Theoretical Computer Science 894 (2021) 50–78

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the effects of hierarchical self-assembly for reducing program-size complexity

Sarah Cannon [a], Erik D. Demaine [b], Martin L. Demaine [b], Sarah Eisenstat [b], David Furcy [c], Matthew J. Patitz [d,*], Robert Schweller [e], Scott M. Summers [c], Andrew Winslow [f]

[a] Department of Mathematical Sciences, Claremont McKenna College, Claremont, CA 91711, USA
[b] Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar St., Cambridge, MA 02139, USA
[c] Department of Computer Science, University of Wisconsin–Oshkosh, Oshkosh, WI 54901, USA
[d] Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701, USA
[e] Department of Computer Science, University of Texas–Rio Grande Valley, Edinburg, TX, 78539, USA
[f] Google LLC, Mountain View, CA 94043, USA

ARTICLE INFO

Article history:
Received 24 February 2021
Received in revised form 2 September 2021
Accepted 3 September 2021
Available online 10 September 2021

Keywords:
Algorithmic self-assembly
Abstract tile assembly model
Hierarchical self-assembly

ABSTRACT

In this paper we present a series of results which show separations between the standard seeded model of self-assembly, Winfree's abstract Tile Assembly Model (aTAM), and the "seedless" 2-Handed Assembly Model (2HAM), which incorporates the dynamics of hierarchical self-assembly. In particular, we focus on the problem of self-assembling various shapes while minimizing the sizes of tile sets, or "programs", in each of these models in order to compare and contrast the models. A high-level overview of a subset of these results was presented in a paper by the authors in STACS 2013, but in this version we expand and improve the set of results related to showing separations between the two models according to their abilities to self-assemble various shapes. We exhibit classes of finite shapes that can be self-assembled more efficiently in each model. We also demonstrate infinite shapes that can self-assemble in one model but not in the other, as well as a shape which cannot self-assemble in either model.

1. Introduction

Self-assembly is a process by which relatively simple components autonomously combine to form more complex structures. It is a process governed solely by the local interactions among components, but yet is capable of producing extremely complex patterns. *Algorithmic* self-assembly is a more specific process in which the components of a self-assembling system are designed so that their growth into an assembly implicitly follows a prescribed algorithm. Mathematical models such as the abstract Tile Assembly Model (aTAM) [1] provide a platform for the design of algorithmic self-assembling systems, and physical implementations via DNA-based tiles have even been realized [2–8]. Theoretical results have proven that systems within many models capable of algorithmic self-assembly are in fact computationally universal [9–17]. This means that for

* Corresponding author.
E-mail addresses: scannon@cmc.edu (S. Cannon), edemaine@mit.edu (E.D. Demaine), mdemaine@mit.edu (M.L. Demaine), seisenst@mit.edu (S. Eisenstat), furcyd@uwosh.edu (D. Furcy), patitz@uark.edu (M.J. Patitz), robert.schweller@utrgv.edu (R. Schweller), summerss@uwosh.edu (S.M. Summers), andrewwinslow@gmail.com (A. Winslow).

https://doi.org/10.1016/j.tcs.2021.09.011
0304-3975/

any algorithm on any arbitrary input, a self-assembling system can be designed which simulates its computation. Nonetheless, it has also been shown that there remain structures, or shapes, that cannot self-assemble in any system of several models capable of algorithmic self-assembly [18–21], showing a fundamental difference between the ability to compute versus the ability to build structures of particular shapes.

A wide variety of models of tile-based self-assembly have been developed to explore the powers and limitations imparted by different dynamics and component types. Each model can be thought of as a programming language, with the definition of a system in a model as a program written in that language. (The assembly process is thus analogous to the execution of the program.) Theoretically (to gain insight into the expressive power of a model) and experimentally (to minimize the cost, complexity, and potential errors in physical systems), it is beneficial to compare the minimal program size required to form given structures across different models. This is equivalent to minimizing the number of unique types of tiles required to self-assemble target structures.

Perhaps two of the most widely studied theoretical models are the aTAM [1,22] and the 2-Handed Assembly Model (2HAM) [9]. In the aTAM, growth of an assembly begins from a specially designated seed tile[1] and proceeds as tiles attach one at a time to the assembly containing the seed. In contrast, in the 2HAM there is no seed and growth begins when any two singleton tiles can combine to each other, and proceeds whenever any assembly which has already formed (including singleton tiles) can combine with either another singleton tile or another single assembly which has already formed. Allowing growth in which two arbitrarily large assemblies can potentially combine at any step is the reason that the 2HAM is also known as a "hierarchical assembly model". As a generalization of the aTAM, the 2HAM allows for a wider range of dynamics [23,24,20,25,26]. Given the major differences in the models, in [27] a subset of the current authors presented a wide array of results which sought to initiate a quantitative comparison of these two models.

In this paper we focus on, and expand, the subset of those results which compare and contrast the program-size complexities, i.e. the tile complexities, of self-assembling several shapes across the two models. By considering simple and complex shapes, as well as finite and infinite shapes, we are able to demonstrate how different techniques, specific to the dynamics of each model, have the potential to allow for reduced tile complexity.

1.1. *Efficiency via the blocking of growth in the aTAM*

In the aTAM, due to the fact that growth must begin from a designated seed assembly and that it is possible to design systems which grow in a fixed sequential manner, "blocking" may be used as a tool. That is, it can be guaranteed that one portion of an assembly has grown before a later portion, and that later portion then grows in such a way that it "crashes" into the earlier portion. This allows the later-growing portion to save on tile complexity by removing the need for unique tile types which keep track of distance, instead allowing repetitive use of (as few as) a single tile type placed over and over until growth is forced to stop when a copy is eventually placed adjacent to a tile from the blocking portion. We are able to exhibit shapes in which such blocking gives the aTAM the advantage in tile complexity, first by just a constant factor, and then asymptotically (by an exponentially large amount).

1.2. *Efficiency via the encoding of information in distance and shape in the 2HAM*

Growth in the 2HAM is accomplished by the repeated combination of pairs of assemblies (called *supertiles*), with the smallest being singleton tiles. Either or both of the pairing supertiles can be of any arbitrary (large but finite) size. While different supertiles may have the same tiles and glues exposed on their perimeters, it may be the case that the shapes of their perimeters and/or the relative positions of sets of their exposed glues differ. If the minimum binding threshold, a.k.a. temperature parameter, is > 1, multiple glues on the perimeter of a supertile may be required for it to bind with another, and careful design may be able to ensure that although the set of perimeter glues is the same for multiple supertiles, their relative distances from each other uniquely distinguish to which other supertiles they are able to bind. In this way, the constant amount of information contained within a fixed set of glues can be implicitly combined with an arbitrary amount of information about the relative glue positions, allowing for the design of systems in which combinations are restricted to only pairs of supertiles that have grown to specifically targeted sizes and/or shapes. Additionally, even in systems with a minimum binding threshold of 1, it may be possible to use the shapes of the perimeters and potential geometric hindrance (rather than multiple glue locations) to differentiate between valid and invalid binding partners. These techniques can be leveraged to design 2HAM systems which are much more efficient, in terms of tile complexity, than any possible aTAM system. We exhibit such finite shapes, and then show how to further leverage the technique to demonstrate an infinite shape which cannot self-assemble in the aTAM, but which can self-assemble in the 2HAM.

1.3. *Our results*

We utilize those design techniques specific to the aTAM and 2HAM to demonstrate how each can provide advantages for self-assembling differing shapes. Specifically, our main results are the following (see Tables 1 and 2 for more details):

[1] The seed is actually allowed to be an assembly consisting of multiple tiles. However, throughout this paper we will only consider systems whose seeds consist of a single tile in order to provide a fair basis of comparison with the 2HAM.

Table 1
Summary of results showing separation between the aTAM and 2HAM with respect to tile complexity of certain classes of finite shapes. The value of a cell denotes the tile complexity. Note that some of our results are asymptotic while others are exact complexities. Under the "Staircases" column, M is a TM, with state set Q, given the empty string as input. The idea is that M can be chosen such that it runs for a number of steps m, which is much greater than a given value of n yet $|Q|$ can be much smaller than n.

	Loops		Counter-with-crasher	Staircases
	$\tau = 1$	$\tau = 2$	$\tau = 2$	$\tau = 2$
aTAM	$n + 5$ (Theorem 3.2)	$n + 3$ (Theorem 3.2)	$O(n)$ (Theorem 3.11)	2^n stair steps: $\Omega(n)$, $O(n^2)$ (Theorem 3.13), (Theorem 3.18)
2HAM	$2n + 2$ (Theorem 3.2)	$\leq n + 3$ (Theorem 3.2)	$\Omega(2^n)$ (Theorem 3.11)	$2^{\Omega(\text{running time of } M)}$ stair steps: $O(|Q|)$ (Theorem 3.19)

Table 2
Summary of results showing separation between the aTAM and 2HAM for some examples of infinite shapes. We say that $\mathcal{T}$ *finitely self-assembles* an infinite shape, say X, if every finite producible assembly of $\mathcal{T}$ can grow into the desired target shape (the term *finite self-assembly* is defined formally in Section 2).

	Infinite staircase		Sierpinski triangle	
	Finite self-assembly	Self-assembly	Finite self-assembly	Self-assembly
aTAM	No (Theorem 4.3)	No (Corollary 4.7)	No (Thm. 3.4 of [18])	No (Thm. 3.4 of [18])
2HAM	Yes ($\tau = 2$) (Theorem 4.2)	Open	No (Theorem 4.9)	No (Corollary 4.12)

1. There is a simple shape (a loop) that can be assembled in the aTAM at temperature $\tau = 1$ using $n + 5$ unique tile types, but any 2HAM system in which the shape assembles at the same temperature requires $2n + 2$ unique tile types, giving a constant factor separation in favor of the aTAM. At temperature $\tau = 2$, the same shape can be assembled in both models using $n + 3$ tile types, erasing that advantage.
2. There is a shape (a "counter-with-crasher") that can self-assemble in the aTAM using only $O(n)$ tile types, but which requires $\Omega(2^n)$ tile types in the 2HAM. This demonstrates an asymptotic advantage in the aTAM.
3. There is a shape (a finite "staircase") that can be built in the 2HAM using a number of unique tile types only asymptotically dependent upon the size of a chosen Turing machine, but in the aTAM the same shape requires a number of unique tile types proportional to the running time of the chosen TM. With $BB(n)$ as the busy beaver function, this allows for a tile complexity separation of $BB(n)$, and demonstrates a (large) asymptotic advantage in the 2HAM. (Note that the counter-with-crasher can also be trivially modified to give a $BB(n)$ separation advantage for the aTAM.)
4. There is an infinite shape (an infinite staircase) that can self-assemble (in a weaker sense) in the 2HAM but not in the aTAM.
5. There is an infinite shape (the Sierpinski triangle) that can neither self-assemble in the aTAM nor the 2HAM.[2]

The structure of this paper is as follows. In Section 2 we introduce notation and give definitions of the aTAM and 2HAM. In Section 3 we present tile complexity results related to finite shapes, and in Section 4 we present results related to infinite shapes. Section 5 provides a brief conclusion.

2. Preliminaries and notation

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

Intuitively, a tile type t is a unit square that can be translated, but not rotated, having a well-defined "side $\vec{u}$" for each $\vec{u} \in U_2$. Each side $\vec{u}$ of t has a "glue" with "label" $\text{label}_t(\vec{u})$–a string over some fixed alphabet–and "strength" $\text{str}_t(\vec{u})$–a nonnegative integer–specified by its type t. Two tiles t and t' that are placed at the points $\vec{a}$ and $\vec{a} + \vec{u}$ respectively, *bind* with *strength* $\text{str}_t(\vec{u})$ if and only if $\left(\text{label}_t(\vec{u}), \text{str}_t(\vec{u})\right) = \left(\text{label}_{t'}(-\vec{u}), \text{str}_{t'}(-\vec{u})\right)$.

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

[2] Previous work in [18] gives the impossibility result for the aTAM.

Fix a finite set T of tile types. A *T-assembly*, sometimes called simply an *assembly* when T is clear from the context, is a partial function $\alpha : \mathbb{Z}^2 \dashrightarrow T$ defined on at least one input, with points $\vec{x} \in \mathbb{Z}^2$ at which $\alpha(\vec{x})$ is undefined interpreted to be empty space, so that $\mathrm{dom}\,\alpha$ is the set of points with tiles. We write $|\alpha|$ to denote $|\mathrm{dom}\,\alpha|$, and we say α is *finite* if $|\alpha|$ is finite. For assemblies α and α', we say that α is a *subassembly* of α', and write $\alpha \sqsubseteq \alpha'$, if $\mathrm{dom}\,\alpha \subseteq \mathrm{dom}\,\alpha'$ and $\alpha(\vec{x}) = \alpha'(\vec{x})$ for all $x \in \mathrm{dom}\,\alpha$.

Each assembly α induces a *binding graph* G_α, a grid graph whose vertices are the points with tiles, with an edge between two points if the tiles at those locations bind in α and the weight of an edge is the strength of the bond it represents. For $\tau \in \mathbb{N}$, an assembly is *τ-stable* if every cut of its binding graph has weight at least τ. That is, the assembly is stable if at least energy τ is required to separate it into two parts. In contrast to the model of Wang tiling [28], the nonnegativity of the strength function implies that glue mismatches between adjacent tiles do not prevent a tile from binding to an assembly, so long as sufficient binding strength is received from the (other) sides of the tile at which the glues match. The binding graph is merely a structure that characterizes the bonds between tiles in an assembly and not the order in which the bonds formed during an assembly sequence.

2.1. Informal description of the abstract tile assembly model (aTAM)

In this section we give an informal description of the aTAM which includes the necessary definitions for this paper. The reader is encouraged to see [22,1,18] for a formal development of the model.

In the aTAM, self-assembly begins with a *seed assembly* σ (typically assumed to be finite and τ-stable) and proceeds asynchronously and nondeterministically, with tiles adsorbing one at a time to the existing assembly in any manner that preserves stability at all times.

An aTAM *tile assembly system* (*TAS*) is an ordered triple $\mathcal{T} = (T, \sigma, \tau)$, where T is a finite set of tile types, σ is a seed assembly with finite domain, and τ is the temperature, a.k.a. minimum binding threshold. An *assembly sequence* in a TAS $\mathcal{T} = (T, \sigma, \tau)$ is a sequence $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ (where $k = \infty$ if it is an infinite assembly sequence) of assemblies in which $\alpha_0 = \sigma$ and each α_{i+1} is obtained from α_i by the "τ-stable" addition of a single tile. The *result* of an assembly sequence $\vec{\alpha}$ is the unique assembly $\mathrm{res}(\vec{\alpha})$ satisfying $\mathrm{dom}\,\mathrm{res}(\vec{\alpha}) = \bigcup_{0 \le i < k} \mathrm{dom}\,\alpha_i$ and, for each $0 \le i < k$, $\alpha_i \sqsubseteq \mathrm{res}(\vec{\alpha})$.

For an assembly sequence $\vec{\alpha}$ in some TAS with result α and a point $\vec{x} \in \mathrm{dom}\,\alpha$, we define the notation $i_{\vec{\alpha}}(\vec{x})$ to represent the index in the assembly sequence $\vec{\alpha}$ at which the point $\vec{x}$ has a tile placed on it.

We write $\mathcal{A}[\mathcal{T}]$ for the *set of all producible assemblies of* $\mathcal{T}$. An assembly α is *terminal*, and we write $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, if no tile can be stably added to it. We write $\mathcal{A}_\Box[\mathcal{T}]$ for the *set of all producible terminal assemblies of* $\mathcal{T}$. A TAS $\mathcal{T}$ is *directed*, or *produces a unique assembly*, if it has exactly one terminal assembly i.e., $|\mathcal{A}_\Box[\mathcal{T}]| = 1$. The reader is cautioned that the term "directed" has also been used for a different, more specialized notion in self-assembly [29]. We interpret "directed" to mean "deterministic", though there are multiple senses in which a TAS may be deterministic or nondeterministic.

Given a connected shape $X \subseteq \mathbb{Z}^2$, we say a TAS $\mathcal{T}$ *self-assembles* X if every producible, terminal assembly places tiles exactly on those positions in X. (Note that this notion is equivalent to *strict* self-assembly as defined in [18].) For an infinite shape $X \subseteq \mathbb{Z}^2$, we say that $\mathcal{T}$ *finitely self-assembles* X if every finite producible assembly of $\mathcal{T}$ has a possible way of growing into an assembly that places tiles exactly on those points in X. Note that if $\mathcal{T}$ self-assembles shape X, then $\mathcal{T}$ finitely self-assembles X (but not necessarily vice versa). Also, it is important to note the difference between directedness and (finite) self-assembly of a shape. If a system (finitely) self-assembles a shape, all terminal assemblies have that shape. However, the system may have multiple assembly sequences which lead to tiles of different types in the same location. Such a system would not be directed.[3]

2.2. Two-handed tile assembly model (2HAM)

The 2HAM [17,9,30,31,12,32] is a generalization of the aTAM in that it allows for two assemblies, both possibly consisting of more than one tile, to attach to each other. Since we must allow that the assemblies might require translation before they can bind, we define a *supertile* to be the set of all translations of a τ-stable assembly, and speak of the attachment of supertiles to each other, modeling that the assemblies attach, if possible, after appropriate translation.

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

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

[3] A trivial example is a system of four tile types, S, A, B, and C where S is the seed, the temperature is 1, and A binds to the east of S, and both B and C are able to bind to the east of A. In this case, there are exactly two terminal assemblies SAB and SAC so the system is not directed. However, since all terminal assemblies have the shape of a 3×1 rectangle, the system self-assembles that shape.

translation preserves cardinality (and note in particular that even though we define $\widetilde{\alpha}$ as a set, $|\widetilde{\alpha}|$ does not denote the cardinality of this set, which is always considered to be $\aleph_0$).

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

It is common with seeded assembly to stipulate an infinite number of copies of each tile, but our definition allows for a finite number of tiles as well. Our definition also allows for the growth of infinite assemblies and finite assemblies to be captured by a single definition, similar to the definitions of [18] for seeded assembly.

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

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

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

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

The *result* of a supertile assembly sequence $\vec{\widetilde{\alpha}}$ is the unique supertile $\mathrm{res}(\vec{\widetilde{\alpha}})$ such that there exist an assembly $\alpha \in \mathrm{res}(\vec{\widetilde{\alpha}})$ and, for each $0 \le i < k$, assemblies $\alpha_i \in \widetilde{\alpha}_i$ such that $\mathrm{dom}\, \alpha = \bigcup_{0 \le i < k} \mathrm{dom}\, \alpha_i$ and, for each $0 \le i < k$, $\alpha_i \sqsubseteq \alpha$. Recall that if α is infinite, then $k = \infty$. For all supertiles $\widetilde{\alpha}, \widetilde{\beta}$, we write $\widetilde{\alpha} \to_{\mathcal{T}} \widetilde{\beta}$ (or $\widetilde{\alpha} \to \widetilde{\beta}$ when $\mathcal{T}$ is clear from context) to denote that there is a supertile assembly sequence $\vec{\widetilde{\alpha}} = (\widetilde{\alpha}_i \mid 0 \le i < k)$ such that $\widetilde{\alpha}_0 = \widetilde{\alpha}$ and $\mathrm{res}(\vec{\widetilde{\alpha}}) = \widetilde{\beta}$. It can be shown using the techniques of [33] for seeded systems that for all two-handed tile assembly systems $\mathcal{T}$ supplying an infinite number of each tile type, $\to_{\mathcal{T}}$ is a transitive, reflexive relation on supertiles of $\mathcal{T}$. We write $\widetilde{\alpha} \to^1_{\mathcal{T}} \widetilde{\beta}$ (or $\widetilde{\alpha} \to^1 \widetilde{\beta}$) to denote an assembly sequence of length 1 from $\widetilde{\alpha}$ to $\widetilde{\beta}$.

A supertile $\widetilde{\alpha}$ is *producible*, and we write $\widetilde{\alpha} \in \mathcal{A}[\mathcal{T}]$, if it is the result of a nascent supertile assembly sequence. A supertile $\widetilde{\alpha}$ is *terminal* if, for all producible supertiles $\widetilde{\beta}$, $C^{\tau}_{\widetilde{\alpha},\widetilde{\beta}} = \varnothing$.[5] Define $\mathcal{A}_{\Box}[\mathcal{T}] \subseteq \mathcal{A}[\mathcal{T}]$ to be the set of terminal and producible supertiles of $\mathcal{T}$. $\mathcal{T}$ is *directed* (a.k.a., *deterministic*, *confluent*) if $|\mathcal{A}_{\Box}[\mathcal{T}]| = 1$.

Let $X \subseteq \mathbb{Z}^2$ be a shape. We say $\mathcal{T}$ *self-assembles* X if, for each $\widetilde{\alpha} \in \mathcal{A}_{\Box}[\mathcal{T}]$, there exists $\alpha \in \widetilde{\alpha}$ such that $\mathrm{dom}\, \alpha = X$; i.e., $\mathcal{T}$ uniquely assembles into the shape X. For an infinite shape $X \subseteq \mathbb{Z}^2$, we say that $\mathcal{T}$ *finitely self-assembles* X if, for each finite $\widetilde{\alpha} \in \mathcal{A}[\mathcal{T}]$, there exists $\alpha \in \widetilde{\alpha}$ such that $\mathrm{dom}\, \alpha \subset X$ and $\widetilde{\alpha} \to_{\mathcal{T}} \widetilde{\alpha}'$ where $\alpha' \in \widetilde{\alpha}'$ and $\mathrm{dom}\, \alpha' = X$.

3. Finite shapes

In this section, we examine classes of finite shapes that can be produced using fewer unique tile types in one self-assembly model (either the aTAM or the 2HAM) than in the other. We first define some related notation.

Given a shape $X \subseteq \mathbb{Z}^2$, we denote by $\mathcal{K}^{\tau}_{\mathrm{aTAM}}(X)$ is the *tile complexity* of X in the aTAM at *temperature* $\tau \in \mathbb{N}$. In other words, $\mathcal{K}^{\tau}_{\mathrm{aTAM}}(X) = \min\{|T| \mid \mathcal{T}$ self-assembles X for some $\mathcal{T} = (T, \sigma, \tau)$ where $|\sigma| = 1\}$. Intuitively, $\mathcal{K}^{\tau}_{\mathrm{aTAM}}(X)$ is the size of the smallest tile set that at temperature τ produces assemblies that place tiles on–and only on–the target shape X in some aTAM system. Let $\mathcal{K}_{\mathrm{aTAM}}(X) = \min\left\{\mathcal{K}^{\tau}_{\mathrm{aTAM}}(X) \,\middle|\, \tau \in \mathbb{N}\right\}$. The quantities $\mathcal{K}^{\tau}_{\mathrm{2HAM}}(X)$ and $\mathcal{K}_{\mathrm{2HAM}}(X)$ are defined analogously.

[4] with the convention that $\infty = \infty + 1 = \infty - 1$.

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

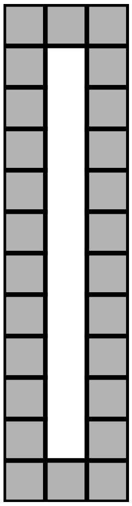


Fig. 1. A loop of size 12.

3.1. Loops

In this subsection, we study the tile complexity of simple loop shapes in the aTAM and 2HAM, defined formally as follows.

Definition 3.1. For any $n \in \mathbb{N}$ such that $n > 2$, define $L_n = (\{0\} \times \{0, \ldots, n-1\}) \cup (\{2\} \times \{0, \ldots, n-1\}) \cup \{(1, 0), (1, n-1)\}$. Intuitively, the set L_n is a "loop of size n." See Fig. 1 for an example.

The first question that we study is: can 2HAM tile assembly systems uniquely produce loops using fewer unique tile types than aTAM tile assembly systems? We will prove that the answer to this question is "no", for possible temperature values $\tau \in \{1, 2\}$.

Throughout this subsection, we *do not* assume that the single seed tile is placed at the origin, nor do we assume that any tile assembly system is directed. Here is our first main theorem.

Theorem 3.2. *For all $n \in \mathbb{N}$ such that $n > 3$, the following hold.*

1. $\mathcal{K}^1_{aTAM}(L_n) = n + 5 < 2n + 2 = \mathcal{K}^1_{2HAM}(L_n)$
2. $\mathcal{K}^2_{2HAM}(L_n) \leq n + 3 = \mathcal{K}^2_{aTAM}(L_n)$

We break the claims of Theorem 3.2 into Lemmas 3.3, 3.5, 3.6, 3.7 and 3.10 in order to prove each of them.

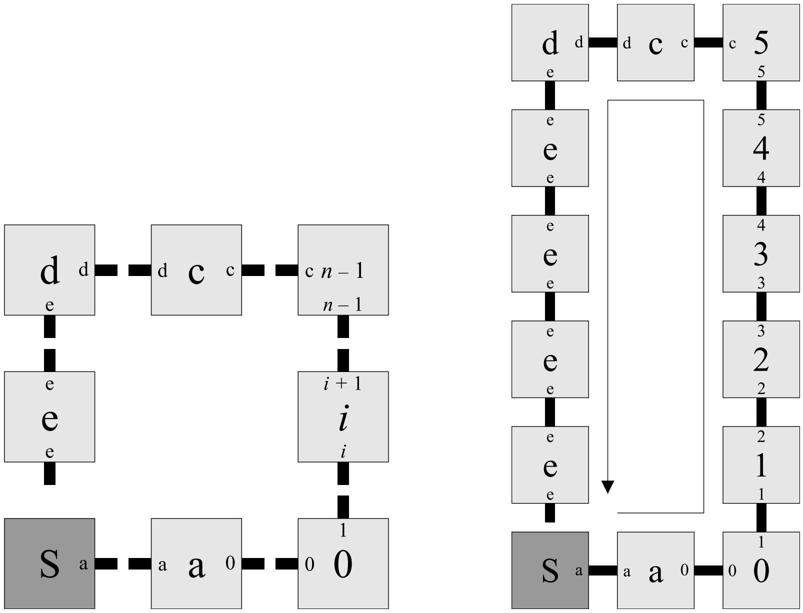
Lemma 3.3. *For all $n \in \mathbb{N}$ such that $n > 2$, $\mathcal{K}^1_{aTAM}(L_n) \leq n + 5$.*

Proof. To see that $\mathcal{K}^1_{\text{aTAM}}(L_n) \leq n + 5$, define the TAS $\mathcal{T}_n = (T_n, \sigma, 1)$, where T_n consists of the tile types given in Fig. 2a. It is easy to see that $\mathcal{T}_n$ uniquely produces the set L_n. Intuitively, starting from the seed 'S', the bottom, right side, and top of the loop assemble from $n + 4$ tile types. Then, since 'S' as the bottom left corner of the loop is guaranteed to already be in place, the left side assembles from a repeating path of 'e' tile types, namely $n - 2$ copies of 'e', until the downward growing column runs into and gets blocked by 'S'. Further copies of 'e' are thus blocked from attaching, making the assembly (uniquely) terminal. □

Before we prove a matching lower bound, we need some additional machinery to simplify reasoning about the self-assembly of loops.

Lemma 3.4. *Let $n \in \mathbb{N}$ such that $n > 2$. If $\mathcal{T} = (T, \sigma, 1)$ in the aTAM self-assembles L_n, then the tiles that $\mathcal{T}$ places at the positions $C = \{(0, 0), (1, 0), (2, 0), (0, n-1), (1, n-1), (2, n-1)\}$ are unique for all terminal assemblies α of $\mathcal{T}$. That is, for any given $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, for every $\vec{x} \in C$, $\left|\{\vec{y} \in \text{dom}\,\alpha \mid \alpha(\vec{y}) = \alpha(\vec{x})\}\right| = 1$. This is also true for any 2HAM TAS $\mathcal{T} = (T, 1)$.*

We call the sets of positions $C = \{(0, 0), (1, 0), (2, 0)\}$ and $\{(0, n-1), (1, n-1), (2, n-1)\}$ the *bottom* and *top* caps of L_n, respectively. Lemma 3.4 follows by a straightforward case analysis (provided below), since the reuse of any tile type from



(a) For $2 < n \in \mathbb{N}$, let $i \in \{1, \ldots, n-2\}$. (b) A loop of size 6.

Fig. 2. Construction for $\mathcal{K}^1_{\text{aTAM}}(L_n) \leq n + 5$.

a cap location in a different location would necessarily allow for the attachment of a tile to the reused cap tile type but which is outside of the intended loop shape. (Note that, as shown in Fig. 3, Lemma 3.4 does not hold at temperature $\tau = 2$.)

Proof. Proof of Lemma 3.4:

1. For any terminal assembly α it must be the case that either (1) every pair of adjacent tiles are connected by a glue bond, or (2) all except one pair of adjacent tiles are connected by a glue bond. This follows immediately from the fact that if more than two adjacent tiles are not connected by a bond, then α isn't τ-stable (in this case $\tau = 1$) and thus is not a valid producible assembly.
2. Given the fact that there can at most be a single pair of adjacent tiles which do not have glue bonds between them, we will inspect all possible cases in which two tiles of the same type might be used in two distinct locations in C, and show that each of them can lead to an assembly sequence where a tile can be placed outside of L_n. This suffices to prove Lemma 3.4 since it must be the case that all terminal assemblies are of shape L_n. In each of the following cases, we will let $\vec{x}$ and $\vec{y}$ refer to the two locations in C in which a tile of the same type is being placed. Of the 6 positions in C, 4 of them are in corners of L_n, and 2 are in the middle of a side (the top or bottom). We will call these *corner* and *middle* tiles, respectively. All of the remaining tiles of L_n are those in the left and right sides between the caps, and we will call these *side* tiles.
   The cases are the following:
   (a) Location $\vec{x}$ is in a corner and is bound to only a single neighbor in α: Suppose that the tile in location $\vec{y}$ is bound to both neighbors in α. Then, the locations of those neighbors relative to $\vec{y}$ must be different than the relative locations of the neighbors of $\vec{x}$ (since every corner is unique). Therefore, the tile in $\vec{x}$ must have a glue exposed in a direction facing a location outside of L_n, and a tile of the same type which attaches to that glue from $\vec{y}$ can attach. This places a tile outside of L_n, so this case cannot be. If the tile in location $\vec{y}$ is not bound to both neighbors, then it must be adjacent to location $\vec{x}$, which means the tile in location $\vec{y}$ is in a middle location. But a tile in a middle location cannot be the same type as that of a tile in a corner, since every tile in a corner is unique.
   (b) Location $\vec{x}$ is in a corner and is bound to both neighbors in α: No matter where location $\vec{y}$ is in L_n, one of the glues which binds to a neighbor of $\vec{x}$ must be exposed to a location adjacent to $\vec{y}$ that is outside of L_n. This follows immediately from the fact that every corner location has a unique pair of sides with adjacent tiles, and all of those pairs differ from those of the middle and side tiles as well. Therefore, for whichever glue is thus exposed from the tile in location $\vec{y}$, a tile of the same type can attach as attaches to that glue from $\vec{x}$. This places a tile outside of L_n, so this case cannot be.
   (c) Location $\vec{x}$ is in a middle position and the tile placed there is bound to only a single neighbor in α: Since there can only be one unbound pair of neighboring tiles, the tile in location $\vec{y}$ must be bound to both neighbors and also $\vec{y}$ is a middle location (since $\vec{x}$ is a middle location). Note that, if the tile in location $\vec{y}$ was only bound to one neighbor,

then that would make it adjacent to $\vec{x}$ and therefore in a corner, which is a possibility we previously ruled out. So, w.l.o.g., assume that the unbound tiles are the top middle and its neighbor to the west. There are two possibilities:

i. The seed is a left corner, in the left side, or the bottom middle: It is possible to grow directly from the seed to the middle bottom and place that tile, which is the same type as the tile placed at $\vec{x}$. Then, the top right corner tile that attaches to the middle top can also attach to the right of the bottom middle. However, then the tile which binds to the south of the top right corner tile can bind below, and that is outside of L_n, so this case cannot be.

ii. The seed is a right corner, in the right side, or the top middle: It is possible to grow directly from the seed to the middle top and place that tile. Then, the bottom left corner tile that attaches to the middle bottom can also attach to the left of the top middle. However, then the tile which binds to the north of the bottom left corner tile can bind above, and that is outside of L_n, so this case cannot be.

(d) Location $\vec{x}$ is in a middle position and the tile placed there is bound to both neighbors in α: The previous case handled when the placed at one of the middle locations is bound to only a single neighbor, so here we can assume that the tile in $\vec{y}$ is also in a middle location and bound to both neighbors. Additionally, at least three corner tiles must be attached to both neighbors (again since there can only be one pair of unbound neighbors). W.l.o.g., assume that the top corners are each attached to two neighbors. No matter where the seed is, it must be possible to grow directly from the seed to the middle bottom tile and then place it before a tile has been placed on its other side (recall that it was bound to two tiles). W.l.o.g., we'll assume it's currently unbound side is the east side. Now, the corner tile, which bound to the east side of the top middle in α, can bind to the bottom middle. Since it was bound to two tiles, it must have a glue to its south. A tile of the same type which bound to its south in α can now bind, but that is outside of (i.e. below) L_n, so this final case also cannot be.

Since all cases where a tile in C could be reused can result in assemblies not in the shape of L_n, no tiles in C may be reused. Thus, Lemma 3.4 is proved. □

The following lemma proves that $n+5$ is a tight bound on $\mathcal{K}^1_{aTAM}(L_n)$.

Lemma 3.5. *For all $n \in \mathbb{N}$ such that $n > 2$, $\mathcal{K}^1_{aTAM}(L_n) \geq n+5$.*

Proof. Assume the aTAM system $\mathcal{T} = (T, \sigma, 1)$ self-assembles L_n. By Lemma 3.4, $|T| \geq 6$. Let $L = \{0\} \times \{1, \ldots, n-2\}$ and $R = \{2\} \times \{1, \ldots, n-2\}$. One of the following must be true of the seed:

1. The seed tile is located in a position in one of the caps, i.e. in C.
   (a) If so, it must be possible to completely grow either L or R (w.l.o.g. assume R) from cap to cap without growing the other (otherwise growth could never get to the other cap).
   (b) Then every tile type in R must be unique. Otherwise, a tile of the same type, say A, must appear in more than one location. Let $\vec{\alpha}_A$ be the assembly subsequence consisting of the placement of the first tile placed after the first tile of type A, followed by each tile up to and including the placement of the second tile of type A. Since R can grow from the side of the cap containing the seed completely before L (by assumption), there exists an assembly sequence such that there are no tiles in place to block the growth of an assembly which grows with an arbitrary number of repetitions of $\vec{\alpha}_A$ (we say $\vec{\alpha}_A$ is *pumped*). That is, after the first placement of a tile of type A, $\vec{\alpha}_A$ can occur, terminating in another A tile. This, and arbitrarily many subsequent occurrences allow yet another copy of $\vec{\alpha}_A$ to occur. Based only on the current assumptions, this results in a system which can produce an infinite assembly, but it may also be able to produce either an infinite number of uniquely shaped finite assemblies or an infinite number of uniquely shaped infinite assemblies. However, we need only note that $n+1$ repetitions of $\vec{\alpha}_A$ creates a subassembly which is taller than L_n and thus can never be a subassembly of L_n (regardless of whatever else may grow), so $\mathcal{T}$ is capable of making (at least one) assembly of a shape other than L_n and so doesn't self-assemble L_n. Thus, every tile type in R must be unique.
2. The seed tile is in L or R (w.l.o.g. assume L). Then, one of the following must be true:
   (a) R can be grown completely by starting from one cap, and thus all tile types in R are unique (following the pumping argument above), or
   (b) Portions of R can grow from each cap. Neither portion can have a repeated tile type, or there would exist an assembly sequence where that portion grows from the seed before the other side grows, and the repeated portion could then be pumped. Additionally, neither portion can use a tile type used by the other, or it would be possible to grow (at least a portion of) one of the caps in the wrong location. This is because if both the upper and lower portions use a tile of the same type, say t_a, then it must be possible for the subassembly that grows the bottom cap then R up to t_a to grow downward from the upper placement of t_a. Since the upper placement of t_a is higher than the lower placement, this will place the bottom cap at a location which is too high. Therefore, all tile types in R must be unique.

So far, we have shown that all tile types of the caps as well as those in R (w.l.o.g.) are unique, thus the tile complexity is $\geq 6+(n-2)=n+4$. To prove Lemma 3.5, we must show that at least one more unique tile type is required, and therefore that the exact same tile types must not be used in L and R. That is, L must use at least one tile type not used by R.

Therefore, assume the opposite, i.e. L uses only tiles of one or more types found in R. Let $\vec{x}$ be some location in L at which tile type t is placed and $\vec{y}$ be some location in R at which tile type t is also placed. If it's possible to either (1) grow only a direct path from the seed to $\vec{x}$ and then to $\vec{y}$, or (2) grow only a direct path from the seed to $\vec{y}$ and then to $\vec{x}$, then we will show that it would be possible to place a tile outside of L_n. Without loss of generality, assume the first. Note that (in either case) when a tile of type t is placed at $\vec{x}$ it must initially bind via the input side that is opposite of the input side when placed at $\vec{y}$ since growth into, then out of, the cap between $\vec{x}$ and $\vec{y}$ flips the direction of growth of the path. Starting from the seed then growing to the first placement of t at $\vec{x}$, it must then be possible (i.e. a valid assembly sequence) to grow the same path that is the reverse of that from $\vec{x}$ to $\vec{y}$ (since both locations have a tile of type t). This causes the path from the seed to $\vec{x}$ to then turn left before getting to the next copy of t rather than right. For the entire shape L_n to have originally grown, it must be that either from $\vec{x}$ there is a path into the bottom cap and to the right (part of which includes the seed) or from $\vec{y}$ there is a path into the bottom cap and to the left. If the former, then growing the rest of that path causes there to be a left turn at the top of L and a right turn at the bottom, which makes a shape not equal to L_n. If the latter, then growing that portion of a path from the end of the second placement of t also causes two opposing turns, also making a shape other than L_n.

Therefore, the seed must be located between $\vec{x}$ and $\vec{y}$ in the sense that there is a direct path from the seed to $\vec{x}$ and another direct path from the seed to $\vec{y}$, where both paths are disjoint from each other. Note that, in this case, when a tile of type t is placed at $\vec{x}$ or $\vec{y}$, it must bind via the same input side in both cases (either from the north, or from the south). It must be possible to extend at least one of these paths up (or down) to a location in a cap. If neither path could be extended in such a way, then the assembly would not be stable. Suppose (w.l.o.g.) that it is possible to extend the path from the seed to $\vec{x}$ to a location in a cap. Take this path extension and use it to extend the path from $\vec{y}$. Such an extension from $\vec{y}$ is valid because a tile of type t is placed at both $\vec{x}$ and $\vec{y}$ and $\vec{x}$ is not contained in a cap. Moreover, such an extension from $\vec{y}$ will result in either (1) the placement of a tile outside of L_n, or (2) the placement of a duplicate tile type within a cap, which violates Lemma 3.4. □

Lemmas 3.3 and 3.5 prove the first equality of part 1 of Theorem 3.2. We now turn our attention to showing the second equality of part 1 of Theorem 3.2.

Lemma 3.6. *For all $n \in \mathbb{N}$ such that $n > 2$, $\mathcal{K}^1_{2HAM}(L_n) = 2n+2$.*

Proof. First, we note that a 2HAM system can be trivially designed to self-assemble L_n by creating a unique tile type for each of its $2n+2$ locations, and thus $\mathcal{K}^1_{2HAM}(L_n) \leq 2n+2$. We now show the lower bound.

Let $\mathcal{T}=(T,1)$ be any 2HAM system in which L_n self-assembles, with terminal assembly α. Lemma 3.4 says that $|T| \geq 6$. Define $L=\{0\}\times\{1,\dots,n-2\}$ and $R=\{2\}\times\{1,\dots,n-2\}$. Then we have $\left|\left\{\alpha\left(\vec{x}\right) \middle| \vec{x}\in L\right\}\right| = n-2$, i.e., every tile that α places in L is unique within L. This is true because, if α placed the same tile type at two distinct locations in L, it would be possible to form an arbitrarily tall assembly consisting of copies of the repeated tile and the tiles in between (if any), copied arbitrarily. Such an assembly could be taller than the loop, contradicting that $\mathcal{T}$ self-assembles the loop. The same is true for R. Thus, we have $|T| \geq 6+n-2=n+4$.

Now suppose that there exist points, say $\vec{a}\in L$ and $\vec{b}\in R$ such that $\alpha\left(\vec{a}\right)=\alpha\left(\vec{b}\right)$. Since $\alpha\left(\vec{a}\right)=\alpha\left(\vec{b}\right)$, it is possible to construct a new connected path of tiles by swapping portions of the assembly which attach above and below the tiles in $\vec{a}$ and $\vec{b}$, to get an example of an assembly $\alpha' \in \mathcal{A}[\mathcal{T}]$ such that α' cannot grow into an assembly α with $\operatorname{dom}\alpha = L_n$ (the path will either extend too far horizontally, or will not complete the loop). Thus, we have $\left|\left\{\alpha\left(\vec{x}\right) \middle| \vec{x}\in L\cup R\right\}\right| = 2n-4$ and $|T| \geq 6+2n-4=2n+2$. □

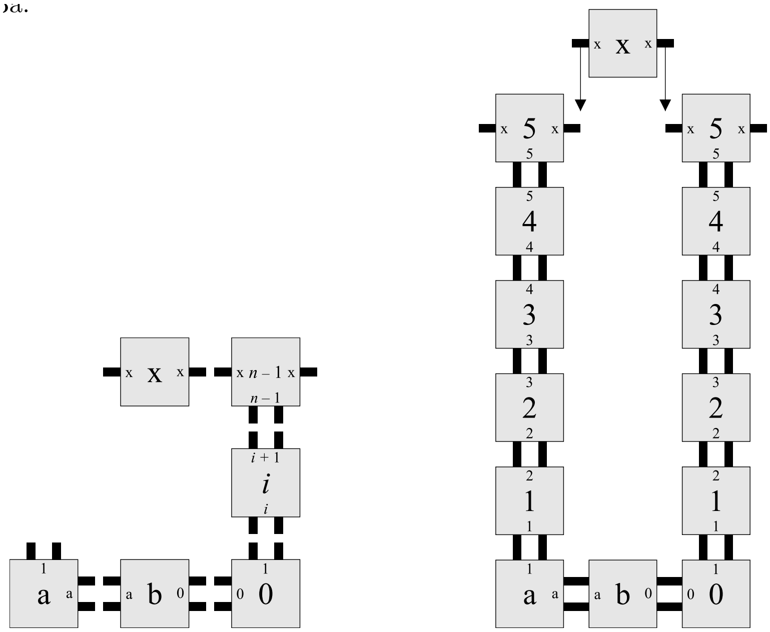
Lemmas 3.3 and 3.6 tell us that there exists a shape (e.g., L_n), along with a temperature value (e.g., $\tau=1$), such that, L_n can be assembled in the aTAM using fewer unique tile types than in the 2HAM. However, as we will see next, if the temperature is $\tau=2$, then there is no difference between the minimum number of unique tile types required to self-assemble L_n in the aTAM versus the 2HAM.

Lemma 3.7. *For all $n \in \mathbb{N}$ such that $n > 2$, $\mathcal{K}^2_{2HAM}(L_n) \leq n+3$.*

Proof. To see that $\mathcal{K}^2_{\text{2HAM}}(L_n) \leq n+3$, define the TAS $\mathcal{T}_n=(T_n,2)$, where T_n consists of the tile types given in Fig. 3a.

It is easy to see that $\mathcal{T}_n$ uniquely produces L_n by building a 'U' shape to which the 'x' tile may attach and close the loop giving L_n. □

Corollary 3.8. *For all $n \in \mathbb{N}$ such that $n > 2$, $\mathcal{K}^2_{aTAM}(L_n) \leq n+3$.*



(a) For $2 < n \in \mathbb{N}$, let $i \in \{1, \ldots, n-2\}$.

(b) A loop of size 6 (almost).

Fig. 3. Construction for $\mathcal{K}^2_{\{\text{aTAM},\text{2HAM}\}}(L_n) \leq n+3$. Note that this construction works in both the aTAM (with the 'a' tile as the seed) and the 2HAM at temperature 2.

Corollary 3.8 follows immediately by creating an aTAM system using the tile set from Fig. 3a and letting a tile of the label 'a' be the seed tile.

Lemma 3.9. *For all $n \in \mathbb{N}$ such that $n > 2$ and all $\tau \in \mathbb{N}$, if the aTAM system $\mathcal{T} = (T, \sigma, \tau)$ self-assembles L_n, then the seed tile only appears once in any terminal assembly α of $\mathcal{T}$.*

Lemma 3.9 follows from a straightforward case analysis. The seed tile type must have at least one strength-2 glue (which we'll call g) in order to allow any growth to proceed from it. In order for a second copy to attach somewhere it must do so via either a strength-2 glue, or two strength-1 glues. Two strength-1 glues are not an option because that would mean that there is a third side with the strength-2 glue g exposed after it attaches. This would allow a tile of whichever type bound to the seed to then attach to that exposed glue, and this tile would have 3 neighbors but there are no tiles in L_n with 3 neighbors. Due to the shape of L_n, there can be at most one tile which attaches via 2 strength-1 glues (since the sides of the shape to either side must be complete before it attaches). Therefore, if a second copy of the seed tile type is able to attach, it must be able to do so via the growth of a direct path of tiles, all connected by strength-2 glues between it and the seed copy.

Either of two cases may occur: (1) along the path between the two copies, the output direction from the seed copy is the same as the input direction for the second copy (i.e. both N), or (2) the directions are different. In case (1), the two copies must be on opposite sides of L_n and it must be the case that another strength-2 glue, say g_2, exists on the seed type other than the one connecting the two copies on the path (otherwise, only that path could grow, which couldn't include all four sides of L_n). However, g_2 must allow a path which turns a corner to grow, but by first growing the path between the two copies, then growing this same "corner-turning" path from both copies (from their identical copies of g_2), the assembly can grow outside of L_n. In case (2), this means that the seed type has two strength-2 glues facing different directions. It must be possible for the path which grows into the second copy of the seed type to instead grow out of the seed copy. By growing a portion of that path as well as a portion of the path growing out of the seed copy toward the second copy, both until their first corners, it must be the case that the produced assembly has a side length incompatible with L_n. Therefore, Lemma 3.9 holds and only one copy of the seed tile type can appear.

The following lemma, along with Lemma 3.7, says that, at temperature 2, loops cannot be used as an example class of shapes that can be self-assembled using fewer unique tile types in the aTAM over the 2HAM.

Lemma 3.10. *For all $n \in \mathbb{N}$ such that $n > 2$, $\mathcal{K}^2_{aTAM}(L_n) \geq n+3$.*

Proof. Let $\mathcal{T} = (T, \sigma, 2)$ be any aTAM TAS in which L_n self-assembles such that every glue on every $t \in T$ has strength either 1 or 2.

The key observation (previously mentioned) is that, since $\tau = 2$, for any $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, there can be at most one point such that the tile placed at this point binds via two strength-1 glues (if there is more than one such point, then α would not be producible at all).

There are two cases to consider.

Case 1. Suppose that, for every $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, for all $\vec{x} \in L_n$, $\alpha\left(\vec{x}\right)$ initially binds via one strength-2 glue (this case will also cover the case where the last tile to attach binds via two strength-2 glues). In this case, every glue strength of every tile type in T could be set to 1 and we would have a TAS $\mathcal{T}' = (T', \sigma, 1)$ in which L_n self-assembles with $|T'| = |T|$ and Lemma 3.5 says that $|T| \geq n + 5 > n + 3$.

Case 2. Suppose that there exists $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, such that there is one such point $\vec{x} \in L_n$ and $t = \alpha\left(\vec{x}\right)$ binds via two strength-1 bonds. Assume, without loss of generality, that $\vec{x} \notin (\{0\} \times \{0, \ldots, n-2\}) \cup \{(1,0), (2,0)\}$. This implies that $\left|\left\{\alpha\left(\vec{x}\right) \mid \vec{x} \in (\{0\} \times \{0, \ldots, n-2\}) \cup \{(1,0), (2,0)\}\right\}\right| = n+1$ and every tile placed at the points in the set $(\{0\} \times \{0, \ldots, n-2\}) \cup \{(1,0), (2,0)\}$ initially binds non-cooperatively (i.e., via a single strength-2 glue). Furthermore, t can only appear once in α, since it binds via two strength-1 glues, and it cannot be the seed tile type (Lemma 3.9). Thus, we have $|T| \geq 1 + 1 + |(\{0\} \times \{0, \ldots, n-2\}) \cup \{(1,0), (2,0)\}| = 2 + n + 1 = n + 3$ (the first "plus 1" represents the tile to which t binds and the second "plus 1" represents t). $\Box$

Theorem 3.2 says that there exists a class of shapes that self-assemble at temperature 1 using fewer unique tile types in the aTAM than in the 2HAM. This violates the intuition that the 2HAM should be able to self-assemble shapes more efficiently because it is a (seedless) generalization of the aTAM, but is possible because of the use of the type of blocking available only in the aTAM. However, the difference is not asymptotic (i.e. for both models the tile complexity is $O(n)$), and furthermore Theorem 3.2 also says that at temperature 2 the tile complexity of loops is not better for the aTAM than the 2HAM. However, one can actually use the technique of blocking to get an asymptotic separation in tile complexity of the aTAM over the 2HAM, as shown by the following result.

Theorem 3.11. *(Counter-with-crasher) For $n \in \mathbb{N}$, there exists a shape $X_n \subseteq \mathbb{Z}^2$, such that $\mathcal{K}^2_{2HAM}(X_n) = \Omega\left(2^n\right)$ and $\mathcal{K}^2_{aTAM}(X_n) = O(n)$.*

Proof. Let $n \in \mathbb{N}$. Define $X_n = \left(\{0, 1, \ldots, n+1\} \times \left\{0, 1, \ldots, 2^n\right\}\right) - \left(\{1\} \times \left\{1, 2, \ldots, 2^n - 1\right\}\right)$. Intuitively, X_n is a $(2^n + 1) \times n$ rectangle with a vertical line of length $2^n - 1$ removed from the middle of the second-to-left column of the rectangle, which we call a *counter-with-crasher*. The tile set that can be inferred from Fig. 4 (and which utilizes a standard binary counter tile set) can be used to construct an aTAM TAS with a single seed tile that uniquely produces X_n at temperature 2 and the size of the tile set is $O(n)$ (it simply requires $\log(2^n) = n$ tile types to represent each of the bit locations of the counter, and a constant additional set of tile types to perform the counting, blocking, etc.). Thus, $\mathcal{K}^2_{aTAM}(X_n) = O(n)$.

Now, suppose that some 2HAM system $\mathcal{T} = (T, \tau)$ self-assembles X_n and $|T| < 2^n - 1$. Suppose further that $\mathcal{T}$ produces α, and the domain of α is X_n. Then, there exist two points $\vec{p}, \vec{q} \in \{0\} \times \left\{1, 2, \ldots, 2^n - 1\right\}$, such that $\alpha\left(\vec{p}\right) = \alpha\left(\vec{q}\right)$. Moreover, all (north/south) glues between (all the tiles in between) $\alpha\left(\vec{p}\right)$ and $\alpha\left(\vec{q}\right)$ must be strength τ, which means just the subassembly in α that consists of the line of all the tiles between and including $\vec{p}$ and $\vec{q}$ can self-assemble in $\mathcal{T}$. Since $\alpha\left(\vec{p}\right) = \alpha\left(\vec{q}\right)$, arbitrarily tall linear assemblies self-assemble in $\mathcal{T}$. Thus, $\mathcal{T}$ produces an assembly whose shape is not X_n, meaning that X_n does not self-assemble in $\mathcal{T}$. In order to avoid such a situation, we must have that any 2HAM TAS $\mathcal{T} = (T, \tau)$ in which X_n self-assembles satisfies $|T| \geq 2^n - 1$. It follows that $\mathcal{K}^\tau_{2HAM}(X_n) = \Omega\left(2^n\right)$. $\Box$

Note that the construction for Theorem 3.11 (see Fig. 4) can be modified so that the northward growth of the assembly depends on the running time of a Turing machine. Making such a modification gives a tile complexity separation between the aTAM and 2HAM that grows asymptotically faster than any computable function, i.e. following the busy beaver function. (See the proof of Theorem 3.19 for details of such a modification.)

As we shall see in the next subsection, in contrast to Theorem 3.11 there exists a class of shapes that self-assemble in the 2HAM using asymptotically fewer unique tile types than what is required in the aTAM.

3.2. Staircases

In this subsection, we will study the tile complexity of shapes that resemble "staircases." We will show that these shapes self-assemble in the 2HAM using asymptotically fewer tile types than what is required for their self-assembly in the aTAM.

Definition 3.12. For each $i, k \in \mathbb{N}$, let $B_{i,k} = (\{0, \ldots, k-1\} \times \{-k, \ldots, 0, \ldots, i+2\}) \cup \{(-1, i+1), (k, 0)\}$ and define, $S_n = \bigcup_{i=0}^{2^n - 1}\left(B_{i,n} + ((n+1)i, 0)\right)$. We refer to each $B_{i,n}$ as the i^{th} stair step.

Intuitively, the set S_n is a "staircase with 2^n steps with each step of width n." See Fig. 5, which depicts S_4. We will use S_n to prove our first main result of this section, which is the following theorem.

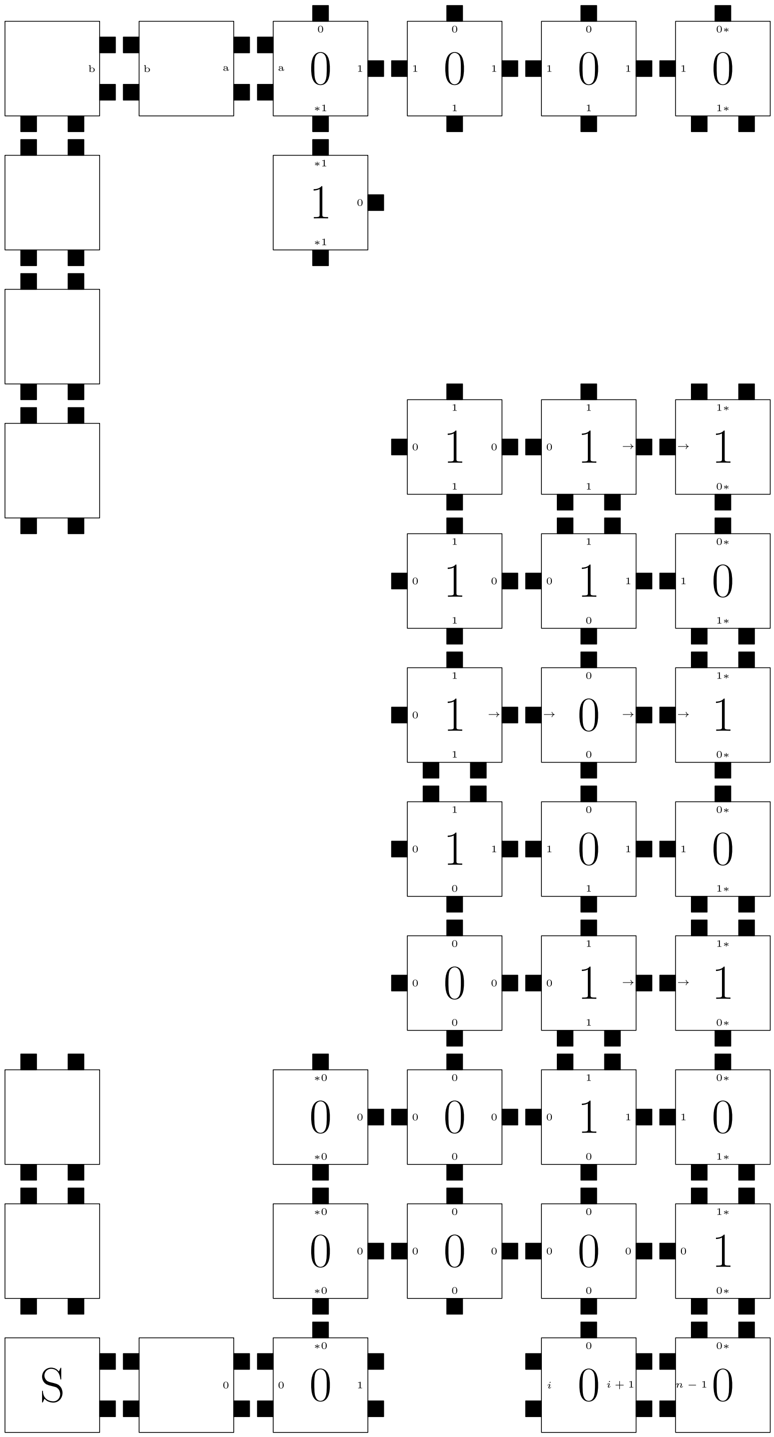


Fig. 4. The tile set shown here implicitly defines a temperature 2 TAS that uniquely produces a shape, called a counter-with-crasher, that requires exponentially many more tile types for unique self-assembly in the 2HAM. The tile types in the bottom row and second-to-right column are defined for $i = 1, \ldots, n-2$. Intuitively, a binary counter self-assembles northward, counting from 0 to $2^n - 1$ then rolling back over to 0, at which point a southward assembling single tile wide path of tiles is initiated, which ultimately gets blocked from further growth by the seed tile.

Theorem 3.13. $\mathcal{K}_{aTAM}(S_n) = \Omega(n)$.

We use a counting argument to prove $\mathcal{K}_{aTAM}(S_n) = \Omega(n)$. It is interesting to note that, if one were to apply the standard, perhaps most obvious information-theoretic argument to prove Theorem 3.13, then one would only obtain a bound of $\Omega\left(\frac{\log n}{\log\log n}\right)$ (this is not to say that an information-theoretic proof of Theorem 3.13 does not exist).

Before we prove Theorem 3.13, we must define some notation.

Notation. For $0 \le i < 2^n - 1$, let $C_i^- = ((n+1)i+n, 0)$ and $C_i^+ = ((n+1)i+n, i+2)$. Let $C_i = \left\{C_i^-, C_i^+\right\}$. We refer to the set C_i as the i^{th} *connector* (column), with C_i^- and C_i^+ denoting the lower and upper connector points, respectively, of the staircase S_n. We call C_i^- and C_i^+ *siblings*.

Definition 3.14. Let $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ be an assembly sequence in some TAS with result α. If $\vec{x}, \vec{y} \in \operatorname{dom}\alpha$ are such that $i_{\vec{\alpha}}(\vec{x}) < i_{\vec{\alpha}}(\vec{y})$ and every path in the binding graph $G_{\alpha_{i_{\vec{\alpha}}(\vec{y})}}$ from the seed to $\vec{y}$ goes through $\vec{x}$, then we write $\vec{x} \prec_{\vec{\alpha}} \vec{y}$ and say that $\vec{y}$ *strictly depends on* $\vec{x}$ *in* $\vec{\alpha}$.

Intuitively, if $\vec{y}$ strictly depends on $\vec{x}$ in $\vec{\alpha}$, then $\vec{x}$ is a kind of "pinch-point" through which all "information" from the seed to $\vec{y}$ must flow immediately prior to the placement of the tile at $\vec{y}$. In a cooperative (i.e. $\tau \ge 2$) system, a tile placement may require no more than one other tile. However, in the following proof(s) we will focus on locations where single τ-strength glues are forced to connect subassemblies and therefore serve as single-tile pinch-points.

Notation. Let $0 \le i < 2^n - 1$.

1. The height of the vertical, one-tile-wide *gap* of C_i is defined as $gap(i) = i+1$. Note that this quantity does not count the points C_i^- and C_i^+, but rather just the number of empty spaces between them.
2. The *height* of the stair step that is immediately east of C_i is defined as $height(i) = gap(i) + (n+3) = i+1+n+3 = i+n+4$.

Definition 3.15. Let $\vec{\alpha}$ be an assembly sequence in $\mathcal{T}$ such that $\operatorname{dom}\operatorname{res}(\vec{\alpha}) = S_n$. For $0 \le i < 2^n - 1$, we say that a point $\vec{x} \in C_i$ is *ambitious* in $\vec{\alpha}$ if there exists a point $\vec{y} = (p, q) \in S_n$ satisfying the following conditions:

1. (Strict dependency) $\vec{x} \prec_{\vec{\alpha}} \vec{y}$,
2. (Sufficient vertical growth) $q = \left\lfloor \frac{gap(i)}{2} \right\rfloor$, and
3. (Restricted horizontal growth) $\vec{y}$ is located in the stair step that is immediately east of C_i if C_i contains or is east of the single seed tile. Otherwise, if C_i is west of the single seed tile, then $\vec{y}$ is contained in the stair step that is immediately west of C_i.

In other words, an ambitious connector point (at which a connector tile is placed) is one that can, without having to wait for a tile to be placed at its sibling connector point, initiate the self-assembly of tiles that grow at least half way "north" (or "south") toward its sibling and the vertical growth is contained in the "very next stair step", where "very next stair step" is defined relative to the position of the seed tile.

Notation. Let $n \in \mathbb{N}$. For $0 \le i < n$, we define the i^{th} *flight of connector columns* of S_n to be the set $\mathcal{F}_i = \bigcup_{j=2^i-1}^{2^{i+1}-2} C_j$.

We will now prove some technical lemmas that will be used to prove Theorem 3.13.

Lemma 3.16. *Let $2 < n \in \mathbb{N}$ and assume the following conditions are true:*

1. *$\mathcal{T} = (T, \sigma, \tau)$ is any singly-seeded TAS in which S_n self-assembles,*
2. *$|T| < \frac{n}{2}$,*
3. *$\vec{\alpha}$ is an assembly sequence in $\mathcal{T}$ with result α and $\operatorname{dom}\alpha = S_n$, and*
4. *n_0 is any positive integer satisfying $2^{n_0} > n$.*

Assuming the conditions listed above, then, for every $n_0 \le i < n$, there exists $\vec{x} \in \mathcal{F}_i$, such that $\vec{x}$ is ambitious in $\vec{\alpha}$.

Lemma 3.16 says that "almost" every flight of connector points contains at least one ambitious connector point, assuming that the tile set T is not "too big".

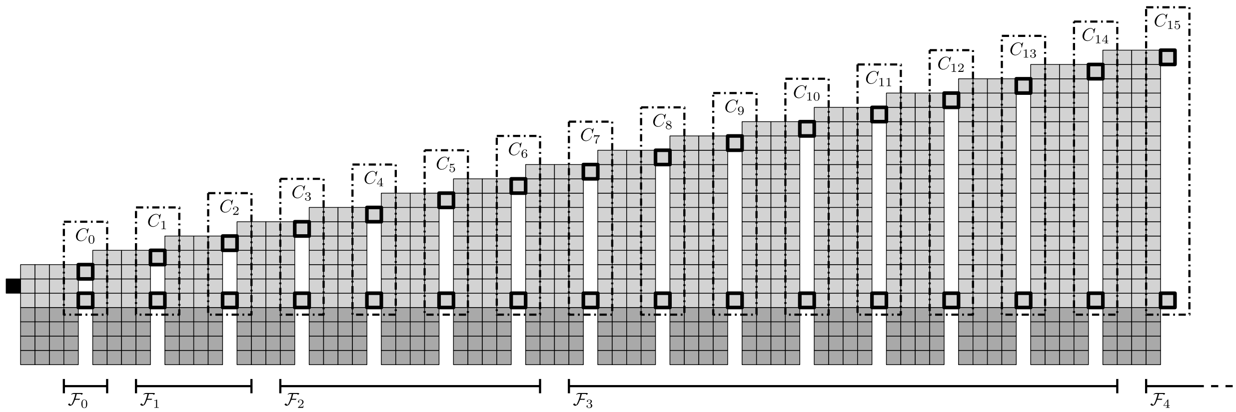


Fig. 5. The staircase S_4 is shown here. The origin is denoted as the black square. The flights and connector columns are indicated.

Proof. Assume, for the sake of obtaining a contradiction, that there is some flight, say $\mathcal{F}_{i^*}$, for some $n_0 \leq i^* < n$, that does not contain an ambitious connector point.

Since, for all $n_0 \leq i < n$, $|\mathcal{F}_i| = 2^i \geq 2^{n_0} > n$, the seed tile must be strictly east (west) of more than $\frac{n}{2}$ connector columns that are all contained within the flight $\mathcal{F}_{i^*}$. Assume, without loss of generality, that the seed is strictly west of more than $\frac{n}{2}$ connector columns that are all contained within the flight $\mathcal{F}_{i^*}$.

Let $C_k \subseteq \mathcal{F}_{i^*}$ be the connector column that either contains the seed tile or is the first connector column that is east of the point at which the seed tile is placed. Assume, without loss of generality, that C_k^- is the first point in C_k to receive a tile in $\vec{\alpha}$.

We will now construct some new assembly sequences in $\mathcal{T}$ and use them to show that $\mathcal{T}$ must place some tile at a point that is not contained in S_n.

First, let $\widehat{\vec{\alpha}}$ be an assembly sequence in $\mathcal{T}$ such that $\widehat{\vec{\alpha}}$ behaves exactly like $\vec{\alpha}$ but only places tiles at locations on which C_k^- strictly depends, until it places a tile at C_k^-, at which point, $\widehat{\vec{\alpha}}$ behaves exactly like $\vec{\alpha}$ but only places tiles at locations that strictly depend on C_k^-. Since $\mathcal{F}_{i^*}$ does not contain an ambitious connector point, it follows that, for all $j = k, k+1, \ldots, 2^{i^*+1} - 2$, C_j^- strictly depends on C_k^- in $\vec{\alpha}$ and in $\widehat{\vec{\alpha}}$. Moreover, since $|T| < \frac{n}{2}$, and $\mathcal{F}_{i^*}$ contains more than $\frac{n}{2}$ connector columns that are all east of the seed and the result of $\vec{\alpha}$ has domain S_n, $\widehat{\vec{\alpha}}$ must place tiles at two connector points, say C_l^- and C_m^-, with l and m satisfying $k \leq l < m < 2^{i^*+1} - 1$ and $\alpha\left(C_l^-\right) = \alpha\left(C_m^-\right)$.

Now, let $\widehat{\widehat{\vec{\alpha}}}$ be an assembly sequence in $\mathcal{T}$ such that $\widehat{\widehat{\vec{\alpha}}}$ behaves exactly like $\widehat{\vec{\alpha}}$ up until $\widehat{\vec{\alpha}}$ places a tile at C_m^-, at which point, $\widehat{\widehat{\vec{\alpha}}}$ repeats the subsequence of tile placements from C_l^- to C_m^- in $\widehat{\vec{\alpha}}$ indefinitely, starting at C_m^- and using the same relative order of placement as $\widehat{\vec{\alpha}}$ uses. Since $\mathcal{F}_{i^*}$ does not contain an ambitious connector point, $\widehat{\widehat{\vec{\alpha}}}$ is a valid, infinite assembly sequence in $\mathcal{T}$, contradicting the fact that S_n self-assembles in $\mathcal{T}$. □

Corollary 3.17. *Let* $21 < n \in \mathbb{N}$, $n_0 = \lceil \log(n+2) \rceil$ *and assume the following conditions are true:*

1. $\mathcal{T} = (T, \sigma, \tau)$ *is any singly-seeded TAS in which* S_n *self-assembles,*
2. $|T| < \frac{n-n_0}{2 \cdot 5}$*, and*
3. $\vec{\alpha}$ *is an assembly sequence in* $\mathcal{T}$ *with result* α *and* $\operatorname{dom} \alpha = S_n$.

Assuming the conditions listed above are true, then there exist natural numbers $0 \leq r < s < n - n_0$ *and corresponding connector points* $\vec{x}_r, \vec{x}_s$ *satisfying the following conditions:*

1. $\vec{x}_r \in \mathcal{F}_{n_0+r}$ *and* $\vec{x}_s \in \mathcal{F}_{n_0+s}$,
2. $\alpha\left(\vec{x}_r\right) = \alpha\left(\vec{x}_s\right)$,
3. $\vec{x}_r$ *and* $\vec{x}_s$ *are either both east of the seed tile or both west of the seed tile,*
4. $\vec{x}_r$ *and* $\vec{x}_s$ *are ambitious in* $\vec{\alpha}$*, and*
5. $r < s - 3$.

Proof. First, note that $|T| < \frac{n-n_0}{2 \cdot 5} < \frac{n}{2}$ and $2^{n_0} = 2^{\lceil \log(n+2) \rceil} \geq 2^{\log(n+2)} = n + 2 > n$, whence Lemma 3.16 guarantees the existence of the following sequence of ambitious points in $\vec{\alpha}$:

$$\vec{x}_0 \in \mathcal{F}_{n_0}, \vec{x}_1 \in \mathcal{F}_{n_0+1}, \ldots, \vec{x}_{n-n_0-1} \in \mathcal{F}_{n-1}$$

Since $|T| < \frac{n-n_0}{2\cdot 5}$, it must be the case that, in the sequence $\alpha(\vec{x_0}), \alpha(\vec{x_1}), \ldots, \alpha\left(\vec{x}_{n-n_0-1}\right)$ of $n - n_0$ tiles, more than $\frac{n-n_0}{\left(\frac{n-n_0}{2\cdot 5}\right)} = 2 \cdot 5$ tiles must be the same type. Of these (at least) $2 \cdot 5 + 1$ tiles, at least 5 must be east (west) of the seed tile. Suppose that these 5 tiles are placed at locations $\vec{x}_r \in \mathcal{F}_{n_0+r}, \ldots, \vec{x}_s \in \mathcal{F}_{n_0+s}$. To complete the proof, we note that $s > r + 3$. $\square$

We are now ready to prove Theorem 3.13.

Proof of Theorem 3.13. Let $\mathcal{T} = (T, \sigma, \tau)$ be any singly-seeded TAS in which S_n self-assembles, for $n > 21$. Assume for the sake of contradiction that $|T| < \frac{n-\lceil\log(n+2)\rceil}{2\cdot 5}$. Let $\vec{\alpha}$ be an assembly sequence in $\mathcal{T}$ whose final assembly is denoted as α, with domain S_n. We will show that it is always possible for $\vec{\alpha}$ to place some tile at a location $\vec{x} \notin S_n$.

Note that the conditions in the hypothesis of Corollary 3.17 are satisfied. Therefore, let $\vec{x}_r$ and $\vec{x}_s$ be the points in S_n given by Corollary 3.17. In what follows, we assume, without loss of generality, that $\vec{x}_r$ and $\vec{x}_s$ are lower connector points and they are both east of the seed tile.

Let $m \in \mathbb{N}$ be the number of tiles that strictly depend on $\vec{x}_s$ in $\vec{\alpha}$ and define $\vec{y}_0, \vec{y}_1, \ldots, \vec{y}_{m-1}$ such that, for all $0 \le j < m$, $\vec{x}_s \prec_{\vec{\alpha}} \vec{y}_j$ and $i_{\vec{\alpha}}\left(\vec{y}_0\right) < i_{\vec{\alpha}}\left(\vec{y}_1\right) < \cdots < i_{\vec{\alpha}}\left(\vec{y}_{m-1}\right)$. We will now construct a new assembly sequence $\widehat{\vec{\alpha}}$ in $\mathcal{T}$ as follows. Let $\widehat{\vec{\alpha}}$ be such that $\widehat{\vec{\alpha}}$ behaves exactly like $\vec{\alpha}$ but only places tiles at locations on which $\vec{x}_r$ strictly depends, until it places a tile at $\vec{x}_r$, at which point, $\widehat{\vec{\alpha}}$ places, for all values $0 \le j < m$, the tile type $\alpha\left(\vec{y}_j\right)$ at $\vec{y}_j - \left(\vec{x}_s - \vec{x}_r\right)$ and in the same relative order according to $\vec{\alpha}$.

Note that $\widehat{\vec{\alpha}}$ is a valid assembly sequence because:

1. For all $0 \le j < m$, $\vec{x}_s \prec_{\vec{\alpha}} \vec{y}_j$,
2. $\alpha\left(\vec{x}_r\right) = \alpha\left(\vec{x}_s\right)$, and
3. immediately after $\widehat{\vec{\alpha}}$ places the tile $\alpha\left(\vec{x}_s\right)$ at $\vec{x}_r$, $\widehat{\vec{\alpha}}$ has yet to place a tile at any location that is east of the column occupied by $\vec{x}_r$. This condition holds because $\widehat{\vec{\alpha}}$ only places tiles at positions on which $\vec{x}_r$ strictly depends, which means $\vec{x}_r$ is the first location in its column to receive a tile under $\widehat{\vec{\alpha}}$.

It is worthy to note that the height of the shortest stair step that is immediately east of $\vec{x}_s$ is $height\left(2^{n_0+s} - 1\right) = 2^{n_0+s} + n + 3$ and the maximum height of the stair step that is immediately east of $\vec{x}_r$ is $height\left(2^{n_0+r+1} - 2\right) = 2^{n_0+r+1} - 1 + n + 3 = 2^{n_0+r+1} + n + 2$.

Since $\vec{x}_s$ is ambitious in $\vec{\alpha}$, it must be the case that there exists some value $0 \le j < m$ such that $\vec{y}_j = \left(p, \left\lfloor \frac{gap\left(2^{n_0+s}-1\right)}{2} \right\rfloor\right)$, $\vec{y}_j$ is contained in the stair step immediately east of $\vec{x}_s$, $\vec{y}_j$ strictly depends on x_s and

$$\begin{aligned} q = \left\lfloor \frac{gap\left(2^{n_0+s} - 1\right)}{2} \right\rfloor &= 2^{n_0+s-1} \\ &> 2^{n_0+r+2} \quad (r < s - 3) \\ &= 2^{n_0+r+1} + 2^{n_0+r+1} > 2^{n_0+r+1} + 2^{n_0} \\ &> 2^{n_0+r+1} + (n+2) = height\left(2^{n_0+r+1} - 2\right) \\ &= \text{“the maximum height of the stair step that is immediately east of } \vec{x}_r\text{”}. \end{aligned}$$

This means that, as $\widehat{\vec{\alpha}}$ tries to mimic $\vec{\alpha}$, it will place at least one tile that is strictly outside-to-the-north of the stair step that is immediately east of $\vec{x}_r$.

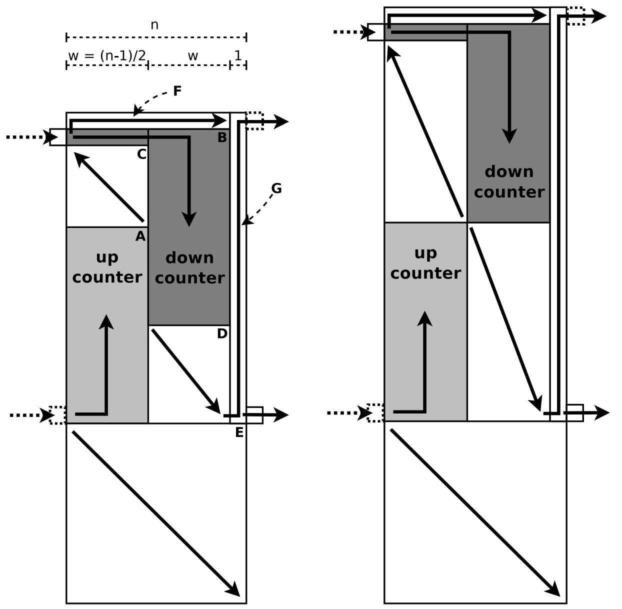
The fact that $\widehat{\vec{\alpha}}$ places a tile at some location $\vec{x} \notin S_n$ is a contradiction to the fact that $\mathcal{T}$ self-assembles S_n. We therefore conclude that, if $\mathcal{T} = (T, \sigma, \tau)$ is any TAS in which S_n self-assembles, then, for all $n > 8$, $|T| \ge \frac{n-\lceil\log(n+2)\rceil}{2\cdot 5} \ge \frac{n-\frac{n}{2}}{2\cdot 5} = \Omega(n)$. $\square$

A natural question to consider is whether or not the bound given in Theorem 3.13 is tight. In what follows, we give a construction for S_n in the aTAM.

Theorem 3.18. $K^2_{aTAM}(S_n) = O(n^2)$.

Proof. We use a square and multiple base-4 counters to build the staircase shape S_n. First, we break down the staircase into flights of stairs. Second, we describe the construction used for the base-4 counters. Finally, we describe the construction of the overall staircase shape.

Preliminaries. Since the staircase is made up of 2^n width-n steps and $2^n - 1 = \sum_{i=0}^{n-1} 2^i$, we can write 2^n as $1 + 1 + 2 + 4 + 8 + \cdots + 2^{n-1}$. Therefore, we can divide the left-to-right sequence of steps into sub-sequences or *flights* of size 1, 1, 2, 4, 8, $\cdots$, 2^{n-1}, respectively. So the leftmost step makes up the first flight, the second step from the left makes up the second flight, $\cdots$, the last or rightmost 2^{n-1} steps make up the last flight.



(a) Any step but the last one in the flight (b) Last (i.e., rightmost) step of the flight

Fig. 6. Construction for a single step of the staircase.

Base-4 counter. Each step in S_n is built using two counters whose width is $w = \frac{n}{2}$ (approximately). Since the width of each counter is (about) half the width of the step but the tallest counter must be (about) as tall as 2^n, we use base-4 counters (note that $4^w = (2^2)^w = 2^{2w} = 2^n$).

We now describe the tile set used to build each w-wide base-4 counter. The seed row contains w distinct tiles and represents the starting value of the counter. In this row, as well as in all subsequent rows, we keep track of the least significant (i.e., rightmost) digit that is not a 3. This digit (represented by a so-called 'increment tile') is the one that will be incremented by 1 (modulo 4) in the next row. The increment tiles are the only ones with strength 2 on their north side.

Therefore, they initiate the growth of each new row. Digits to the left of the increment tile (if any) are simply copied (using so-called 'copy tiles') from the previous row. Digits to the right of the increment tile are all 3s and are reset to 0 (using so-called 'reset tiles'). When the increment tile reaches the value 3, the new least significant non-3 digit must be found (using so-called 'find tiles') to the left of the increment tile. Finally, when all digits are 3s, counting stops but we add one last, topmost row to cap the counter with glues needed for the overall construction. Note, that for all categories of tile, we use distinct tiles types to distinguish the leftmost and rightmost digits from the middle digits.

The number of tiles in this construction is computed as follows:

Tile type classification	Number of tile types	Notes
Seed row	w	One tile type per digit in the initial value of the counter
Increment tiles	9	Three tiles (incrementing each one of the digits 0, 1 and 2; note that incrementing 3 is done below using a reset tile) each for the leftmost, middle and rightmost digits
Copy tiles	8	Four tiles (0 through 3) each for the leftmost and middle digits
Find tiles	8	Four tiles (0 through 3) each for the leftmost and middle digits
Reset tiles	2	One tile each for the middle and rightmost digits
Topmost tiles	3	One tile for the leftmost, middle and rightmost digits

Therefore, the total number of tile types for building a w-wide base-4 counter is $w + 30$, which is $\Theta(n)$.

Staircase shape. Fig. 6 depicts the way each step is built, using one square, two base-4 counters and several types of filler tiles. Fig. 6(a) labels each section of the step with a letter from A to G.

In this construction, the rightmost column in each step (section G) is used to communicate to the next step, through both the upper and lower connectors, one bit of information, namely whether to continue the current flight or start a new

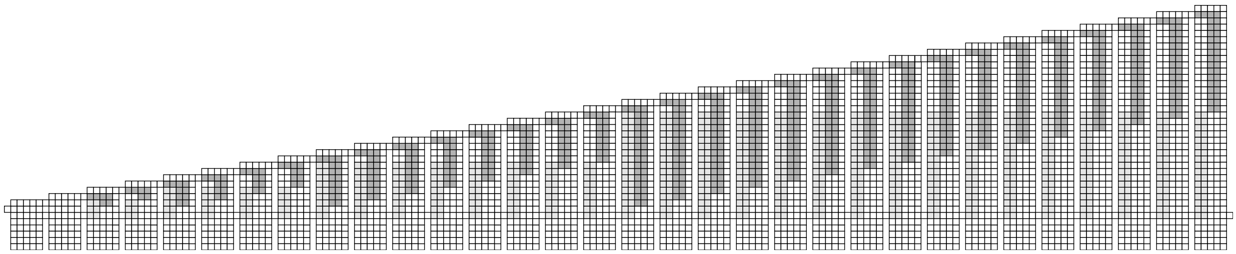


Fig. 7. Staircase construction for $n = 5$. This image depicts the pattern of the up and down-growing counters. The two initial steps are hard-coded and therefore do not contain counters.

flight. As a special case, in the last flight, the bit of information represents whether or not the current flight is complete or not.

The bulk of each step is made up of one $n \times n$ square positioned under two counters that grow in opposite directions, namely upward or downward. If n is odd, then each counter has width $w = \frac{n-1}{2}$. If n is even, then one of the counters has width $w_1 = \frac{n}{2}$ and the other has width $w_2 = \frac{n-2}{2}$. From now on, we assume that n is odd (the other case is similar). Since the first two flights (numbered 0 and 1) of the staircase always contain exactly one step each, we hardwire their construction (without using counters) with $\Theta(n)$ tile types. Furthermore, we use a single, linear-size tile set to build the bottommost squares in all steps.

If $1 < i < n$, then all steps in flight i use the same two counters. One counter grows upward (shown in light gray as section A in Fig. 6(a)) while the other grows downward (shown in dark gray as section B). In all steps but the last one in the flight, the two counters grow past each other. In other words, the west side of the upper-right tile in the up counter is adjacent to the east side of some leftmost tile in the down counter. In this case, section D uses filler tiles to propagate the fact that the next step should use the same counters (i.e., it is still part of the current flight). In contrast, in the last step of the flight (see Fig. 6(b)), the two counters barely touch diagonally, in which case section D uses filler tiles to propagate the fact that this step is the last one in the current flight. Fig. 7 shows the resulting construction for $n = 5$.

To summarize the construction of each step:

1. The up and down counters are built asynchronously (using $\Theta(n)$ tile types) from the upper connector and lower connector, respectively. Note that the down counter is hooked to the upper-left connector using a single $\Theta(n)$ row of tiles shown in dark gray above section C (the 'hook section' not identified by a letter),
2. While the counters are being built, the bottommost square can grow from the lower-left tile in the up counter and the step's topmost row (section F) can grow (using $\Theta(1)$ tile types) from the leftmost tile in the hook section.
3. Once the counters meet, section C fills up using $\Theta(1)$ tile types (in fact, exactly one tile type).
4. Similarly, once the counters meet, section D fills up using $\Theta(1)$ tile types.
5. Finally, once sections D and the top row of section E are both complete, section G is built (using $\Theta(1)$ tile types) starting from its bottommost tile.

In conclusion, each flight uses $\Theta(n)$ tile types and the whole construction uses $\Theta(n^2)$ tile types. □

In the 2HAM, it is possible to beat the $\Omega(n)$ lower bound for the aTAM from Theorem 3.13. To achieve this, we self-assemble staircase shapes using a simple Turing machine simulation (the simulation is used to specify the width of each stair step). In doing so, if we use a Turing machine that always halts – but perhaps not for a very long time – we get an asymptotic separation in tile complexity between the aTAM and 2HAM. More formally, we have the following.

Theorem 3.19. *If $M = (Q, \Sigma, \Gamma = \Sigma \cup \{\sqcup\}, \delta, q_0, q_h)$ is a deterministic Turing machine and t denotes the number of steps M takes to halt on the empty string, then $\mathcal{K}^2_{2HAM}\left(S_{3(3|Q\times\Gamma|+4|\Gamma|)} + 2(t+1)\right) = O(|Q|)$.*

Intuitively, our construction for Theorem 3.19 works as follows. We first build a square using a "standard" aTAM Turing machine simulation construction (very similar to the construction that Rothemund and Winfree give in Theorem 5 of [22]). Our Turing machine construction has the property that, to the top of the completed simulation square, tiles that represent either a 0 or a 1 may attach nondeterministically. This topmost row of tiles is used as a seed for a binary counter, which counts from the nondeterministically chosen starting value, say w, up to the next highest power of 2, minus one. The stair steps attach to each other in a two-handed fashion, at temperature $\tau = 2$, via two "connector tile types" that are located at opposite corners of each stair step.

Proof. In our construction, all possible stair steps can be thought to self-assemble in parallel and then fully-formed stair steps of the right height bind together in the correct order to form the staircase.

Construction of an individual stair step has three logical phases: the tile collector gadget, simulation of M, and binary counter.

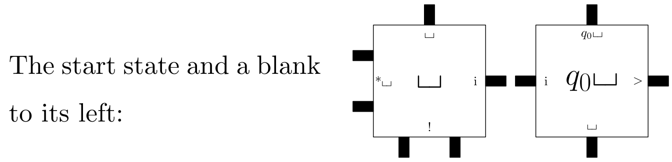


Fig. 8. Input row tile types. The input row consists of a leftmost blank symbol, the start state and as many blank symbols (see Fig. 10) to the right of the start state to get to the rightmost edge of the tile type collector square. We assume that the input tile types attach to the top of the tile collector square.

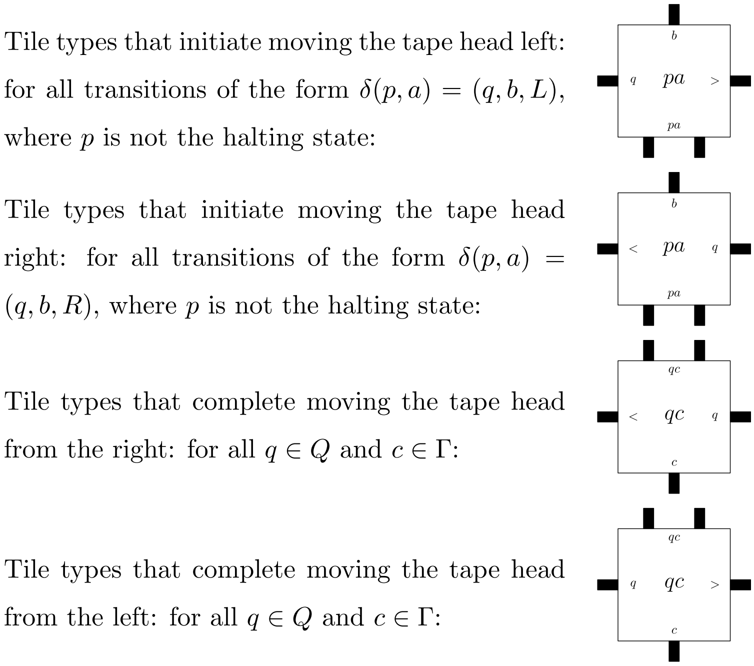


Fig. 9. Tape head movement tile types.

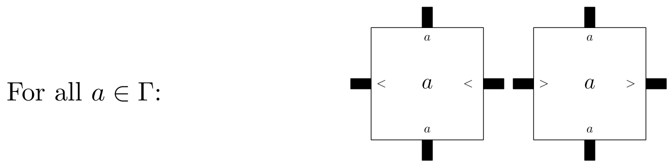


Fig. 10. Copy tile types. These tile types copy the previous TM configuration up to the next row. Each copy tile "knows" whether it is to the left or right of the tape head.

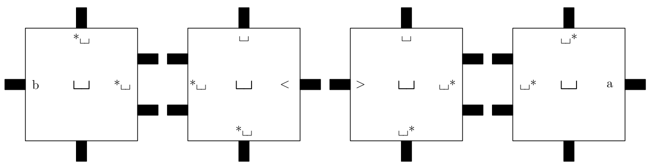


Fig. 11. These tile types expand the tape (on the right and the left, respectively) by one blank symbol.

Simulation. The simulation tile types are given in Figs. 8–13.

Our 2HAM simulation of M essentially works the same as an aTAM simulation of M, except we assume that the input row, which encodes the initial configuration of M (see the tile types in Fig. 8), attaches in a cooperative fashion to the top of some kind of seed square. In our case, the seed square to which the input row attaches is the tile collector gadget, which we describe next. It is clear that the tile complexity of the simulation tile types is dominated by $O(|Q|)$, assuming a fixed, constant-size Σ.

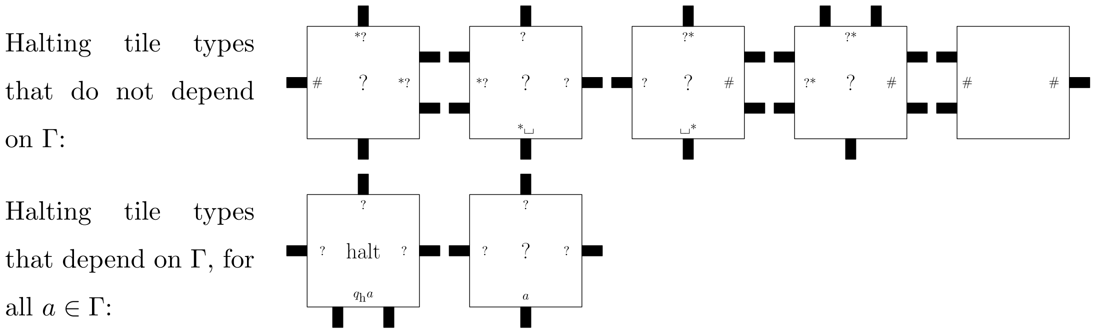


Fig. 12. Halting row tile types. The connector tile type (top row, rightmost tile type) is designed to attach to the east of the rightmost tile in the halting row via the '#' glue label. The purpose of the connector tile type is to connect the rightmost tile in the halting row of a stairstep of height h to the leftmost tile in the halting row of a stairstep of height $h+1$.

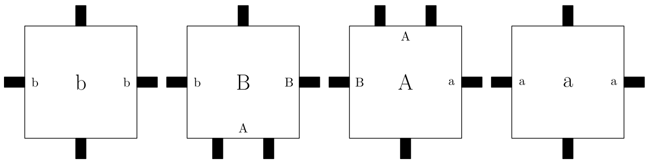


Fig. 13. Filler tile types. These tile types fill in the square around the TM simulation. These tile types are utilized in both the tile collector gadget and the simulation phase.

Tile collector gadget. We will not explicitly define each tile type in the tile collector gadget, but rather give a high-level overview of how it works.

At its core, the tile collector gadget is a diagonal path of tiles with notches at regular intervals to which certain classes of simulation tile types may attach, specifically copy, transition and halting tile types (see Figs. 10, 12 and 9). The reason we use a tile collector gadget in our construction is because it is possible for M to not execute all possible transitions as it computes, which means some simulation tile types in our construction have the potential to not be utilized. In the aTAM, such unutilized simulation tile types are not considered producible, since they do not bind to the seed-containing-assembly. However, in the seedless 2HAM, unutilized simulation tile types that were created to handle transitions that never execute represent terminal assemblies of size one, which obviously are not of the desired staircase shape. Therefore, in order for our construction to produce only terminal assemblies in the shape of the desired staircase, we must ensure that every tile type in our construction can grow into a terminal assembly in the shape of the desired staircase.

In our simulation construction (defined previously), there are seven classes of tile types that have the potential to not be used in the simulation of M. These seven classes of tile types consist of copy, transition and halting tile types (see Figs. 10, 12 and 9). Therefore, the tile collector gadget consists of a path of tiles with a notch for each type of transition, copy and halting tile. We design the tile collector gadget to ensure that each simulation tile binds to its respective notch via the same input sides that it would use if it were actually being utilized in the simulation of M (see the grey tiles in Fig. 14). The notches in the path are two tiles "deep" to account for the fact that some simulation tile types expose a north-facing, strength-2 output glue, to which some other simulation tile type may attach, e.g., the "qc" transition tile types that accept the state from either the left or the right. While there are only seven logical classes of such tile types, there may, depending on the definition of M, be many individual tile types, whence the actual length (number of tiles in the path) of the tile collector gadget is $3(3|Q \times \Gamma| + 4|\Gamma|)$. The glues of the tiles along the path are uniquely hard-coded so that the tile collector gadget will correctly assemble via any order of attachment. It is important to note that the tile complexity of the tile collector gadget is $O(|Q|)$, assuming a fixed, constant-size Σ.

We use filler tile types (see the 'a' and 'b' tile types in Fig. 13) to fill in a square around the diagonal tile collector path. As the filler tiles attach, two signals are propagated (the '@' symbols in Fig. 14): one of these signals is sent up the leftmost column and the other signal is sent right-to-left along the topmost row. When these two signals meet, they cooperate to place a tile (see the trio of tiles in the upper left corner of Fig. 14), to which the first input row tile type may attach (i.e., the leftmost tile type in Fig. 8). At this point, the initial configuration of M assembles along the top of the tile collector gadget, now a square, from left-to-right and the simulation tile types carry out the simulation of M. In each row of the simulation of M, the tape is expanded in both directions by one tape cell. As the space-time configuration history of M assembles on top of the tile collector square, a path of tiles assembles diagonally, down and to the left, starting at the lower left corner of the tile collector square, using the 'A' and 'B' tile type duo from Fig. 13. See Fig. 15 for a high-level example of the simulation of M.

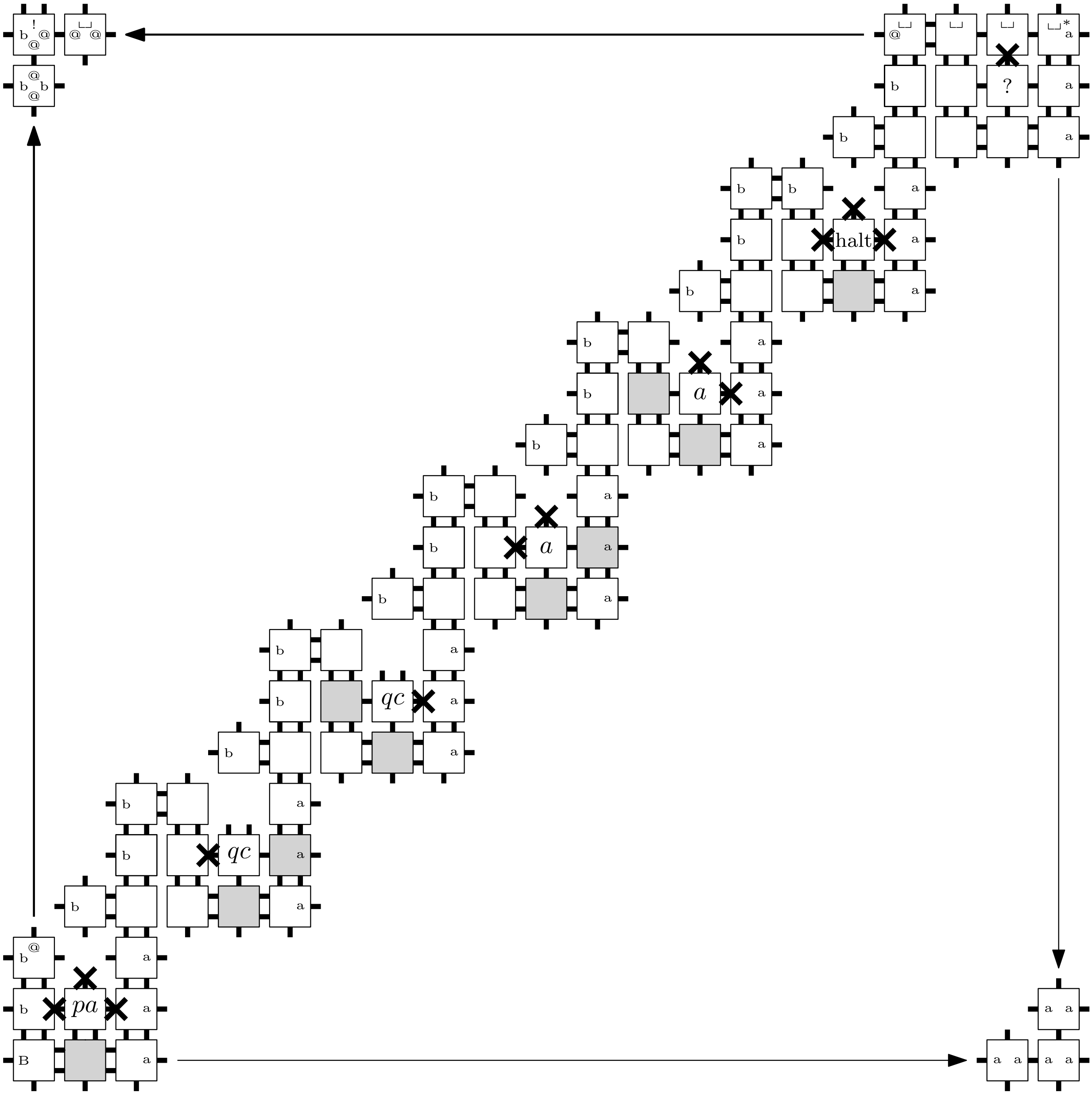


Fig. 14. The diagonal path of the tiles that make up the tile collector gadget. Also shown is a trio of tiles (in the upper left corner) that cooperate to initiate the assembly of the initial configuration of the simulation of M. The seven classes of simulation tile types are represented along the collector gadget path. Each simulation tile binds to the tile collector gadget via the same input sides that it would use if it were actually being utilized in the simulation of M. In this figure, we denote glue mismatches (between simulation tile types and the tile collector gadget) with a thick black X. Note that that each class of simulation tile types may, depending on the definition of M, consist of several individual tile types. Therefore, the actual length (number of tiles along the path) of the tile collector gadget is $3\,(3|Q \times \Gamma| + 4|\Gamma|)$. Note that the collector gadget, as presented here, and therefore our construction, does not uniquely produce a terminal assembly, but it can be modified to do so. It is also possible to modify the collector gadget to nondeterministically self-assemble into a constant size assembly to which only certain subsets of simulation (super)tiles may attach. However, for each combination of simulation supertiles, there will be a corresponding constant size collector gadget.

Binary counter. The TM simulation phase results in a Turing machine simulation square of size $W = 3\,(3|Q \times \Gamma| + 4|\Gamma|) + 2(t+1)$. To the topmost row of the Turing machine simulation square, "binary counter initialization" tile types (see Fig. 16 for detailed tile type definitions) attach nondeterministically. These tile types are essentially guessing a bit string $w \in \{0,1\}^W$, which is encoded along the north-facing glues of the topmost row of the TM simulation square (see the north-facing glues of the tile types defined in Fig. 16).

The remaining logical groups of tile types are shown in Figs. 16, 17 and 18.

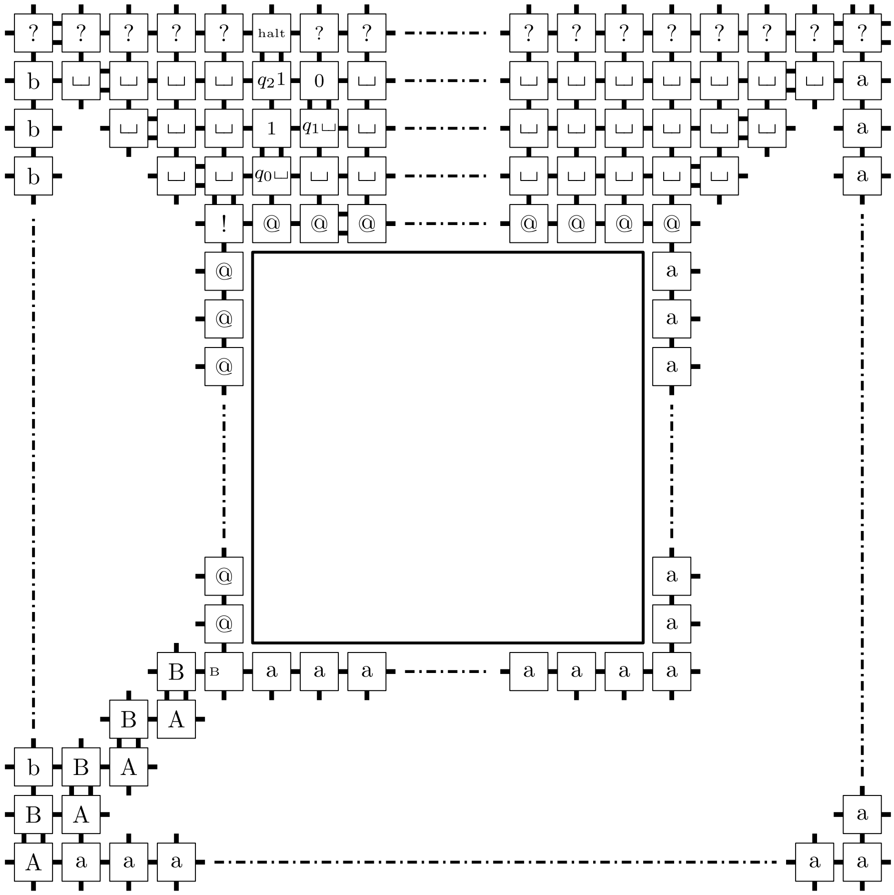


Fig. 15. The simulation of M is carried out on top of the tile collector square gadget. In this figure, the tile collector gadget is represented by the empty square, along with the single-tile-wide ring of tiles that encircle it. The size of the resulting square (tile collector gadget + simulation of M) is $3\,(3|Q \times \Gamma| + 4|\Gamma|) + 2(t+1)$, where t is the number of steps that M takes to halt on the empty string (three in this example).

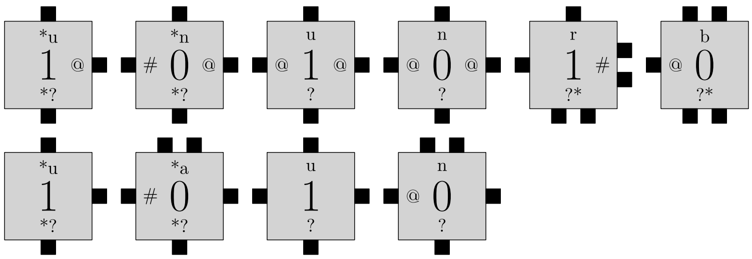


Fig. 16. These tile types assemble from right-to-left along the top of a TM simulation square. They mark the rightmost '0' bit in the nondeterministically chosen value, say w, and also initiate a vertical (optimal, non-zig-zag) binary counter [34,9] that counts from w up to $2^W - 1$.

Finally, the tile types shown in Figs. 17 and 18 build the rest of the stair step directly on top of the row of binary counter initialization tiles (see Fig. 19).

Putting it all together. Note that, in the 2HAM, individual stair steps may assemble completely and independently of other stair steps. By way the binary counter initialization tile types nondeterministically attach to the topmost row of the Turing machine simulation square, there is a one-to-one correspondence between (heights of) stair steps that are able to form and strings over the set $\{0,1\}^W$. Thus, we have 2^W total stair steps and, by definition, each stair step has some height $h \in \{3+W, \ldots, 2^W + W\}$. By the placement of the connector tile types, a stair step of height h may bind to the left side of another stair step if and only if the latter has height $h+1$. □

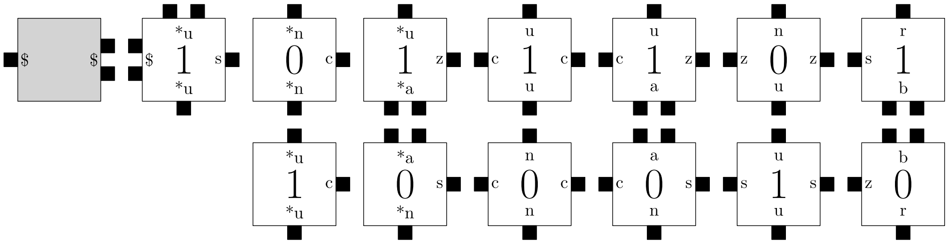


Fig. 17. These binary counter tile types implement an optimal, non-zig-zag binary counter, modified so that only special tile types are allowed to attach along the leftmost edge of the counter, i.e., the tile types whose north glues are prefixed with '*'. The purpose of the leftmost tile type in the top row is to connect the rightmost tile in the topmost row of a stairstep of height h to the leftmost tile in the second-from-the-topmost row of a stairstep of height $h+1$.

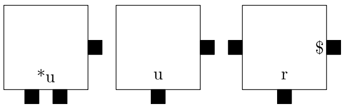


Fig. 18. These tile types cap off each stair step with an additional row of tiles to which one of the connector tiles will bind.

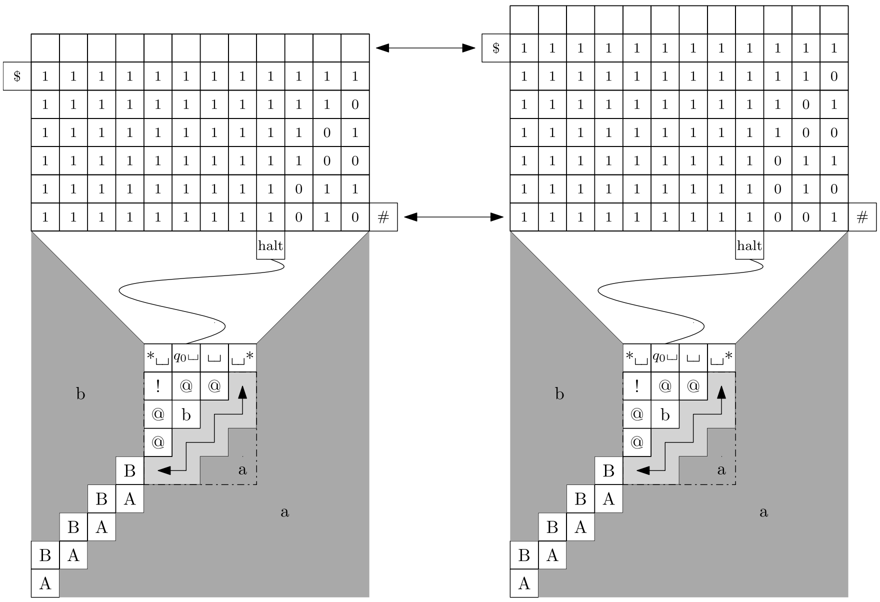


Fig. 19. Two consecutive stair steps coming together. All glue details are omitted. The light grey area is the diagonal path of the tile collector gadget. The dark grey area is filled in using the 'a' and 'b' filler tiles.

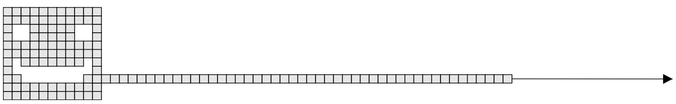


Fig. 20. A blob with an infinite tail.

Theorem 3.19 says that, at temperature $\tau = 2$, the 2HAM can be used to self-assemble certain shapes much $\left(\text{much}^{\text{much}^{\text{much}^{\cdot^{\cdot^{\cdot}}}}}\right)$ more efficiently (with respect to tile complexity) than what is possible in the aTAM.

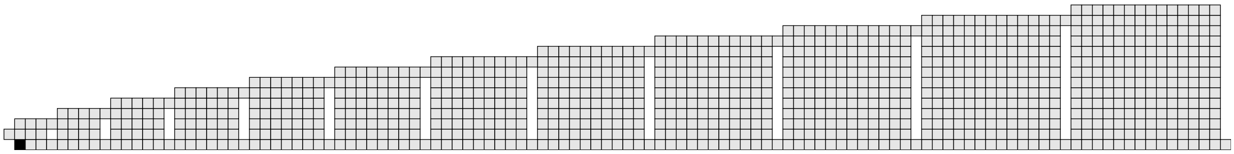


Fig. 21. A finite portion of the infinite staircase, denoted as S_∞. The black square represents the origin.

4. Infinite shapes

In this section, we examine a class of infinite shapes and whether or not they can (finitely) self-assemble in the two models.

We first note that it is easy to exhibit a class of infinite shapes that self-assemble in the aTAM but do not self-assemble in the 2HAM. Simply take any finite shape $X \subset \mathbb{Z}^2$ and union it with a one-way infinite line to get a kind of "blob with an infinite tail" (see Fig. 20 for an example of such a shape). Such shapes do not self-assemble in the 2HAM via a straightforward pumping lemma argument on the infinite tail portion of the shape (since the assembly sequence could proceed infinitely without attaching to the blob). However, we note that it is easy to take any such blob+tail shape and exhibit an aTAM TAS in which that shape self-assembles. To see this, simply create hard-coded tile types for the finite blob portion (with the seed tile placed at some location in the blob) and then have a single tile type that repeats infinitely in one direction for the tail portion. This construction also testifies to the finite self-assembly of a blob+tail shape in the 2HAM. In what follows, we will define an infinite shape that does not (finitely) self-assemble in the aTAM but does finitely self-assemble in the 2HAM.

Definition 4.1. For each $i \in \mathbb{N}$, let $B_i = (\{0,\ldots,i+2\} \times \{0,\ldots,i+2\}) \cup \{(i+3,0),(i+3,i+2)\}$ and $S_\infty = \{(-1,1)\} \cup \bigcup_{i=0}^{\infty}\left(B_i + \left(\frac{i(i+7)}{2},0\right)\right)$. Intuitively, the set S_∞ is essentially a succession of larger and larger squares that are connected by pairs of tiles positioned at the top right and bottom right of each square. See Fig. 21 for an example.

We now show how to finitely self-assemble infinite staircases in the 2HAM.

Theorem 4.2. *The infinite staircase S_∞ finitely self-assembles in the 2HAM.*

Intuitively, our construction for Theorem 4.2 proceeds as follows. We first assemble horizontal lines using three tile types: one to start the line, one to keep it going and one to stop the line. The tile that stops the line may attach nondeterministically at any step, whence lines of every length are able to form and any finite length line can grow into a line which stops (this is necessary for the system to finitely self-assemble S_∞). Each line of length k ultimately grows into a $k \times k$ square. Connector-tiles that attach to the left and right of each square ensure that only a $(k-1)\times(k-1)$ square may attach to the left of a $k \times k$ square.

Proof. Our proof is by construction, i.e., we will describe a 2HAM TAS $\mathcal{T} = (T,2)$ in which S_∞ finitely self-assembles. Our tile set T simply consists of two logical groups of tile types, which are shown in Figs. 22a and 22b. Intuitively, $\mathcal{T}$ finitely self-assembles S_∞ because if one simply assumes that the seed row tiles may only grow finite rows of tiles, then the construction works in essentially the same way as the construction for Theorem 3.19. See Fig. 23 for an example of two consecutive square stair steps coming together to bind with exactly strength 2. □

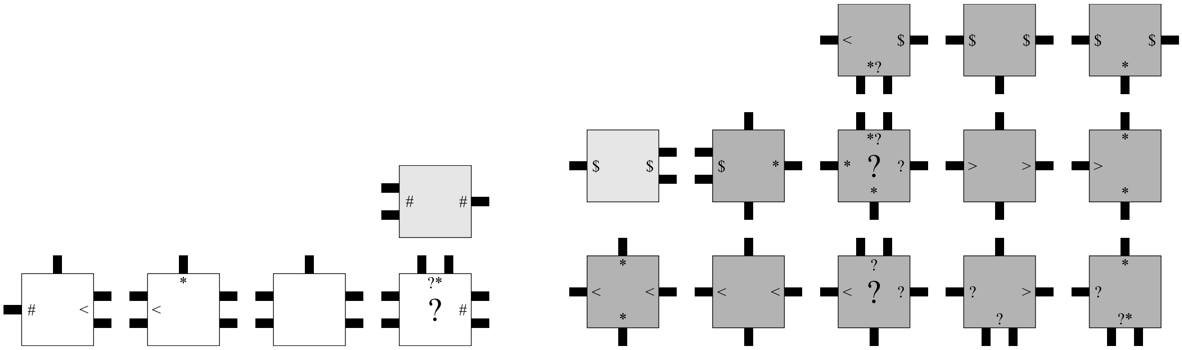
Note that S_∞ does not self-assemble in $\mathcal{T}$ in our 2HAM TAS because the seed row tile types (see Fig. 22a) could produce an infinite horizontal line assembly that does not contain a tile type with a "?". This infinite line structure would be terminal (in the limit) as a single, infinite row of tiles which does not grow into an assembly with shape S_∞.

It does not seem obvious whether S_∞ self-assembles in the 2HAM (in some other tile assembly system than the one we are describing here). Next, we have the following impossibility result for S_∞ in the aTAM.

Theorem 4.3. *The infinite staircase S_∞ does not finitely self-assemble in the aTAM.*

Before we prove Theorem 4.3, we define some notation and state some useful observations.

Notation. For all $i \in \mathbb{N}$, let $C_i^- = \left(\frac{(i+1)(i+8)}{2} - 1, 0\right)$ and $C_i^+ = \left(\frac{(i+1)(i+8)}{2} - 1, i+2\right)$ be the lower and upper connector points for column i, respectively. Let $C_i = \left\{C_i^-, C_i^+\right\}$. We call C_i^- and C_i^+ *siblings*.



(a) These seed row tile types nondeterministically assemble a row of tiles (possibly infinitely long) that specifies the dimension of the square to build. The rightmost edge of the row is specially marked as well as the second-to-leftmost tile.

(b) These square builder tile types build the remainder of the square whose size is defined by a given seed row. The remainder of the square is formed by "shifting" the '?' symbol from the rightmost tile in the seed row up and to the left until it reaches the upper left corner. We use the '*' symbol to indicate the rightmost edge of the square and the second-to-leftmost column so that the left connector tile type is properly placed. The self-assembly process of each square is essentially the same as the self-assembly process shown in Figure 4b in [22].

Fig. 22. The two logical groups of tile types that comprise the entirety of our tile set T.

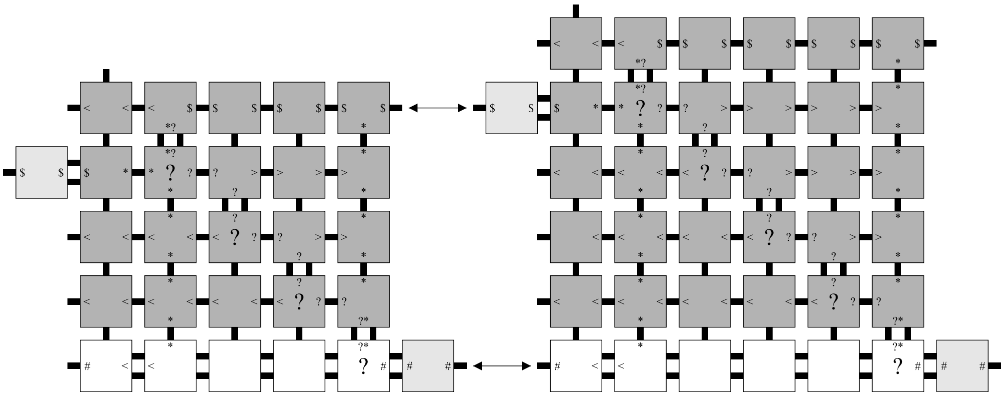


Fig. 23. An example of two consecutive square stair steps coming together.

In our proof of Theorem 4.3, we will use the *gap* function as it was originally defined in Section 3.2, i.e., $gap(i) = i + 1$.

Definition 4.4. Let $\vec{\alpha}$ be an assembly sequence in an aTAM TAS $\mathcal{T}$, such that, $\text{dom}\,\text{res}(\vec{\alpha}) = S_\infty$. For some $i \geq 0$, we say that a point $\vec{x} \in C_i$ is *ambitious* in $\vec{\alpha}$ if there exists a point $\vec{y} = (p, q) \in S_\infty$ satisfying the following conditions:

1. $\vec{x} \prec_{\vec{\alpha}} \vec{y}$,
2. $q \in \left\{ \left\lfloor \frac{gap(i)}{2} \right\rfloor, \left\lfloor \frac{gap(i)}{2} \right\rfloor + 1 \right\}$, and
3. $p < \frac{(i+2)(i+9)}{2} - 1$.

In other words, an ambitious point (at which a connector tile is placed) is one that can grow at least half way "up" (or "down") toward its sibling connector tile and it can do so without going through the next connector column C_{i+1}. This definition of an ambitious point is simply an adaptation of Definition 3.15 to S_∞ for the purposes of this proof.

Observation 4.5. Let $\vec{\alpha}$ be an assembly sequence in an aTAM TAS $\mathcal{T}$, with result α, such that, $\text{dom}\,\alpha = S_\infty$. If $\mathcal{C}^- = \left\langle C_0^-, C_1^-, \ldots \right\rangle$ and $\mathcal{C}^+ = \left\langle C_0^+, C_1^+, \ldots \right\rangle$, then, either $\mathcal{C}^-$ contains an infinite subsequence of points $\left\langle C_{i_0}^-, C_{i_1}^-, \ldots \right\rangle$, such that, for all $j \in \mathbb{N}$, $C_{i_j}^-$ is ambitious in $\vec{\alpha}$, or $\mathcal{C}^+$ contains a similarly defined infinite sequence of ambitious points in $\vec{\alpha}$.

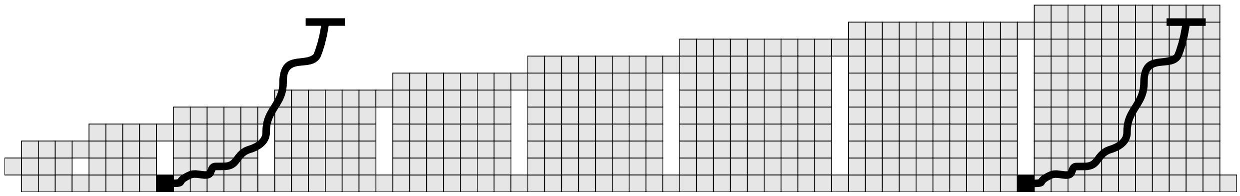


Fig. 24. Proof idea of Theorem 4.3. The black tiles represent the locations C_r^- and C_s^- respectively (C_r^- on the left and C_s^- on the right). The easternmost squiggly thick black line represents an ambitious placement of a tile by some assembly sequence, say $\vec{\alpha}$. The westernmost squiggly thick black line represents another assembly sequence, say $\widehat{\vec{\alpha}}$, trying to mimic $\vec{\alpha}$ and, in doing so, erroneously places a tile outside of S_∞.

Note that, in Observation 4.5, if neither $\mathcal{C}^-$ nor $\mathcal{C}^+$ contain an infinite subsequence of ambitious points, then S_∞ cannot self-assemble in $\vec{\alpha}$. To see this, observe that, if $\vec{\alpha}$ places a tile in the "middle" row of a square of S_∞, then there must be some ambitious point that facilitates the placement of such a tile (here, we say "middle" because we are off by one by the way we define *gap*). However, if there are only finitely many ambitious connector points, then, eventually, $\vec{\alpha}$ will not be able to place tiles in the "middle" rows of squares of S_∞, i.e., the square immediately east of C_i in S_∞, for some value of i, will not have tiles placed at any points with y-values of either $\left\lfloor \frac{gap(i)}{2} \right\rfloor$ or $\left\lfloor \frac{gap(i)}{2} \right\rfloor + 1$.

From this point on, we will assume, without loss of generality, that $\mathcal{C}^-$ contains an infinite subsequence of ambitious points $\left\langle C_{i_0}^-, C_{i_1}^-, \ldots \right\rangle$.

Observation 4.6. Let $\vec{\alpha}$ be an assembly sequence in an aTAM TAS $\mathcal{T}$, with result α, such that, $\operatorname{dom}\alpha = S_\infty$. If $\left\langle C_{i_0}^-, C_{i_1}^-, \ldots \right\rangle$ is an infinite subsequence of $\mathcal{C}^-$, such that, for all $j \geq 0$, $C_{i_j}^-$ is an ambitious point in $\vec{\alpha}$, then there exist indices $r = i_j, s = i_k \in \mathbb{N}$ satisfying the following conditions:

1. $s \geq 24$,
2. the points C_r^- and C_s^- are east of the seed tile of $\mathcal{T}$
3. $\alpha\left(C_r^-\right) = \alpha\left(C_s^-\right)$, and
4. $gap(s) > 5\left(gap(r) + 3\right)$.

We are now ready to prove Theorem 4.3.

Proof of Theorem 4.3. Let $\mathcal{T}$ be an arbitrary aTAM TAS and assume, for the sake of contradiction, that $\mathcal{T}$ finitely self-assembles S_∞. Then there exists an assembly sequence in $\mathcal{T}$, say $\vec{\alpha}$, with result α, such that $\operatorname{dom}\alpha = S_\infty$. We will derive a contradiction by showing that there is necessarily some finite producible assembly $\widehat{\alpha} \in \mathcal{A}[\mathcal{T}]$, such that, $\operatorname{dom}\widehat{\alpha} - S_\infty \neq \varnothing$, which violates finite self-assembly.

By Observation 4.5, the infinite sequence $\mathcal{C}^- = \left\langle C_0^-, C_1^-, \ldots \right\rangle$ contains an infinite subsequence of ambitious points in $\vec{\alpha}$, say $\left\langle C_{i_0}^-, C_{i_1}^-, \ldots \right\rangle$. Therefore, let $r, s \in \mathbb{N}$ be the indices given by Observation 4.6. By the definition of *gap* and the relationship between r and s, it follows that

$$gap(s) > 5\left(gap(r) + 3\right) \Leftrightarrow r < \frac{s - 19}{5}. \tag{4.1}$$

A sketch of the remainder of the proof is given in Fig. 24.

Let $m \in \mathbb{N}$ be the number of locations in the square immediately east of C_s^- that strictly depend on C_s^- and define the locations $\vec{y}_0, \vec{y}_1, \ldots, \vec{y}_{m-1}$, such that, for all $0 \leq j < m$, $C_s^- \prec_{\vec{\alpha}} \vec{y}_j$ and $i_{\vec{\alpha}}\left(\vec{y}_0\right) < i_{\vec{\alpha}}\left(\vec{y}_1\right) < \cdots < i_{\vec{\alpha}}\left(\vec{y}_{m-1}\right)$. We will now construct a new assembly sequence $\widehat{\vec{\alpha}}$ in $\mathcal{T}$ as follows. Let $\widehat{\vec{\alpha}}$ be such that $\widehat{\vec{\alpha}}$ behaves exactly like $\vec{\alpha}$ but only places tiles at locations on which C_r^- strictly depends, until it places a tile at C_r^-, at which point, $\widehat{\vec{\alpha}}$ places, for all $0 \leq j < m$, the tile type $\alpha\left(\vec{y}_j\right)$ at $\vec{y}_j - \left(\vec{x}_s - \vec{x}_r\right)$ and in the same relative order. Note that $\widehat{\vec{\alpha}}$ is a valid assembly sequence because:

1. each assembly of $\widehat{\vec{\alpha}}$ is finite (under finite self-assembly, an infinite assembly, whose domain is a strict subset of S_∞, could potentially be terminal),
2. for all $0 \leq j < m$, $C_s^- \prec_{\vec{\alpha}} \vec{y}_j$,
3. $\alpha\left(C_r^-\right) = \alpha\left(C_s^-\right)$, and
4. immediately after $\widehat{\vec{\alpha}}$ places the tile type $\alpha\left(C_s^-\right)$ at C_r^-, $\widehat{\vec{\alpha}}$ has yet to place a tile at any location that is east of C_r^-. This condition holds because $\widehat{\vec{\alpha}}$ only places tiles at positions on which C_r^- strictly depends, which means C_r^- is the first location in its column to receive a tile under $\widehat{\vec{\alpha}}$.

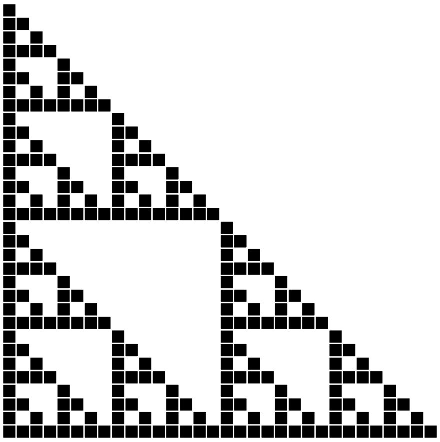


Fig. 25. Discrete Sierpinski triangle, denoted **S**.

Since C_s^- is ambitious, there must be an index l, such that, $\vec{y}_l$ is located in the square immediately east of the column C_s. Therefore, if $\vec{y}_l = (p, q)$, then

$$\begin{aligned} q &\geq \left\lfloor \frac{gap(s)}{2} \right\rfloor = \left\lfloor \frac{s+1}{2} \right\rfloor > \frac{s+1}{2} - 1 \\ &> \frac{2s+5}{5} \text{ (for } s > 15) \\ &= \frac{s-19}{5} + 4 + \frac{s+4}{5} > r + 4 + \frac{s+4}{r+4} \quad \text{(by (4.1) and assuming} r > 0) \\ &> r + 3 + \left\lceil \frac{s+4}{r+4} \right\rceil = gap(r) + 2 + \left\lceil \frac{gap(s)+3}{gap(r)+3} \right\rceil . \end{aligned}$$

By the definition of ambitious (Definition 4.4), when placing a tile at the point $\vec{y}_l$, $\vec{\alpha}$ can only grow east from C_s^- by at most $gap(s) + 3$ points. Thus, the number of squares, east of the point C_r^-, through which $\widehat{\vec{\alpha}}$ may grow is less than $\left\lceil \frac{gap(s)+3}{gap(r)+3} \right\rceil$. By the definition of S_∞, the height of the square that is $\left\lceil \frac{gap(s)+3}{gap(r)+3} \right\rceil$ squares to the east of C_r^- is $gap(r) + 2 + \left\lceil \frac{gap(s)+3}{gap(r)+3} \right\rceil$. Therefore, by the above chain of inequalities, when placing a tile at $\vec{y}_l = (p, q)$, $\widehat{\vec{\alpha}}$ will have no choice but to grow too far "up", i.e., at least to the point (p, q), and hence out of S_∞ – even if it tries to grow "east" as far as it possibly can and into a taller square, past C_r^-, before growing "up" to place $\vec{y}_l$. □

Corollary 4.7. *The infinite staircase S_∞ does not self-assemble in the aTAM.*

Proof. Self-assembly implies finite self-assembly. Theorem 4.3 says that S_∞ does not finitely self-assemble in the aTAM, therefore, it does not self-assemble in the aTAM. □

We previously showed that the infinite staircase finitely self-assembles in the 2HAM. We also mentioned that "blobs with infinite tails" also finitely self-assemble in the 2HAM. We now show, while perhaps not overly-surprising, that not every infinite shape finitely self-assembles in the 2HAM, i.e., there is an example of a shape, e.g., the discrete Sierpinski triangle (see Fig. 25) that does not finitely self-assemble in the 2HAM.

Definition 4.8. Let $V = \{(1, 0), (0, 1)\}$ be the *generator* of the discrete Sierpinski triangle and define the stages $S_0 = \{(0, 0)\}$, and for integers $i > 0$, $S_{i+1} = S_i \cup (S_i + 2^i V)$, where $A + cB = \{\vec{m} + c\vec{n} \mid \vec{m} \in A \text{ and } \vec{n} \in B\}$. The *discrete Sierpinski triangle*, denoted as **S**, is defined as the infinite union $\mathbf{S} = \bigcup_{i=0}^{\infty} S_i$. A finite portion of this infinite shape is shown in Fig. 25.

Theorem 4.9. *The discrete Sierpinski triangle does not finitely self-assemble in the 2HAM (at any temperature).*

Before we prove Theorem 4.9, we must first develop some machinery, starting with the following observation about non-cooperative self-assembly in the 2HAM.

Observation 4.10. Let $\mathcal{T} = (T, 1)$ be any 2HAM TAS and $\widetilde{\alpha} \in \mathcal{A}[\mathcal{T}]$. If $\alpha \in \widetilde{\alpha}$, then, for any subassembly $\alpha' \sqsubseteq \alpha$, $\widetilde{\alpha}' \in \mathcal{A}[\mathcal{T}]$.

Observation 4.10 is obvious since, by definition of a subassembly, α' must be stable and therefore every tile must be bound to at least one other. Also, since any subassembly is a tree, and at $\tau = 1$ any bond is sufficient for tiles to attach, it's possible for any subassembly to grow one tile at a time.

We now define notation for certain substructures of **S**.

Notation. For all $i \geq 1$, the i^{th} *vertical branch* of **S** is the line $\{2^i\} \times \{0, \ldots, 2^i - 1\}$.

The following is obvious by inspection.

Observation 4.11. For all $k \geq 1$, S_k contains $k - 1$ vertical branches.

We are now ready to prove Theorem 4.9.

Proof of Theorem 4.9. To prove Theorem 4.9, for the sake of contradiction, we assume that there exists some 2HAM TAS $\mathcal{T} = (T, \tau)$ that finitely self-assembles the discrete Sierpinski triangle **S**. We will derive a contradiction by showing that there is some finite producible supertile $\widetilde{\widehat{\alpha}} \in \mathcal{A}[\mathcal{T}]$, such that, for some $\widehat{\alpha} \in \widetilde{\widehat{\alpha}}$, $\operatorname{dom} \widehat{\alpha} - \mathbf{S} \neq \varnothing$, assuming $\widehat{\alpha}$ is translated so that its lower left most tile is placed at the origin. (By the definition of **S**, namely its particular tree structure, any producible super tile of $\mathcal{T}$ must have a well-defined lower-leftmost location in its domain.) This would violate one of the conditions of finite self-assembly, because $\widetilde{\widehat{\alpha}}$ would be an example of a finite producible supertile that cannot grow into a supertile with shape **S**.

Since the underlying grid graph of **S** is a tree, we may assume that $\mathcal{T} = (T, \tau = 1)$ because all glues must be τ-strength. Let $\tilde{\vec{\alpha}}$ be a supertile assembly sequence in $\mathcal{T}$ with $\operatorname{res}(\tilde{\vec{\alpha}}) = \widetilde{\alpha}$ and, $\alpha \in \widetilde{\alpha}$ where $(0, 0) \in \alpha$, and $\operatorname{dom} \alpha = \mathbf{S}$. Let $k \geq |T| + 2$. Consider an assembly α_k satisfying the following conditions: (1) $\alpha_k \sqsubseteq \alpha$ and (2) $\operatorname{dom} \alpha_k = S_k$. By Observation 4.10, $\widetilde{\alpha}_k \in \mathcal{A}[\mathcal{T}]$, i.e., the supertile induced by α_k is a producible supertile in $\mathcal{T}$. Note that the tile type placed at the lower leftmost position of α_k must have a south glue of either strength-0 or which does not match the north glue label/strength of any tile type in T and a west glue of either strength-0 or which doesn't match the east glue label/strength of any tile type in T.

By Observation 4.11, and our choice of k, S_k contains at least $k - 1 \geq |T| + 2 - 1 = |T| + 1$ vertical branches. Thus, there exist two numbers, say i and j, satisfying $1 \leq i < j \leq k$, such that in α_k the north glue of $\alpha_k\left(2^i, 0\right)$ is the same as the north glue of $\alpha_k\left(2^j, 0\right)$.

Define a subassembly of α_k, β_i, as follows: for all $\vec{x} \in \{2^i\} \times \{1, \ldots, 2^i - 1\}$, $\beta_i\left(\vec{x}\right) = \alpha_k\left(\vec{x}\right)$, and β_i is undefined at all other points. By Observation 4.10, $\widetilde{\beta_i} \in \mathcal{A}[\mathcal{T}]$. Moreover, the north glue of $\beta_i\left(2^i, 2^i - 1\right)$ has strength 0 or does not match the south label/strength of any other tile type in T, else a tile could attach to it, which would be outside of **S**.

Next, define the assembly $\widehat{\alpha}_k$, such that, for all $\vec{x} \in \{0, \ldots, 2^k - 1\} \times \{0\}$, $\widehat{\alpha}_k\left(\vec{x}\right) = \alpha_k\left(\vec{x}\right)$ and $\widehat{\alpha}_k$ is undefined at all other points. By Observation 4.10, $\widetilde{\widehat{\alpha}}_k \in \mathcal{A}[\mathcal{T}]$.

Since the south glue of $\beta_i\left(2^i, 1\right)$ is the same as the north glue of $\alpha_k\left(2^j, 0\right)$ and we have $\tau = 1$, the translated assembly $\beta_i + \left(2^j - 2^i, 0\right)$ can attach to $\widehat{\alpha}_k$ via the south glue of $\beta_i\left(2^i, 1\right)$ and the north glue of $\alpha_k\left(2^j, 0\right)$.

Let $\widehat{\alpha}$ denote the assembly resulting from attaching $\beta_i + \left(2^j - 2^i, 0\right)$ to $\widehat{\alpha}_k$ in the previously described fashion, i.e., $\widehat{\alpha} = \widehat{\alpha}_k \cup \left(\beta_i + \left(2^j - 2^i, 0\right)\right)$. Intuitively, (the shape of) $\widehat{\alpha}$ is the bottom row of stage k of the Sierpinski triangle to which an undergrown vertical branch is attached in the wrong location (and it cannot grow taller because its north most glue either has strength 0 or does not match the south glue of any other tile type. Thus, $\widetilde{\widehat{\alpha}}$ is an example of a finite supertile, producible in $\mathcal{T}$ that cannot grow into a supertile, say $\widetilde{\alpha}$, where $\operatorname{dom} \widetilde{\alpha} = \mathbf{S}$. $\square$

Corollary 4.12. *The discrete Sierpinski triangle does not self-assemble in the 2HAM.*

Proof. Self-assembly implies finite self-assembly. Theorem 4.9 says that **S** does not finitely self-assemble in the 2HAM, therefore, it does not self-assemble in the 2HAM. $\square$

5. Conclusion

In this paper, we studied the effect that hierarchical self-assembly has on reducing tile complexity of various shapes. We leave open the following questions:

1. We showed $\mathcal{K}^2_{\text{aTAM}}(S_n) = \Omega(n)$ (see Theorem 3.13) and $\mathcal{K}^2_{\text{aTAM}}(S_n) = O\left(n^2\right)$ (see Theorem 3.18). What are tight bounds on $K^2_{\text{aTAM}}(S_n)$?
2. We showed that there is a finite shape that self-assembles more efficiently asymptotically in the aTAM than it does in the 2HAM (see Theorem 3.11). Here, the aTAM tile set makes use of blocking, where a repeating path of tiles growing to the south eventually crashes into a previously placed tile. Of course, the terminal assembly in this example was not fully connected. If the final assembly is required to be fully connected, i.e., no glue mismatches, then is there a shape that can be self-assembled asymptotically more efficiently in the aTAM than in the 2HAM?

3. The infinite staircase S_∞ finitely self-assembles in the 2HAM but not the aTAM. Does it self-assemble in the 2HAM?

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank the hard work of the anonymous reviewers who worked so hard to provide valuable feedback which greatly improved this paper.

The work of Sarah Cannon was supported in part by National Science Foundation grants CCF-0830734 and CBET-0941538. The work of Erik Demaine, Martin Demaine, and Sarah Eisenstat was supported in part by National Science Foundation grant CDI-0941538. The work of Matthew Patitz and Robert Schweller was supported in part by National Science Foundation grant CCF-1117672, and Matthew Patitz's work was also supported in part by National Science Foundation grant CAREER-1553166. The work of Andrew Winslow was supported in part by National Science Foundation grants CCF-0830734 and CBET-0941538.

References

[1] E. Winfree, Algorithmic self-assembly of DNA, Ph.D. thesis, California Institute of Technology, June 1998.
[2] D. Woods, D. Doty, C. Myhrvold, J. Hui, F. Zhou, P. Yin, E. Winfree, Diverse and robust molecular algorithms using reprogrammable DNA self-assembly, Nature 567 (2019) 366–372.
[3] H.-L. Chen, R. Schulman, A. Goel, E. Winfree, Reducing facet nucleation during algorithmic self-assembly, Nano Lett. 7 (9) (2007) 2913–2919, https://doi.org/10.1021/nl070793o.
[4] P.W.K. Rothemund, N. Papadakis, E. Winfree, Algorithmic self-assembly of DNA Sierpinski triangles, PLoS Biol. 2 (12) (2004) e424.
[5] C.G. Evans, E. Winfree, DNA sticky end design and assignment for robust algorithmic self-assembly, in: D. Soloveichik, B. Yurke (Eds.), Proceedings, DNA Computing and Molecular Programming - 19th International Conference, DNA 19, Tempe, AZ, USA, September 22-27, 2013, in: Lecture Notes in Computer Science, vol. 8141, Springer, 2013, pp. 61–75.
[6] C.G. Evans, Crystals that count! Physical principles and experimental investigations of DNA tile self-assembly, Ph.D. thesis, California Institute of Technology, 2014.
[7] R. Schulman, E. Winfree, Programmable control of nucleation for algorithmic self-assembly, SIAM J. Comput. 39 (4) (12/2009) 1581–1616, https://doi.org/10.1137/070680266.
[8] R.D. Barish, R. Schulman, P.W. Rothemund, E. Winfree, An information-bearing seed for nucleating algorithmic self-assembly, Proc. Natl. Acad. Sci. 106 (15) (2009) 6054–6059, https://doi.org/10.1073/pnas.0808736106.
[9] Q. Cheng, G. Aggarwal, M.H. Goldwasser, M.-Y. Kao, R.T. Schweller, P.M. de, Espanés, complexities for generalized models of self-assembly, SIAM J. Comput. 34 (2005) 1493–1515.
[10] J.E. Padilla, M.J. Patitz, R.T. Schweller, N.C. Seeman, S.M. Summers, X. Zhong, Asynchronous signal passing for tile self-assembly: fuel efficient computation and efficient assembly of shapes, Int. J. Found. Comput. Sci. 25 (4) (2014) 459–488.
[11] M.J. Patitz, R.T. Schweller, S.M. Summers, Exact shapes and Turing universality at temperature 1 with a single negative glue, in: Proceedings of the 17th International Conference on DNA Computing and Molecular Programming, DNA'11, 2011, pp. 175–189.
[12] B. Fu, M.J. Patitz, R.T. Schweller, R. Sheline, Self-assembly with geometric tiles, in: Proceedings of the 39th International Colloquium on Automata, Languages and Programming, ICALP, 2012, pp. 714–725.
[13] J. Hendricks, M.J. Patitz, T.A. Rogers, S.M. Summers, The power of duples (in self-assembly): It's not so hip to be square, Theor. Comput. Sci. (2015), https://doi.org/10.1016/j.tcs.2015.12.008, http://www.sciencedirect.com/science/article/pii/S030439751501169X.
[14] S.P. Fekete, J. Hendricks, M.J. Patitz, T.A. Rogers, R.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, 2015, pp. 148–167.
[15] O. Gilbert, J. Hendricks, M.J. Patitz, T.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, 2016, pp. 937–956.
[16] B. Behsaz, J. Maňuch, L. Stacho, Turing universality of step-wise and stage assembly at temperature 1, in: D. Stefanovic, A. Turberfield (Eds.), DNA Computing and Molecular Programming, in: Lecture Notes in Computer Science, vol. 7433, Springer, Berlin Heidelberg, 2012, pp. 1–11.
[17] E.D. Demaine, M.L. Demaine, S.P. Fekete, M. Ishaque, E. Rafalin, R.T. Schweller, D.L. Souvaine, Staged self-assembly: nanomanufacture of arbitrary shapes with $O(1)$ glues, Nat. Comput. 7 (3) (2008) 347–370.
[18] J.I. Lathrop, J.H. Lutz, S.M. Summers, Strict self-assembly of discrete Sierpinski triangles, Theor. Comput. Sci. 410 (2009) 384–405.
[19] M.J. Patitz, S.M. Summers, Self-assembly of discrete self-similar fractals, Nat. Comput. 1 (2010) 135–172.
[20] J. Hendricks, J. Obseth, M.J. Patitz, S.M. Summers, Hierarchical growth is necessary and (sometimes) sufficient to self-assemble discrete self-similar fractals, in: Proceedings of the 24th International Conference on DNA Computing and Molecular Programming, (DNA 24), Shandong Normal University, Jinan, China October 8-12, Shandong Normal University, Jinan, China, 2018, pp. 87–104.
[21] D. Furcy, S.M. Summers, Scaled pier fractals do not strictly self-assemble, Nat. Comput. 16 (2) (2017) 317–338.
[22] P.W.K. Rothemund, E. 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, ACM, Portland, Oregon, United States, 2000, pp. 459–468.
[23] M.J. Patitz, T.A. Rogers, R.T. Schweller, S.M. Summers, A. Winslow, Resiliency to multiple nucleation in temperature-1 self-assembly, in: Proceedings of the 22nd International Conference on DNA Computing and Molecular Programming, (DNA 22), September 4-8, 2016, Ludwig-Maximilians-Universität, Munich, Germany, 2016, pp. 98–113.
[24] E.D. Demaine, M.J. Patitz, T.A. Rogers, R.T. Schweller, S.M. Summers, D. Woods, The two-handed tile assembly model is not intrinsically universal, Algorithmica 74 (2) (2016) 812–850, https://doi.org/10.1007/s00453-015-9976-y.
[25] J. Hendricks, M.J. Patitz, T.A. Rogers, The simulation powers and limitations of higher temperature hierarchical self-assembly systems, Fundam. Inform. 155 (1–2) (2017) 131–162, https://doi.org/10.3233/FI-2017-1579.

[26] R.T. Schweller, A. Winslow, T. Wylie, Nearly constant tile complexity for any shape in two-handed tile assembly, Algorithmica 81 (8) (2019) 3114–3135, https://doi.org/10.1007/s00453-019-00573-w.
[27] S. Cannon, E.D. Demaine, M.L. Demaine, S. Eisenstat, M.J. Patitz, R.T. Schweller, S.M. Summers, A. Winslow, Two hands are better than one (up to constant factors): self-assembly in the 2HAM vs. aTAM, in: N. Portier, T. Wilke (Eds.), STACS, in: LIPIcs, vol. 20, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013, pp. 172–184.
[28] H. Wang, Proving theorems by pattern recognition – II, Bell Syst. Tech. J. XL (1) (1961) 1–41.
[29] L.M. Adleman, J. Kari, L. Kari, D. Reishus, P. Sosík, The undecidability of the infinite ribbon problem: implications for computing by self-assembly, SIAM J. Comput. 38 (6) (2009) 2356–2381, https://doi.org/10.1137/080723971.
[30] D. Doty, M.J. Patitz, D. Reishus, R.T. Schweller, S.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), 2010, pp. 417–426.
[31] Z. Abel, N. Benbernou, M. Damian, E.D. Demaine, M.L. Demaine, R. Flatland, S.D. Kominers, R.T. Schweller, Shape replication through self-assembly and RNAse enzymes, in: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, 2010, pp. 1045–1064.
[32] E.D. Demaine, M.J. Patitz, R.T. Schweller, S.M. Summers, Self-assembly of arbitrary shapes using RNAse enzymes: meeting the Kolmogorov bound with small scale factor (extended abstract), in: T. Schwentick, C. Dürr (Eds.), 28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011), in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 9, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011, pp. 201–212, http://drops.dagstuhl.de/opus/volltexte/2011/3011.
[33] P.W.K. Rothemund, Theory and experiments in algorithmic self-assembly, Ph.D. thesis, University of Southern California, December 2001.
[34] L. Adleman, Q. Cheng, A. Goel, M.-D. Huang, Running time and program size for self-assembled squares, in: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, Hersonissos, Greece, 2001, pp. 740–748.