Self-Assembly of Decidable Sets (extended abstract)*

Matthew J. Patitz, and Scott M. Summers**

Department of Computer Science
Iowa State University
Ames, IA 50011, U.S.A.
{mpatitz,summers}@cs.iastate.edu

Abstract. The theme of this paper is computation in Winfree's Abstract Tile Assembly Model (TAM). We first review a simple, well-known tile assembly system (the "wedge construction") that is capable of universal computation. We then extend the wedge construction to prove the following result: if a set of natural numbers is decidable, then it and its complement's canonical two-dimensional representation self-assemble. This leads to a novel characterization of decidable sets of natural numbers in terms of self-assembly. Finally, we prove that our construction is, in some "natural" sense, optimal with respect to the amount of space it uses.

1 Introduction

In his 1998 Ph.D. thesis, Erik Winfree [9] introduced the (abstract) Tile Assembly Model (TAM) - a mathematical model of laboratory-based nanoscale self-assembly. The TAM is also an extension of Wang tiling [7, 8]. In the TAM, molecules are represented by un-rotatable, but translatable two-dimensional square "tiles," each side of which having a particular glue "color" and "strength" associated with it. Two tiles that are placed next to each other *interact* if the glue colors on their abutting sides match, and they *bind* if the strength on their abutting sides matches, and is at least a certain "temperature." Extensive refinements of the TAM were given by Rothemund and Winfree in [5, 4], and Lathrop et. al. [3] gave an elegant treatment of the model that does not discriminate against the self-assembly of infinite structures.

In this paper, we explore the notion of *computation* in the TAM - what is it, and how is it accomplished? Despite its deliberate over-simplification, the TAM is a computationally expressive model. For instance, Winfree proved [9] that in two or more spatial dimensions, the TAM is equivalent to Turing-universal computation. In other words, it is possible to construct, for any Turing machine

* This research was supported in part by National Science Foundation Grants 0652569 and 0728806

** This author's research was supported in part by NSF-IGERT Training Project in Computational Molecular Biology Grant number DGE-0504304

M and any input string w, a finite assembly system (i.e., finite set of tile types) that tiles the first quadrant, and encodes the set of all configurations that M goes through when processing the input string w. This implies that the process of self-assembly can (1) be directed algorithmically, and (2) be used to evaluate computable functions.

One can also regard the process of self-assembly itself as computation that, takes as input some initial configuration of tiles, and produces output in the form of some particular connected shape, and *nothing* else (i.e., *strict* self-assembly [3]). The self-assembly of shapes, and their associated Kolmogorov (shape) complexity, was studied extensively by Soloveichik and Winfree in [6], where they proved the counter-intuitive fact that, sometimes fewer tile types are required to self-assemble a "scaled-up" version of a particular shape as opposed to the actual shape.

Another flavor of computation in the TAM is the self-assembly of a language $A \subseteq \mathbb{N}$. Of course, one must make some additional assumptions about the self-assembly of A, since A is one-dimensional, and not necessarily connected. In this case, it only makes sense to talk about the *weak* self-assembly [3] of A. We say that A weakly self-assembles if "black" tiles are placed on, and only on, the points that are in A. One can also view weak self-assembly as painting a picture of the set A onto a much larger canvas of tiles. It is clear that if A weakly self-assembles, then A is necessarily computably enumerable. Moreover, Lathrop et. al. [2] discovered that the converse of the previous statement holds in the following sense. If the set A is computably enumerable, then a "simple" representation of A as points along the x-axis weakly self-assembles.

In this paper, we continue the work of Lathrop et. al. [2]. Specifically, we focus our attention on the self-assembly of decidable sets in the TAM. We first reproduce Winfree's proof of the universality of the TAM [9] in the form of a simple construction called the "wedge construction." The wedge construction self-assembles the *computation history* of an arbitrary TM M on input w in the space to the right of the y-axis, above the x-axis, and above the line $y = x - |w| - 2$. Our first main result follows from a straight-forward extension of the wedge construction, and gives a new characterization of decidable languages of natural numbers in terms of self-assembly. We prove that a set $A \subseteq \mathbb{N}$ is decidable if and only if $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble. Technically speaking, our characterization is (exactly) the first main theorem from Lathrop et. al. [2] with "computably enumerable" replaced by "decidable," and $f(n) = n$. Finally, we establish that, if $A \subseteq \mathbb{N}$ is a decidable set having sufficient space complexity, then it is impossible to "naturally" self-assemble the set $A \times \{0\}$ without placing tiles in more than one quadrant.

2 The Tile Assembly Model

We now give a brief intuitive sketch of the abstract TAM. See [9, 5, 4, 3] for other developments of the model. We work in the 2-dimensional discrete Euclidean space. We write $U_2 = \{(0,1),(1,0),(0,-1),(-1,0)\}$. We refer to the *first quadrant* $\mathbb{N}^2$ as Q_1, the *second quadrant* as Q_2, etc..

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

Given a set T of tile types, an *assembly* is a partial function $\alpha : \mathbb{Z}^2 \dashrightarrow T$. An assembly is *stable* if it cannot be broken up into smaller assemblies without breaking bonds of total strength at least $\tau = 2$. If α is an assembly, and $X \subseteq \mathbb{Z}^2$, then we write the *restriction of* α *to* X as $\alpha \upharpoonright X$.

Self-assembly begins with a *seed assembly* σ 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. A *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 $\tau = 2$ is the temperature. An assembly α is *terminal*, and we write $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, if no tile can be stably added to it. A TAS $\mathcal{T}$ is *directed*, or *produces a unique assembly*, if it has exactly one terminal assembly.

A set $X \subseteq \mathbb{Z}^2$ *weakly self-assembles* [3] if there exist a TAS $\mathcal{T} = (T, \sigma, \tau)$ and a set $B \subseteq T$ such that $\alpha^{-1}(B) = X$ holds for every terminal assembly α. That is, there is a set B of "black" tile types such that every terminal assembly has black tiles on points in the set X and only X.

An *assembly sequence* in a TAS $\mathcal{T} = (T, \sigma, \tau)$ is an infinite sequence $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \alpha_2, \ldots)$ 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. We define the *result* of an assembly sequence $\boldsymbol{\alpha}$ to be the unique assembly $\alpha = \mathrm{res}(\boldsymbol{\alpha})$ satisfying $\mathrm{dom}\ \alpha = \bigcup_{0 \leq i < k} \mathrm{dom}\ \alpha_i$, where $k \in \mathbb{Z}^+ \cup \{\infty\}$. The *precedence graph* [1] $G_{\boldsymbol{\alpha}}$ of $\boldsymbol{\alpha}$, with $\alpha = \mathrm{res}(\boldsymbol{\alpha})$, is defined as the directed graph $G = (\mathrm{dom}\ \alpha, E)$, where $(\boldsymbol{u}, \boldsymbol{v}) \in E$ if and only if (1) $\alpha(\boldsymbol{u})$ and $\alpha(\boldsymbol{v})$ interact, and (2) $\boldsymbol{\alpha}$ assigns a tile to $\boldsymbol{u}$ before it assigns a tile to $\boldsymbol{v}$.

To prove that a particular TAS $\mathcal{T} = (T, \sigma, \tau)$ is directed, it suffices to exhibit a locally deterministic [6] assembly sequence in $\mathcal{T}$. To save space here, we refer the reader to [6] for a detailed discussion of local determinism.

3 The Wedge Construction

In this section, we review the "wedge construction" - a simple, well-known TAS that simulates an arbitrary Turing machine on some binary string in the first quadrant of the discrete Euclidean plane. We will later use the wedge construction to prove our main result.

Construction 1 (wedge construction) *Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R)$ be a standard TM, $x \in \{0,1\}^*$, and define the TAS $\mathcal{T}_{M(x)} = \left(T_{M(x)}, \sigma, \tau\right)$, where*

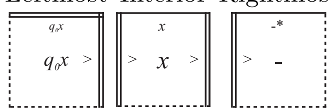
$T_{M(x)}$ *is the set of tile types defined in section 3.1,* σ *is the seed assembly satisfying* $\mathrm{dom}\,\sigma = (\{0, \ldots, |x| - 1\} \times \{0\})$ *that encodes the initial configuration of* M*, and* $\tau = 2$.

3.1 Tile Types for Construction 1

We construct the set of tile types $T_{M(x)}$ as follows.

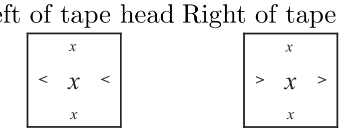
1. For all $x \in \Gamma$, add the seed row tile types:

Leftmost Interior Rightmost



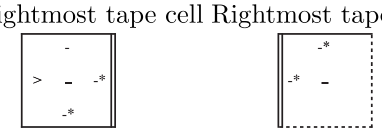
2. For all $x \in \Gamma$, add the tile types:

Left of tape head Right of tape head



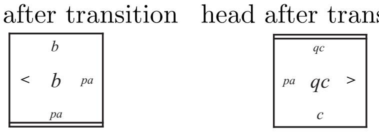
3. Add the following two tile types that grow the tape to the right:

2nd rightmost tape cell Rightmost tape cell



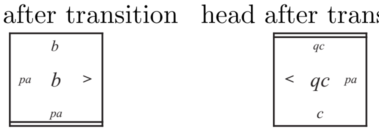
4. For all $p, q \in Q$, and all $a, b, c \in \Gamma$ satisfying $(q, b, R) = \delta(p, a)$ and $q \notin \{q_A, q_R\}$ (i.e. for each transition moving the tape head to the right into a non-accepting state), add the tile types:

Tape cell with output value after transition Cell that receives tape head after transition

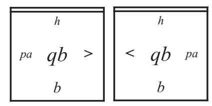


5. For all $p, q \in Q$, and all $a, b, c \in \Gamma$ satisfying $(q, b, L) = \delta(p, a)$ and $q \notin \{q_A, q_R\}$ (i.e. for each transition moving the tape head to the left into a non-accepting state), add the tile types:

Tape cell with output value after transition Cell that receives tape head after transition



6. For all $p \in Q$, $a, b \in \Gamma$, and all $h \in \{\text{ACCEPT}, \text{REJECT}\}$ satisfying $\delta(q, b) \in \{q_A, q_R\} \times \Gamma \times \{L, R\}$ (i.e. for each transition moving the tape head into a halting state), add the tile types:



3.2 Proof of Correctness

Lemma 1. *If M is a standard TM, and $x \in \{0,1\}^*$, then the TAS $\mathcal{T}_{M(x)}$ is locally deterministic.*

Proof (Proof sketch). It is straightforward to define an assembly sequence $\boldsymbol{\alpha}$, leading to a terminal assembly $\alpha = \text{res}(\boldsymbol{\alpha})$, in which (1) the j^{th} configuration C_j of M is encoded in the row $R_j = (\{0, \ldots, |x| - 1 + j\} \times \{j\})$, and (2) $\boldsymbol{\alpha}$ self-assembles C_i in its entirety before C_j if $i < j$. It follows easily from Construction 1 that every tile that binds in $\boldsymbol{\alpha}$ does so deterministically, and with exactly strength 2, whence $\mathcal{T}_{M(x)}$ is locally deterministic.

4 A New Characterization of Decidable Languages

We now turn our attention to the self-assembly of decidable sets of positive integers in the TAM. We will modify the wedge construction from the previous section in order to prove that, for every decidable set $A \subseteq \mathbb{N}$, there exists a directed TAS $\mathcal{T}_{A \times \{0\}} = (T_{A \times \{0\}}, \sigma, \tau)$ in which $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble. Throughout our discussion, we assume that $M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R)$ is a standard, total TM having '-' as its blank symbol, and satisfying $L(M) = A$. Our proof relies on the simple observation that, for every input $w \in \mathbb{N}$, there exists a $t \in \mathbb{N}$ such that M halts on w after t steps. This means that we can essentially stack wedge constructions one on top of the other. Intuitively, our main construction is the "self-assembly version" of the following enumerator.

while $0 \le n < \infty$ **do**
  simulate M on the binary representation of n
  if M *accepts* **then**
    output 1
  else
    output 0
  end if
  $n := n + 1$
end while

Just as the above enumerator prints the characteristic sequence of A, our construction will self-assemble the characteristic sequence of A along the positive x-axis.

4.1 Rigorous Construction of $\mathcal{T}_{A\times\{0\}}$

In this section we present a full definition of the tile set $T_{A\times\{0\}}$, and in the next section we provide a higher level description of the behavior of our tile set. Note that in both sections we will be discussing a version of $\mathcal{T}_{A\times\{0\}}$ in which the simulations of M proceed from the bottom up since it is often more natural to think about this particular orientation. However, to be technically consistent we ultimately rotate all of the tile types in $T_{A\times\{0\}}$ by 270 degrees, and then assign the seed tile to the location $(-1, 0)$. The full construction is implemented in C++, and is available at the following URL: `http://www.cs.iastate.edu/~lnsa`.

In our construction, we use the following sets of strings (where '$*$' and '-' simply represent the literal characters).

$$
\begin{aligned}
C &= \{\mathrm{M0{*}L}, \mathrm{M1}, \mathrm{M1{*}L}, \mathrm{M1{*}}, \mathrm{0{*}L}, \mathrm{1L}, 0, 0{*}, 1, \text{-}, \text{-}{*}\}\\
C[\text{no blank}] &= \{\mathrm{M0{*}L}, \mathrm{M1}, \mathrm{M1{*}L}, \mathrm{M1{*}}, \mathrm{0{*}L}, \mathrm{1L}, 0, 0{*}, 1\}\\
C[*] &= \{\mathrm{M0{*}L}, \mathrm{M1{*}L}, \mathrm{M1{*}}, \mathrm{0{*}L}, 0{*}\}\\
C[\text{no } *] &= C[\text{no blank}] - C[*]\\
M &= \{x \in C \mid \mathrm{M} \sqsubseteq x\}\\
N &= C[\text{no blank}] - M
\end{aligned}
$$

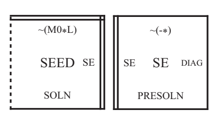
Intuitively, the set C contains the glue colors that appear on the north and south edges of some set of tile types that self-assembles a log-width binary counter (i.e., a binary counter that counts from 1 to infinity, and the width of each row is proportional to the log of the number it represents). We will embed these strings, and hence the behavior of a binary counter, into the tile types of the wedge construction. We will do so as follows.

Let T be the set of tile types given in Construction 1 that are not in groups (1) or (3). For each tile type $t \in T$, $c \in C$, and $\boldsymbol{u} \in U_2$, define the tile type t_c such that

$$
t_c(\boldsymbol{u}) = \begin{cases} (\mathrm{col}_t(\boldsymbol{u}), \mathrm{str}_t(\boldsymbol{u})) & \text{if } \boldsymbol{u} \in \{(1,0), (-1,0)\}\\ (\mathrm{col}_t(\boldsymbol{u}) \circ (c), \mathrm{str}_t(\boldsymbol{u})) & \text{otherwise,} \end{cases}
$$

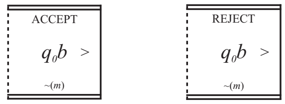
Note that "$\mathrm{col}_t(\boldsymbol{u}) \circ (c)$" means concatenate the string c, surrounded by parentheses, to the end of the string $\mathrm{col}_t(\boldsymbol{u})$. The set $\{t_c \mid t \in T \text{ and } c \in C\}$ makes up part of the tile set $T_{A\times\{0\}}$, and we define the remaining tile types as follows.

1. The following are seed tile types.

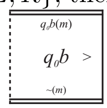


2. The following are the tile types for the initial configuration of M on some input.
   (a) Tile types that store the location of the tape head. For all $m \in M$, and all $b \in \{0, 1\}$,

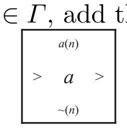
i. If there exists $h \in \{q_{\mathrm{A}}, q_{\mathrm{R}}\}$ such that $\delta(q_0, b) \in \{h\} \times \Gamma \times \{L, R\}$,
If $h = q_{\mathrm{A}}$, add: If $h = q_{\mathrm{R}}$, add:



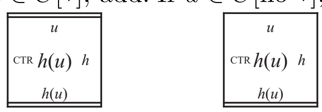
ii. If $\delta(q_0, b) \notin \{q_{\mathrm{A}}, q_{\mathrm{R}}\} \times \Gamma \times \{L, R\}$, then add the following tile types:



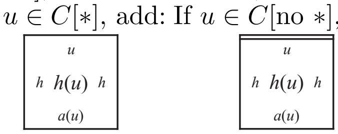
(b) Tile types that represent the tape contents to the right of the tape head. For all $n \in N \cup \{\text{-}\}$, and all $a \in \Gamma$, add the following tile types:



3. Halting row tile types. For all $h \in \{\mathrm{ACCEPT}, \mathrm{REJECT}\}$, add the following tile types:
   (a) The following tile types initiate the halting signal. For all $u \in C[\text{no blank}]$,
   If $u \in C[*]$, add: If $u \in C[\text{no } *]$, add:

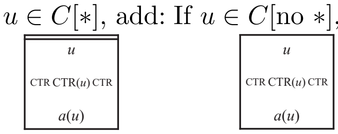


   (b) The following tile types propagate the halting signal to the right edge. For all $u \in C[\text{no blank}]$, and for all $a \in \Gamma$,
   If $u \in C[*]$, add: If $u \in C[\text{no } *]$, add:

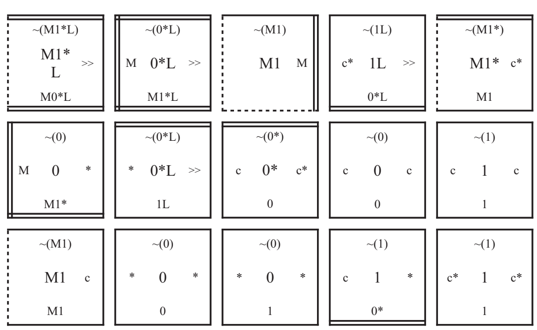


4. These are also halting row tile types, and fill in the space to the left of the initial halting tile. For all $u \in C[\text{no blank}]$, add the following tile types:

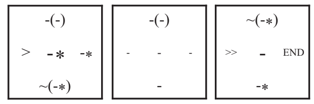
   If $u \in C[*]$, add: If $u \in C[\text{no } *]$, add:



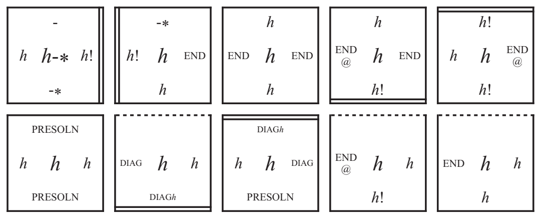
5. These are the tile types that perform counter increment operations.



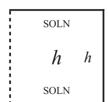
6. The following tile types propagate blank tape cells to the north



7. The following tile types self-assemble a one-tile-wide path from the halting configuration to some location on the positive x-axis. For all $h \in \{\text{ACCEPT}, \text{REJECT}\}$, add the following tile types:



8. The following are solution tiles. For all $h \in \{\text{ACCEPT}, \text{REJECT}\}$, add the tile types:



Construction 2 *Let $\mathcal{T}_{A\times\{0\}} = (T_{A\times\{0\}}, \sigma, \tau)$ be the TAS, where,*

$T_{A\times\{0\}} = \{\, t_c \mid t \in T$ *and* $c \in C \} \cup \{\, t \mid t$ *is a tile type defined in the above list*$\}$,

$\tau = 2$, *and* σ *consists of the leftmost tile type in group (1) of the above list placed at the point* $(0, 1)$.

4.2 Overview of Construction 2

This section gives a high level, intuitive description of Construction 2. Note that $\mathcal{T}_{A\times\{0\}}$ is singly-seeded, with the leftmost tile in group (1) of Section 4.1 being the seed tile type placed at the point $(0, 1)$.

The tile set $T_{A\times\{0\}}$ is constructed in two phases. First, we use the definition of the TM M to generate $T_{M(x)}$ as in Construction 1. We then "embed" a binary counter directly into these tile types in order to simulate the self-assembly version of a loop. This creates a tile set which can simulate M on every input $x \in \mathbb{N}$ (assuming A is decidable), while passing the values of a binary counter up through the assembly. These are the tiles that form the yellow portion of the structure shown in Figure 1, and labeled $M(0)$, $M(1)$, and $M(2)$.

In order to provide M with a one-way, infinite-to-the-right work tape, every row in our construction that represents a computation step grows the tape by one tape cell to the right. The binary counter used to simulate a loop, running M on

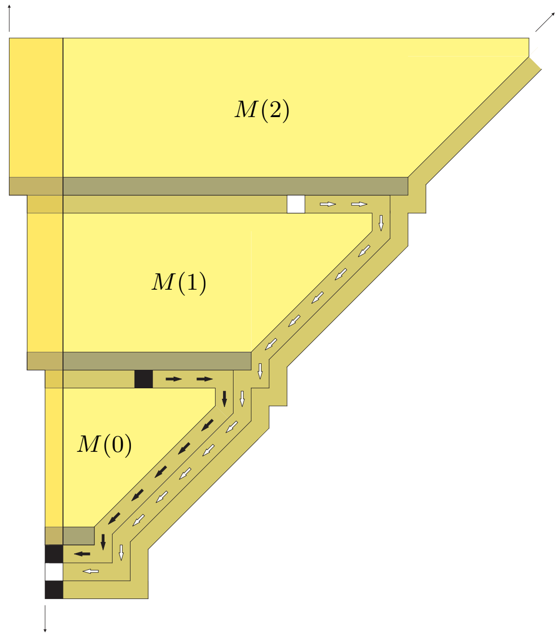


Fig. 1: The left-most (orange) bars represent a binary counter that is embedded into the tile types of the TM; the darkest (green) rows represent the initial configuration of M on inputs 0, 1, and 2; and the (green) horizontal rows that contain a white/black tile represent halting configurations of M. Although this image seems to imply that the embedded binary counter increases its width (to the left) on each input, this is not true in our construction. This image merely depicts the conceptual "shape" of the log-width counter that is embedded in our construction.

each input, is log-width and grows left into the second quadrant (represented by the slightly brighter yellow tiles on the leftmost side of Figure 1). An increment operation is performed immediately above each halting configuration of M.

The tile types that represent the initial configuration of M (on some input x) are shown in group (2) of Section 4.1. These tile types initiate each computation by using the value of x, embedded in the tile types of the binary counter, to construct a TM configuration with x located in its leftmost portion and q_0 reading the leftmost symbol of x.

Next, we construct the tile types for the ACCEPT and REJECT rows (i.e., halting configurations of M). To do this, we construct tile types that form a row immediately above the any row that represents a halting configuration of M. Conceptual examples of these rows are shown in Figure 1 as those with the black and white tiles, which represent ACCEPT and REJECT signals, respectively. The tile types that make up halting configurations are constructed in groups (3) and (4) of Section 4.1.

It is straightforward to construct the set of tile types that self-assemble a row that increments the value of the embedded binary counter (on top of the row that represents the halting configuration of M on x). These tile types are shown in group (5) of Section 4.1. After the counter increments, it initiates the simulation of M on input $x+1$. We prefix the north edge colors of the tile types that make up a counter row with '$\sim$' so as to signal that the next row should be the initial configuration of M on $x+1$. This has the effect of simulating M on $x+1$ directly on top of the simulation of M on x.

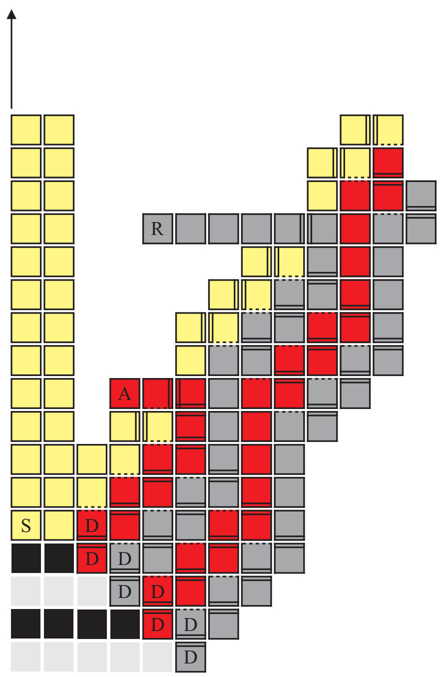


Fig. 2: The lightest (yellow) tiles represent successive simulations of M. When M halts and accepts, an accept signal (darkest shade of grey or red tiles) is sent down along the right side of the assembly to the appropriate point on the negative y-axis. The reject signals (middle shade of grey tiles) operate in the same fashion. The diagonal (D) signal allows each halting signal to essentially "turn" the corner.

The tile types in group (6) of Section 4.1 simply allow the blank symbol to propagate up through the assembly.

The final component of $T_{A\times\{0\}}$ is a group of tile types that carry the ACCEPT and REJECT signals to the appropriate location on the x-axis. These tile types are shown in groups (7) and (8) of Section 4.1, and their functionality can be seen in Figure 2.

4.3 Proof of First Main Theorem

Lemma 2. *Let $A \subseteq \mathbb{N}$ be decidable. The set $A \times \{0\}$ weakly self-assembles in the locally deterministic TAS $\mathcal{T}_{A\times\{0\}}$.*

Proof. The details of this proof are tedious, and therefore omitted from this version of the paper.

The following technical result is a primitive self-assembly simulator.

Lemma 3. *Let $A \subseteq \mathbb{Z}^2$. If A weakly self-assembles, then there exists a TM M_A with $L(M_A) = A$.*

Proof. Assume that A weakly self-assembles. Then there exists a TAS $\mathcal{T} = (T, \sigma, \tau)$ in which the set A weakly self-assembles. Let B be the set of "black" tile types given in the definition of weak self-assembly. Fix some enumeration $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 \ldots$ of $\mathbb{Z}^2$, and let M_A be the TM, defined as follows.

Require: $\boldsymbol{v} \in \mathbb{Z}^2$
  $\alpha := \sigma$
  while $\boldsymbol{v} \notin \operatorname{dom} \alpha$ **do**
    choose the least $j \in \mathbb{N}$ such that some tile can be added to α at $\boldsymbol{a}_j$
    choose some $t \in T$ that can be added to α at $\boldsymbol{a}_j$
    add t to α at $\boldsymbol{a}_j$
  end while
  if $\alpha(\boldsymbol{v}) \in B$ **then**
    accept
  else
    reject
  end if

It is routine to verify that M_A accepts A.

Lemma 4. *Let $A \subseteq \mathbb{N}$. If $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble, then A is decidable.*

Proof. Assume the hypothesis. Then by Lemma 3, there exist TMs $M_{A\times\{0\}}$ and $M_{A^c\times\{0\}}$ satisfying $L\left(M_{A\times\{0\}}\right) = A \times \{0\}$, and $L\left(M_{A^c\times\{0\}}\right) = A^c \times \{0\}$, respectively. Now define the TM M as follows.

Require: $n \in \mathbb{N}$
  Simulate both $M_{A\times\{0\}}$ and $M_{A^c\times\{0\}}$ on input $(n, 0)$ in parallel.
  if $M_{A\times\{0\}}$ *accepts* **then**
    accept
  end if
  if $M_{A^c\times\{0\}}$ *accepts* **then**
    reject
  end if

It is clear that M is a decider for A.

Lemma 5. *Let $A \subseteq \mathbb{N}$. If the set A is decidable, then $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble.*

Proof. This follows immediately from Construction 2 and Lemma 2. Note that the choice of the set B determines whether the set $A \times \{0\}$ or $A^c \times \{0\}$ weakly self-assembles.

We now have the machinery to prove our main result.

Theorem 1 (first main theorem). *Let $A \subseteq \mathbb{N}$. The set A is decidable if and only if $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble.*

Proof. This follows from Lemmas 4 and 5.

In the next section, we will prove that our construction is optimal in some natural sense with respect to the amount of space that it uses.

5 Two Quadrants are Sufficient and Necessary

In the proof of Theorem 1, we exhibited a directed TAS that placed at least one tile in each of three different quadrants. This leads one to ask the natural question: is it possible to do any better than three quadrants? In other words, does Theorem 1 hold if only two quadrants of space are allowed?

It turns out that the answer to the previous question is yes. Namely, if we simply shift the embedded binary counter in our construction to the right as its width grows, then we only need two quadrants of space to self-assemble the set $A \times \{0\}$. (There is enough space to accommodate the counter bits because the right edge of the TM simulation grows to the right faster than the right edge of the counter.) Note that the modifications to the tile set are straightforward, requiring the modification of only five tile types.

Now one must ask the question: does Theorem 1 hold when no more than one quadrant of space is available? First note that Winfree [9] proved one spatial dimension is sufficient to self-assemble $A \times \{0\}$ if A is regular. It is also easy to see that if $A \in \mathrm{DSPACE}(n)$, then it is possible to modify our construction to weakly self-assemble $A \times \{0\}$ using only one quadrant of space. However, and in the remainder of this section, we will prove that, if $A \notin \mathrm{DSPACE}\left(2^n\right)$, then it is impossible to weakly self-assemble the set $A \times \{0\}$ in any "natural" way without using more than one quadrant.

Note that, because of space-constraints, we merely sketch the proof of our second main theorem in this version of the paper.

Definition 1. *Let $A \subseteq \mathbb{N}$ be a decidable set and $\mathcal{T}$ be a TAS in which the set $A \times \{0\}$ weakly self-assembles. We say that $\mathcal{T}$ row-computes A if, for every $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, the following conditions hold.*

1. *Let $\boldsymbol{\alpha}$ be an assembly sequence of $\mathcal{T}$ with $\alpha = res(\boldsymbol{\alpha})$. For all $n \in \mathbb{N}$, there exists a unique point $(x_0, y_0) \in Q_1 \cup Q_2$ such that there is a path*

$$P_n = \langle (x_0, y_0), (x_1, y_1), \ldots, (x_{l-1}, y_{l-1}) \rangle$$

in the precedence graph $G_{\boldsymbol{\alpha}}$, where $(x_{l-1}, y_{l-1}) = (n, 0)$ and $y_0 > y_1 \geq \cdots \geq y_{l-1} = 0$.

2. *Let $P = \bigcup_{n=1}^{\infty} P_n$, and $\alpha' = \alpha \restriction (\mathrm{dom}\ \alpha - P)$. For all $m \in \mathbb{N}$, there is a finite assembly sequence $\boldsymbol{\alpha} = (\alpha_i \mid 0 \leq i < k)$ satisfying $\alpha_0 = \alpha' \restriction (\mathbb{Z} \times \{0, \ldots m-1\})$, and $\mathrm{dom}\ res(\boldsymbol{\alpha}) = \alpha' \restriction (\mathbb{Z} \times \{0, \ldots, m\})$.*

We assume that if $\mathcal{T}$ row-computes a set $A \subseteq \mathbb{N}$, then every terminal assembly α of $\mathcal{T}$ consists of two components: a simulation of some TM M with $L(M) = A$, and the paths that determine the fate of the points along the x-axis. Intuitively, condition (1) says that for every point $(n, 0)$ along the x-axis, there is a unique point in the first or second quadrant, and the path P_n that connects the former point to the latter carries the answer to the following question: "Is $n \in A$?" For technical reasons, we assume that the path P_n never grows "up." Finally, condition (2) says that the simulation component of α can self-assemble one row at a time.

It is clear that, for any decidable set $A \subseteq \mathbb{N}$, the construction that we outlined at the beginning of this section row-computes A.

Theorem 2 (second main theorem). *Let $A \subseteq \mathbb{N}$. If $A \notin DSPACE(2^n)$, and $\mathcal{T}$ is any TAS that row-computes A, then for all $\alpha \in \mathcal{A}_{\Box}[\mathcal{T}]$, $\alpha\left(Q_1\right) \cap \alpha\left(Q_1^c\right) \neq \emptyset$.*

Proof (Proof sketch). Assume for the sake of contradiction that for every terminal assembly α of $\mathcal{T}$, $\mathrm{dom}\ \alpha \subseteq Q_1$. Since $\mathcal{T}$ row-compute A, there must be a path P in G_α from some point $(x_0, y_0) \in Q_1$ to some point along the x-axis. Moreover, the path P must "turn left" at some point. If this were not the case for every such path, then it is possible to use condition (2) in the definition of row-computes to show that $A \in \mathrm{DSPACE}(n)$, which contradicts the fact that $A \notin \mathrm{DSPACE}\left(2^n\right)$.

Since there is one path that, en route to the x-axis, turns left (at some point), every successive path must do so. Because $\mathrm{dom}\ \alpha \subseteq Q_1$, there exists $n \in \mathbb{N}$ for which a path terminating at the point $(n, 0)$ goes through the point $(n+1, 0)$. This clearly violates condition (1) of the definition of row-computes. Hence, our initial assumption must be wrong, and the theorem follows.

In other words, Theorem 2 says that if A has sufficient space complexity, then it is impossible to weakly self-assemble the set $A \times \{0\}$ in any "natural" way with the entire assembly being contained entirely in the first quadrant. This is the sense in which the construction that we outlined at the beginning of this section is *optimal*.

6 Conclusion

In this paper, we investigated the self-assembly of decidable sets of natural numbers in the TAM. We first proved that, for every decidable language $A \subseteq \mathbb{N}$, $A \times \{0\}$ and $A^c \times \{0\}$ weakly self-assemble. This implied a novel characterization of decidable sets in terms of self-assembly. Our second main theorem established that in order to achieve this compactness (i.e., self-assembly of $A \times \{0\}$

as opposed to $f(A) \times \{0\}$ for some function f) for spatially complex languages, any "natural" construction will inevitably utilize strictly more than one quadrant of space. In fact, we conjecture that Theorem 2 holds for *any* TAS $\mathcal{T}$ in which $A \times \{0\}$ weakly self-assembles. Our results continue to expose the rich interconnectedness between geometry and computation in the TAM.

Acknowledgment Both authors wish to thank Dave Doty for pointing out simplifications to Section 5.

References

1. Qi Cheng, Ashish Goel, and Pablo Moisset de Espanés, *Optimal self-assembly of counters at temperature two*, Proceedings of the First Conference on Foundations of Nanoscience: Self-assembled Architectures and Devices, 2004.
2. James I. Lathrop, Jack H. Lutz, Matthew J. Patitz, and Scott M. Summers, *Computability and complexity in self-assembly*, Proceedings of The Fourth Conference on Computability in Europe (Athens, Greece, June 15-20, 2008). To appear., 2008.
3. James I. Lathrop, Jack H. Lutz, and Scott M. Summers, *Strict self-assembly of discrete Sierpinski triangles*, Proceedings of The Third Conference on Computability in Europe (Siena, Italy, June 18-23, 2007), 2007.
4. Paul W. K. Rothemund, *Theory and experiments in algorithmic self-assembly*, Ph.D. thesis, University of Southern California, December 2001.
5. Paul W. K. Rothemund and Erik Winfree, *The program-size complexity of self-assembled squares (extended abstract).*, Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, 2000, pp. 459–468.
6. David Soloveichik and Erik Winfree, *Complexity of self-assembled shapes*, SIAM Journal on Computing 36, 2007, pp. 1544–1569.
7. Hao Wang, *Proving theorems by pattern recognition – II*, The Bell System Technical Journal **XL** (1961), no. 1, 1–41.
8. ———, *Dominoes and the AEA case of the decision problem*, Proceedings of the Symposium on Mathematical Theory of Automata (New York, 1962), Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., 1963, pp. 23–55.
9. Erik Winfree, *Algorithmic self-assembly of DNA*, Ph.D. thesis, California Institute of Technology, June 1998.