m_{(x',y',z')}(i_x, i_y, i_z) = \alpha(mx' + i_x, my' + i_y, mz' + i_z)$ for $0 \leq i_x, i_y, i_z < m$. For some tile set S, a partial function $R : B^S_m \dashrightarrow T$ is said to be a *valid m-block macrotile representation* from S to T if for any $\alpha, \beta \in B^S_m$ such that $\alpha \sqsubseteq \beta$ and $\alpha \in \text{dom } R$, then $R(\alpha) = R(\beta)$.

For a given valid m-block macrotile representation function R from tile set S to tile set T, define the *assembly representation function*[2] $R^* : \mathcal{A}^S \to \mathcal{A}^T$ such that $R^*(\alpha') = \alpha$ if and only if $\alpha(x, y, z) = R\left(\alpha'^m_{(x,y,z)}\right)$ for all $(x, y, z) \in \mathbb{Z}^3$. For an assembly $\alpha' \in \mathcal{A}^S$ such that $R^*(\alpha') = \alpha$, α' is said to map *cleanly* to $\alpha \in \mathcal{A}^T$ under R^* if for all non empty blocks $\alpha'^m_{(x,y,z)}$, $(x, y, z) + (u_x, u_y, u_z) \in \text{dom } (\alpha)$ for some $(u_x, u_y, u_z) \in U_3$ such that $u_x^2 + u_y^2 + u_z^2 \leq 1$, or if α' has at most one non-empty m-block $\alpha^m_{0,0}$. In other words, α' may have tiles on macrotile blocks representing empty space in α, but only if that position is adjacent to a tile in α. We call such growth "around the edges" of α' *fuzz* and thus restrict it to be adjacent to only valid macrotiles, but not diagonally adjacent (i.e. we do not permit *diagonal fuzz*).

In the following definitions, let $\mathcal{T} = (T, \sigma_T, \tau_T)$ be a TAS, let $\mathcal{S} = (S, \sigma_S, \tau_S)$ be a TAS, and let R be an m-block representation function $R : B^S_m \to T$.

▶ **Definition 1.** *We say that $\mathcal{S}$ and $\mathcal{T}$ have* equivalent productions *(under R), and we write $\mathcal{S} \Leftrightarrow \mathcal{T}$ if the following conditions hold:*

1. $\{R^*(\alpha') | \alpha' \in \mathcal{A}[\mathcal{S}]\} = \mathcal{A}[\mathcal{T}]$.
2. $\{R^*(\alpha') | \alpha' \in \mathcal{A}_\square[\mathcal{S}]\} = \mathcal{A}_\square[\mathcal{T}]$.
3. *For all $\alpha' \in \mathcal{A}[\mathcal{S}]$, α' maps cleanly to $R^*(\alpha')$.*

▶ **Definition 2.** *We say that $\mathcal{T}$* follows *$\mathcal{S}$ (under R), and we write $\mathcal{T} \dashv_R \mathcal{S}$ if $\alpha' \to^{\mathcal{S}} \beta'$, for some $\alpha', \beta' \in \mathcal{A}[\mathcal{S}]$, implies that $R^*(\alpha') \to^{\mathcal{T}} R^*(\beta')$.*

The next definition essentially specifies that every time $\mathcal{S}$ simulates an assembly $\alpha \in \mathcal{A}[\mathcal{T}]$, there must be at least one valid growth path in $\mathcal{S}$ for each of the possible next steps that $\mathcal{T}$ could make from α which results in an assembly in $\mathcal{S}$ that maps to that next step. While this definition is unfortunately dense, it accommodates subtle situations such as where $\mathcal{S}$ must "commit to" a subset of possible representations in $\mathcal{T}$ before being explicitly mapped, under R, to any one in particular.

▶ **Definition 3.** *We say that $\mathcal{S}$* models *$\mathcal{T}$ (under R), and we write $\mathcal{S} \models_R \mathcal{T}$, if for every $\alpha \in \mathcal{A}[\mathcal{T}]$, there exists $\Pi \subset \mathcal{A}[\mathcal{S}]$ where $\Pi \neq \varnothing$ and $R^*(\alpha') = \alpha$ for all $\alpha' \in \Pi$, such that, for every $\beta \in \mathcal{A}[\mathcal{T}]$ where $\alpha \to^{\mathcal{T}} \beta$, (1) for every $\alpha' \in \Pi$ there exists $\beta' \in \mathcal{A}[\mathcal{S}]$ where $R^*(\beta') = \beta$ and $\alpha' \to^{\mathcal{S}} \beta'$, and (2) for every $\alpha'' \in \mathcal{A}[\mathcal{S}]$ where $\alpha'' \to^{\mathcal{S}} \beta'$, $\beta' \in \mathcal{A}[\mathcal{S}]$, $R^*(\alpha'') = \alpha$, and $R^*(\beta') = \beta$, there exists $\alpha' \in \Pi$ such that $\alpha' \to^{\mathcal{S}} \alpha''$.*

[2] Note that R^* is a total function since every assembly of S represents *some* assembly of T; the functions R and α are partial to allow undefined points to represent empty space.

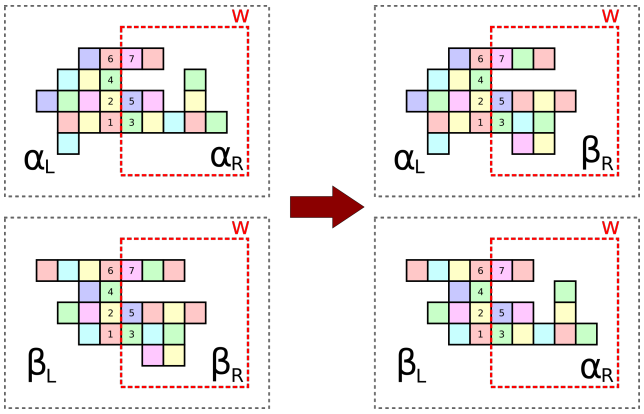

Figure 2 An illustration of the window movie lemma. On the left are two producible assemblies $\alpha = \alpha_L \cup \alpha_R$ and $\beta = \beta_L \cup \beta_R$ made from the same tile set, which are each divided into two subassemblies by the window w. For both assemblies, the window w has the same window movie, i.e. the order in which tiles present glues along the window, depicted by numbers on the tiles describing the relative order in which they attached. Since all growth within the windowed regions depends only on the glues presented along the window, we can splice these assemblies to get $\alpha_L \cup \beta_R$ or $\beta_L \cup \alpha_R$ (illustrated on the right). The window movie lemma then guarantees that both of these assemblies are producible.

▶ **Definition 4.** *We say that* $\mathcal{S}$ intrinsically simulates $\mathcal{T}$ *(under* R*) if* $\mathcal{S} \Leftrightarrow_R \mathcal{T}$ *(equivalent productions),* $\mathcal{T} \dashv_R \mathcal{S}$ *and* $\mathcal{S} \models_R \mathcal{T}$ *(equivalent dynamics).*

2.4 Window Movie Lemma

In [15], the authors proved the Window Movie Lemma, a pumping lemma of sorts for the aTAM (and its variants) which has since seen much use as a powerful tool for proving that certain tile-assembly simulations are impossible. Since it appears in several of our proofs, we first informally describe the lemma, then explicitly state it. A *window* is an edge cut which partitions the lattice graph ($\mathbb{Z}^2$ in 2D or $\mathbb{Z}^3$ in 3D) into two regions. Given some window w and some assembly sequence $\vec{\alpha}$ in a TAS $\mathcal{T}$, a *window movie* M is defined to be the ordered sequence of glues presented along w by tiles in $\mathcal{T}$ during the assembly sequence $\vec{\alpha}$. Informally, if we think of the window w as a thin pane dividing two regions of tile locations and imagine stepping through the assembly sequence $\vec{\alpha}$ one tile attachment at a time, M is constructed by recording the glues which appear on the surface of the pane and their relative order. More formally, a *window movie* is the sequence $M_w^{\vec{\alpha}} = \{(v_i, g_i)\}$ of pairs of grid graph vertices v_i and glues g_i, given by order of appearance of the glues along window w during $\vec{\alpha}$. Furthermore, if k glues appear along w during the same assembly step in $\vec{\alpha}$, then these glues appear contiguously and are listed in lexicographical order of the unit vectors describing their orientation in $M_w^{\vec{\alpha}}$.

Informally, the Window Movie Lemma states that any tile attachments that occur within the region bounded by a window are possible in a region bounded by the same window (up to translation) with an identical window movie. This allows us to splice assembly sequences together and, consequently, pump a sequence of tile attachments so long as we can ensure the existence of identical window movies. Figure 3 illustrates how the Window Movie Lemma can be used to pump growth.

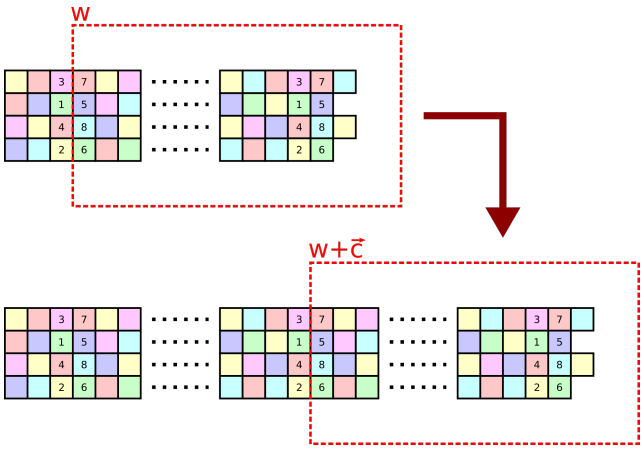

Figure 3 Using the Window Movie Lemma to "pump" assembly sequences. The top assembly depicts a ribbon of tiles growing horizontally to the right and numbers on tiles describe a relative order of attachment. If such a ribbon of tiles grows long enough, then by pigeonhole principle, eventually there must exist two identical vertical slices along its length. Because every tile attachment inside a window w depends only on the tiles and their relative order of attachment along the window, we can thus find an assembly sequence where growth repeats after the second identical vertical slice. This can be performed indefinitely to "pump" the ribbon.

Window Movie Lemma

Let $\vec{\alpha} = \{\alpha_i\}$ and $\vec{\beta} = \{\beta_i\}$ be assembly sequences in TAS $\mathcal{T}$ and let α, β be the result assemblies of each respectively. Let w be a window that partitions α into two configurations α_L and α_R and let $w' = w + \vec{c}$ be a translation of w that partitions β into two configurations β_L and β_R (with α_L and β_L being the configurations containing their respective seed tiles). Furthermore define $M_w^{\vec{\alpha}}$ and $M_w^{\vec{\beta}}$ to be the window movies for $\vec{\alpha}, w$ and $\vec{\beta}, w'$ respectively. Then if $M_w^{\vec{\alpha}} = M_w^{\vec{\beta}}$, the assemblies $\alpha_L \cup \beta'_R$ and $\beta'_L \cup \alpha_R$ (where $\beta'_L = \beta_L - \vec{c}$ and $\beta'_R = \beta_R - \vec{c}$) are also producible.

3 Results

In this section we sketch our results. Detailed proofs can be found in the full version on arXiv [8]. We begin with some trivial observations which allow us to fill in several boxes from Table 1.

▶ **Observation 5.** *If there exists a directed system $\mathcal{T}$ in tile-assembly model M which cannot be simulated by any system in tile-assembly model M', then (1) there exists a system in M which cannot be simulated by any system in M', (2) there exists a system in M which cannot be simulated by any directed system in M', and (3) there exists a directed system in M which cannot be simulated by any directed system in M'.*

▶ **Observation 6.** *There exists systems, both directed and undirected, in the 3D models (3DaTAM and SaTAM) which cannot be simulated by any systems in any of the 2D models (aTAM and PaTAM, both directed and undirected).*

Observation 5 holds because the set of directed systems in a model is a subset of all systems in that model. Consequently, $\mathcal{T}$ is a system in both M and in the directed subset of M. By assumption, $\mathcal{T}$ cannot be simulated by any system in M' and therefore cannot

be simulated by any subset of systems of M', particularly the subset of directed systems. Regarding Observation 6, while we restrict the notion of simulation to use square macrotiles, simulations of systems on triangular lattices have been implemented using roughly hexagonal macrotiles made from square tiles [1], so one might imagine the possibility that by loosening our definition of simulation to use more interesting macrotiles, it could be possible to capture the geometry of 3D square tiles using 2D tiles. In our case however, we note that there can exist no planar embedding of the lattice graph of $\mathbb{Z}^3$ as a consequence of Kuratowski's theorem. Consequently, there can be no way to divide $\mathbb{Z}^2$ into connected regions of macrotile locations which preserves the adjacency of points in $\mathbb{Z}^3$ and therefore simulation could not be possible even if we generalized our notion of macrotiles. This is true for any 3D systems which have producible assemblies whose domains, as graphs, are non-planar as is trivially possible in all 3D models considered.

3.1 Simulations using existing tile sets

In [7], it was shown that there exists IU tile sets for the 3DaTAM, SaTAM, and both models' subsets of directed systems. While the main focus of that result was intrinsic simulation within a model, those IU tile sets can be used to trivially fill in a few boxes of Table 1. First we note that any aTAM system can also be thought of as a 3DaTAM system (or even SaTAM system since tiles occupying only a single plane of 3D space can't constrain a 3D region) with glues only appearing on 4 of the 6 faces of any tile. Second, we note that the IU tile sets for the 3DaTAM and SaTAM differed only by the addition of a few tile types responsible for growing a wall around each face of a macrotile before resolving. This was necessary for intrinsic universality in the SaTAM since without them, the tiles making up a macrotile were sparse enough to necessarily allow a diffusion path for tiles to pass through a resolved macrotile. Consequently, if we don't include those tile types, then the IU tile set can simulate 3DaTAM systems even in the SaTAM since without walls surrounding each macrotile, the diffusion restriction does not interfere with the attachment of any tiles. Finally, by design, this tile set preserves directedness when simulating a directed system. Therefore, using the IU tile set and proofs from [7], the following observations hold.

▶ **Observation 7.** *There exists a universal tile set in both the 3DaTAM and SaTAM which intrinsically simulates all systems in the aTAM, preserving directedness.*

▶ **Observation 8.** *There exists a universal tile set in the SaTAM which intrinsically simulates all systems in the 3DaTAM, preserving directedness.*

3.2 Directed systems cannot simulate undirected systems

▶ **Theorem 9.** *There exist systems in the aTAM, 3DaTAM, PaTAM, and SaTAM, which cannot be simulated by any directed system in any of these models.*

Whereas directed systems only have one terminal assembly, undirected systems can have several. Figure 4 illustrates the tile set and terminal assemblies of a simple undirected system $\mathcal{T}$ which can be a system in the aTAM, 3DaTAM, PaTAM, or SaTAM without modification as it does not use any dynamics unique to any of those models. Because directed systems can only have a single terminal assembly, any directed system attempting to simulate $\mathcal{T}$ would necessarily fail since any assembly representation function R^* could not map one terminal assembly to both terminal assemblies of $\mathcal{T}$.

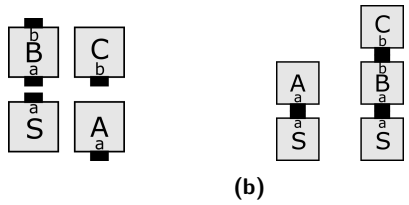

Figure 4 (a) Tile set of an undirected system for the proof of Theorem 9 and (b) Its two terminal assemblies.

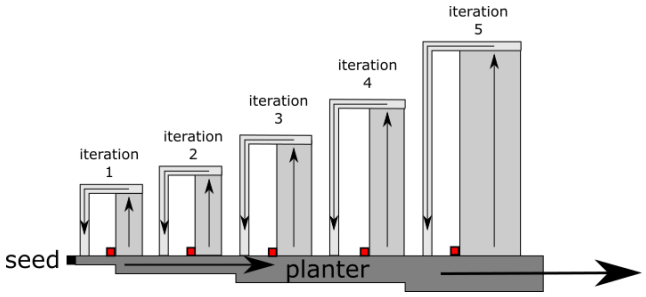

Figure 5 System $\mathcal{T}$ of the proof of Theorem 10. An infinite planter grows to the east from the seed and initiates upward growth of an infinite series of counters, each taller than the last, which initiate single-tile-wide paths that grow to the left then crash downward into the planter. To the left of each counter, at its base, it is possible for a red tile to attach.

3.3 The PaTAM cannot simulate the aTAM

Here we show that there are aTAM systems which cannot be correctly simulated by any PaTAM systems. To show this, we take advantage of the fact that aTAM systems are capable of growth inside of constrained regions while PaTAM systems are not. Specifically, we show that the PaTAM can't simulate the directed aTAM and, by Observation 5, note that this also implies that the PaTAM can't simulate the aTAM.

▶ **Theorem 10.** *There exists a system $\mathcal{T}$ which is a directed aTAM system, and therefore also an aTAM system, which cannot be simulated by any PaTAM system.*

Figure 5 is a schematic diagram of the terminal assembly of $\mathcal{T}$, a directed aTAM system which we claim is impossible to simulate in the PaTAM. Note that $\mathcal{T}$ is more complex than a system in which tiles attach to constrain a region which could have another tile attach within. This is because the definition of intrinsic simulation allows for macrotiles to resolve even when they aren't completely filled with tiles. Consequently, while macrotiles may map to tiles constraining a region, the tiles making up the macrotiles may not constrain a region. Our construction is designed to ensure that at some point, any supposed simulating system must constrain a region before the tiles inside are able to attach. In our directed aTAM system $\mathcal{T}$, this is done by first initiating the growth of a *planter*, a gadget that counts up in binary as it grows eastward, initiating the growth of increasingly tall arms at defined intervals. These arms are essentially binary counter gadgets which each grow upward to a distance, encoded in the glues of the tiles provided by the planter, and initiate the growth of thin arms when they finish. The thin arms are just a single tile wide and begin by growing a fixed distance to the west before growing south to crash into the planter below. By this process, each arm initiated by the planter constrains increasingly large regions of space which each contain a single location between the planter and arms, in which a single tile

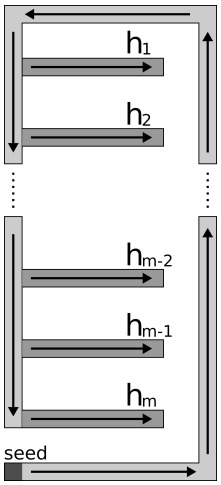

Figure 6 A schematic of system $\mathcal{P}$ for the proof of Theorem 11. Tiles grow in a rectangular shape, periodically spawning arms which can crash into the walls and constrain a region. It is undirected and its size depends non-deterministically on the number of tiles that attach between each corner.

can cooperatively attach (denoted by the red squares in Figure 5). Each of the tiles making up the southward growing portion of the thin arms are of the same tile type, each with identical glues on their north and south faces. While it is possible for different macrotiles to map to the same tile in $\mathcal{T}$, there are only so many combinations of tiles that make up a macrotile. Consequently, regardless of scale factor, if we look far enough down the planter, there will be an arm which grows tall enough that the simulating set must repeat a macrotile representation in two places along the same thin arm. We can then use the Window Movie Lemma to show that this arm "pumps" in our supposed simulating system, before crashing into the planter. It is therefore impossible for any simulating PaTAM system to prevent a region from becoming constrained before the macrotile inside is able to resolve, yielding terminal assemblies which aren't correctly mapped to a terminal assembly in $\mathcal{T}$.

3.4 The aTAM cannot simulate the PaTAM

Given that the PaTAM is just the aTAM with an added restriction on tile attachment, it's not terribly surprising that the PaTAM can't simulate the full dynamics of the aTAM; however, less obvious is the fact that the planarity restriction *also* gives the PaTAM some capabilities not possible in the aTAM, namely the ability to constrain a region and stop growth within. We utilize this ability in our proof of Theorem 11 which is sketched here. Also, by Observation 5, this also holds for the directed aTAM.

▶ **Theorem 11.** *There exists a PaTAM system $\mathcal{P}$ which cannot be simulated by any aTAM system.*

As with the proof for Theorem 10, in the definition of intrinsic simulation, we consider all possible representation functions and scale factors to prove impossibility. Figure 6 is a schematic diagram of PaTAM system $\mathcal{P}$ which is impossible to correctly simulate in the aTAM. Growth of $\mathcal{P}$ begins with tiles attaching in a row growing east. The length of this row is non-deterministic as at any point along the row, it's possible for a corner tile to attach, initiating growth to the north. Consequently, $\mathcal{P}$ is an undirected system so any potential simulating system must be able to simulate all possible assemblies of $\mathcal{P}$. Similarly,

northward and eastward growing rows of tiles attach with some length depending on how many tiles attached before each corner. Finally, a column of tiles begins growing south and, as it does, initiates the growth of several arms eastward, each spaced 4 tiles apart. Both the southward growing column of tiles and the arms continue growth until they are constrained or crash into another part of the assembly. To show that $\mathcal{P}$ cannot be simulated in the aTAM, we assume the existence of a simulating aTAM system $\mathcal{T}$ and prove that it must admit some assembly sequences which don't correspond to those in $\mathcal{P}$. To do this, we consider an assembly sequence in $\mathcal{P}$ where the rectangle of tiles grows to a size, based on the scale factor of the simulation, so that a sufficiently large number of sufficiently long arms are spawned by the south growing column of tiles. We also choose an assembly sequence where the south growing column will eventually collide with the seed tile, constraining the region containing the arms. Because we've chosen the assembly to be sufficiently large, each arm is capable of being "pumped" as per the window movie lemma. We then grow the bottom arm until just after it has collided with the east wall and note that, while $\mathcal{T}$ is an aTAM system and can still grow tiles inside of the constrained region, tiles on the inside and outside will no longer be able to affect each other's growth. There are a few cases to be considered, depending on whether or not the representation function has resolved the last tile of the bottom arm, but essentially we then show that we can continue the growth of the west wall until its macrotiles have resolved to tiles in $\mathcal{P}$ that constrain the rectangle's interior. By a counting argument and our choice of the number of arms, we can then show that one of the other arms must be able to continue growth within the constrained region, and that the assembly sequence in $\mathcal{T}$ maps to one invalid in $\mathcal{P}$.

3.5 The 3DaTAM cannot simulate the SaTAM

The proof of Theorem 12 is similar in principle to the proof of Theorem 11, albeit with a slightly different system which takes advantage of the differences between 2D and 3D. We sketch the proof here.

▶ **Theorem 12.** *There exists an SaTAM system $\mathcal{S}$ which cannot be simulated by any 3DaTAM system.*

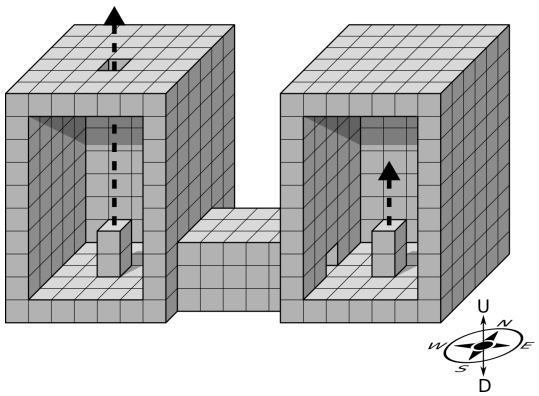

Figure 7 Cut-away view of system $\mathcal{S}$ from the proof of Theorem 12. Two chambers are connected by a thin tunnel. Pillars growing inside the outer west chamber will eventually constrain the region within the chambers, at which point, the pillar growing in the inner east chamber will no longer be able to continue growth.

The system $\mathcal{S}$ for this result, as illustrated in Figure 7, initially grows 2 nearly sealed chambers connected by a thin tunnel which allows for a diffusion path between them. These chambers both have a fixed base size of 9×9, but they can grow to have an arbitrary height in a way similar to the frame of the system used in the proof of Theorem 11. Once fully grown, the ceiling of one chamber contains a single tile wide opening which is the only way for tiles to diffuse into the chambers from outside; we call the chamber with this hole the *outer chamber* and the other one the *inner chamber*. Additionally, from the bottoms of both chambers, pillars can grow upwards to an arbitrary height by the attachment of copies of tiles with identical tile types. The pillar in the inner chamber will eventually crash into the ceiling *or* until the pillar in the outer chamber grows tall enough to plug the opening in its ceiling and constrain the space inside. We show that $\mathcal{S}$ cannot be simulated by any 3DaTAM system by showing that, in any potential simulating system, under the right conditions, although unwanted, it must still possible for the inner chamber pillar to continue growth even after the outer chamber pillar has sealed the chambers. To do this, we note that during some supposed simulation, the only way for the pillar in the inner chamber to "know" that the chambers have been sealed, is for tiles to attach inside of the tunnel. Consequently, because the tunnel is thin with a cross-section made of a hollow 3×3 square, the chambers can only communicate with each other a finite amount of times during a simulation. Specifically, if the scale factor of the simulation is c, then the number of tiles that can be placed in any x-coordinate corresponding to the tunnel is bounded by $5c \times 5c$ which includes any potential tiles growing in the fuzz adjacent to the macrotiles of the tunnel. Therefore, by a simple counting argument, if we initially grew our chambers to have a sufficiently large height, then there must exist some assembly sequence where both pillars grow by any desired number of macrotiles (which we choose to be long enough to allow pumpable growth) and during which no tile is placed in the center of the tunnel. Using the Window Movie Lemma, we then construct an assembly sequence where the outer chamber pumps to constrain the chambers. Because during this assembly sequence, no tiles are placed in the center of the tunnel, there is nothing to stop the inner chamber pillar from also being pumped. Such an assembly sequence must be possible in any 3DaTAM system which supposedly simulates our system $\mathcal{S}$, and since this assembly sequence corresponds to one which is invalid in the SaTAM, such a simulation is impossible.

3.6 The PaTAM can simulate the directed PaTAM

▶ **Theorem 13.** *There exists a universal Planar aTAM tile set S that can simulate any directed PaTAM system.*

Despite the fact that both the PaTAM and directed PaTAM are not intrinsically universal for themselves[7], using tools from [7] and [2] we are able to construct a PaTAM tile set capable of simulating arbitrary directed PaTAM systems. Here we outline the process by which a PaTAM tileset S can simulate any given directed PaTAM system $\mathcal{T}$. The tileset S is universal, meaning that regardless of the directed PaTAM system $\mathcal{T}$, the same tileset will be used at a fixed binding threshold, with only the seed of the simulating system changing to accommodate $\mathcal{T}$.

Given a directed PaTAM system $\mathcal{T}$, we define a simulating system $\mathcal{S}$ using a fixed tile set at binding threshold 2. The seed of $\mathcal{S}$ consists of already-resolved macrotiles in the same configuration as the seed of $\mathcal{T}$. Each macrotile in $\mathcal{S}$ consists of a 9×9 grid of structures we call *component blocks* (CBs) which are each made of many smaller tile-based constructions and which each store an encoding of the system $\mathcal{T}$ along with a bit of extra data in the form

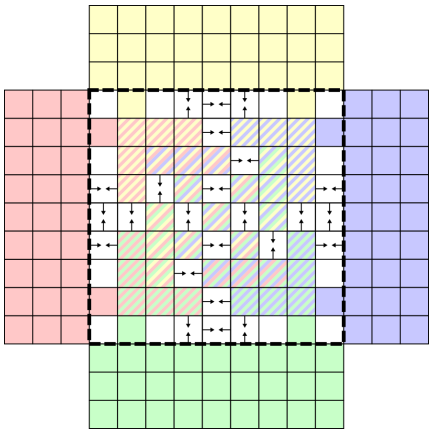

Figure 8 A schematic describing the 9×9 grid of potential component blocks which may appear in a macrotile location. Squares containing two arrows indicate a grid location which may contain a probe region. The surrounding macrotiles are illustrated using colored tiles to represent their relative direction from the current macrotile. Colors of CB locations indicate which surrounding macrotile the CB may have information about.

of specific glues on some of its tiles. The CBs of a macrotile each perform calculations using tiles which emulate Turing machines to determine how they should grow and whether or not the macrotile can resolve given the current information regarding the surrounding macrotiles.

Each CB essentially behaves like an individual tile on the 9×9 grid and we can think of CBs as growing in one of two ways. Either the CB grows using tile attachments from another adjacent CB in a way analogous to a τ-strength tile attachment, or a CB can grow in the gap between two adjacent CBs in certain locations of the grid designated as *probe regions*. This is analogous to a tile attachment that occurs by cooperative binding between two opposing tiles (which we refer to as *across-the-gap* cooperation). These "cooperative attachments" between CBs are used to consolidate information between the CBs. For instance, one CB might contain information encoded about the north adjacent macrotile and one might contain information about the west; in the probe region between them, a new CB can grow which will contain the information about both which it can then use to determine if a tile attachment in $\mathcal{T}$ would be possible in the tile location corresponding to the macrotile. Figure 8 illustrates the layout of a macrotile into CB locations with these probe regions indicated by squares with two opposing arrows.

Probe regions are CB locations in which two adjacent CBs, on opposite sides, can present structures called *probes* which are long, thin structures that grow from the surrounding CBs towards the center of a CB location. Each probe that grows in a probe region, indicates some possible combination of information from surrounding macrotiles and grows in a unique position according to this information. The length of a probe is chosen to be just shy of the center of the CB location, so that when two probes align from opposing sides of the probe region, there will be exactly a single tile wide gap between them. This gap allows a tile to cooperatively attach and grow along the sides of the probes to recover the information from both. Otherwise, if no probes in a probe region align, there will be enough room for the components that make up a CB to squeeze in between the probes from one side of the probe region to another. Figure 9 illustrates two scenarios involving probe regions.

Probe regions were introduced in [2] to solve the problem illustrated in Figure 10. Naively when simulating a tile system, to check for macrotiles which may cooperate across-the-gap, tiles must grow to query both adjacent macrotiles and determine if the attachment is possible.

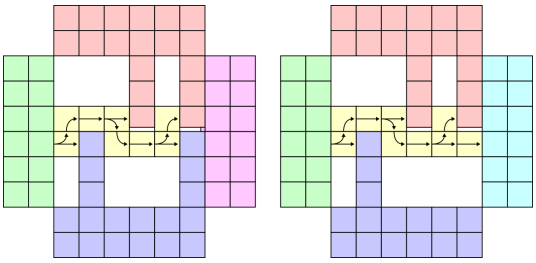

Figure 9 Probe regions between component blocks. The red and blue CBs grow probes to the center of the CB location in the middle while the green CB attempts to grow through the probe region. On the left, two probes happen to align, in which case a path of tiles containing information from the green CB cannot pass and the CB to the east results from the cooperative tile attachment between the probes. On the right, no probes align meaning the path of tiles from the green CB can squeeze between the probes to influence the growth of the CB to the east.

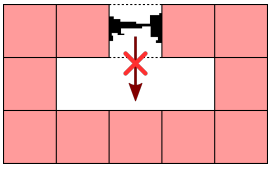

Figure 10 When checking for across-the-gap cooperation during a simulation, tiles can't naively span the entire gap without disconnecting two regions of space.

This however necessarily separates regions of space and in the case of planar systems also constrains one before it has been determined if the attachment can even occur. If it cannot, then tiles will no longer be able to attach in the constrained region and the simulation will likely end up being invalid. Probes avoid this problem by aligning exactly when across-the-gap cooperation is possible while still allowing tile structures to grow through if they don't align.

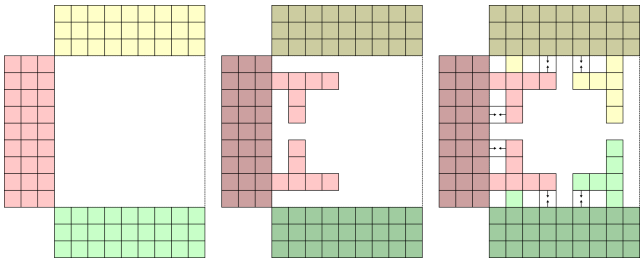

Figure 11 Hands made of component blocks growing from surrounding macrotiles.

Now that we have an idea of how the component blocks and probe regions behave we describe the protocol for resolving a macrotile by highlighting a few important cases. Growth within a macrotile begins when one or more of the surrounding macrotiles resolve and tiles begin to attach within the macrotile. From a surrounding macrotile, the protocol always begins by the growth of two "T"-shaped structures made from CBs called *hands* illustrated in Figure 11. Note that two adjacent surrounding macrotiles may both attempt to grow hands in the same location. This is handled by a single point of competition and the first surrounding macrotile for placing a tile in the closest corner of the shared hand locations is allowed to place theirs. Between the hands and the surrounding macrotiles probes are grown in the regions indicated on the right of Figure 11 which allows a CB to "attach"

cooperatively to combine information from both the hand and nearby macrotile. In some cases this information may be redundant, but with two or more surrounding macrotiles at least one location will always be able to combine information from two macrotiles.

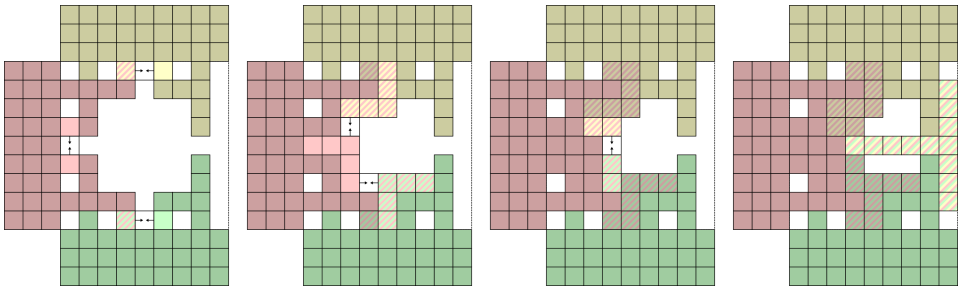

Figure 12 Once the hands have grown, CBs cooperate until information from all sides has been combined into a single CB. Then the macrotile can resolve.

The CBs resulting from cooperation between the hands and surrounding macrotiles then cooperate once again and CBs grow along the hands to form clockwise elbows with additional probe regions between them. CBs then cooperatively attach between these elbows and cooperate again near the center of the tile to eventually combine all of the information from the surrounding macrotiles. Once this occurs, the CB which "attaches" in the center of the tile contains the information from all sides. If the surrounding macrotiles represent tiles in $\mathcal{T}$ capable of placing a tile, additional CBs can grow to the remaining sides to present this information to the remaining sides and repeat the procedure in the adjacent macrotile locations.

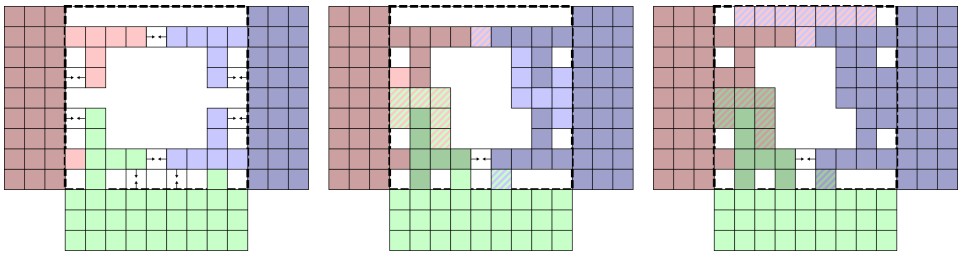

Figure 13 Probe regions between opposing macrotiles can check for across-the-gap cooperation.

In the case that an across-the-gap cooperation is possible in $\mathcal{T}$, the protocol deviates slightly. Illustrated in Figure 13, if across-the-gap cooperation is possible between the east and west macrotiles, their hands will share a probe region with aligned probes. Consequently, a CB can grow in that location and resolve the macrotile. This growth may constrain the region to the south, halting any tile attachments and CB growth in the south side of the macrotile, but this doesn't matter since the macrotile will only need to start the process in adjacent macrotiles that haven't yet resolved. The described protocol is robust to different orders of hand growth and different numbers of surrounding macrotiles, including those that don't end up contributing to macrotile resolution. If at any time a CB has sufficient information to determine how the macrotile should resolve, it begins growth to the center and then surrounding edges of the macrotile. This process will not be interrupted by other CBs since we are simulating a directed system where at most one unique tile can attach in each location.

References

1 John Calvin Alumbaugh, Joshua J Daymude, Erik D Demaine, Matthew J Patitz, and Andréa W Richa. Simulation of programmable matter systems using active tile-based self-assembly. In *International Conference on DNA Computing and Molecular Programming*, pages 140–158. Springer, 2019.

2 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 *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2012, pages 302–310, 2012.

3 Shawn M. Douglas, Hendrik Dietz, Tim Liedl, Björn Högberg, Franziska Graf, and William M. Shih. Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature*, 459:414–418, May 2009. doi:10.1038/nature08016.

4 Constantine Glen Evans. *Crystals that count! Physical principles and experimental investigations of DNA tile self-assembly.* PhD thesis, California Institute of Technology, 2014.

5 Bin Fu, Matthew J. Patitz, Robert T. Schweller, and Robert Sheline. Self-assembly with geometric tiles. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming – 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *LNCS*, pages 714–725. Springer, 2012.

6 Hongzhou Gu, Jie Chao, Shou-Jun Xiao, and Nadrian C. Seeman. A proximity-based programmable dna nanoscale assembly line. *Nature*, 465(7295):202–205, May 2010. doi:10.1038/nature09026.

7 Daniel Hader, Aaron Koch, Matthew J. Patitz, and Michael Sharp. The impacts of dimensionality, diffusion, and directedness on intrinsic universality in the abstract tile assembly model. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2607–2624. SIAM, 2020.

8 Daniel Hader and Matthew J. Patitz. The impacts of dimensionality, diffusion, and directedness on intrinsic cross-model simulation in tile-based self-assembly, 2023. arXiv:2305.01877.

9 Jacob Hendricks, Matthew J. Patitz, and Trent A. Rogers. Universal simulation of directed systems in the abstract tile assembly model requires undirectedness. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016), New Brunswick, New Jersey, USA* October 9-11, 2016, pages 800–809, 2016.

10 Ming-Yang Kao and Robert T. Schweller. Randomized self-assembly for approximate shapes. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (1)*, volume 5125 of *Lecture Notes in Computer Science*, pages 370–384. Springer, 2008. doi:10.1007/978-3-540-70575-8_31.

11 Yonggang Ke, Luvena L Ong, William M Shih, and Peng Yin. Three-dimensional structures self-assembled from DNA bricks. *Science*, 338(6111):1177–1183, 2012.

12 Wenyan Liu, Hong Zhong, Risheng Wang, and Nadrian C. Seeman. Crystalline two-dimensional dna-origami arrays. *Angewandte Chemie International Edition*, 50(1):264–267, 2011. doi:10.1002/anie.201005911.

13 Kyle Lund, Anthony J. Manzo, Nadine Dabby, Nicole Michelotti, Alexander Johnson-Buck, Jeanette Nangreave, Steven Taylor, Renjun Pei, Milan N. Stojanovic, Nils G. Walter, Erik Winfree, and Hao Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465(7295):206–210, May 2010. doi:10.1038/nature09012.

14 Kyle Lund, Anthony T. Manzo, Nadine Dabby, Nicole Micholotti, Alexander Johnson-Buck, Jeanetter Nangreave, Steven Taylor, Renjun Pei, Milan N. Stojanovic, Nils G. Walter, Erik Winfree, and Hao Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465:206–210, 2010.

15 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 *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA 2014), (Portland, OR, USA, January 5-7, 2014)*, pages 752–771, 2014.

16 Jennifer E. Padilla, Matthew J. Patitz, Raul Pena, Robert T. Schweller, Nadrian C. Seeman, Robert Sheline, Scott M. Summers, and Xingsi Zhong. Asynchronous signal passing for tile self-assembly: Fuel efficient computation and efficient assembly of shapes. In *UCNC*, volume 7956 of *Lecture Notes in Computer Science*, pages 174–185. Springer, 2013. doi:10.1007/978-3-642-39074-6_17.

17 Jennifer E. Padilla, Ruojie Sha, Martin Kristiansen, Junghuei Chen, Natasha Jonoska, and Nadrian C. Seeman. A signal-passing DNA-strand-exchange mechanism for active self-assembly of DNA nanostructures. *Angewandte Chemie International Edition*, 54(20):5939–5942, March 2015.

18 Paul W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, March 2006. doi:10.1038/nature04586.

19 Paul W. K. Rothemund and Erik Winfree. The program-size complexity of self-assembled squares (extended abstract). In *STOC '00: Proceedings of the thirty-second annual ACM Symposium on Theory of Computing*, pages 459–468, Portland, Oregon, United States, 2000. ACM.

20 Paul WK Rothemund, Nick Papadakis, and Erik Winfree. Algorithmic self-assembly of dna sierpinski triangles. *PLoS biology*, 2(12):e424, 2004.

21 David Soloveichik and Erik Winfree. Complexity of self-assembled shapes. *SIAM Journal on Computing*, 36(6):1544–1569, 2007. doi:10.1137/S0097539704446712.

22 Erik Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, June 1998.

23 Damien Woods. Intrinsic universality and the computational power of self-assembly. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 373(2046), 2015. doi:10.1098/rsta.2014.0214.

24 Damien Woods, David Doty, Cameron Myhrvold, Joy Hui, Felix Zhou, Peng Yin, and Erik Winfree. Diverse and robust molecular algorithms using reprogrammable dna self-assembly. *Nature*, 567(7748):366–372, 2019.

25 Yin Zhang, Angus McMullen, Lea-Laetitia Pontani, Xiaojin He, Ruojie Sha, Nadrian C. Seeman, Jasna Brujic, and Paul M. Chaikin. Sequential self-assembly of dna functionalized droplets. *Nature Communications*, 8(1):21, 2017. doi:10.1038/s41467-017-00070-0.