

Powers and Limitations of Synchronous Self-Assembly

Non-cooperative Assemblies and Limited Synchronization

Florent Becker ✉ 

Laboratoire d'Informatique Fondamentale d'Orléans (UR4022),
Université d'Orléans and INSA Centre-Val-de-Loire, France

Phillip Drake ✉ 

University of Arkansas, Fayetteville, AR, USA

Matthew J. Patitz ✉ 

University of Arkansas, Fayetteville, AR, USA

Ryder Smith ✉ 

Hass Hall Academy, Bentonville, AR, USA

Abstract

In abstract models of algorithmic self-assembly, synchronization between attachments has emerged as a crucial distinction between the classical asynchronous model (aTAM) and a new synchronous model, the syncTAM. This paper presents recent advances in gauging the additional power afforded by the syncTAM. While it is known that the syncTAM and the aTAM are each unable to fully simulate the other, this paper offers evidence that the syncTAM is computationally significantly more powerful than the aTAM, especially in the non-cooperative setting.

The additional power of the non-cooperative syncTAM is witnessed by the following constructions, all impossible in the non-cooperative aTAM: a flagpole, a strict self-assembly of a variant of the discrete Sierpinski triangle, and the ability to build the same assemblies (modulo scale factor) as directed aTAM systems.

The second topic is that of limited synchronization, wherein, when the number of attachments is smaller than some threshold l , they happen synchronously, but attachments in excess of that number must wait. In that context, the precise value of l is crucial, and changes to that value prevent simulation and can change which shapes can be obtained.

2012 ACM Subject Classification Theory of computation → Models of computation

Keywords and phrases self-assembly, noncooperative self-assembly, models of computation, tile assembly systems

Digital Object Identifier 10.4230/LIPIcs.DNA.32.2

Related Version *Full Version:* <https://arxiv.org/abs/2605.27604> [3]

Supplementary Material *Software (Source Code):* [https://self-assembly.net/wiki/index.php/Synchronous_Tile_Assembly_Model_\(syncTAM\)#Examples_and_Simulation_Software](https://self-assembly.net/wiki/index.php/Synchronous_Tile_Assembly_Model_(syncTAM)#Examples_and_Simulation_Software)

Funding *Florent Becker:* Research made at a public, state-funded university, thanks to the guarantee of full academic freedom and perennial research funding.

Phillip Drake: This author's work was supported in part by NSF grant 2329908

Matthew J. Patitz: This author's work was supported in part by NSF grant 2329908

1 Introduction and Outline

Crafting artifacts with intricate small-scale details has been a mark of sophistication in a wide variety of human cultures. Contemporary researchers have turned this endeavor into the field of nanotechnology, with the aim to extend the realm of controlled manipulation



© Florent Becker, Phillip Drake, Matthew J. Patitz, and Ryder Smith;
licensed under Creative Commons License CC-BY 4.0

32nd International Conference on DNA Computing and Molecular Programming (DNA 32).

Editors: Dominic Scalise and Robert Schweller; Article No. 2; pp. 2:1–2:23

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

down to the molecular and atomic scales. As it turns out, DNA is a very handy vehicle for that, especially as its ability to hold data can be harnessed for algorithmic control of its assembly. Thus was born the field of (DNA) Algorithmic Self-Assembly.

The principle is to come up with basic building blocks which are able to selectively bind and grow patterns of interest. Theoretical models serve to predict and engineer this binding and assembly. There is a variety of them and this paper will be dealing with *abstract* models of self-assembly, which tend to be easier to reason about while giving broad brush insights into the possibilities of molecular self-assembly.

Arguably, the first abstract model of algorithmic self-assembly was the abstract Tile Assembly Model (aTAM) [28]. Since its introduction, a wide variety of results have exhibited its powerful algorithmic potential [4, 9, 19, 25–27].

One striking feature which sets it apart from more classical models of computation is its asynchronous nature. In order to claim correctness for a system in the aTAM, one has to prove that it obeys one’s fancy whatever order the building blocks (known as tiles) come to interact with the growing assembly. On the other hand, there is a natural synchronous variant of the aTAM, called the syncTAM which works synchronously, and thus much closer to more classical models of computation such as cellular automata. The first investigations of the syncTAM in [2] have shown that the two models are uncomparable (each can do things the other cannot). In this paper, we set out to refine that distinction.

It would seem natural that a synchronous model affords its user more control, which in turn means they need fewer resources in order to implement algorithms. In particular, the *temperature* parameter (a.k.a. minimum binding threshold) is a very important source of control in both the aTAM and the syncTAM. Informally, the temperature of an aTAM system tells whether tiles can do a bit of local synchronization through cooperation when they assemble (at temperature 2 and more), or not (at temperature 1). It is well-known in the aTAM that systems with temperature 1 are less powerful than systems with temperature 2 and above. They can be distinguished by intrinsic universality, defined in [9], allowing [21] to state that there are temperature 2 systems which no temperature 1 system can simulate. Moreover, in two dimensions, temperature 1 systems are limited in their computational capacities, as they must be *fragile* or *pumpable* [22]. On the other hand, effectively implementing temperature 2 is a tough experimental challenge. For this reason, there have been many theoretical explorations of variations of the aTAM that exhibit at least some of the power of cooperative, temperature-2, self-assembly at temperature-1 [7, 10–12, 14, 18, 24]. We set out to show that the synchronicity of the syncTAM can act as a kind of substitute for temperature 2 by exploring the power of temperature 1 syncTAM systems. Namely, we show that these systems are able to simulate a flagpole, thus overcoming the simulation obstacle that was identified in [21] for the aTAM. The cooperative synchronization gadget used in this system can act as a basic brick for 2-dimensional computation, allowing the strict self-assembly of a shape close to the Sierpinski Triangle. Sierpinski Triangles have been known since [1] to belong to the class of fractals which cannot be strictly self-assembled in the aTAM [4]. This witnesses that in addition to cooperation and thus computation, the temperature 1 syncTAM is able to *detect the absence* of some part of the assembly. If synchronization was a complete substitute for cooperation, then temperature 1 systems in the syncTAM would be able to build the same assemblies (at least modulo some scale factor) as temperature 2 systems in the aTAM. We show that this is the case in the absence of non-determinism: when restricted to aTAM systems which are directed, the claim holds.

How much of that power remains if, instead of full synchronicity, one assumes that tiles come all at once, *up to some maximal amount l*? This is what we investigate next, introducing the L-syncTAM. Looking for intermediates between the syncTAM and the aTAM is a natural

way to try and make sense of more plausible models of algorithmic DNA self-assembly. The hope is that any property which is robust to changes along the continuum between the syncTAM and the aTAM will hold in experiments. The investigation of the L-syncTAM for various values of l reveals that altering that parameter either up or down affects the dynamics of the model, preventing universal simulation between different values of l , and even affecting which shapes can be assembled.

Section 2 introduces the reader to the aTAM, syncTAM and L-syncTAM models, as well as to intrinsic simulation for these models. Section 3 exposes the power of the syncTAM model by way of the flagpole and Sierpinski triangle constructions. Section 4 examines directed systems in the syncTAM, up to the “simulation” at temperature 1 of directed aTAM systems. Finally, Section 5 is the investigation of limited synchronicity through the L-syncTAM model.

2 Preliminary Definitions and Models

In this section we define the models and terminology used throughout the paper.

2.1 The abstract Tile-Assembly Model

We work within the abstract Tile-Assembly Model [28] in 2 and 3 dimensions. The abbreviation *aTAM* refers to the 2D model. These definitions are borrowed from [15] and we note that [26] and [20] are good introductions to the model for unfamiliar readers.

Let $\mathbb{N}$ be the set of nonnegative integers, for $l, u \in \mathbb{Z}$, let $\llbracket l, u \rrbracket = \{k \in \mathbb{Z} \mid l \leq k < u\}$. For $n \in \mathbb{N}$, let $[n] = \llbracket 0, n \rrbracket = \{0, 1, \dots, n-2, n-1\}$.

Fix 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*. There is a single glue of strength 0, referred to as the *null* glue. A *tile type* is a tuple $t \in (\Sigma^* \times \mathbb{N})^4$, thought of as a unit square 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 $\mathbb{Z}^2$ lattice, each called a *tile*. A *glue set* G_T is the set of all glues associated with a tile set T . Given a direction $d \in \{N, E, S, W\} = \mathcal{D}$ and a tile set T , $G_{T,d}$ is the set of all glues on the edges of tiles in direction d and $g_{t,d}$ is the glue g on tile t in direction d . We write $\text{str}_t(d)$ to denote the strength of $g_{t,d}$.

Given a tile set T , a *configuration* is an arrangement (possibly empty) of tiles in the lattice $\mathbb{Z}^2$, i.e. a partial function $\alpha : \mathbb{Z}^2 \dashrightarrow T$. 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* of $X \subset \mathbb{Z}^2$ is $\{X + (x, y) \mid x, y \in \mathbb{Z}\}$, i.e. the set of all its translations. The shape of an assembly α is the shape S_α of its domain: two assemblies have the same shape if they have the same domain modulo translation. An assembly α contains a shape S if there is $s \in S$ such that $s \subset \text{dom } \alpha$. 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 $\text{dom } \alpha \subseteq \text{dom } \beta$ and for all $p \in \text{dom } \alpha$, $\alpha(p) = \beta(p)$ (i.e., they have tiles of the same types in all locations of $\text{dom } \alpha$).

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* (a.k.a. *temperature*). 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 $\alpha \rightarrow_1^{\mathcal{T}} \beta$) if $\alpha \sqsubseteq \beta$ and $|S_\beta \setminus S_\alpha| = 1$. That is, $\alpha \rightarrow_1^{\mathcal{T}} \beta$ if β differs from α by the addition of a single tile. The *$\mathcal{T}$ -frontier* is the set $\partial^{\mathcal{T}} \alpha = \bigcup_{\alpha \rightarrow_1^{\mathcal{T}} \beta} S_\beta \setminus S_\alpha$ of locations in which a tile could τ -stably attach to α . When $\mathcal{T}$ is clear from context we simply refer to these as the *frontier* locations.

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 $\alpha_0, \alpha_1, \dots$ over $\mathcal{A}^T$ is called a *$\mathcal{T}$ -assembly sequence* if, for all $1 \leq i < k$, $\alpha_{i-1} \rightarrow_1^{\mathcal{T}} \alpha_i$. The *result* $\lim \alpha$ of an assembly sequence α 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 $\alpha \rightarrow^{\mathcal{T}} \beta$) if there is a $\mathcal{T}$ -assembly sequence starting with α whose result is β . We say α is *$\mathcal{T}$ -producible* if $\sigma \rightarrow^{\mathcal{T}} \alpha$ 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 that is $\mathcal{T}$ -producible from α . We denote the set of $\mathcal{T}$ -producible and $\mathcal{T}$ -terminal assemblies by $\mathcal{A}_\square[\mathcal{T}]$. If $|\mathcal{A}_\square[\mathcal{T}]| = 1$, i.e., there is exactly one terminal assembly, we say that $\mathcal{T}$ is *directed*. When $\mathcal{T}$ is clear from context, we may omit $\mathcal{T}$ from notation.

Given TAS $\mathcal{T}$, we define $\mathcal{S}_\square[\mathcal{T}] = \{S_\alpha \mid \alpha \in \mathcal{A}_\square[\mathcal{T}]\}$ as the set of all shapes of terminal assemblies in $\mathcal{T}$. Clearly, if $\mathcal{T}$ is directed $|\mathcal{S}_\square[\mathcal{T}]| = 1$. However, even if $\mathcal{T}$ is not directed it may be the case that $|\mathcal{S}_\square[\mathcal{T}]| = 1$ (see [6] for an example). In such a case we call the system *shape directed* and say that $\mathcal{T}$ *strictly self-assembles* S_α , where $S_\alpha \in \mathcal{S}_\square[\mathcal{T}]$.

2.2 The Synchronous Tile Assembly Model

The Synchronous Tile Assembly Model (syncTAM) is the variant of the aTAM where at every step of assembly, rather than a single location of the frontier being non-deterministically selected to receive a tile, *all* frontier locations simultaneously receive a tile each. As the frontier of a system can grow arbitrarily large, the number of tiles added per assembly step may also grow arbitrarily large. As in the aTAM, if any single frontier location allows τ -strength binding of tiles of multiple types, one of those types is non-deterministically chosen for each such location.

We formally define the syncTAM in the same way as the aTAM with the exception of the definition of a single step in the assembly sequence. Given a syncTAS $\mathcal{S} = (T, \sigma, \tau)$ and two τ -stable assemblies α and β , we say that α *$\mathcal{S}$ -produces β in one step*, written $\alpha \rightarrow_1^{\mathcal{S}} \beta$, if $\alpha \sqsubseteq \beta$ and $\text{dom}(\beta) \setminus \text{dom}(\alpha) = \partial^{\mathcal{S}}(\alpha)$. Note that like in the aTAM, we define $\partial^{\mathcal{S}}(\alpha)$ to be the set of locations where a tile can τ -stably attach to α . The syncTAM inherits all other definitions and notation from the aTAM.

To distinguish between a tile assembly system (a.k.a. TAS) in the aTAM versus one in the syncTAM, we will use the term *syncTAS* to refer to a TAS in the syncTAM.

2.3 Limited Synchronicity Tile Assembly Model

The Limited Synchronicity Tile Assembly Model (L-syncTAM) is a variant of the syncTAM (introduced in [2]) wherein instead of all frontier locations simultaneously receiving a tile each step, some predefined nonzero integer number of frontier locations each receive a tile. We define this number as the *synchronicity* for a given system, and denote it as $l \in \mathbb{Z}^+$. At every assembly step, l frontier locations are filled (unless the frontier is smaller than l , in which case all frontier locations are filled).

Formally, we define the L-syncTAM similarly to the syncTAM. We call a tile assembly system $\mathcal{L}$ in the L-syncTAM an *L-syncTAS* and define it as $\mathcal{L} = (T, \sigma, \tau, l)$, where l is its synchronicity and T , σ , and τ are defined as previously. As with the syncTAM, given a τ -stable assembly α in $\mathcal{L}$, we define $\partial^{\mathcal{L}}(\alpha)$ to be the set of locations where a tile can τ -stably attach to α , and the L-syncTAM inherits all other definitions and notations from the aTAM, with the following refinement to the definition of an assembly step.

Given two τ -stable assemblies α and β in $\mathcal{L}$, we say that α *$\mathcal{L}$ -produces β in one step*, written $\alpha \rightarrow_1^{\mathcal{L}} \beta$ if $\alpha \sqsubseteq \beta$ and one of the two following are true:

1. $\text{dom}(\beta) \setminus \text{dom}(\alpha) = \partial^{\mathcal{L}}(\alpha)$ and $|\partial^{\mathcal{L}}(\alpha)| \leq l$
2. $\text{dom}(\beta) \setminus \text{dom}(\alpha) \subseteq \partial^{\mathcal{L}}(\alpha)$ and $|\text{dom}(\beta) \setminus \text{dom}(\alpha)| = l$

The first case holds when the frontier is no larger than l . In the second case, when the frontier is larger, the particular subset of frontier locations that receive tiles at a step is non-deterministically selected from the full set of frontier locations. Note that since binding locations are selected only from the current frontier, $\partial^{\mathcal{L}}(\alpha)$, where tiles can bind to the existing assembly α with strength $\geq \tau$, for a tile in location $\vec{l} \in \mathbb{Z}^2$, where $\vec{l} \notin \text{dom} \alpha$, to attach to α it must form one or more bonds with strengths totaling $\geq \tau$ with one or more tiles already in α , regardless of how many bonds it may form with other tiles attaching during the same step.

Note that the aTAM is a special case of the L-syncTAM where $l = 1$, and the syncTAM can be thought of as being a special case where l is allowed to be ∞ .

2.4 Scaled shapes

Given S , a connected set of points in $\mathbb{Z}^2$, we define a version of S *scaled by factor c* , and denote it by S^c , as $S^c = \{(x, y) \in \mathbb{Z}^2 \mid (\lfloor \frac{x}{c} \rfloor, \lfloor \frac{y}{c} \rfloor) \in S\}$. Intuitively, S^c is a version of S expanded by replacing each point of S by a $c \times c$ square of points. Likewise, for a set A of shapes, A^c is defined as $\{S^c \mid S \in A\}$.

Given two TASs $\mathcal{T}$ and $\mathcal{T}'$, if $\mathcal{S}_{\square}[\mathcal{T}] = \mathcal{S}_{\square}[\mathcal{T}']$, then we say that $\mathcal{T}$ *shape-matches* $\mathcal{T}'$. Similarly, if $\mathcal{S}_{\square}[\mathcal{T}]^c = \mathcal{S}_{\square}[\mathcal{T}']^d$ for some $c, d \in \mathbb{Z}^+$, we say that $\mathcal{T}$ *scale shape-matches* $\mathcal{T}'$.

2.5 Synchronous intrinsic simulation

Intrinsic universality via intrinsic simulation has been studied in a variety of tile-assembly models (e.g. [8, 9, 15–17, 23]). Essentially, intrinsic simulation deals with one system simulating the behavior of another modulo scale factor, with squares of tiles in the simulating system representing individual tiles of the simulated system. Intrinsic universality (IU) for a model (or class of systems within a model) occurs when there is a single tile set (along with associated functions, to be discussed) that can be used to create a seed structure from its tiles for any of the systems in the model (or class) and then intrinsically simulate them. Now we provide formal definitions which are largely taken from [16] but are modified to incorporate the dynamic of synchronicity of the L-syncTAM.

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 $\alpha : \mathbb{Z}_m^2 \dashrightarrow T$, where $\mathbb{Z}_m = \{0, 1, \dots, m-1\}$. Let B_m^T 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 $\alpha : \mathbb{Z}^2 \dashrightarrow T$ and $(x', y') \in \mathbb{Z}^2$, define $\alpha_{(x', y')}^m$ to be the m -block macrotile defined by $\alpha_{(x', y')}^m(i_x, i_y) = \alpha(mx' + i_x, my' + i_y)$ for $0 \leq i_x, i_y < m$. For some tile set S , a partial function $R : B_m^S \dashrightarrow T$ is said to be a *valid m-block macrotile representation* from S to T if for any $\alpha, \beta \in B_m^S$ 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*¹ $R^* : \mathcal{A}^S \rightarrow \mathcal{A}^T$ such that $R^*(\alpha') = \alpha$ if and only if $\alpha(x, y) = R\left(\alpha'_{(x,y)}^m\right)$ for all $(x, y) \in \mathbb{Z}^2$. 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 $\alpha'_{(x,y)}^m$, $(x, y) + (u_x, u_y) \in \text{dom}(\alpha)$ for some $(u_x, u_y) \in \{(0, \pm 1), (\pm 1, 0)\}$, or if α' has at most one non-empty m -block $\alpha'_{0,0}^m$. In other words, α' may have tiles in macrotile blocks representing empty space in α , but only if that position is adjacent to a macrotile that represents 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 $\mathcal{T} = (T, \sigma_T, \tau_T, l_T)$ and $\mathcal{S} = (S, \sigma_S, \tau_S, l_S)$ each be an L-syncTAS, and let R be an m -block representation function $R : B_m^S \rightarrow T$. We will define how $\mathcal{S}$ *synchronously simulates* $\mathcal{T}$ at *spatial scale factor* m and *temporal scale factor* $s \in \mathbb{Z}^+$.

► **Definition 1.** Let $\mathcal{L} = (T, \sigma, \tau, l)$ be an L-syncTAS, and let α and β be two τ -stable assemblies in $\mathcal{L}$. We say that β is *equivalent modulo l* to α , and write $\beta \approx_l \alpha$, if and only if $\exists \gamma$ such that γ is a τ -stable assembly in $\mathcal{L}$ and the following conditions hold:

1. $\alpha \rightarrow_1^{\mathcal{L}} \gamma$
2. $\alpha \sqsubseteq \beta \sqsubseteq \gamma$
3. $\exists i$, where $0 \leq i < l$, such that $|\text{dom}(\alpha)| + i = |\text{dom}(\beta)| = |\text{dom}(\gamma)| - l + i$
4. $\text{dom}(\beta) \setminus \text{dom}(\alpha) \subseteq \partial^{\mathcal{L}}(\alpha)$

Essentially, if $\beta \approx_l \alpha$, β can be thought of as an “intermediate” assembly between α and γ (where α produces γ in one step) where only i (which is $< l$) tiles have been added to α . While such an assembly β is not producible in $\mathcal{L}$ if $i > 0$, it will be useful to define such an assembly in the context of a simulator that may require many tile additions to simulate each individual tile addition of $\mathcal{L}$.

► **Definition 2.** 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. $\forall \alpha' \in \mathcal{A}[\mathcal{S}], \exists \alpha \in \mathcal{A}[\mathcal{T}]$ such that $R^*(\alpha') \approx_{l_T} \alpha$
2. $\forall \alpha \in \mathcal{A}[\mathcal{T}], \exists \alpha' \in \mathcal{A}[\mathcal{S}]$ such that $R^*(\alpha') = \alpha$
3. $\{R^*(\alpha') \mid \alpha' \in \mathcal{A}_{\square}[\mathcal{S}]\} = \mathcal{A}_{\square}[\mathcal{T}]$.
4. For all $\alpha' \in \mathcal{A}[\mathcal{S}]$, α' maps cleanly to $R^*(\alpha')$.

► **Definition 3.** We say that $\mathcal{T}$ *follows* $\mathcal{S}$ (under R), and we write $\mathcal{T} \dashv_R \mathcal{S}$, if $\alpha' \rightarrow^{\mathcal{S}} \beta'$ for some $\alpha', \beta' \in \mathcal{A}[\mathcal{S}]$ implies that $\exists \alpha'', \beta'' \in \mathcal{A}[\mathcal{S}]$ such that $R^*(\alpha') \approx_l R^*(\alpha'')$, $R^*(\beta') \approx_l R^*(\beta'')$ and $R^*(\alpha'') \rightarrow^{\mathcal{T}} R^*(\beta'')$.

► **Definition 4.** We say that $\mathcal{T}$ *synchronously follows* $\mathcal{S}$ (under R with temporal scale factor s), and we write $\mathcal{T} \dashv_R^s \mathcal{S}$, if $\mathcal{T} \dashv_R \mathcal{S}$ and, for every assembly sequence $\vec{\beta} = \beta_0, \beta_1, \beta_2, \dots$ in $\mathcal{S}$, there exists assembly sequence $\vec{\alpha} = \alpha_0, \alpha_1, \alpha_2, \dots$ in $\mathcal{T}$ such that:

1. $\alpha_0 = \sigma$ and $R^*(\beta_0) = \sigma$
2. For every $0 \leq i < |\vec{\alpha}|$, there is $i' \in [i \cdot (s - 1), i \cdot (s + 1)]$ such that $R^*(\beta_{i'}) = \alpha_i$

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

¹ 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.

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 5.** 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 $\Pi \neq \emptyset$ and $R^*(\alpha') = \alpha$ for all $\alpha' \in \Pi$, such that, for every $\beta \in \mathcal{A}[\mathcal{T}]$ where $\alpha \rightarrow^{\mathcal{T}} \beta$, (1) for every $\alpha' \in \Pi$ there exists $\beta' \in \mathcal{A}[\mathcal{S}]$ where $R^*(\beta') = \beta$ and $\alpha' \rightarrow^{\mathcal{S}} \beta'$, and (2) for every $\alpha'' \in \mathcal{A}[\mathcal{S}]$ where $\alpha'' \rightarrow^{\mathcal{S}} \beta'$, $\beta' \in \mathcal{A}[\mathcal{S}]$, $R^*(\alpha'') \approx_{\mathcal{T}} \alpha$, and $R^*(\beta') = \beta$, there exists $\alpha' \in \Pi$ such that $\alpha' \rightarrow^{\mathcal{S}} \alpha''$.

► **Definition 6.** We say that $\mathcal{S}$ synchronously simulates $\mathcal{T}$ (under R and temporal scale factor s) if $\mathcal{S} \Leftrightarrow_R \mathcal{T}$ (equivalent productions), $\mathcal{T} \dashv_R^s \mathcal{S}$ and $\mathcal{S} \models_R \mathcal{T}$ (equivalent dynamics).

3 Shapes

In this section we show how the syncTAM, even at temperature-1, is capable of forms of cooperative behavior and detection of empty locations. This allows the syncTAM to implement a flagpole, the shibboleth shape for cooperation, at temperature 1. This flagpole relies on *implementing* cooperation thanks to synchronization by a *cooperation gadget*. Synchronization also allows for parts of the syncTAM assembly to “probe” regions of space in order to assess whether or not some other part of the assembly has / has not arrived in that region within a fixed window of time. As such, syncTAM spaces may use empty space as input for computation. This sidesteps the gap between weak and strict self-assembly of shapes [4, 5, 13], and enables the strict self-assembly of a (modified) discrete Sierpinski Triangle. Last, the cooperation gadget proves to be enough to build the same assemblies as any *directed* aTAM system with a non-cooperative system running in the syncTAM.

First, we show how temperature-1 syncTAM systems can simulate cooperative attachments.

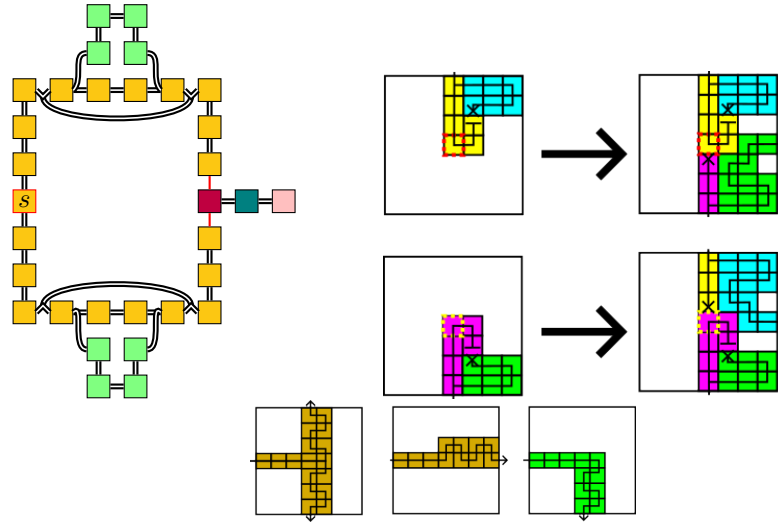
3.1 Simulating a flagpole at temperature-1

From the earliest investigations on temperature 1 and intrinsic universality [21], it has been known that the aTAM temperature 1 system cannot simulate systems which exhibit cooperation, i.e. systems with a temperature at least 2. In particular, let $\mathcal{F}$ be an aTAM system at temperature 2 as shown in Figure 1, which is a slight variation of the one in [21].

As in [21], this system features two arms growing from left to right, one at the top on line $y = 3$, and one at the bottom on line $y = -3$. Each of these arms will turn back towards the $y = 0$ line at a non-deterministic position. If both arms do so in the same column, a three tile flag (in purple, teal and pink) will grow from their meeting point. Our variation adds green tiles, which allow the arms to dally while they move right: for every 4 column they advance, they can do so either straight with the yellow tiles, in 4 time steps, or with a detour through the green tiles, which takes them 6 time steps total. Because of this, even if both arms pick the same column $x = c$ to turn back to the center, they do not necessarily reach position $(c, \pm 1)$ at the same time.

Let F be the set of shapes assembled by $\mathcal{F}$ in the aTAM at temperature 2. Note that in the syncTAM at temperature 2, $\mathcal{F}$ assembles the same set of shapes.

► **Lemma 7.** The TAS $\mathcal{F}$ cannot be scale shape-matched at temperature 1 in the aTAM: there are no integer $c \geq 1$ and TAS $\mathcal{F}_1$ at temperature 1 such that $\mathcal{A}_{\square}[\mathcal{F}_{\infty}] = c \cdot F$.



■ **Figure 1** Our variation $\mathcal{F}$ on the “Flagpole” system from [21], with the TAS on the left. Then, going clockwise, some growth patterns. The first two are in the cooperative gadget (red in the figure to the left), with either the top or bottom neighboring tile arriving first, then the other one arriving and initiating the growth of the flag. If the top arrives first, the flag can only be initiated by the arrival of the path from the bottom because growth from the top consists of two paths. The yellow, if unblocked, blocks the growth of the aqua. The bottom works symmetrically, and if it arrives first its pink path blocks its own green path but also the top’s yellow path, which allows the aqua path to continue, unblocked, to form the flag. Below, the macrotiles for the non-cooperative parts: all paths through the macrotiles are of length 11.

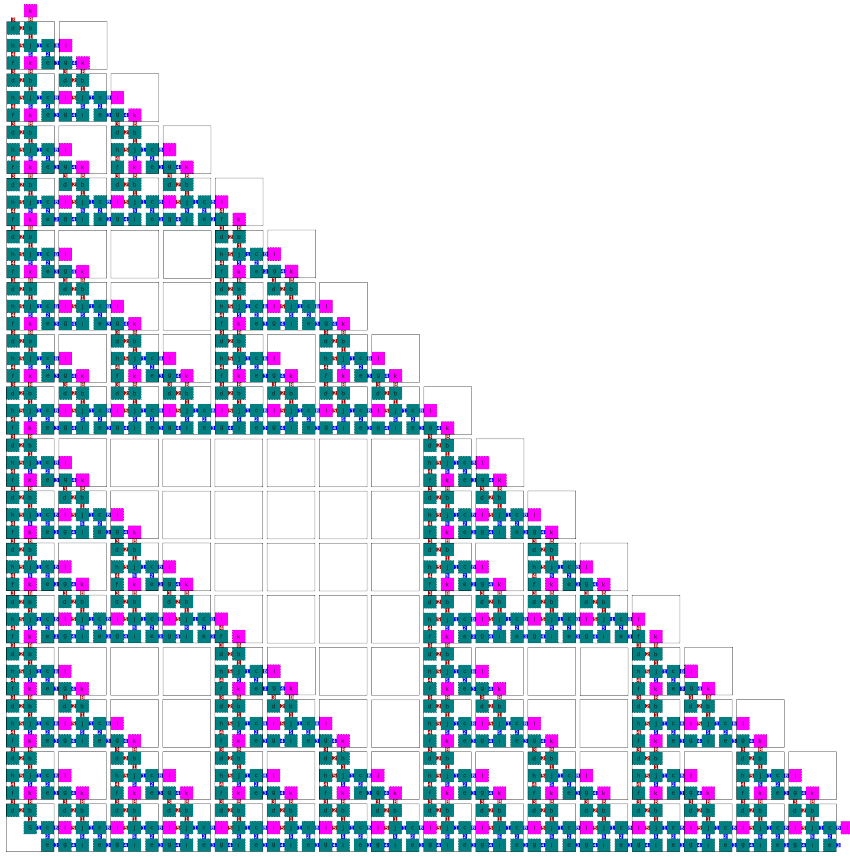
Proof. Since the system $\mathcal{F}$ is a trivial modification of the system in [21], proving Lemma 7 requires only a trivial application of the Window Movie Lemma that was used to show that the original system could not be simulated because any system attempting to simulate it would have to be able to grow assemblies of invalid shapes. Essentially, the horizontal arms can grow to arbitrary lengths and therefore each can be “pumped” once grown past some “pumping” length. If a flag can grow in a case where the top and bottom arm are of the same length, it must be able to do so via glue connections to only the top or bottom arm. By pumping only the arm with the connections to the flag tiles, that arm can be made longer than the other, but it must still be able to grow the flag. This is an invalid shape. ◀

On the other hand, there is a temperature 1 TAS $\mathcal{F}_{\text{sync}}$ which simulates $\mathcal{F}$ in the syncTAM, and thus has $\mathcal{A}_{\square}[\mathcal{F}_{\text{sync}}] = c \cdot F$. A depiction of the system with $c = 7$ and the gadgets that simulate the cooperative behavior of the tile at the base of the flagpole can be seen in Figure 1.

3.2 A modified Sierpinski triangle at temperature-1

In this section we show how a modified version of the Sierpinski triangle at scale factor 3 can self-assemble in the syncTAM. This construction is used to demonstrate the ability of systems in the syncTAM to have gadgets that detect the lack of growth in an area, even without glue cooperativity (i.e. at temperature-1) and continue growth only in that absence of growth.

Let Δ be the shape defined in Figure 2.



■ **Figure 2** A portion of the infinite modified Sierpinski triangle that strictly self-assembles in the syncTAM at temperature 1.

► **Theorem 8.** *The set Δ strictly self-assembles in the syncTAM at temperature 1 and scale 3 but does not strictly self-assemble in the aTAM at any temperature and scale.*

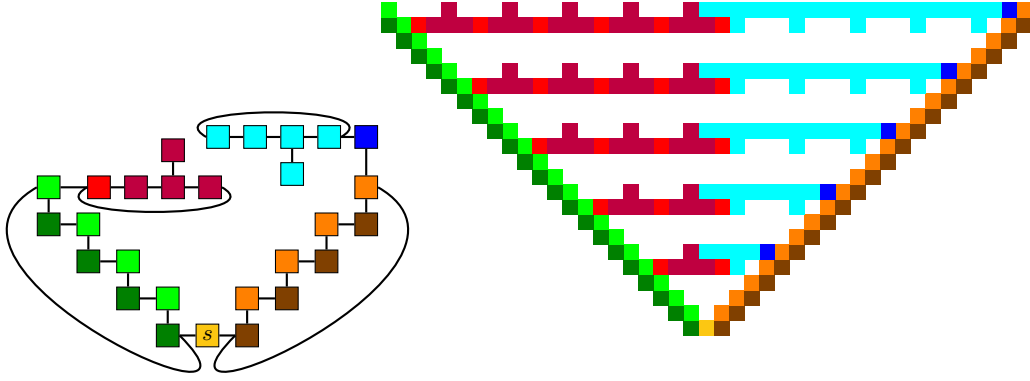
Proof. Lemma 9 shows that Δ does strictly self-assemble at temperature 1 in the syncTAM.

To see that it does not strictly self-assemble in the aTAM, notice that like the actual Sierpinski Triangle, for each k this shape features vertical lines where it has groups of tiles of constant size (here, 2) separated by gaps of size more than k , yet for no vector $\vec{p}$ does it have $\vec{p}$ -periodic, square, non-empty subpatterns of arbitrary size. Then, by the Tree Pump Theorem [5], it cannot be strictly self-assembled in the aTAM. The proof is the same as that of Lemma 31 in that paper, as only these hypothesis on the shape is used. ◀

► **Lemma 9.** *There is a syncTAM system which strictly self-assembles Δ at temperature 1 and scale-factor 3.*

We prove Lemma 9 by providing a syncTAM system which strictly self-assembles Δ .

A portion of the infinite shape can be seen in Figure 2, and the syncTAM system can be downloaded at [http://self-assembly.net/wiki/index.php/Synchronous_Tile_Assembly_Model_\(syncTAM\)](http://self-assembly.net/wiki/index.php/Synchronous_Tile_Assembly_Model_(syncTAM)), which also provides a link to the simulator for simulating the system. Our syncTAM system uses 3×3 *macrotiles* to logically “simulate” the behaviors of the aTAM tiles. The synchronous nature of the syncTAM allows us to use the timing of different portions of the assembly’s growth to guarantee that growth will be blocked from



■ **Figure 3** A system which is directed in the syncTAM but not in the aTAM. The tiles of the temperature 1 syncTAM system $\mathcal{S}$ (the seed is the yellow s tile), and a portion of the infinite shape V assembled by $\mathcal{S}$ in the syncTAM. Growth begins from the seed tile at the bottom and proceeds with the diagonal “arms” growing upward to the left and right, and with red rows growing right from the left arm, and blue rows growing left from the right arm, at every 4th step. The green part is L , the orange and brown part is R , the blue parts are the $B_k^+ \cup T_k^+$, while the red parts are the $B_k^- \cup T_k^-$.

proceeding whenever growth of a stage in one direction reaches the correct length because it crashes into growth from the other direction that has also reached the correct length. In the aTAM, it cannot be guaranteed that the growth of both portions will proceed at identical rates, and thus the necessary blocking may not occur. However, by adding tiles in every frontier location at every step the syncTAM provides such a guarantee and thus provides a powerful tool for designing systems. A detailed description of the system can be found in the appendix of the full version of this paper online [3].

4 Matching Assemblies of Directed aTAM Systems With Non-cooperative syncTAM Systems

Any system which is directed in the aTAM is also directed in the syncTAM, since any syncTAM production must be an aTAM production [2]. In this directed case, it is actually possible to assemble a scaled version of the aTAM system’s terminal assembly. Note that being directed in the aTAM is a stronger condition than being directed in the syncTAM: in order to be directed in the aTAM, a TAS needs to be deterministic in the face of asynchronism of tile additions. For example, the system in Figure 3 is directed in the syncTAM, assembling the production depicted on the figure, while in the aTAM, the meeting point of the two parts of each rung varies based on the particular assembly sequence chosen.

► **Theorem 10.** *If $\mathcal{S}$ is directed in the aTAM, there is a temperature 1 syncTAM system $\mathcal{S}'$ which assembles a scaled version of $\alpha \in \mathcal{A}_{\square}[\mathcal{S}]$, for some scale factor s and macrotile representation function R .*

The proof of this theorem can be found in Appendix A of the appendix, with more detail available in the appendix of the full version of this paper [3]. The construction leverages the cooperation gadget of Section 3.1, using the directedness of $\mathcal{S}$ to argue away timing problems due to the need to simulate the asynchronicity of $\mathcal{S}$.

5 L-SyncTAM Results

First we present Lemma 11, which will be useful for several of the proofs in this section. Lemma 11 shows that when a certain set of requirements are met for subassemblies produced by systems in the aTAM or L-syncTAM, all such subassemblies grow independently of each other.

Given a shape S let $\text{bound}(S)$ refer to the set of unit vectors in $\mathbb{Z}^2$ that form its perimeter, or *boundary*. Given two points $l_1, l_2 \in \mathbb{Z}^2$, let $\text{dist}(l_1, l_2)$ be a function that returns the minimal Manhattan distance between points l_1 and l_2 .

► **Lemma 11** (Eventually Independent Subassemblies). *Let $\mathcal{T} = (T, \sigma, \tau, l)$ be a shape-directed L-syncTAS in the L-syncTAM for any $l \geq 1$ and let S be the shape that $\mathcal{T}$ strictly self-assembles. If there exists a cut C of $\mathbb{Z}^2$ that partitions S into shapes $s_0, s_1, \dots, s_n$ such that:*

1. $\text{dom } \sigma \subseteq s_0$, [The seed is completely contained in s_0]
2. $\forall 0 < i \leq n, |s_i| = \infty$, [Other than perhaps s_0 , all must be infinite]
3. For each $0 < i \leq n$, $|C \cap \text{bound}(s_i)| < \infty$, [C has a finite intersection with the boundary of each except maybe s_0]
4. For all $0 < i \neq j \leq n$, $\forall l_1 \in s_i, \forall l_2 \in s_j, \text{dist}(l_1, l_2) \geq 3$. [The minimum distance between points in any pair of shapes (not including s_0) is at least 3.]
5. For each $0 < i \leq n, C \cap \text{bound}(s_i) = \text{bound}(s_0) \cap \text{bound}(s_i)$, [C contains the entire shared boundary of each s_i with s_0]

then, the following holds: Let $\alpha \in \mathcal{A}_{\square}[\mathcal{T}]$ be any terminal assembly of $\mathcal{T}$ and let $\vec{\alpha}$ be any arbitrary assembly sequence that produces α . Recall that $\text{dom } \alpha = S$. There exists some $f \in \mathbb{N}$ such that for every assembly step of $\vec{\alpha}$ whose index (i.e. location in the sequence) is greater than f , $\forall 0 < i \leq n$ there is a frontier location inside of s_i .

Proof. To prove Lemma 11, we first note that since the seed σ is completely contained in s_0 , no location of s_i initially has a tile, and for all $0 < i \leq n$, the first location in s_i that receives a tile must be adjacent to cut C since that cut separates the subassembly in s_i (which we will call α_i) from the rest of the assembly. For any (necessarily empty) location $\vec{l} \in \mathbb{Z}^2$ to become a frontier location, at least one neighboring location $\vec{l}'$ must have been a frontier location that became filled with a tile. If $\vec{l}$ first becomes a frontier location during some assembly step s , then some neighboring location must have received a tile during step $s - 1$. This means that any location in s_i that receives a tile must have been included in a path of successive, adjacent frontier locations that originated adjacent to the seed and at some point crossed C .

By the conditions of the lemma, the cut C only overlaps with a finite portion of the perimeter of s_i . Also, since each such s_i has infinite size that is filled (in the limit) by tiles, there must be an infinite number of assembly steps during which there exists a frontier location within s_i during $\vec{\alpha}$. For any assembly step s during which there is not a frontier location in s_i , since there must exist a later step where a location in s_i receives a tile, there must exist a later step $s' > s$ where there exists a frontier location adjacent to C , as this is the only boundary into which a path of frontier locations can enter s_i . Since the locations adjacent to C are finite, this can only happen a finite number of times. Therefore, there exists some constant f_i such that after the f_i th assembly step no more tiles are placed adjacent to C , and since the number of tiles placed within s_i is infinite, for the infinite number of steps greater than f_i , there must be frontier locations within s_i .

Finally, since the argument holds for each s_i , each has its own such constant f_i and the f of the lemma is simply the maximum of them. Corollary 12 directly follows from the fact that each shape s_i for $0 < i \leq n$ is separated by a distance greater than that of fuzz. ◀

► **Corollary 12.** *Let $\mathcal{S} = (T, \sigma, \tau, l)$ be an L -syncTAS in the L -syncTAM for any $l \geq 1$. If $\mathcal{S}$ synchronously simulates some $\mathcal{T}$ such that Lemma 11 holds for $\mathcal{T}$, then Lemma 11 holds for $\mathcal{S}$.*

Corollary 12 follows immediately from Lemma 11 since the scaled shape assembled by $\mathcal{S}$ must have a similarly scaled and translated cut C' for which all of the same properties hold (noting that the condition that the distance required between shapes by Item 4 ensures sub-shapes are still disjoint even accounting for fuzz in the simulation), and therefore so does the proof.

5.1 The class of L-syncTAM systems is not IU for any $l > 1$

For any fixed $l \geq 2$, the class of L-syncTAM systems with synchronicity l is not intrinsically universal (for synchronous simulation). For any fixed candidate simulator, there are systems that cannot be simulated by that simulator.

► **Lemma 13.** *For all $l \geq 2$, there is an infinite family $(\mathcal{S}_n^l)_{n \in \mathbb{N}}$ of L -syncTAM systems with synchronicity l such that for each n , $\mathcal{S}_n^l$ has $n + 5$ tiles and cannot be synchronously simulated in the L -syncTAM at synchronicity l with any temporal scale factor s and any spatial scale factor greater than 1.*

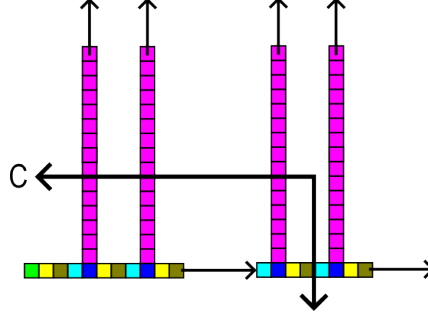
Proof. Fix n . Consider the system of Figure 4, with the green seed tile fixed at $(0, 0)$, and 4 tile types (yellow, gold, aqua, and blue) that grow a row with a periodic sequence of those 4 tile types infinitely far to the right. From each blue tile, a periodic sequence of n unique tile types attach (all shown as pink) infinitely often, to create infinite vertical columns. This system is directed, and strictly-self assembles the shape shown in Figure 4. Call this system $\mathcal{S}_n^l = (T, \sigma, 1, l)$ and note that $|T| = n + 5$.

We will prove Lemma 13 by contradiction, so assume that L-syncTAM system $\mathcal{S}'$ synchronously simulates $\mathcal{S}_n^l$ at scale factor $c > 1$ and some temporal scale factor s . We now inspect a particular assembly sequence in $\mathcal{S}_n^l$ and show that $\mathcal{S}'$ does not synchronously follow $\mathcal{S}_n^l$.

Let $\vec{\alpha}$ be an assembly sequence in $\mathcal{S}_n^l$ that proceeds as follows. The horizontal row grows to a length $> 4((3s + 1)l + (l - 1)) + 2$ by starting from the seed and attaching $(3s + 1)l + (l - 1)$ copies of the 4-tile repeating segment. From each blue tile, a vertical column begins to grow. The columns can grow in any order, resulting in $(3s + 1)l + (l - 1)$ vertical columns which will each ultimately be infinite in height. Note that the horizontal row will continue to grow and initiate further upward columns, but we will not need to include them in our analysis.

There must be an assembly sequence $\vec{\alpha}'$ in $\mathcal{S}'$ that models $\vec{\alpha}$. We let $\vec{\alpha}'$ proceed until it has not only grown sufficiently far horizontally (i.e. $> (c \cdot 4((3s + 1)l + (l - 1)) + 2)$), but also continued growth so that all subassemblies mapping to the $(3s + 1)l + (l - 1)$ vertical arms are of sufficient height that Corollary 12 of Lemma 11 can be applied. This results in cut C separating the $(3s + 1)l + (l - 1)$ vertical columns into shapes $s_1, s_2, \dots, s_{(3s+1)l+(l-1)}$ of the lemma, so that each vertical column is guaranteed to have a frontier location in all further assembly steps. The portion of the horizontal bar containing the seed will be s_0 , and the rest of the assembly to the east of the cut will be $s_{(3s+1)l+l}$. We extend assembly sequence $\vec{\alpha}'$ as follows.

We will refer to each vertical column as V_i for $0 \leq i < (3s + 1)l + (l - 1)$, and to the rightmost $l - 1$ vertical columns of that group as *buffer* columns. Since $\mathcal{S}'$ has synchronicity l , there must be l tiles placed during every assembly step. Therefore, starting with $i = 0$, we perform assembly steps such that each places one tile in each of the buffer columns (using



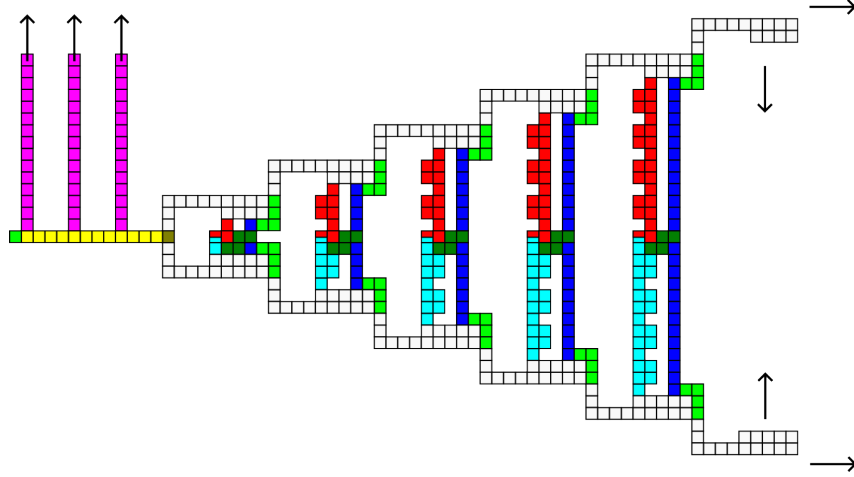
■ **Figure 4** A schematic depiction of a TAS in the L-syncTAM with synchronicity l that cannot be simulated by a system that is IU for the L-syncTAM at synchronicity l . The seed is the green tile on the bottom left. An infinite number of repetitions of 4 tiles attach to its right. From each blue tile, a column grows upward infinitely, with repetitions of a series of n unique tile types attaching. Cut C of Lemma 11 is also shown.

$l - 1$ of the tile placements), and one tile in V_i . We repeat this until there is a frontier location, which we will refer to as a *nonresolving frontier location*, in V_i so that when a tile is placed there, no new macrotile resolves to a tile of $\mathcal{S}_n^l$. That is, assuming the assembly α' in $\mathcal{S}'$ maps under R to some assembly α in $\mathcal{S}_n^l$ (i.e. $R(\alpha') = \alpha$), when a tile is placed in that nonresolving frontier location to create assembly α'' in $\mathcal{S}'$, it is still the case that $R(\alpha'') = \alpha$. This must be possible because the scale factor of the simulation is $c > 1$ and therefore macrotiles must have dimensions > 1 , meaning that any path from the south to the north of a macrotile must be of length > 1 and a macrotile only resolves during a single assembly step. Thus, there must be at least one step during which there is a tile placed that does not cause the macrotile to resolve, and that tile's location is a nonresolving frontier location immediately prior to that.

Once it occurs that a nonresolving frontier location exists in V_i , we increment to $i + 1$ and repeat until nonresolving frontier locations exist in each of the $(3s + 1)l$ leftmost vertical columns. At this point, let the current assembly be β' , and let $\beta = R(\beta')$ be the assembly in $\mathcal{S}_n^l$ to which it maps. We now perform $3s + 1$ assembly steps as follows. For $0 \leq p < 3s + 1$, each step consists of placing a single tile in each V_q , for $lp \leq q < (l+1)p$, into the nonresolving frontier locations. In this way, each column in each set of l columns starting from the left receives a single tile. However, since all locations were nonresolving frontier locations, that means that the assembly sequence proceeded $3s + 1$ steps without any change in the assembly to which it maps in $\mathcal{S}_n^l$. This breaks Definition 4, i.e. the simulated system does not synchronously follow the simulator because it must continue to add l tiles per assembly step, but even allowing for the maximum amount of time within the temporal scaling of s , which is a window of size $3s$, $\mathcal{S}'$ resolves to no additional tiles. This is a contradiction that $\mathcal{S}'$ synchronously simulates $\mathcal{S}_n^l$ and thus Lemma 13 is proven. ◀

► **Theorem 14.** *For all $l \geq 2$, the class of L-syncTAM systems with synchronicity l is not intrinsically universal.*

Proof. Let $l \in \mathbb{N}$. Any fixed L-syncTAM system at synchronicity l with t tile types can only simulate systems with at most $(t + 1)^s$ tiles at scale s , as that is the cardinality of the domain of the representation function. But by Lemma 13, there are infinitely many L-syncTAM systems at synchronicity l which can only be simulated at scale 1. Hence, regardless of the size of the tile set of a fixed system simulator, since it must be finite, no fixed system can simulate all of them and the class of L-syncTAM systems with synchronicity l is not intrinsically universal. ◀



■ **Figure 5** Initial portion of shape S_5 (i.e. S_l for $l = 5$) from the proof of Theorem 15. The seed tile (green) initiates growth of the yellow line which initiates the growth of $l - 2$ arms (pink) that each grow infinitely upward. The gold tile at the end of the yellow row initiates growth of the cone (white) that grows infinitely to the right, with rungs (red, aqua, green, blue, and lime green) that periodically grow between the sides of the cone. A depiction of the growth sequence of the cone can be seen in Figure 7.

5.2 Shapes that cannot self-assemble with less synchronization

We now present a result that shows that for any synchronicity value $l > 1$, there exists a shape which cannot be made by any system with lower synchronicity. This is a much stronger statement than simply saying there is a system with synchronicity l that cannot be simulated by any with lower synchronicity. This instead shows that, even without the constraints of following a system's dynamics, the shape simply cannot be strictly produced.

For any $l > 1$, the shape pictured on Figure 5 strictly self-assembles in an L-syncTAM system with synchronicity l , but not in any system with lower synchronicity.

► **Theorem 15.** *For each $l > 1$, there exists a shape S_l such that:*

1. *There exists L-syncTAM system $\mathcal{L}_l = (T, \sigma, 2, l)$ such that $\mathcal{L}_l$ strictly self-assembles S_l .*
2. *For all $l' < l$, there exists no L-syncTAM system $\mathcal{L}' = (T', \sigma', \tau, l')$ and scale factor $c \in \mathbb{Z}^+$ such that $\mathcal{L}'$ strictly self-assembles $c \cdot S_l$.*

The proof of Theorem 15 relies on observing that any simulator for this system must have $l - 2$ frontier positions for the infinite arms growing upwards. On the other hand, by a pumping argument, it cannot encode the positions within the cone in order to be able to find the middle without keeping pairs of tile additions in sync. Hence synchronicity $\geq l$ is needed in order to assemble that shape. The proof may be found in Appendix B of the appendix, with more detail available in the appendix of the full version of this paper [3].

5.3 Systems that cannot be simulated by systems with greater synchronization

The following theorem states that for any $l \geq 1$, there is a system in the L-syncTAM with synchronicity l that cannot be simulated by any L-syncTAM system whose synchronicity l' is greater than l , i.e. the additional synchronization causes any system to fail to correctly simulate.

► **Theorem 16.** *For each $l \geq 1$, there exists an L -syncTAM system $\mathcal{S}_l$ with synchronicity l such that, for any $l' > l$, there exists no L -syncTAM system $\mathcal{S}'$ with synchronicity l' which simulates $\mathcal{S}_l$.*

Proof. We prove Theorem 16 by first describing a directed L -syncTAS $\mathcal{S}_l$, for arbitrary $l \geq 1$. A schematic depiction can be seen in Figure 6. It consists of a green seed tile fixed at $(0,0)$, and 4 tile types (yellow, gold aqua, and blue) that grow a row with a periodic sequence of those 4 tile types infinitely far to the right. From each blue tile, a single tile (pink) attaches to itself infinitely often, to create infinite vertical columns. This system is similar to that which was shown in the proof of Lemma 13, with the only modification being that each infinite vertical column only contains a single repeating tile type. Call this system $\mathcal{S}_l = (T, \sigma, 1, l)$ and note that $|T| = 6$.

Now, let l' be an integer $> l$. We will prove Theorem 16 by contradiction, so assume that $\mathcal{S}'$ is an L -syncTAS which synchronously simulates $\mathcal{S}_l$ at some temporal scale factor s and spatial scale factor c . Let us now inspect a particular assembly sequence of $\mathcal{S}'$.

Define $k = (3s + 1)l' + (l' - 1)$, a constant whose value will become clear later. Let $\vec{\alpha}$ be an assembly sequence in $\mathcal{S}_l$ that proceeds as follows. The horizontal row grows to a length $> 4k + 2$ by starting from the seed and attaching k copies of the 4-tile repeating segment. From each blue tile, a vertical column begins to grow, resulting in k vertical columns which will ultimately be infinite in height. Note that the horizontal row will continue to grow and initiate further upward columns, but we do not need to include them in our analysis.

There must be an assembly sequence $\vec{\alpha}'$ in $\mathcal{S}'$ that models $\vec{\alpha}$. Let $\vec{\alpha}'$ proceed until it has not only grown sufficiently far horizontally (i.e. $> c \cdot (4k + 2)$), but also continues growth so that all subassemblies mapping to the k vertical columns are of sufficient height that Corollary 12 of Lemma 11 may be applied, with cut C separating k vertical columns into shapes $s_1, s_2, \dots, s_k$, such that each vertical column is guaranteed to have a frontier location in all future assembly steps. The portion of the horizontal bar containing the seed will be s_0 , and the rest of the assembly to the east of the cut will be s_{k+1} . We extend assembly sequence $\vec{\alpha}'$ as follows.

We'll refer to each vertical column as V_i for $0 \leq i < k$, and to the rightmost $l' - 1$ of those as *buffer* columns. Since $\mathcal{S}'$ has synchronicity l' , there must be l' tiles placed during each assembly step. Therefore, starting with $i = 0$, we perform assembly steps such that each places one tile in each of the buffer columns (using $l' - 1$ of the tile placements in each assembly step), and one in V_i . We repeat this until there is a frontier location, which we will refer to as a *resolving frontier location*, in V_i so that there exists a tile type in $\mathcal{S}'$ that may be placed there whose attachment causes a new macrotile to resolve to a tile in $\mathcal{S}_l$. We will refer to this tile as a *resolving tile*. That is, assuming the assembly α' in $\mathcal{S}'$ maps under R to some assembly α in $\mathcal{S}$ (i.e. $R(\alpha') = \alpha$), when a resolving tile is placed in that resolving frontier location to create an assembly α'' in $\mathcal{S}'$, it is the case that $R(\alpha'') \neq \alpha$.

Once it occurs that a resolving frontier location exists in V_i , we increment to $i + 1$ and repeat until resolving frontier locations exist in each of the $(3s + 1)l'$ leftmost vertical columns. At this point, let the current assembly be β' , and let $\beta = R(\beta')$ be the assembly in $\mathcal{S}$ to which it maps. We now perform $3s + 1$ assembly steps as follows. For $0 \leq p < 3s + 1$, each step consists of placing a single resolving tile into the resolving frontier locations in each V_q for $l'p \leq q < (l' + 1)p$. In this way, each column in each set of l' columns starting from the left receives a single tile, causing l' macrotiles to resolve. This breaks Definition 4, i.e. the simulated system does not synchronously follow the simulator because the simulator adds $> l$ tiles in every assembly step for each of $3s + 1$ consecutive steps, causing the dynamics to differ even beyond compensation by the temporal scaling factor s . This is a contradiction of the claim that $\mathcal{S}'$ simulates $\mathcal{S}_l$, and thus Theorem 16 is proven. ◀

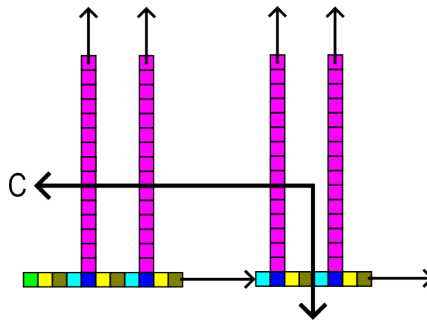


Figure 6 A depiction of a TAS in the L-syncTAM with synchronicity l that cannot be simulated by any TAS in the L-syncTAM with synchronicity $l' > l$. The seed is the green tile on the bottom left, and an infinite number of repetitions of 4 tiles can attach to its right. From each blue tile, a column grows upward infinitely, with copies of the same pink tile type attaching. Cut C of Lemma 11 is also shown.

References

- 1 Kimberly Barth, David Furcy, Scott M. Summers, and Paul Totzke. Scaled tree fractals do not strictly self-assemble. In *Unconventional Computation & Natural Computation (UCNC) 2014*, University of Western Ontario, London, Ontario, Canada July 14-18, 2014, pages 27–39, 2014. doi:10.1007/978-3-319-08123-6_3.
- 2 Florent Becker, Phillip Drake, Matthew Patitz, and Trent Rogers. Synchronous Versus Asynchronous Tile-Based Self-Assembly. In *DNA 2025*, Lyon, France, August 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DNA.31.9.
- 3 Florent Becker, Phillip Drake, Matthew J. Patitz, and Ryder Smith. Powers and limitations of synchronous self-assembly, 2026. arXiv:2605.27604.
- 4 Florent Becker, Daniel Hader, and Matthew J. Patitz. Strict self-assembly of discrete self-similar fractals in the abstract tile assembly model. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'25)*, New Orleans, USA, pages 2387–2466. SIAM, 2025. doi:10.1137/1.9781611978322.80.
- 5 Florent Becker, Daniel Hader, and Matthew J. Patitz. Strict self-assembly of discrete self-similar fractals in the abstract tile assembly model. *Algorithmica*, 88(1):18, 2026. doi:10.1007/S00453-025-01340-W.
- 6 Nathaniel Bryans, Ehsan Chiniforooshan, David Doty, Lila Kari, and Shinnosuke Seki. The power of nondeterminism in self-assembly. *Theory of Computing*, 9:1–29, 2013. doi:10.4086/toc.2013.v009a001.
- 7 Matthew Cook, Yunhui Fu, and Robert T. Schweller. Temperature 1 self-assembly: Deterministic assembly in 3D and probabilistic assembly in 2D. In *SODA 2011: Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2011. doi:10.1137/1.9781611973082.45.
- 8 Erik D. Demaine, Matthew J. Patitz, Trent A. Rogers, Robert T. Schweller, Scott M. Summers, and Damien Woods. The two-handed assembly model is not intrinsically universal. In *40th International Colloquium on Automata, Languages and Programming, ICALP 2013, Riga, Latvia, July 8-12, 2013*, Lecture Notes in Computer Science. Springer, 2013. doi:10.1007/978-3-642-39206-1_34.
- 9 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. doi:10.1109/FOCS.2012.76.

- 10 David Doty, Matthew J. Patitz, Dustin Reishus, Robert T. Schweller, and Scott M. Summers. Strong fault-tolerance for self-assembly with fuzzy temperature. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 417–426, 2010. doi:10.1109/FOCS.2010.47.
- 11 Sándor P. Fekete, Jacob Hendricks, Matthew J. Patitz, Trent A. Rogers, and Robert T. Schweller. Universal computation with arbitrary polyomino tiles in non-cooperative self-assembly. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015), San Diego, CA, USA, January 4-6, 2015*, pages 148–167, 2015. doi:10.1137/1.9781611973730.12.
- 12 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. doi:10.1007/978-3-642-31594-7_60.
- 13 David Furcy and Scott M. Summers. Scaled pier fractals do not strictly self-assemble. *Natural Computing*, 16(2):317–338, 2017. doi:10.1007/S11047-015-9528-Z.
- 14 Oscar Gilbert, Jacob Hendricks, Matthew J. Patitz, and Trent A. Rogers. Computing in continuous space with self-assembling polygonal tiles. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016), Arlington, VA, USA January 10-12, 2016*, pages 937–956, 2016.
- 15 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, January 5-8, 2020*, pages 2607–2624, Salt Lake City, UT, USA, 2020. SIAM. doi:10.1137/1.9781611975994.159.
- 16 Daniel Hader and Matthew J. Patitz. The impacts of dimensionality, diffusion, and directedness on intrinsic cross-model simulation in tile-based self-assembly. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023*, volume 261 of *LIPIcs*, pages 71:1–71:19, Paderborn, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2023.71.
- 17 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. doi:10.1109/FOCS.2016.90.
- 18 Jacob Hendricks, Matthew J. Patitz, Trent A. Rogers, and Scott M. Summers. The power of duples (in self-assembly): It’s not so hip to be square. *Theoretical Computer Science*, 743:148–166, 2018. doi:10.1016/J.TCS.2015.12.008.
- 19 James I. Lathrop, Jack H. Lutz, Matthew J. Patitz, and Scott M. Summers. Computability and complexity in self-assembly. *Theory Comput. Syst.*, 48(3):617–647, 2011. doi:10.1007/s00224-010-9252-0.
- 20 James I. Lathrop, Jack H. Lutz, and Scott M. Summers. Strict self-assembly of discrete Sierpinski triangles. *Theoretical Computer Science*, 410:384–405, 2009. doi:10.1016/J.TCS.2008.09.062.
- 21 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. doi:10.1137/1.9781611973402.56.
- 22 Pierre-Étienne Meunier, Damien Regnault, and Damien Woods. The program-size complexity of self-assembled paths. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, June 22-26, 2020*, pages 727–737, Chicago, IL, USA, 2020. ACM. doi:10.1145/3357713.3384263.

- 23 Pierre-Étienne Meunier and Damien Woods. The non-cooperative tile assembly model is not intrinsically universal or capable of bounded turing machine simulation. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 328–341, New York, NY, USA, 2017. ACM. doi:10.1145/3055399.3055446.
- 24 Matthew J. Patitz, Robert T. Schweller, and Scott M. Summers. Exact shapes and Turing universality at temperature 1 with a single negative glue. In Luca Cardelli and William M. Shih, editors, *DNA Computing and Molecular Programming - 17th International Conference, DNA 17, Pasadena, CA, USA, September 19-23, 2011. Proceedings*, volume 6937 of *Lecture Notes in Computer Science*, pages 175–189. Springer, 2011. doi:10.1007/978-3-642-23638-9_15.
- 25 Matthew J. Patitz and Scott M. Summers. Self-assembly of decidable sets. *Natural Computing*, 10(2):853–877, 2011. doi:10.1007/s11047-010-9218-9.
- 26 Paul W. K. Rothmund 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. doi:10.1145/335305.335358.
- 27 David Soloveichik and Erik Winfree. Complexity of self-assembled shapes. *SIAM Journal on Computing*, 36(6):1544–1569, 2007. doi:10.1137/S0097539704446712.
- 28 Erik Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, June 1998.

A Proof of Theorem 10

► **Theorem 10.** *If $\mathcal{S}$ is directed in the aTAM, there is a temperature 1 syncTAM system $\mathcal{S}'$ which assembles a scaled version of $\alpha \in \mathcal{A}_{\square}[\mathcal{S}]$, for some scale factor s and macrotile representation function R .*

Proof. In the aTAM as well as in the syncTAM, any tile attachment takes place thanks to a set of *predecessors*. The set of directions of those predecessors can be (up to rotation) one of $\{N\}, \{N, E\}, \{N, S\}, \{N, E, S\}, \{N, E, S, W\}$. In this proof, these sets of directions are represented as a binary tree, respectively as follows: N , $N \wedge E$, $N \wedge S$, $(N \wedge E) \wedge S$, $(N \wedge E) \wedge (S \wedge E)$. Likewise, for each set I of predecessors, the value of the input glues of the position is a function $i : I \rightarrow G$, will be considered as the binary tree obtained by mapping i on the leaves of the representation of i . For instance, the input mapping glues n, e, s on the N, E , and S , respectively, $\{N \mapsto n, E \mapsto e, S \mapsto s\} : \{N, E, S\} \rightarrow G$, is represented as the binary tree $(n \wedge e) \wedge s$.

Let $\mathcal{S} = (T, \sigma, \tau)$ be a TAS which is directed in the aTAM, and G its set of glues. Let $g = |G|$ be the number of glues in T . The syncTAM system $\mathcal{S}' = (T', \sigma', 1)$ defined below, with set of glues G' , creates a scaled version of $\alpha \in \mathcal{A}_{\square}[\mathcal{S}]$, the terminal assembly of $\mathcal{S}$, in the syncTAM at scale $s = 10g^4 + 1$ with the representation function $R : B_m^{T'} \dashrightarrow T$ also defined below.

Interface between macrotiles and invariants. The set of glues G' of $\mathcal{S}'$ is made of two parts: $E \uplus I$. The set E of *external* glues contains a glue g' (of strength 1) for each $g \in G$ and $d \in \{N, E, S, W\}$. The set I of *internal* glues will be defined below, as the need for those glues arises in the description of the construction. For each tile $t \in T$, there is a unique tile $\hat{t} \in T'$.

Given scale factor s , the macrotiles of $\mathcal{S}'$ that map under R to tiles of $\mathcal{S}$ will be $s \times s$ regions. Let $i^* = \lceil s/2 \rceil + 1$; on each side of a macrotile, the i^* -th position (counting clockwise) is the *input position* in that direction, with the position immediately before it (counting clockwise) being the *output position*. (These are depicted in Figure 7 of the full version of this paper [3] by the arrows pointing into and out of each side.)

There is an invariant associated with the representation of the glues which $\mathcal{S}'$ will uphold:

for any assembly α of $\mathcal{S}'$ where some $s \times s$ square M maps to a tile t under R , there is an α' such that $\alpha \rightarrow_{\mathcal{S}'} \alpha'$ and the sequence $\vec{g}$ of glues (read clockwise) on each side d of M satisfies either $\vec{g}_{i^*} = t_d$ or $\vec{g}_{s-i^*} = t_d$.

That is, every macrotile in $\mathcal{S}'$ that maps to a tile in $\mathcal{S}$ will have (or be able to grow into a macrotile that has) on each side a glue, in either the input or output position, that represents the glue on the corresponding side of the tile in $\mathcal{S}$ that the macrotile maps to.

Because $\mathcal{S}$ is directed in the aTAM, for any attachment of a tile of type t at position z in a production α of $\mathcal{S}$ and any subset of neighbors of z which are filled before z and have glues with binding strength summing to τ or more facing towards z , there must not be any tile other than t that is attachable thanks to this subset of neighbors.

Macrotile Anatomy. Each macrotile is divided into regions. The *input position* is the i^* -th position of each side of the macrotile (counting clockwise), while the *output position* is the $(s - i^*)$ -th. Thus, each input position of a macrotile is aligned with the output position of the neighboring macrotile. For each possible set of inputs I , there is a *computation region*, with room for a *computation zone* for each combination of glues $i \in G^I$. All these zones and regions are disjoint and separated by the *wire region*, to be used for communication between them. As an exception to this rule, the wire region also passes through the computation zones in the $\{N, S\}$ and $\{E, W\}$ computing regions. (These are special cases because they handle the situation when a tile in $\mathcal{S}$ attaches via the cooperation of two input glues across-the-gap, i.e. from N and S or from E and W , as opposed to there being a single input glue or a set of input glues that are all adjacent to each other.) The *center* of the macrotile is the position at $(\lfloor s/2 \rfloor, \lfloor s/2 \rfloor)$; it is outside all regions. The tile there is always one of the unique $\hat{t}$ tiles that corresponds to a tile of T . The representation function is defined by $R(M) = t$ iff the center position of M is occupied by $\hat{t}$.

Input. For each glue $g \in G$ and direction d , there is a set of hard-coded tile types such that from glue g' next to the input position in direction d , a path grows to the computation zone associated with $(d \mapsto g)$ within the d computation region, where it starts the growth of the computation gadget for $(d \mapsto g)$.

Computation gadgets. A computation gadget for a set of inputs I and the values $i \in G^I$ behaves thus: if there is a unique possible attachment in $\mathcal{S}$ with inputs i , say of tile type t , have the macrotile output t (see below for how output proceeds). Otherwise, send i to some computation gadgets having i as a child (within the regions associated with trees having I as a child):

- those with two adjacent inputs
- those which would result in a unique attachment

The computations gadgets for singletons in each direction d are trivial. If glue g has strength at least τ and there is a unique tile type t with glue g on side d , then the gadget is hard-coded to have the macrotile start outputting tile type t (see below). Otherwise, the gadget is hard-coded to forward the input $d \mapsto i$ to the relevant computation gadgets.

For non-singleton sets of input directions, the computation gadget uses a mechanism similar to that of Figure 1 which allows it to wait for both of its inputs to arrive. When both of its inputs have arrived, the behavior is the same as in the singleton case: a unique candidate tile attachment t triggers the output of t , otherwise the input value is transferred to the parent computation gadgets (if there are any).

Output. When a computation gadget logically outputs the attachment of a tile t , it grows a path towards the center of the macrotile and from there to each side, where it passes through the input position before outputting glue t'_d at the output position. The paths from the center to the output positions are designed in such a way that all input and output positions are reached at the same time (i.e. the paths are of the same length, which may require some to “stall” by growing indirectly toward their destinations for short distances).

Wires through the NS and EW computation zones. Wires which need to cross a computation zone within the $\{N, S\}$ or $\{E, W\}$ computation region send two probes, each trying to cross a different input wire of the computation zone then meeting again once they have passed it. The only case where both are blocked is if both inputs are present and they beget a unique tile attachment. In that case, the macrotile is already outputting the value of that attachment (which must be the same value since $\mathcal{S}$ is directed), so the blocking does not prevent correct output.

The claim is that this system builds a scaled version of $\alpha \in \mathcal{A}_{\square}[\mathcal{S}]$. First note that before each output position is reached, all input positions have been blocked. This means that there cannot be a path of attachments between an output position of a macrotile and an input position of the same macrotile.

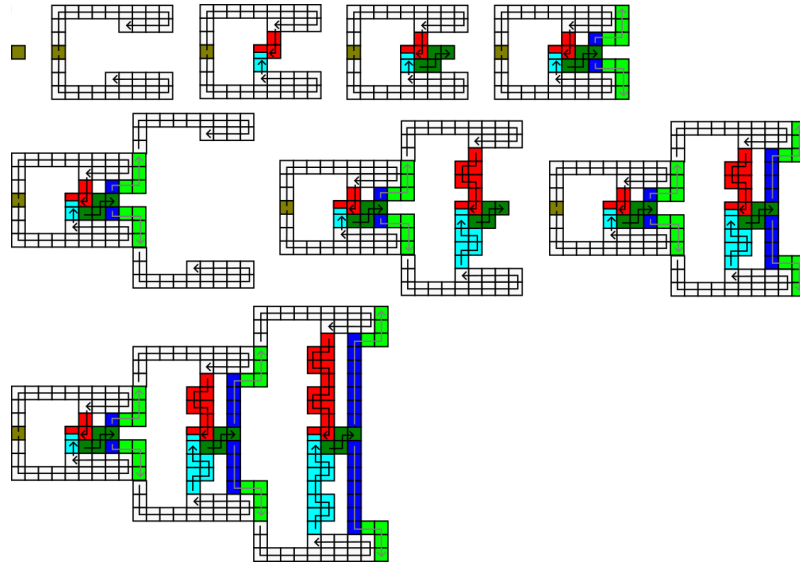
Moreover, conflicts between paths can only arise if two different computing gadgets have found a unique attachment. In this case, because $\mathcal{S}$ is directed, they are racing to output the same tile, so the behavior downstream of the center of the macrotile will be the same as if there was no conflict. It is crucial that $\mathcal{S}$ is directed in the aTAM and not just in the syncTAM, as there might be a race between the macrotile blocking its input positions on output sides and said neighbors outputting a glue there. In $\mathcal{S}$ running in the syncTAM, such a race is always won by whichever sets of inputs gets there first, so it does not preclude directedness. In the aTAM on the other hand, the attachment can be delayed until both sets inputs are complete, so by directedness, they must all agree on the same unique output. Because there cannot be a dependency path from an output position to an input position of the same macrotile location, any sets of inputs of a macrotile of $\mathcal{S}'$ which is involved in a conflict between paths must correspond to a competition between attachments which can happen in the aTAM.

If the seed σ of $\mathcal{S}$ consists of a single tile, then the seed σ' for $\mathcal{S}'$ can be created by simply placing the tile $\hat{t}$ that corresponds to the seed tile type t of σ into the center location of the macrotile representing σ . From that, the output paths will grow to the sides of the macrotile, and then (following the argument above) growth of all subsequent macrotiles will correctly match the growth of the tiles of $\mathcal{S}$ by always growing macrotiles that map, under R , to the tile types in the corresponding locations of the terminal assembly $\alpha \in \mathcal{A}_{\square}[\mathcal{S}]$. If instead σ consists of multiple tiles, then σ' simply consists of the corresponding $\hat{t}$ tiles in the centers of the corresponding macrotile locations, all connected by a path of tiles that cross through and block their adjacent input and output positions. Thus, $\mathcal{S}'$ produces a scaled version of the terminal assembly of $\mathcal{S}$ for scale factor s and macrotile representation function R . ◀

B Proof of Theorem 15

► **Theorem 15.** *For each $l > 1$, there exists a shape S_l such that:*

1. *There exists L -syncTAM system $\mathcal{L}_l = (T, \sigma, 2, l)$ such that $\mathcal{L}_l$ strictly self-assembles S_l .*
2. *For all $l' < l$, there exists no L -syncTAM system $\mathcal{L}' = (T', \sigma', \tau, l')$ and scale factor $c \in \mathbb{Z}^+$ such that $\mathcal{L}'$ strictly self-assembles $c \cdot S_l$.*



■ **Figure 7** A depiction of the growth sequence of the initial part of the **cone** portion of the shape S_5 from Figure 5.

Proof. We prove Theorem 15 by first defining the shape S_l and providing an L-syncTAM system that strictly self-assembles it.

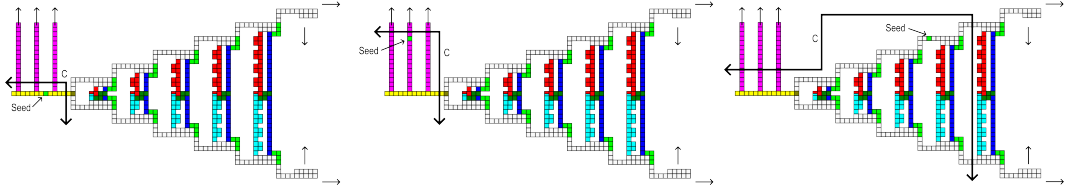
Given some $l > 1$, S_l is an infinite shape consisting of $l - 2$ **arms** that are single-tile-wide columns that each grow upward infinitely, and a **cone** that is a “sideways V” that grows infinitely to the right, with its sides growing further apart and **rungs** periodically growing between the two spreading arms of the V. An initial portion of S_5 (i.e. S_l for $l = 5$) is shown in Figure 5.

The system $\mathcal{L}_l = (T, \sigma, 2, l)$ strictly self-assembles S_l as follows:

1. From the seed, which is a single tile, a set of $4(l - 2)$ tiles grow a row of length $4(l - 2)$ to the right. At each position i to the right of the seed where $(i \bmod 4) == 0$, the yellow tile at that position has a north facing strength-2 glue that attaches to a pink tile type.
2. An infinite number of copies of a single pink tile type attach to the north of the yellow tiles at positions $(i \bmod 4) == 0$ and then to each other, growing $l - 2$ **arms** infinitely upward.
3. To the right of the rightmost yellow tile, a tile of the gold tile type attaches, to initiate growth of the **cone** (whose growth process can be seen in Figure 7).

When growth of S_l begins from the seed, the frontier is initially of size 1. Then, as the yellow row grows and initiates growth of the **arms**, the size of the frontier increases until, when the yellow portion completes, it is $l - 1$ (i.e. one location for each of the $l - 2$ **arms**, and one to the right of the rightmost yellow tile). Then, as the **cone** grows from the gold tile, it consistently has two frontier locations (one on the top half and one on the bottom half) except during the growth of the 4-tile-long green segments in the middle of each **rung**, during which there is 1 frontier location in the **cone**. Thus, once growth of the **cone** has begun, there are always either l frontier locations or $l - 1$ (during the growth of the green segments).

Since there are never more than l frontier locations, the entire frontier can be filled during each assembly step. This ensures that growth of the top and bottom portions of the **cone** can always remain synchronized, and the lengths of the hard-coded sections of the cone are designed so that, even as the rungs grow taller and taller, the red and aqua segments



■ **Figure 8** Examples of the shape S_l from the proof of Theorem 15 being grown in $\mathcal{L}'$ with the seed in different locations and possible cuts (labeled as C in each) used to apply Lemma 11. Note that $\mathcal{L}'$ is allowed to scale S_l to instead self-assemble $c \cdot S_l$, but that doesn't change the location of the cuts or the fact that they still each meet the criteria of Lemma 11 since the scaling only adds a constant amount of thickness across each portion of the cut.

are always guaranteed to collide in the central row. Because of this, the shape S_l is always formed, with bumps extending to the left of the top halves on the left sides of **rungs** and to the right of the bottom halves on the left sides of **rungs**. Therefore, $\mathcal{L}_l$ strictly self-assembles S_l , completing the first portion of the proof.

We now present the second portion of the proof, showing that for all $l' < l$, there exists no L-syncTAM system $\mathcal{L}' = (T', \sigma', \tau, l')$ and scale factor $c \in \mathbb{Z}^+$ such that $\mathcal{L}'$ strictly self-assembles $c \cdot S_l$. We prove this portion by contradiction. Therefore, assume there do exist $l' < l$, $\mathcal{L}' = (T', \sigma', \tau, l')$, and $c \in \mathbb{Z}^+$ such that $\mathcal{L}'$ strictly self-assembles $c \cdot S_l$.

1. $\mathcal{L}'$ is allowed to place its seed anywhere within the shape $c \cdot S_l$. However, since each **arm** and the **cone** are infinite, it must be possible to follow an assembly sequence in $\mathcal{L}'$ that grows portions of each **arm** and the **cone** so that it is possible to make a cut that separates the seed-bearing portion of the assembly from infinite segments of the **arms** and **cone**. (See Figure 8 for examples.)
2. We make a cut C through $c \cdot S_l$ that separates it into sub-shapes $s_0, s_1, \dots, s_{l-1}$, such that s_0 consists of the seed and the yellow segment (plus possibly finite portions of the **arms** and **cone**), s_1 through s_{l-2} each consist of the infinite remainder of one of the **arms** that isn't contained in s_0 , and s_{l-1} consists of the infinite portion of the **cone** that is not contained within s_0 .
3. We now allow $\mathcal{L}'$ to proceed following an arbitrary assembly sequence until the number of steps f , given by Lemma 11 occur. By Lemma 11 it must be the case that, after some assembly step f , each **arm** must always have at least one frontier location, and the **cone** must always have at least one, for a total of at least $l - 1 \geq l'$ frontier locations during all of the subsequent assembly steps (which are infinite in count).
4. From this point on, we choose an assembly sequence, continuing after step f , that proceeds as follows: for every step $f' > f$, a single tile is placed in the **cone**, and $k' - 1$ tiles are placed, each in a unique **arm**.
5. Let g be the number of unique glue types in T' , the tile set of $\mathcal{L}'$. Given the scale factor c at which $\mathcal{L}'$ is self-assembling S_l to achieve $c \cdot S_l$, we let $h = ((g + 1)^{6c} \cdot (6c)! + 1) \cdot 3 + 6$. Justification for this number can be found in the proof of Theorem 1.2 of [21], and it is the count of all possible “window movies” across cuts of the upward or downward growing segments in the locations where they are one (macro)tile wide. This means that this is the maximum possible number of ways that glues can cross a cut across such a segment.
6. We let growth in $\mathcal{L}'$ continue beyond f' until a new period of the **cone** receives its first tile, and then further until the first time after that during which either a red segment grows downward a distance of $2h$ tiles or an aqua segment grows upward a distance of $2h$. This must occur since the **rungs** become arbitrarily tall, and for a **rung** to form, at least one (if not both) of the red and aqua segments must grow at least to the midpoint.

7. We now rewind the assembly sequence until both the red and aqua segments are shorter than $1.5h$ (noting that one of them already may be shorter than that), and at least one is of length $> h$. Since the **cone** is growing by the addition of a single tile during each assembly step, we can apply the Window Movie Lemma (WML) of [21] and the abstract Tile Assembly Model. Without loss of generality, assume that the red segment is of length $> h$. The WML guarantees that since a window movie must have repeated during its growth to the current length, the segment in between the repeated window movies can be “pumped” to create a red segment of arbitrary length (or at least until it crashes into the aqua segment). Since this results in the red segment extending past the midpoint, the shape of the **rung** is invalid in terms of $c \cdot S_l$ with bumps on the left past the midpoint.
8. We can create a new, but valid, assembly sequence that continues to place a single tile in each of $k' - 1$ unique **arms** and a single tile in the red segment of the current period during each step. This results in a **rung** of the shape described, and therefore the self-assembly of a shape that does not match $c \cdot S_l$, which is a contradiction to the statement that $\mathcal{L}'$ strictly self-assembles $c \cdot S_l$. Therefore, no system at synchronicity $l' < l$ can strictly self-assemble S_l at any scale factor, and thus Theorem 15 is proven. ◀