# Improved Lower and Upper Bounds on the Tile Complexity of Uniquely Self-Assembling a Thin Rectangle Non-Cooperatively in 3D

**David Furcy**
Computer Science Department, University of Wisconsin Oshkosh, WI, USA

**Scott M. Summers**
Computer Science Department, University of Wisconsin Oshkosh, WI, USA

**Logan Withers**
Computer Science Department, University of Wisconsin Oshkosh, WI, USA

## Abstract

We investigate a fundamental question regarding a benchmark class of shapes in one of the simplest, yet most widely utilized abstract models of algorithmic tile self-assembly. More specifically, we study the directed tile complexity of a $k \times N$ thin rectangle in Winfree's ubiquitous abstract Tile Assembly Model, assuming that cooperative binding cannot be enforced (temperature-1 self-assembly) and that tiles are allowed to be placed at most one step into the third dimension (just-barely 3D). While the directed tile complexities of a square and a scaled-up version of any algorithmically specified shape at temperature 1 in just-barely 3D are both asymptotically the same as they are (respectively) at temperature 2 in 2D, the (nearly tight) bounds on the directed tile complexity of a thin rectangle at temperature 2 in 2D are not currently known to hold at temperature 1 in just-barely 3D. Motivated by this discrepancy, we establish new lower and upper bounds on the directed tile complexity of a thin rectangle at temperature 1 in just-barely 3D. The proof of our upper bound is based on the construction of a novel, just-barely 3D temperature-1 self-assembling counter. Each value of the counter is comprised of $k-2$ digits, represented in a geometrically staggered fashion within $k$ rows. This nearly optimal digit density, along with the base of the counter, which is proportional to $N^{\frac{1}{k-1}}$, results in an upper bound of $O\left(N^{\frac{1}{k-1}} + \log N\right)$, and is an asymptotic improvement over the previous state-of-the-art upper bound. On our way to proving our lower bound, we develop a new, more powerful type of specialized Window Movie Lemma that lets us bound the number of "sufficiently similar" ways to assign glues to a set (rather than a sequence) of fixed locations. Consequently, our lower bound, $\Omega\left(N^{\frac{1}{k}}\right)$, is also an asymptotic improvement over the previous state-of-the-art lower bound.

**2012 ACM Subject Classification** Theory of computation → Models of computation

**Keywords and phrases** Self-assembly, algorithmic self-assembly, tile self-assembly

**Digital Object Identifier** 10.4230/LIPIcs.DNA.27.4

## 1 Introduction

A key objective in algorithmic self-assembly is to characterize the extent to which an algorithm can be converted to an efficient self-assembling system comprised of discrete, distributed and disorganized units that, through random encounters with, and locally-defined reactions to each other, coalesce into a terminal assembly having a desirable form or function. In this paper, we study a fundamental theoretical question regarding a benchmark class of shapes in one of the simplest yet most popular abstract models of algorithmic self-assembly.

Ubiquitous throughout the theory of tile self-assembly, Erik Winfree's abstract Tile Assembly Model (aTAM) [26] is a discrete mathematical model of DNA tile self-assembly [23] that augments classical Wang tiling [25] with a mechanism for automatic growth. In the


aTAM, a DNA tile is represented by a unit square (or cube) tile type that may neither rotate, reflect, nor fold. Each side of a tile type is decorated with a glue consisting of both a non-negative integer strength and a string label, the symbols of which are drawn from some fixed alphabet. A tile set is a finite set of tile types, from which infinitely many tiles of each type may be instantiated. If one tile is positioned at an unoccupied location Manhattan distance 1 away from another tile and their opposing glues are equal, then the two tiles bind with the strength of the adjacent glues. A special seed tile type is designated and a seed tile, which defines the seed-containing assembly, is placed at some fixed location. During the process of self-assembly, a sequence of tiles bind to and never detach from the seed-containing assembly, provided that each one, in a non-overlapping fashion, binds to one or more tiles in the seed-containing assembly with total strength at least a certain positive integer value called the temperature. If the temperature is greater than or equal to 2, then it is possible to enforce cooperative binding, where a tile may be prevented from binding at a certain location until at least two adjacent locations become occupied by tiles. Otherwise, only non-cooperative binding is allowed (temperature-1 self-assembly). A fundamental theoretical question in tile self-assembly is determining the effect of the value of the temperature on the computational and geometric expressiveness of tile self-assembly.

To that end, temperature-1 self-assembly has been shown to hinder the efficient self-assembly of shapes when tile assemblies are required to be fully connected [22] or contain no glue mismatches [16]. Temperature-1 self-assembly is also neither intrinsically universal [17,19], nor capable of bounded Turing computation [19]. Recently and quite remarkably, Meunier, Regnault and Woods [18] established a general pumping lemma for temperature-1 self-assembly, nearly proving a conjecture by Doty, Patitz and Summers [6] on the computational weakness of temperature-1 self-assembly.

Interestingly, temperature-1 self-assembly does not limit the computational or geometric expressiveness of generalizations of the aTAM [4, 5, 7, 8, 12]. This is also true even when the generalization only adds a small number of additional features to the model, like a single negative glue [20], duple tiles [13], or another plane in which (cubic) 3D tiles are allowed to be placed [3, 9–11]. The latter variant is colloquially known as "just-barely" 3D self-assembly. In this paper, we study the limitations of temperature-1 self-assembly for unique shape-building in the just-barely 3D aTAM. We are specifically interested in studying the *directed tile complexity* of a given target shape, or the size of the smallest tile set that, regardless of the order in which tiles bind to the seed-containing assembly, always self-assembles into a unique terminal assembly of tiles that are placed on and only on points of a given target shape.

Although temperature-1 self-assembly cannot enforce cooperative binding, there is a striking resemblance of its computational and geometric expressiveness in just-barely 3D, to that of temperature-2 self-assembly in 2D, with respect to the directed tile complexity of two benchmark shapes: a square and a scaled-up version of any algorithmically specified shape. Adleman, Cheng, Goel and Huang [1] proved, using optimal base conversion, that the directed tile complexity of an $N \times N$ square at temperature 2 in 2D is $O\left(\frac{\log N}{\log \log N}\right)$, matching a corresponding lower bound for all Kolmogorov-random $N$ and all positive temperature values, set by Rothemund and Winfree [22]. The lower bound also holds in just-barely 3D. An $O\left(\frac{\log N}{\log \log N}\right)$ upper bound for the directed tile complexity of an $N \times N$ square at temperature 1 in just-barely 3D was established by Furcy, Micka and Summers [9] via a just-barely 3D, optimal encoding construction at temperature 1. Just-barely 3D, optimal encoding at temperature 1 was inspired by, achieves the same result as, but is drastically different from the 2D optimal encoding at temperature 2 developed by Soloveichik and Winfree [24], who proved that the directed tile complexity of a scaled-up version of any

algorithmically specified shape $X$ at temperature 2 is $\Theta\left(\frac{K(X)}{\log K(X)}\right)$, where $K(X)$ is the size of the smallest Turing machine that outputs the list of points in $X$. This tight bound for temperature-2 self-assembly in 2D was shown to hold for temperature-1 self-assembly in just-barely 3D by Furcy and Summers [10].

Another benchmark shape, for positive integers $k, N$, is the $k \times N$ rectangle, where $k < \frac{\log N}{\log\log N - \log\log\log N}$, making it "thin". A thin rectangle is an interesting testbed because its restricted height creates a limited channel through which tiles may propagate information, for example, the current value of a self-assembling counter. In fact, Aggarwal, Cheng, Goldwasser, Kao, Moisset de Espanés and Schweller [2] used an optimal, base-$\left\lceil N^{\frac{1}{k}} \right\rceil$ counter that uniquely self-assembles within the restricted confines of a thin rectangle to derive an upper bound of $O\left(N^{\frac{1}{k}} + k\right)$ on the directed tile complexity of a $k \times N$ thin rectangle at temperature 2 in 2D. They then leveraged the limited bandwidth of a thin rectangle in a counting argument for a corresponding lower bound of $\Omega\left(\frac{N^{\frac{1}{k}}}{k}\right)$.

The previous theory for a square and an algorithmically specified shape would suggest that these thin rectangle bounds should hold at temperature 1 in just-barely 3D. Yet, we currently do not know if this is the case. Thus, the power of temperature-1 self-assembly in just-barely 3D resembles that of temperature-2 self-assembly in 2D, with respect to the directed tile complexities of a square and a scaled-up version of any algorithmically specified shape, but not a thin rectangle.

Motivated by this theoretical discrepancy, we prove new lower and upper bounds on the directed tile complexity of a thin rectangle at temperature 1 in just-barely 3D, where $R^3_{k,N}$ is a just-barely 3D $k \times N$ rectangle if it satisfies $\{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, k-1\} \times \{0\} \subseteq R^3_{k,N} \subseteq \{0, 1 \ldots, N-1\} \times \{0, 1 \ldots, k-1\} \times \{0, 1\}$. See Tables 1 and 2 for a quick summary of our results and how they compare with previous state-of-the-art results.

**Table 1** State-of-the-art directed tile complexity for benchmark shapes in the aTAM, where $K(X)$ is the size of the smallest Turing machine that outputs the list of points in $X$.

| | 2D Temperature 2 | | Just-barely 3D Temperature 1 | |
|---|---|---|---|---|
| | Lower bound | Upper bound | Lower bound | Upper bound |
| $N \times N$ Square | $\Theta\left(\frac{\log N}{\log\log N}\right)$ | | Same as 2D Temperature 2 | |
| Algorithmically-defined shape $X$ | $\Theta\left(\frac{K(X)}{\log K(X)}\right)$ | | Same as 2D Temperature 2 | |
| $k \times N$ rectangle | $\Omega\left(\frac{N^{\frac{1}{k}}}{k}\right)$ | $O\left(N^{\frac{1}{k}} + k\right)$ | $\Omega\left(\frac{N^{\frac{1}{2k}}}{k}\right)$ | $O\left(N^{\frac{1}{\lfloor \frac{k}{3} \rfloor}} + \log N\right)$ |

**Table 2** In this table, we highlight our improved lower and upper bounds on the directed tile complexity of rectangles, the two main contributions of this paper, and compare them with corresponding bounds in 2D at temperature 2. Note that, for thin rectangles, the additive logarithmic term disappears, since, for a thin rectangle, $k < \frac{\log N}{\log\log N - \log\log\log N}$, which implies that $\log N < N^{\frac{1}{k}} < N^{\frac{1}{k-1}}$, for sufficiently large $k$ and $N$.

| | 2D Temperature 2 | | Just-barely 3D Temperature 1 | |
|---|---|---|---|---|
| | Lower bound | Upper bound | Lower bound | Upper bound |
| $k \times N$ rectangle | $\Omega\left(\frac{N^{\frac{1}{k}}}{k}\right)$ | $O\left(N^{\frac{1}{k}} + k\right)$ | $\Omega\left(N^{\frac{1}{k}}\right)$ | $O\left(N^{\frac{1}{k-1}} + \log N\right)$ |

First, we have our upper bound:

▶ **Theorem 1.** *The directed tile complexity of a just-barely 3D $k \times N$ rectangle at temperature 1 is $O\left(N^{\frac{1}{k-1}} + \log N\right)$.*

Theorem 1 is an asymptotic improvement over the previous state-of-the-art upper bound: $O\left(N^{\frac{1}{\lfloor \frac{k}{3} \rfloor}} + \log N\right)$ [11]. The latter bound is based on the self-assembly of a just-barely 3D counter that uniquely self-assembles at temperature 1, but whose base $M$ depends on the dimensions of the target rectangle, where each digit is represented geometrically and in binary within a just-barely 3D region of space comprised of $\Theta(\log N)$ columns and 3 rows. In a construction like this, the number of rows used to represent each digit affects the base of the counter, which, for a thin rectangle, turns out to be the asymptotically-dominating term in the tile complexity. For example, in the Furcy, Summers and Wendlandt construction, the number of rows per digit is 3, so the base is set to $\Theta\left(N^{\frac{1}{\lfloor \frac{k}{3} \rfloor}}\right)$. Intuitively, "squeezing" more digits into the counter for the same rectangle of height $k$ will result in a decrease in the base and therefore the tile complexity.

Our construction for Theorem 1 is based on the self-assembly of a just-barely 3D counter similar to the Furcy, Summers and Wendlandt construction, but the geometric structure of our counter is organized according to digit regions, or just-barely 3D regions of space comprised of $k$ rows and $\Theta\left(N^{\frac{1}{k-1}}\right)$ columns, in which $k-2$ base-$\Theta\left(N^{\frac{1}{k-1}}\right)$ digits are represented in a staggered fashion. This increase in digit density is the main reason why the "$\lfloor \frac{k}{3} \rfloor$" term from the Furcy, Summers and Wendlandt upper bound is replaced by a "$k-1$" term in Theorem 1. Finally, we have our lower bound:

▶ **Theorem 2.** *The directed tile complexity of a just-barely 3D $k \times N$ rectangle at temperature 1 is $\Omega\left(N^{\frac{1}{k}}\right)$.*

Theorem 2 is an asymptotic improvement over the previous state-of-the-art lower bound: $\Omega\left(\frac{N^{\frac{1}{2k}}}{k}\right)$. Technically, the latter bound is not explicitly proved (or even stated) and therefore cannot be referenced, but it can be derived via a straightforward application of the standard Window Movie Lemma introduced in [17]. On our way to proving Theorem 2, we prove Lemma 5, which is essentially a new, more powerful type of Window Movie Lemma technique, specifically designed for temperature-1 self-assembly within a just-barely 3D, rectangular region of space. We conjecture that the conclusion of Lemma 5 can be generalized to give a powerful tool for proving even better lower bounds on the directed tile complexity of 3D non-rectangular shapes at temperature-1 than what would otherwise be possible with the standard Window Movie Lemma. Lemma 5 lets us develop a more refined counting argument based on upper bounding the number of "sufficiently similar" ways for an assembly sequence to assign glues to a fixed set of locations abutting a plane. Intuitively, two assignments are sufficiently similar if, up to translation, they respectively agree on: the set of locations to which glues are assigned, the local order in which certain consecutive pairs of glues appear, and the glues that are assigned to a certain set (of roughly half) of the locations.

## 2 Formal definition of the abstract Tile Assembly Model

In this section, we briefly sketch a strictly 3D version of Winfree's abstract Tile Assembly Model (see also [14, 21, 22]).

All logarithms in this paper are base-2. A *grid graph* is an undirected graph $G = (V, E)$, where $V \subset \mathbb{Z}^3$, such that, for all $\left\{\vec{a}, \vec{b}\right\} \in E$, $\vec{a} - \vec{b}$ is a 3-dimensional unit vector. The *full grid graph* of $V$ is the undirected graph $G_V^{\text{f}} = (V, E)$, such that, for all $\vec{x}, \vec{y} \in V$, $\{\vec{x}, \vec{y}\} \in E \iff \|\vec{x} - \vec{y}\| = 1$, i.e., if and only if $\vec{x}$ and $\vec{y}$ are adjacent in the 3-dimensional integer Cartesian space.

A 3-dimensional *tile type* is a tuple $t \in (\Sigma^* \times \mathbb{N})^6$, e.g., a unit cube, with six sides, listed in some standardized order, and each side having a *glue* $g \in \Sigma^* \times \mathbb{N}$ consisting of a finite string *label* and a nonnegative integer *strength*. We assume a finite set of tile types, but an infinite number of copies of each tile type, each copy referred to as a *tile*. A *tile set* is a set of tile types and is usually denoted as $T$.

A *configuration* is a (possibly empty) arrangement of tiles on the integer lattice $\mathbb{Z}^3$, i.e., a partial function $\alpha : \mathbb{Z}^3 \dashrightarrow T$. Two adjacent tiles in a configuration *bind*, *interact*, or are *attached*, if the glues on their abutting sides are equal (in both label and strength) and have positive strength. Each configuration $\alpha$ induces a *binding graph* $G_\alpha^{\text{b}}$, a grid graph whose vertices are positions occupied by tiles, according to $\alpha$, with an edge between two vertices if the tiles at those vertices bind. An *assembly* is a connected, non-empty configuration, i.e., a partial function $\alpha : \mathbb{Z}^3 \dashrightarrow T$ such that $G_{\text{dom } \alpha}^{\text{f}}$ is connected and $\text{dom } \alpha \neq \emptyset$. Given $\tau \in \mathbb{Z}^+$, $\alpha$ is *$\tau$-stable* if every cut-set of $G_\alpha^{\text{b}}$ has weight at least $\tau$, where the weight of an edge is the strength of the glue it represents. When $\tau$ is clear from context, we say $\alpha$ is *stable*. Given two assemblies $\alpha, \beta$, we say $\alpha$ is a *subassembly* of $\beta$, and we write $\alpha \sqsubseteq \beta$, if $\text{dom } \alpha \subseteq \text{dom } \beta$ and, for all points $\vec{p} \in \text{dom } \alpha$, $\alpha(\vec{p}) = \beta(\vec{p})$.

A 3-dimensional *tile assembly system* (TAS) is a triple $\mathcal{T} = (T, \sigma, \tau)$, where $T$ is a tile set, $\sigma : \mathbb{Z}^3 \dashrightarrow T$ satisfying $|\text{dom } \sigma| = 1$ is the *seed assembly* (trivially $\tau$-stable), and $\tau \in \mathbb{Z}^+$ is the *temperature*. Given two $\tau$-stable assemblies $\alpha, \beta$, we write $\alpha \to_1^{\mathcal{T}} \beta$ if $\alpha \sqsubseteq \beta$ and $|\text{dom } \beta \backslash \text{dom } \alpha| = 1$. In this case we say $\alpha$ *$\mathcal{T}$-produces $\beta$ in one step*. If $\alpha \to_1^{\mathcal{T}} \beta$, $\text{dom } \beta \backslash \text{dom } \alpha = \{\vec{p}\}$, and $t = \beta(\vec{p})$, we write $\beta = \alpha + (\vec{p} \mapsto t)$. The *$\mathcal{T}$-frontier* of $\alpha$ is the set $\partial^{\mathcal{T}} \alpha = \bigcup_{\alpha \to_1^{\mathcal{T}} \beta} (\text{dom } \beta \backslash \text{dom } \alpha)$, i.e., the set of empty locations at which a tile could stably attach to $\alpha$. The *$t$-frontier* of $\alpha$, denoted $\partial_t^{\mathcal{T}} \alpha$, is the subset of $\partial^{\mathcal{T}} \alpha$ defined as $\left\{ \vec{p} \in \partial^{\mathcal{T}} \alpha \;\middle|\; \alpha \to_1^{\mathcal{T}} \beta \text{ and } \beta(\vec{p}) = t \right\}$.

Let $\mathcal{A}^T$ denote the set of all assemblies of tiles from $T$, and let $\mathcal{A}_{<\infty}^T$ denote the set of finite assemblies of tiles from $T$. A sequence of $k \in \mathbb{Z}^+ \cup \{\infty\}$ assemblies $\vec{\alpha} = (\alpha_0, \alpha_1, \ldots)$ over $\mathcal{A}^T$ is a *$\mathcal{T}$-assembly sequence* if, for all $1 \leq i < k$, $\alpha_{i-1} \to_1^{\mathcal{T}} \alpha_i$. The *result* of an assembly sequence $\vec{\alpha}$, denoted as $\text{res}(\vec{\alpha})$, is the unique limiting assembly (for a finite sequence, this is the final assembly in the sequence). We write $\alpha \to^{\mathcal{T}} \beta$, and we say $\alpha$ *$\mathcal{T}$-produces* $\beta$ (in 0 or more steps), if there is a $\mathcal{T}$-assembly sequence $\alpha_0, \alpha_1, \ldots$ of length $k = |\text{dom } \beta \backslash \text{dom } \alpha| + 1$ such that (1) $\alpha = \alpha_0$, (2) $\text{dom } \beta = \bigcup_{0 \leq i < k} \text{dom } \alpha_i$, and (3) for all $0 \leq i < k$, $\alpha_i \sqsubseteq \beta$. We say $\alpha$ is *$\mathcal{T}$-producible* if $\sigma \to^{\mathcal{T}} \alpha$, and we write $\mathcal{A}[\mathcal{T}]$ to denote the set of $\mathcal{T}$-producible assemblies. An assembly $\alpha$ is *$\mathcal{T}$-terminal* if $\alpha$ is $\tau$-stable and $\partial^{\mathcal{T}} \alpha = \emptyset$. We write $\mathcal{A}_\square[\mathcal{T}] \subseteq \mathcal{A}[\mathcal{T}]$ to denote the set of $\mathcal{T}$-producible, $\mathcal{T}$-terminal assemblies. If $|\mathcal{A}_\square[\mathcal{T}]| = 1$ then $\mathcal{T}$ is said to be *directed*. We say that a TAS $\mathcal{T}$ *uniquely self-assembles* a shape $X \subseteq \mathbb{Z}^3$ if $\mathcal{A}_\square[\mathcal{T}] = \{\alpha\}$ and $\text{dom } \alpha = X$.

The *directed tile complexity* of a shape $X$ at temperature $\tau$ is the minimum number of distinct tile types of any TAS that uniquely self-assembles (USA) $X$, denoted by $K_{USA}^{\tau}(X) = \min \{ n \mid \mathcal{T} = (T, \sigma, \tau), |T| = n \text{ and } \mathcal{T} \text{ uniquely self-assembles } X \}$.

## 3 The upper bound

In this section, we prove Theorem 1, our upper bound, namely that $K^1_{USA}\left(R^3_{k,N}\right) = O\left(N^{\frac{1}{k-1}} + \log N\right)$. In order to do so, we construct a TAS that uniquely self-assembles a sufficiently large rectangle (of any height $k \geq 3$) $R^3_{k,N}$. Specifically, we construct a TAS $\mathcal{T} = (T, \sigma, 1)$ so that it simulates a base $B = \left\lceil N^{\frac{1}{k-1}} \right\rceil$, $W = k - 2$ digit counter, henceforth referred to as *the counter*, that starts counting at a specified starting value and stops after the maximum value is incremented, before rolling over to 0. In the remainder of this section, we will describe the self-assembly of the counter.

Each $W$-digit, base-$B$ value of the counter is represented in a corresponding just-barely 3D rectangular region of space called a digit region. There are two types of digit regions, one for each type of counter step: *copy* and *increment*. The former duplicates the value from the previous increment region and the latter increments the value from the previous copy region. The counter alternates between increment and copy steps.

Counter digits are represented geometrically and in binary, using the *bit bump* technique by Cook, Fu and Schweller [3]. Each digit is comprised of $b + 2 = \lceil \log B \rceil + 2$ bit bumps that protrude from a row of tiles. Each bit bump geometrically encodes one bit. The two most significant (westernmost) bits of a digit are its *indicator bits*: 10 – most significant, 01 – least significant, 00 – neither, 11 – both. The rest of the bits represent a base-$B$ value. Bumps of a digit in increment (copy) regions protrude to the south (north). If the bump is in the $z = 0$ ($z = 1$) plane, then it represents a 0 (1). The $W$ digits in a digit region are staggered like the steps of a staircase, descending (ascending) in a copy (increment) region. Figure 1 shows the layout of the two types of digit regions, positioned consecutively as they would be in the counter, which is self-assembling to the east.

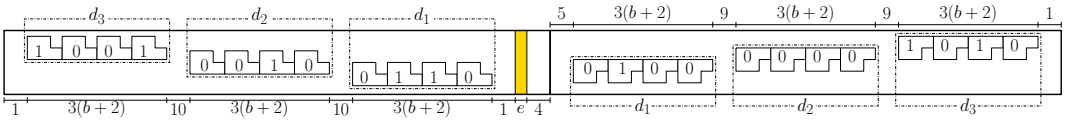



**Figure 1** A copy region (west) and an increment region (east). The numeric quantities below and above each region indicate a number of columns. Note that $W = 3$, so $k = 5$ (not drawn to scale).

**Digit regions (Figure 1):** We assume that the most significant bit of a digit is represented by the westernmost bit bump (to the east of the indicator bits), which means that we have $d_1 = 2$ (least significant digit), $d_2 = 2$ and $d_W = d_3 = 1$ (most significant digit). Thus, the base-3 value 122 in the copy region is incremented to 200 in the increment region. The orange column represents a variable number of $e = B - (2W(3(b + 2)) + 50)$ *extra* columns. The extra columns ensure that the combined width of two consecutive digit regions is $B$. The idea is that the terminal assembly of $\mathcal{T}$ is a sequence of copy-increment digit region pairs with one pair per value of the counter. Since each value of the counter corresponds to a pair of consecutive digit regions and $e$ is such that the width of two consecutive digit regions is $B$ columns, assuming the counter starts counting at 0, the width of the $k$-row terminal assembly that $\mathcal{T}$ will produce is $B \cdot B^W = \left\lceil N^{\frac{1}{k-1}} \right\rceil \cdot \left\lceil N^{\frac{1}{k-1}} \right\rceil^{k-2} \geq N^{\frac{1}{k-1}} \cdot N^{\frac{k-2}{k-1}} \geq N$. Then, $\mathcal{T}$ can be modified to produce a unique terminal assembly of height $k$ and width $N$, by using a positive starting value and $O(N \mod B)$ additional tile types.

Figures 2 through 10 illustrate the self-assembly of an increment step for an artificial example with $B = 3$, $k = 5$ and starting value 122.

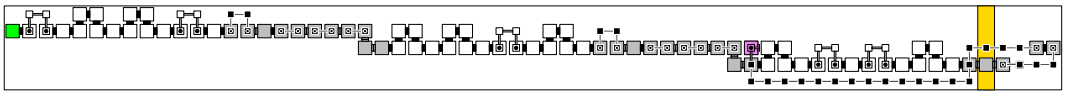


**Figure 2** The initial value assembly.

**The initial value (Figure 2):** The initial value is represented inside a copy region. The green tile is the seed tile. We use big (small) squares to represent tiles placed in the $z = 0$ ($z = 1$) plane. A glue between a $z = 0$ tile and $z = 1$ tile is denoted as a small disk. Glues between $z = 1$ ($z = 0$) tiles are denoted as thin (thick) lines. Each three-tile-wide bit bump geometrically encodes one bit. The bits of a digit are comprised of white tiles. In every copy region, the bump in the $z = 1$ plane immediately east of digit $d_i$ for $i > 1$ does not represent a bit, but rather a portion of the assembly that will eventually block the self-assembly of a subsequent path of repeating tiles (for example, the path of red tiles in Figure 6). The two easternmost tiles in the $z = 0$ plane will also block the self-assembly of a subsequent path of repeating tiles (for example, the path of blue tiles in Figure 4). The tiles that traverse the orange column represent paths of $e$ tiles. In general, we can *hard-code* a path of tiles that uniquely self-assembles a corresponding initial value assembly, from the seed to the $z = 0$ purple tile, where the glues of each tile type along the path encode the relative location of the tile in the path. In general, such a path contributes $O(e + kb) = O(B + \log N)$ tile types to $T$. After the initial value self-assembles, the counter executes an increment step. A bit indicating the presence of an arithmetic carry, *the carry bit*, initially set to 1, is introduced in the east-facing glues of both purple tiles that specifically start reading $d_1$ for an increment step. The purple tiles that start reading $d_i$ for $i > 1$ propagate the carry bit.

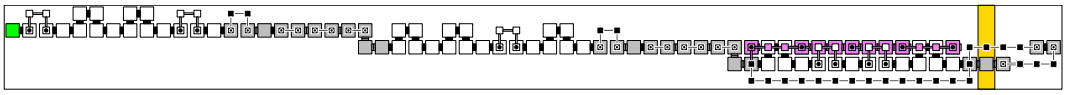


**Figure 3** The purple tiles are reading $d_1$ from most to least significant bit (west to east).

**Read digit (Figure 3):** The purple tiles are reading the four bits of $d_1$ in the copy region, starting with its most significant (westernmost) bit. In general, the counter reads the value of a digit as follows. Below the westernmost purple tile in the $z = 1$ plane, another purple tile attaches in the $z = 0$ plane (see also the purple tiles in Figure 2) and they both have east-facing glues. The east-facing glue on the $z = 1$ ($z = 0$) tile encodes 0 (1), but, due to the presence of the westernmost bit bump, only one is exposed. For each $1 \leq i < b + 2$ and $x \in \{0, 1\}^{i-1}$, there corresponds a *reader gadget* that reads a 0 in bit position $i$ (for notational convenience, $i = 1$ is the position of the most significant bit of a digit). Such a reader gadget self-assembles a horizontal path of three tiles in the $z = 1$ plane and a tile below the easternmost $z = 1$ tile (in the $z = 0$ plane), such that, the west-facing *input* glue of its westernmost tile encodes $x0$, the east-facing *output* glue of its easternmost $z = 0$ tile encodes $x01$ and the east-facing output glue of its easternmost $z = 1$ tile encodes $x00$. A reader gadget that reads a 1 is defined similarly. All reader gadgets also propagate the carry bit. A reader gadget that reads the bit in position $b + 2$ is similar to previous reader gadgets but has only one output glue, which is on a tile in the $z = 0$ plane and encodes the value of its input glue. It also initiates the self-assembly of a path of repeating tiles, along which

the value of the digit that was just read, the carry bit and a bit indicating whether $d_1$ was just read are propagated. The presence of all $b+2$ bit bumps ensures a unique assembly sequence of $b+2$ corresponding reader gadgets. In general, for each $1 \leq i \leq b+2$, we need $O\left(2^i\right)$ reader gadgets that read a bit in position $i$. Thus, in general, all the reader gadgets contribute $O\left(2^b\right) = O(B)$ tile types to $T$. Our reader gadgets are inspired by the simulation macro tiles depicted in Figures 3 and 6 of [3].

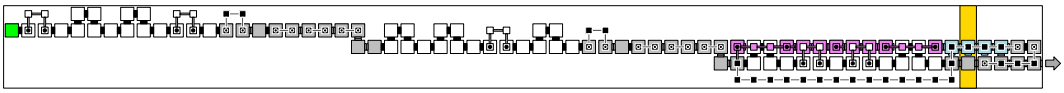


**Figure 4** The path of repeating blue tiles propagates $d_1$ and the carry bit to the increment region.

**Propagate digit (Figure 4):** After the bits of any digit in a copy region are read, a blue tile type with equal west and east glues self-assembles in a path of repeating tiles to the east, propagating the value of the digit that was just read and the carry bit. A previous portion of the assembly blocks the self-assembly of this path, at which point the south-facing glue of the easternmost blue tile in the path is exposed, from which a fixed size, hard-coded path of gray tiles self-assembles to the location immediately west of the most significant bit of $d_1$ in the increment region. The path of repeating blue tiles, all of the same type, propagate the value of a base-$B$ digit, the carry bit and a bit indicating whether $d_1$ was just read, thus, in general, contributing $O(B)$ tile types to $T$. Similarly, the hard-coded path, in general, contributes $O(B)$ tile types to $T$. We use two different types of hard-coded paths: one for $d_1$ and another for $d_i$ for $i > 1$. The latter self-assembles a bump in the $z = 1$ plane that eventually blocks a subsequent path of repeating tiles.

**Figure 5** The bits of $d_1$, read in Figure 3, are written.

**Write digit (Figure 5):** In this example, the value of $d_1$ is incremented from 2 and rolls over to 0, resulting in a carry out. The bits are written using fixed size *writer* gadgets, that work in a fashion similar to the reader gadgets, where we have a corresponding writer gadget for each bit position and each one propagates the carry bit (the first writer gadget receives the carry bit, updates it accordingly and propagates it). Thus, just like the reader gadgets, the writer gadgets contribute $O(B)$ tile types to $T$. The two easternmost $z = 0$ tiles will eventually block the self-assembly of a subsequent path of repeating blue tiles (for example, the blue tiles in Figure 4 but after reading a different digit). The path of tiles, starting with the gray tile immediately east of the least significant bit and ending at the westernmost $z = 1$ black tile is hard-coded and propagates the carry bit, thus, in general, contributing $O(b) = O(\log N)$ tile types to $T$. Note that, in general, the same tile types are used for the self-assembly of similar paths that self-assemble after writing every digit except $d_W$ in an increment region.

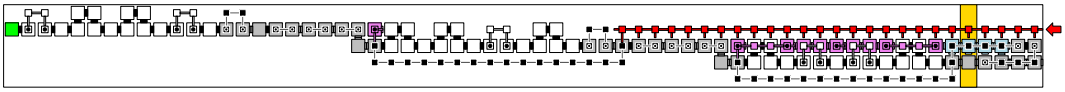


**Figure 6** Return to the copy region to read the next digit, $d_2$.

**Return to read another digit (Figure 6):** We use a red tile type with equal west and east glues to self-assemble a path of repeating tiles to the west, starting six tiles to the west of the most significant bit of the digit that was just written in the increment region. A previous portion of the assembly, namely the bump in the $z = 1$ plane that is east of the least significant bit of each digit $d_i$ for $i > 1$ in a copy region blocks the self-assembly of the path of repeating red tiles, at which point the south-facing glue of the westernmost red tile in the path is exposed, from which a hard-coded path of tiles self-assembles, ending at the $z = 0$ purple tile. The red tile types propagate the carry bit, thus, in general, contributing $O(1)$ tile types to $T$. The hard-coded path also propagates the carry bit, thus, in general, contributing $O(b) = O(\log N)$ tile types to $T$. Note that the same tile types are used for the self-assembly of similar paths that self-assemble after writing every digit except $d_W$ in an increment region.

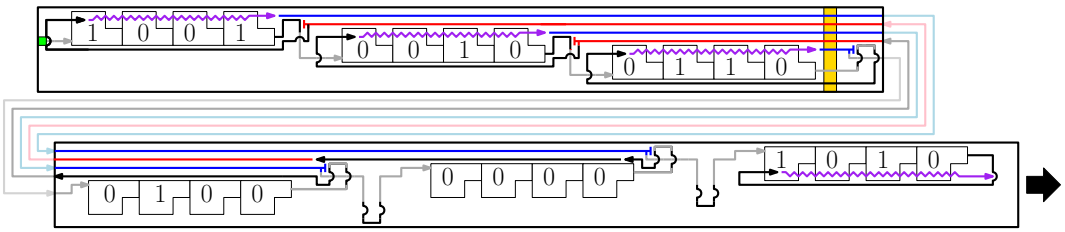



**Figure 7** A conceptual depiction of the complete self-assembly of an increment region (bottom), given the initial value copy region (top).

**Increment step high-level assembly sequence and corresponding full assembly (Figures 7 and 8):** The assembly sequence begins at the location of the green seed tile in the copy region. The gray (black) lines are hard-coded paths of tiles in the $z = 0$ ($z = 1$) plane. The blue (red) lines are paths of repeating tiles in the $z = 0$ ($z = 1$) plane. The purple zig-zag lines represent the reader gadgets. A corresponding full assembly is shown in Figure 8. The counter concludes an increment step after it writes $d_W$ in the increment region. The purple zig-zag line through $d_W$ in the increment region represents the first sequence of tile placements for the next copy step. Note that if the carry bit is 1 after writing $d_W$ in the increment region, then the counter can stop counting.

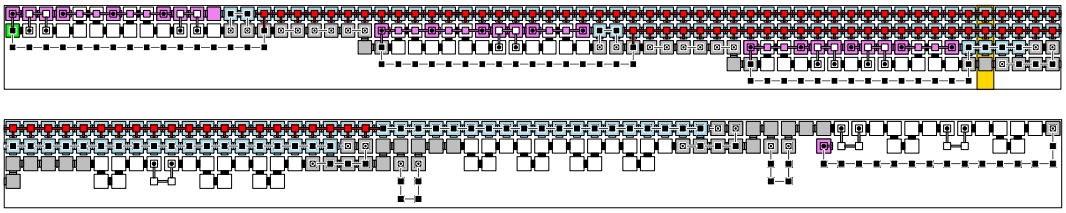


**Figure 8** The full assembly of the initial value copy region is on the top and the next increment region is on the bottom. A producible, non-terminal assembly results when the latter is translated so that its westernmost column is immediately east of the easternmost column of the former.

**Copy step high-level assembly sequence: (Figures 9 and 10):** After an increment step concludes, a copy step is executed. A copy step is carried out in a fashion similar to an increment step, using a specific set of tile types. In a copy step, the digits are read from the previous increment region and written in the next copy region in the order $d_W$ to $d_1$ (reverse

of an increment step). Regardless, the tile types for a copy step, in general, contribute $O(B + \log N)$ tile types to $T$. Moreover, all the tile types used for an increment step are defined to be disjoint from those for a copy step, which has no effect on the asymptotic size of $T$.

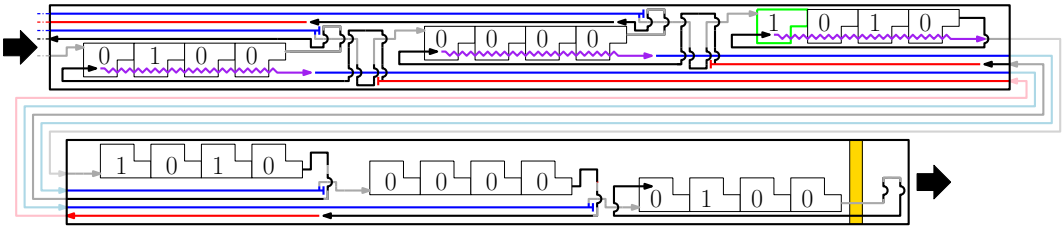



**Figure 9** A conceptual depiction of the self-assembly of a copy region (bottom) from a given increment region (top), which begins at the most significant bit of $d_W$ in the increment region (outlined in green). A copy step concludes after the self-assembly of the hard-coded path of tiles, starting at the least and ending at the most significant bit of $d_1$ (copy region). This hard-coded path contributes $O(e)$ tile types to $T$.

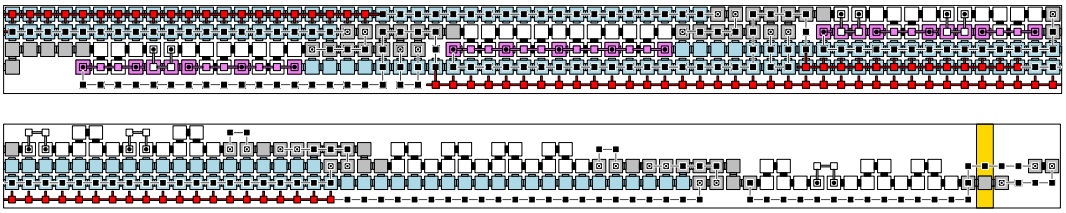


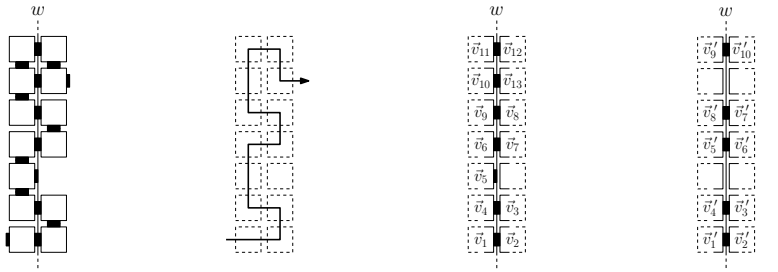
**Figure 10** The full assembly of the completed increment region is on the top (see also the bottom assembly of Figure 8) and the next copy region is on the bottom. A producible, non-terminal assembly results when first the latter is translated so that its westernmost column is immediately east of the easternmost column of the former and then this assembly is translated so its westernmost column is immediately east of the easternmost column of the top assembly shown in Figure 8.

**Tile complexity and correctness:** From the preceding discussion, generalized to arbitrary $k$ and $N$, we have $|T| = O(B + \log N) = O\left(N^{\frac{1}{k-1}} + \log N\right)$, from which the bound for Theorem 1 follows. Formal correctness can be shown using the method of Conditional Determinism by Shutters and Lutz [15].

## 4 The lower bound

In this section, we prove Theorem 2, our lower bound, namely that $K^1_{USA}\left(R^3_{k,N}\right) = \Omega\left(N^{\frac{1}{k}}\right)$. We first give some notation that will be used throughout the remainder of this section. Let $\mathcal{T} = (T, \sigma, \tau = 1)$ be a 3D TAS with $\alpha \in \mathcal{A}_\square[\mathcal{T}]$. Assume $s = (\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_m)$ is a simple path in $G^{\text{b}}_\alpha$, such that, $\vec{x}_0 = \text{dom } \sigma$. We say that $\vec{\alpha}$ *follows* $s$ if there is a $\mathcal{T}$-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \leq i < m + 1)$ that satisfies the next two conditions: $\alpha_0 = \sigma$, and for all $0 < i \leq m$, $\text{dom } \alpha_i \backslash \text{dom } \alpha_{i-1} = \{\vec{x}_i\}$.

This paragraph contains definitions that were taken directly from [17]. A *window* $w$ is a set of edges forming a cut-set of the full grid graph of $\mathbb{Z}^3$. Given a window $w$ and an assembly $\alpha$, a window that *intersects* $\alpha$ is a partitioning of $\alpha$ into two configurations (i.e., after being split into two parts, each part may or may not be disconnected). In this case




**(a)** A subassembly of $\alpha$ and a window $w$ induced by a translation of the $y$-axis.

**(b)** A portion of the simple path $s$ through $G^b_\alpha$.

**(c)** The glue window movie $M_{\vec{\alpha},w}$.

**(d)** The restricted glue window submovie $M_{\vec{\alpha},w} \upharpoonright s$.

**Figure 11** An assembly, a simple path, and two types of glue window movies in 2D. Here, we have $M_{\vec{\alpha},w} = (\vec{v}_1, g_1)$, $(\vec{v}_2, g_2)$, $(\vec{v}_3, g_3)$, $(\vec{v}_4, g_4)$, $(\vec{v}_5, g_5)$, $(\vec{v}_6, g_6)$, $(\vec{v}_7, g_7)$, $(\vec{v}_8, g_8)$, $(\vec{v}_9, g_9)$, $(\vec{v}_{10}, g_{10})$, $(\vec{v}_{11}, g_{11})$, $(\vec{v}_{12}, g_{12})$, $(\vec{v}_{13}, g_{13})$, where $g_1 = g_2$, $g_3 = g_4$, $g_6 = g_7$, $g_8 = g_9$, $g_{11} = g_{12}$ and $g_{13} = g_{10}$. Note that $M_{\vec{\alpha},w} \upharpoonright s$ only includes the location-glue pairs where the glues actually form bonds between locations in $s$. For example, $\vec{v}_{10}$ and $\vec{v}_{13}$ are excluded from $M_{\vec{\alpha},w} \upharpoonright s$ because the glues that connect them are not part of the path of glue that follow $s$.

we say that the window $w$ cuts the assembly $\alpha$ into two non-overlapping configurations $\alpha_L$ and $\alpha_R$, satisfying, for all $\vec{x} \in \text{dom } \alpha_L$, $\alpha(\vec{x}) = \alpha_L(\vec{x})$, for all $\vec{x} \in \text{dom } \alpha_R$, $\alpha(\vec{x}) = \alpha_R(\vec{x})$, and $\alpha(\vec{x})$ is undefined at any point $\vec{x} \in \mathbb{Z}^3 \backslash (\text{dom } \alpha_L \cup \text{dom } \alpha_R)$. Given a window $w$, its translation by a vector $\vec{\Delta}$, written $w + \vec{\Delta}$ is simply the translation of each one of $w$'s elements (edges) by $\vec{\Delta}$. All windows in this paper are assumed to be induced by some translation of the $yz$-plane. Each window is thus uniquely identified by its $x$ coordinate. For a window $w$ and an assembly sequence $\vec{\alpha}$, we define a *glue window movie* $M$ to be the order of placement, position and glue type for each glue that appears along the window $w$ in $\vec{\alpha}$, regardless of whether the glue (eventually) forms a bond. Given an assembly sequence $\vec{\alpha}$ and a window $w$, the associated glue window movie is the maximal sequence $M_{\vec{\alpha},w} = (\vec{v}_1, g_1), (\vec{v}_2, g_2), \ldots$ of pairs of grid graph vertices $\vec{v}_i$ and glues $g_i$, given by the order of appearance of the glues along window $w$ in the assembly sequence $\vec{\alpha}$. We write $M_{\vec{\alpha},w} + \vec{\Delta}$ to denote the translation by $\vec{\Delta}$ of $M_{\vec{\alpha},w}$, yielding $\left(\vec{v}_1 + \vec{\Delta}, g_1\right), \left(\vec{v}_2 + \vec{\Delta}, g_2\right), \ldots$.

If $\vec{\alpha}$ follows $s$, then the notation $M_{\vec{\alpha},w} \upharpoonright s$ denotes the *restricted* glue window submovie (*restricted to* $s$), which consists of only those steps of $M_{\vec{\alpha},w}$ that place glues that immediately form positive-strength bonds that cross $w$ at locations belonging to the simple path $s$. Let $\vec{v}$ denote the location of the starting point of $s$ (i.e., the location of $\sigma$). Let $\vec{v}_i$ and $\vec{v}_{i+1}$ denote two consecutive locations in $M_{\vec{\alpha},w} \upharpoonright s$ that are located across $w$ from each other. We say that these two locations define a *crossing* of $w$, where a crossing has exactly one direction. We say that this crossing is *away from* $\vec{v}$ (or *away from* $\sigma$) if the $x$ coordinates of $\vec{v}$ and $\vec{v}_i$ are equal or the $x$ coordinate of $\vec{v}_i$ is between the $x$ coordinates of $\vec{v}$ and $\vec{v}_{i+1}$. In contrast, we say that this crossing is *toward* $\vec{v}$ (or *toward* $\sigma$) if the $x$ coordinates of $\vec{v}$ and $\vec{v}_{i+1}$ are equal or the $x$ coordinate of $\vec{v}_{i+1}$ is between the $x$ coordinates of $\vec{v}$ and $\vec{v}_i$. See Figure 11 for 2D examples of $M_{\vec{\alpha},w}$ and $M_{\vec{\alpha},w} \upharpoonright s$, where $\sigma$ is located west of $w$ and the locations $\vec{v}_1$ and $\vec{v}_2$ form an away crossing, whereas the locations $\vec{v}_3$ and $\vec{v}_4$ form a crossing toward $\sigma$.

We say that two restricted glue window submovies are "sufficiently similar" if they have the same (odd) number of crossings, the same set of crossing locations (up to horizontal translation), the same crossing directions at corresponding crossing locations, and the same glues in corresponding "away crossing" locations.

▶ **Definition 3.** *Assume: $\mathcal{T} = (T, \sigma, 1)$ is a 3D TAS, $\alpha \in \mathcal{A}[\mathcal{T}]$, $s$ is a simple path in $G^b_\alpha$ starting from the location of $\sigma$, $\vec{\alpha}$ is a sequence of $\mathcal{T}$-producible assemblies that follows $s$, $w$ and $w'$ are windows, $\sigma$ is not located between $w$ and $w'$, $\vec{\Delta} \neq \vec{0}$ is a vector satisfying $w' = w + \vec{\Delta}$, $e$ and $e'$ are two odd numbers, and $M = M_{\vec{\alpha},w} \upharpoonright s = (\vec{v}_1, g_1), \ldots, (\vec{v}_{2e}, g_{2e})$ and $M' = M_{\vec{\alpha},w'} \upharpoonright s = (\vec{v}'_1, g'_1), \ldots, (\vec{v}'_{2e'}, g'_{2e'})$ are both non-empty restricted glue window submovies. We say that $M$ and $M'$ are* sufficiently similar *if the following are satisfied:*

1. *same number of crossings: $e = e'$,*
2. *same set of crossing locations (up to translation):* $\left\{ \vec{v}_i + \vec{\Delta} \;\middle|\; 1 \leq i \leq 2e \right\} = \left\{ \vec{v}'_j \;\middle|\; 1 \leq j \leq 2e \right\}$,
3. *same crossing directions at corresponding crossing locations:* $\left\{ \vec{v}_{4i-2} + \vec{\Delta} \;\middle|\; 1 \leq i \leq \frac{e+1}{2} \right\} = \left\{ \vec{v}'_{4j-2} \;\middle|\; 1 \leq j \leq \frac{e+1}{2} \right\}$, *and*
4. *same glues in corresponding "away crossing" locations: for all $1 \leq i, j \leq \frac{e+1}{2}$, if $\vec{v}'_{4j-2} = \vec{v}_{4i-2} + \vec{\Delta}$, then $g'_{4j-2} = g_{4i-3}$.*

See Figure 12 for an example of two restricted glue window submovies that are sufficiently similar. The following result basically says that we must examine only a "small" number of distinct restricted glue window submovies in order to find two different ones that are sufficiently similar.

▶ **Lemma 4.** *Assume: $\mathcal{T} = (T, \sigma, 1)$ is a 3D TAS, $G$ is the set of all glues in $T$, $k, N \in \mathbb{Z}^+$, $s$ is a simple path starting from the location of $\sigma$ such that $s \subseteq R^3_{k,N}$, $\vec{\alpha}$ is a sequence of $\mathcal{T}$-producible assemblies that follows $s$, $m \in \mathbb{Z}^+$, for all $1 \leq l \leq m$, $w_l$ is a window, for all $1 \leq l < l' \leq m$, $\vec{\Delta}_{l,l'} \neq \vec{0}$ satisfies $w_{l'} = w_l + \vec{\Delta}_{l,l'}$, and for all $1 \leq l \leq m$, there is an odd $1 \leq e_l < 2k$ such that $M_{\vec{\alpha},w_l} \upharpoonright s$ is a non-empty restricted glue window submovie of length $2e_l$. If $m > |G|^k \cdot k \cdot 16^k$, then there exist $1 \leq l < l' \leq m$ such that $e_l = e_{l'} = e$ and $M_{\vec{\alpha},w_l} \upharpoonright s = (\vec{v}_1, g_1), \ldots, (\vec{v}_{2e}, g_{2e})$ and $M_{\vec{\alpha},w_{l'}} \upharpoonright s = (\vec{v}'_1, g'_1), \ldots, (\vec{v}'_{2e}, g'_{2e})$ are sufficiently similar non-empty restricted glue window submovies.*

To prove Lemma 4, we first count the number of ways to choose the set $\{\vec{v}_1, \ldots, \vec{v}_{2e}\}$. Then, we count the number of ways to choose the set $\left\{ \vec{v}_{4i-2} \;\middle|\; 1 \leq i \leq \frac{e+1}{2} \right\}$. Finally, we count the number of ways to choose the sequence $\left( g_{\vec{x}_i} \;\middle|\; i = 1, \ldots, \frac{e+1}{2} \right)$. After summing over all odd $e$, we get the indicated lower bound on $m$ that notably neither contains a "factorial" term nor a coefficient on the "$k$" in the exponent of "$|G|$". The full proof of Lemma 4 is omitted from this version of the paper.

The following result is the cornerstone of our lower bound machinery. It basically says that if, for some directed TAS $\mathcal{T}$, two distinct restricted glue window submovies are sufficiently similar, then $\mathcal{T}$ does not self-assemble $R^3_{k,N}$.

▶ **Lemma 5.** *Assume: $\mathcal{T}$ is a directed, 3D TAS, $k, N \in \mathbb{Z}^+$, $s \subseteq R^3_{k,N}$ is a simple path, in the full grid graph of $R^3_{k,N}$, from the location of the seed of $\mathcal{T}$ to some location in the furthest extreme column of $R^3_{k,N}$, $\vec{\alpha}$ is a $\mathcal{T}$-assembly sequence that follows $s$, $w$ and $w'$ are windows, such that, $\vec{\Delta} \neq \vec{0}$ is a vector satisfying $w' = w + \vec{\Delta}$, and $e$ is an odd number satisfying $1 \leq e < 2k$. If $M = M_{\vec{\alpha},w} \upharpoonright s = (\vec{v}_1, g_1), \ldots, (\vec{v}_{2e}, g_{2e})$ and $M' = M_{\vec{\alpha},w'} \upharpoonright s = (\vec{v}'_1, g'_1), \ldots, (\vec{v}'_{2e}, g'_{2e})$ are sufficiently similar non-empty restricted glue window submovies, then $\mathcal{T}$ does not self-assemble $R^3_{k,N}$.*

See Figures 12 and 13 for a 2D example of Lemma 5. We now give some notation that will be useful for proving Lemma 5. The definitions and notation in the following paragraph are inspired by notation that first appeared in [17].

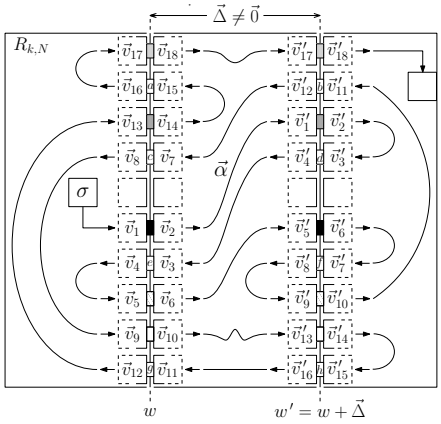



**Figure 12** A 2D example of the hypothesis of Lemma 5 for $k = 10$ and $e = 9$. Since the example is 2D, we use $R_{k,N} = \{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, k-1\}$, rather than $R^3_{k,N}$. Note that $\vec{\alpha}$ follows a simple path $s$ from the location of $\sigma$ to a location in the furthest extreme column. The restricted glue window movies are sufficiently similar because their glues are at the same locations (up to translation), oriented in the same direction (away or toward $\sigma$), and each pair of glues that are placed by $\vec{\alpha}$ at an "away crossing" of one of the windows is equal to its translated counterpart in the other window, e.g., the two topmost glues that touch $w$ and $w'$ are both light gray. The same constraint holds for all glue pairs shown with a solid shade of gray or a striped pattern. On the other hand, the glues adjacent to $w'$ that are placed by $\vec{\alpha}$ at a "toward crossing", for example $g'_{11}$ and $g'_{12}$, are decorated with a letter in order to represent the fact that we do not assume that these glues are equal to their translated counterparts that touch $w$ (i.e., $g_{15}$ and $g_{16}$).

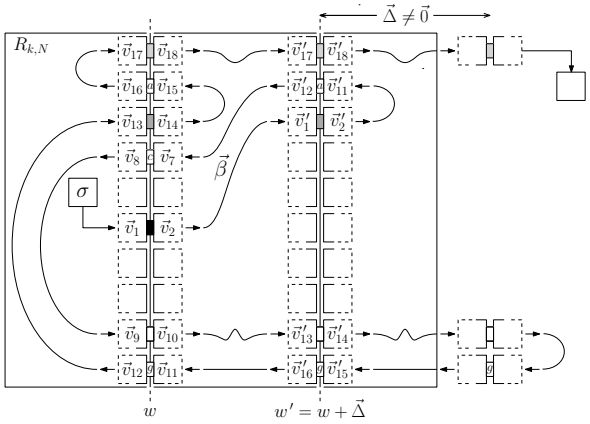



**Figure 13** A 2D example of the conclusion of Lemma 5, corresponding to example of the hypothesis from Figure 12. Given the fact that $\mathcal{T}$ is directed and the way $\vec{\beta}$ is defined, every pair of glues that touch $w$ must be equal to the corresponding pair of glues that touch $w'$ (if any). Thus, e.g., the glue pairs labelled $b$ and $h$ in Figure 12 must really be equal to the glue pairs $a$ and $g$, respectively. After $\vec{\beta}$ places a tile at location $\vec{v}'_{17}$, it will mimic how $\vec{\alpha}$ got from $\vec{v}_{18}$ to the tile in the extreme column of $R_{k,N}$, as depicted in Figure 12. Since $\vec{\Delta} \neq \vec{0}$, this always results in at least one tile placement outside of $R_{k,N}$. In this example, $\beta$ also happens to exit $R_{k,N}$ earlier in its assembly sequence, i.e., in the sub-path from $\vec{v}'_{14}$ to $\vec{v}'_{15}$.

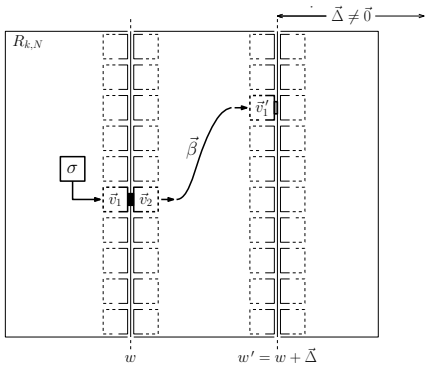
For a $\mathcal{T}$-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \leq i < l)$, we write $|\vec{\alpha}| = l$. We write $\vec{\alpha}[i]$ to denote $\vec{x} \mapsto t$, where $\vec{x}$ and $t$ are such that $\alpha_{i+1} = \alpha_i + (\vec{x} \mapsto t)$. We write $\vec{\alpha}[i] + \vec{\Delta}$, for some vector $\vec{\Delta}$, to denote $\left(\vec{x} + \vec{\Delta}\right) \mapsto t$. If $\alpha_{i+1} = \alpha_i + (\vec{x} \mapsto t)$, then we write $Pos\left(\vec{\alpha}[i]\right) = \vec{x}$ and $Tile\left(\vec{\alpha}[i]\right) = t$. Assuming $|\vec{\alpha}| > 0$, the notation $\vec{\alpha} = \vec{\alpha} + (\vec{x} \mapsto t)$ denotes a *tile placement step*, namely the sequence of configurations $(\alpha_i \mid 0 \leq i < l + 1)$, where $\alpha_l$ is the configuration satisfying, $\alpha_l\left(\vec{x}\right) = t$ and for all $\vec{y} \neq \vec{x}$, $\alpha_l\left(\vec{y}\right) = \alpha_{l-1}\left(\vec{y}\right)$. Note that the "+" in a tile placement step is different from the "+" used in the notation "$\beta = \alpha + (\vec{p} \mapsto t)$". However, since the former operates on an assembly sequence, it should be clear from the context which operator is being invoked. The definition of a tile placement step does not require that the sequence of configurations be a $\mathcal{T}$-assembly sequence. After all, the tile placement step $\vec{\alpha} = \vec{\alpha} + (\vec{x} \mapsto t)$ could be attempting to place a tile at a location that is not even adjacent to (a location in the domain of) $\alpha_{l-1}$. Or, it could be attempting to place a tile at a location that is in the domain of $\alpha_{l-1}$, which means a tile has already been placed at $\vec{x}$. So we say that such a tile placement step is *correct* if $(\alpha_i \mid 0 \leq i < l + 1)$ is a $\mathcal{T}$-assembly sequence. If $|\vec{\alpha}| = 0$, then $\vec{\alpha} = \vec{\alpha} + (\vec{x} \mapsto t)$ results in the $\mathcal{T}$-assembly sequence $(\alpha_0)$, where $\alpha_0$ is the assembly such that $\alpha_0\left(\vec{x}\right) = t$ and $\alpha_0\left(\vec{y}\right)$ is undefined at all other locations $\vec{y} \neq \vec{x}$.

**Algorithm 1** The algorithm for $\vec{\beta}$.

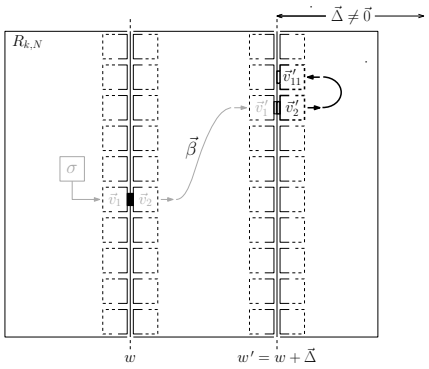
```
1  Initialize j = 1, n = 0 and β⃗ = ()
2  while Pos(α⃗[n]) ≠ v⃗'_{4j-2} do                          /* Loop 1 */
3  |  β⃗ = β⃗ + α⃗[n]
4  |  n = n + 1
5  while v⃗'_{4j-2} ≠ v⃗_{2e} + Δ⃗ do                         /* Loop 2 */
6  |  Let i be such that 4i − 2 is the index of v⃗'_{4j-2} − Δ⃗ in M
7  |  Let n be such that Pos(α⃗[n]) = v⃗_{4i-2}
8  |  while Pos(α⃗[n]) ≠ v⃗_{4i} do                          /* Loop 2a */
9  |  |  β⃗ = β⃗ + (α⃗[n] + Δ⃗)
10 |  |  n = n + 1
11 |  Let j' be such that 4j' is the index of v⃗_{4i} + Δ⃗ in M'
12 |  Let n be such that Pos(α⃗[n]) = v⃗'_{4j'}
13 |  while Pos(α⃗[n]) ≠ v⃗'_{4j'+2} do                      /* Loop 2b */
14 |  |  β⃗ = β⃗ + α⃗[n]
15 |  |  n = n + 1
16 |  j = j' + 1
17 Let n be such that Pos(α⃗[n]) = v⃗_{2e}
18 while n < |α⃗| do                                        /* Loop 3 */
19 |  β⃗ = β⃗ + (α⃗[n] + Δ⃗)
20 |  n = n + 1
21 return β⃗
```

The proof of Lemma 5 relies on Algorithm 1 that uses $\vec{\alpha}$ to construct a new assembly sequence $\vec{\beta}$ such that the tile placement steps by $\vec{\beta}$ on the far side of $w'$ from the seed mimic a (possibly strict) subset of the tile placements by $\vec{\alpha}$ on the far side of $w$ from the seed.

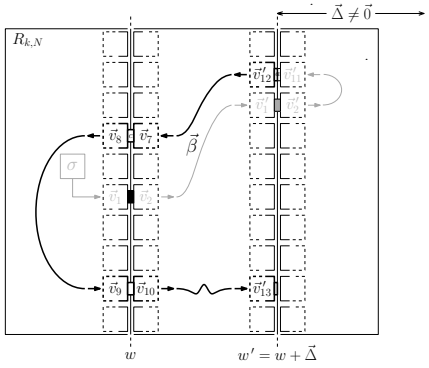
When $\vec{\beta}$ is on the near side of $w'$ to the seed, it mimics $\vec{\alpha}$, although $\vec{\beta}$ does not necessarily mimic every tile placement by $\vec{\alpha}$ on the near side of $w'$ to the seed. When $\vec{\beta}$ crosses $w'$, going away from the seed, by placing tiles at $\vec{v}'_{4j-3}$ and $\vec{v}'_{4j-2}$ in this order, then the tile it places



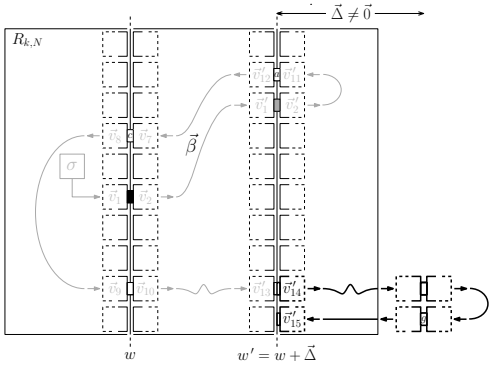

**(a)** Right after Loop 1 has completed: The $\vec{\alpha}$ sub-path from $\sigma$ to $\vec{v}'_1$ was used to initialize $\vec{\beta}$.



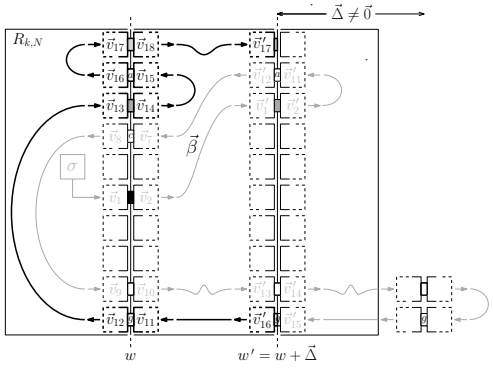

**(b)** Right after Loop 2a has completed for the first time: The $\vec{\alpha}$ sub-path from $\vec{v}_{14}$ to $\vec{v}_{15}$ was translated by $\vec{\Delta}$ and appended to $\vec{\beta}$.



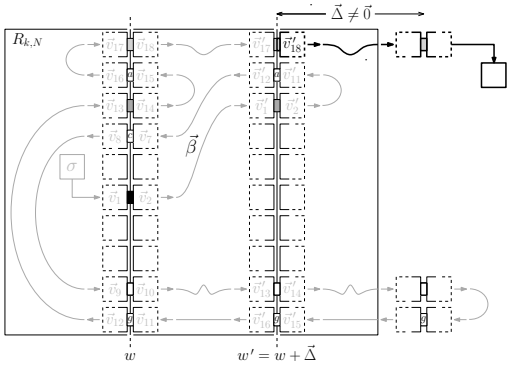

**(c)** Right after Loop 2b has completed for the first time: The $\vec{\alpha}$ sub-path from $\vec{v}'_{12}$ to $\vec{v}'_{13}$ was appended to $\vec{\beta}$.




**(d)** Right after Loop 2a has completed for the second time: The $\vec{\alpha}$ sub-path from $\vec{v}_{10}$ to $\vec{v}_{11}$ was translated by $\vec{\Delta}$ and appended to $\vec{\beta}$.




**(e)** Right after Loop 2b has completed for the second time: The $\vec{\alpha}$ sub-path from $\vec{v}'_{16}$ to $\vec{v}'_{17}$ was appended to $\vec{\beta}$.




**(f)** Right after Loop 3 (and the algorithm) has completed: The $\vec{\alpha}$ suffix starting with $\vec{v}_{18}$ was translated by $\vec{\Delta}$ and appended to $\vec{\beta}$.

**Figure 14** The trace of Algorithm 1 when applied to the assembly sequence $\vec{\alpha}$ shown in Figure 12. In each sub-figure, the new sub-path is bolded and is a continuation of the sub-path in the previous one. The last sub-figure above shows the same assembly sequence $\vec{\beta}$ depicted in Figure 13.

at $\vec{v}'_{4j-2}$ is of the same type as the tile that $\vec{\alpha}$ places at $\vec{v}_{4i-2} = \vec{v}'_{4j-2} - \vec{\Delta}$. After $\vec{\beta}$ crosses $w'$ by placing a tile at $\vec{v}'_{4j-2}$, $\vec{\beta}$ places tiles that $\vec{\alpha}$ places along $s$ from $\vec{v}_{4i-2}$ to $\vec{v}_{4i-1}$, but the tiles $\vec{\beta}$ places are translated to the far side of $w'$ from the seed. When $\vec{\beta}$ is about to cross $w'$, going toward the seed, by placing a tile at $\vec{v}'_{4j-1}$, then, since $\mathcal{T}$ is directed, the type of tile that it places at this location is equal to the type of tile that $\vec{\alpha}$ places at $\vec{v}'_{4j-1}$. This means that $\vec{\beta}$ may continue to follow $s$ but starting from $\vec{v}'_{4j}$. Eventually, $\vec{\beta}$ will finish crossing $w'$ going away from the seed for the last time by placing a tile at $\vec{v}_{2e} + \vec{\Delta}$. Then, $\vec{\beta}$ places tiles that $\vec{\alpha}$ places along $s$, starting from $\vec{v}_{2e}$, but the tiles that $\vec{\beta}$ places are translated to the far side of $w'$ from the seed. Since $\vec{\Delta} \neq \vec{0}$, $\vec{\beta}$ will ultimately place a tile that is not in $R^3_{k,N}$, which means $\mathcal{T}$ does not self-assemble $R^3_{k,N}$.

We illustrate the behavior of this algorithm in Figure 14, where we apply it to the assembly sequence $\vec{\alpha}$ shown in Figure 12. The full proof of Lemma 5 is omitted from this version of the paper. The following result combines Lemmas 4 and 5 and we will use its contrapositive to prove our main lower bound.

▶ **Lemma 6.** *Assume: $\mathcal{T} = (T, \sigma, 1)$ is a 3D TAS, $G$ is the set of all glues in $T$, $k, N \in \mathbb{Z}^+$, $s \subseteq R^3_{k,N}$ is a simple path, in the full grid graph of $R^3_{k,N}$, from the location of $\sigma$ to some location in the furthest extreme column of $R^3_{k,N}$, $\vec{\alpha}$ is a $\mathcal{T}$-assembly sequence that follows $s$, $m \in \mathbb{Z}^+$, for all $1 \leq l \leq m$, $w_l$ is a window, for all $1 \leq l < l' \leq m$, $\vec{\Delta}_{l,l'} \neq \vec{0}$ satisfies $w_{l'} = w_l + \vec{\Delta}_{l,l'}$, and for all $1 \leq l \leq m$, there is an odd $1 \leq e_l < 2k$ such that $M_{\vec{\alpha},w_l} \upharpoonright s$ is a non-empty restricted glue window submovie of length $2e_l$. If $m > |G|^k \cdot k \cdot 16^k$, then $\mathcal{T}$ does not self-assemble $R^3_{k,N}$.*

The proof of Lemma 6 is omitted from this version of the paper. Here is our main lower bound:

▶ **Theorem 2.** $K^1_{USA}\left(R^3_{k,N}\right) = \Omega\left(N^{\frac{1}{k}}\right)$.

**Proof.** Assume $\mathcal{T} = (T, \sigma, \tau = 1)$ is a directed, 3D TAS that self-assembles $R^3_{k,N}$. Assume $\alpha \in \mathcal{A}_\square[\mathcal{T}]$ with dom $\alpha = R^3_{k,N}$. Let $s = (\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_m)$ be a simple path in $G^{\text{b}}_\alpha$, such that, $\vec{x}_0 =$ dom $\sigma$ and $\vec{x}_m$ is in the furthest extreme (westernmost or easternmost) column of $R^3_{k,N}$ from the location of $\sigma$, in either $z$ plane. Since $\tau = 1$, there is a $\mathcal{T}$-assembly sequence $\vec{\alpha}$ that follows $s$. Assume $N \geq 3$. Since $s$ is a simple path from the location of the seed to some location in the furthest extreme column of $R^3_{k,N}$, in either $z$ plane, there is some positive integer $m \geq \left\lfloor \frac{N}{2} \right\rfloor \geq \frac{N}{3}$ such that, for all $1 \leq l \leq m$, $w_l$ is a window that cuts $R^3_{k,N}$, for all $1 \leq l < l' \leq m$, there exists $\vec{\Delta}_{l,l'} \neq \vec{0}$ satisfying $w_{l'} = w_l + \vec{\Delta}_{l,l'}$, and for each $1 \leq l \leq m$, there exists a corresponding odd number $1 \leq e_l < 2k$ such that $M_{\vec{\alpha},w_l} \upharpoonright s$ is a non-empty restricted glue window submovie of length $2e_l$. Since $\mathcal{T}$ self-assembles $R^3_{k,N}$, (the contrapositive of) Lemma 6 says that $m \leq |G|^k \cdot k \cdot 16^k$. We also know that $\frac{N}{3} \leq m$, which means that $\frac{N}{3} \leq |G|^k \cdot k \cdot 16^k$. Thus, we have $N \leq 3 \cdot |G|^k \cdot k \cdot 16^k$ and it follows that $|T| \geq \frac{|G|}{6} \geq \frac{1}{6} \frac{N^{\frac{1}{k}}}{(3 \cdot k \cdot 16^k)^{\frac{1}{k}}} \geq \frac{1}{6} \frac{N^{\frac{1}{k}}}{(3^k \cdot 2^k \cdot 16^k)^{\frac{1}{k}}} = \frac{1}{6} \frac{N^{\frac{1}{k}}}{96} = \Omega\left(N^{\frac{1}{k}}\right)$. ◀

Lemma 5, upon which our proof of Theorem 2 crucially depends (via Lemma 6), assumes that $\mathcal{T}$ is directed. If $\mathcal{T}$ is not assumed to be directed, then it is possible to construct an undirected 3D TAS $\mathcal{T}$ that satisfies all the other conditions of the hypothesis of Lemma 5, but $\mathcal{T}$ self-assembles $R^3_{k,N}$. The full construction of such an undirected $\mathcal{T}$ is omitted from this version of the paper.

## 5 Conclusion

In this paper, we gave improved lower and upper bounds on $K^1_{USA}\left(R^3_{k,N}\right)$, namely $\Omega\left(N^{\frac{1}{k}}\right)$ and $O\left(N^{\frac{1}{k-1}} + \log N\right)$. We leave open the question of determining tight bounds for $K^1_{USA}\left(R^3_{k,N}\right)$ as well as for $K^1_{SA}\left(R^3_{k,N}\right)$.

## References


1 Leonard M. Adleman, Qi Cheng, Ashish Goel, and Ming-Deh A. Huang. Running time and program size for self-assembled squares. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing (STOC)*, pages 740–748, 2001.

2 Gagan Aggarwal, Qi Cheng, Michael H. Goldwasser, Ming-Yang Kao, Pablo Moisset de Espanés, and Robert T. Schweller. Complexities for generalized models of self-assembly. *SIAM Journal on Computing (SICOMP)*, 34:1493–1515, 2005.

3 Matthew Cook, Yunhui Fu, and Robert T. Schweller. Temperature 1 self-assembly: Deterministic assembly in 3D and probabilistic assembly in 2D. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 570–589, 2011.

4 Erik D. Demaine, Martin L. Demaine, Sándor P. Fekete, Mashhood Ishaque, Eynat Rafalin, Robert T. Schweller, and Diane L. Souvaine. Staged self-assembly: nanomanufacture of arbitrary shapes with $O(1)$ glues. *Natural Computing*, 7(3):347–370, 2008. doi:10.1007/s11047-008-9073-0.

5 David Doty, Matthew J. Patitz, Dustin Reishus, Robert T. Schweller, and Scott M. Summers. Strong fault-tolerance for self-assembly with fuzzy temperature. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 417–426, 2010.

6 David Doty, Matthew J. Patitz, and Scott M. Summers. Limitations of self-assembly at temperature 1. *Theoretical Computer Science*, 412:145–158, 2011.

7 Sándor P. Fekete, Jacob Hendricks, Matthew J. Patitz, Trent A. Rogers, and Robert T. Schweller. Universal computation with arbitrary polyomino tiles in non-cooperative self-assembly. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 148–167, 2015.

8 Bin Fu, Matthew J. Patitz, Robert T. Schweller, and Robert Sheline. Self-assembly with geometric tiles. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 714–725, 2012.

9 David Furcy, Samuel Micka, and Scott M. Summers. Optimal program-size complexity for self-assembled squares at temperature 1 in 3D. *Algorithmica*, 77(4):1240–1282, 2017.

10 David Furcy and Scott M. Summers. Optimal self-assembly of finite shapes at temperature 1 in 3D. *Algorithmica*, 80(6):1909–1963, 2018.

11 David Furcy, Scott M. Summers, and Christian Wendlandt. Self-assembly of and optimal encoding within thin rectangles at temperature-1 in 3D. *Theoretical Computer Science*, 872:55–78, 2021.

12 Oscar Gilbert, Jacob Hendricks, Matthew J. Patitz, and Trent A. Rogers. Computing in continuous space with self-assembling polygonal tiles (extended abstract). In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 937–956, 2016.

13 Jacob Hendricks, Matthew J. Patitz, Trent A. Rogers, and Scott M. Summers. The power of duples (in self-assembly): It's not so hip to be square. *Theoretical Computer Science*, 743:148–166, 2018.

14 James I. Lathrop, Jack H. Lutz, and Scott M. Summers. Strict self-assembly of discrete Sierpinski triangles. *Theoretical Computer Science*, 410:384–405, 2009.

15 Jack H. Lutz and Brad Shutters. Approximate self-assembly of the sierpinski triangle. *Theory Comput. Syst.*, 51(3):372–400, 2012.

16 Ján Manuch, Ladislav Stacho, and Christine Stoll. Two lower bounds for self-assemblies at temperature 1. *Journal of Computational Biology*, 17(6):841–852, 2010.

17 P.-E. Meunier, M. J. Patitz, S. M. Summers, G. Theyssier, A. Winslow, and D. Woods. Intrinsic universality in tile self-assembly requires cooperation. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 752–771, 2014.

18 Pierre-Étienne Meunier, Damien Regnault, and Damien Woods. The program-size complexity of self-assembled paths. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 727–737, 2020.

19 Pierre-Étienne Meunier and Damien Woods. The non-cooperative tile assembly model is not intrinsically universal or capable of bounded turing machine simulation. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 328–341, 2017.

20 Matthew J. Patitz, Robert T. Schweller, and Scott 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, pages 175–189, Berlin, Heidelberg, 2011. Springer-Verlag. URL: `http://dl.acm.org/citation.cfm?id=2042033.2042050`.

21 Paul W. K. Rothemund. *Theory and Experiments in Algorithmic Self-Assembly*. PhD thesis, University of Southern California, December 2001.

22 Paul W. K. Rothemund and Erik Winfree. The program-size complexity of self-assembled squares (extended abstract). In *The Thirty-Second Annual ACM Symposium on Theory of Computing (STOC)*, pages 459–468, 2000.

23 Nadrian C. Seeman. Nucleic-acid junctions and lattices. *Journal of Theoretical Biology*, 99:237–247, 1982.

24 David Soloveichik and Erik Winfree. Complexity of self-assembled shapes. *SIAM Journal on Computing (SICOMP)*, 36(6):1544–1569, 2007.

25 Hao Wang. Proving theorems by pattern recognition – II. *The Bell System Technical Journal*, XL(1):1–41, 1961.

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