Strict self-assembly of discrete Sierpinski triangles

by

Scott Martin Summers

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Jack H. Lutz, Major Professor
James I. Lathrop
Krishna B. Athreya

Iowa State University

Ames, Iowa

2007

UMI Number: 1443125

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

TABLE OF CONTENTS

ACKNOWLEDGMENTS iii

1 INTRODUCTION 1

2 NOTATION AND TERMINOLOGY 5

3 THE TILE ASSEMBLY MODEL 6

4 LOCAL DETERMINISM 14

5 ZETA-DIMENSION 16

6 THE STANDARD DISCRETE SIERPINSKI TRIANGLE S 17

7 IMPOSSIBILITY OF STRICT SELF-ASSEMBLY OF S 19

8 THE FIBERED SIERPINSKI TRIANGLE T 23

9 STRICT SELF-ASSEMBLY OF T 26

The Standard Optimal Counter 27

A Modified Optimal Counter 29

A Tile Assembly System for **T** 36

Proof of Correctness 37

10 CONCLUSION 40

BIBLIOGRAPHY 41

ACKNOWLEDGMENTS

There are several individuals (other than the author) to whom this thesis owes its existence, and I would like to take this opportunity to thank each of them.

First, I would like to thank my major professor Dr. Jack H. Lutz for his guidance, encouragement, and most importantly, his patience throughout the course of my research and the writing of this thesis.

I also owe thanks to Dr. James I. Lathrop for not only agreeing to serve on my committee but also for his frequent visits to Atanasoff B23 in which many (always useful and often research-related) discussions took place.

I would also like to thank Dr. Krishna B. Athreya for agreeing to serve on my committee and evaluate my research.

I also thank my parents and grandparents for their encouragement in and generous financial support of my education. It is to them that I owe my interest in science.

Finally, I dedicate this thesis to my fiancé Diana, without whose infinite patience, continuous emotional support and unconditional love, this document would most certainly not exist.

1 INTRODUCTION

The Tile Assembly Model (TAM), an extension of Wang tiling [16, 17, 5], is an abstract mathematical model of nanoscale self-assembly. The TAM was first introduced by Winfree [18] and further developed by Rothemund and Winfree [13, 12]. Similar developments of the TAM can be found in [1, 11, 15, 3]. In the TAM, molecules are represented by tile types (i.e., squares that cannot be rotated) that each have a "glue strength" and a "color" on all four sides. There are infinitely many copies of each tile type yet only finitely many tile types. Self-assembly begins with a special "seed" tile type. A tile type can bind to the existing structure given that its edge colors and glue strengths match those of all abutting tiles, and that the total glue strength is at least a certain temperature. The following is a simple example of a tile set from [13].

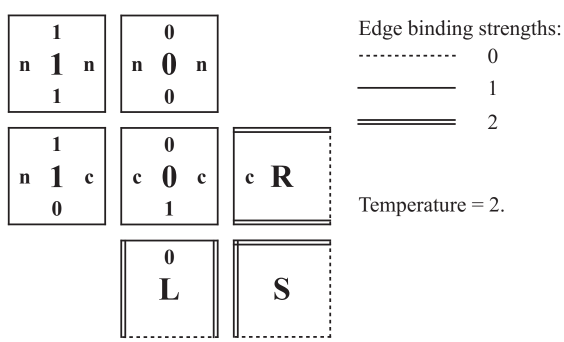


Figure 1.1 A simple example of a set of tile types.

It is easy to see that the tile set given in the above figure self-assembles an infinite binary counter in the second quadrant as follows. Note that this particular structure is not "terminal" in the sense that there are several locations at which new tiles can attach.

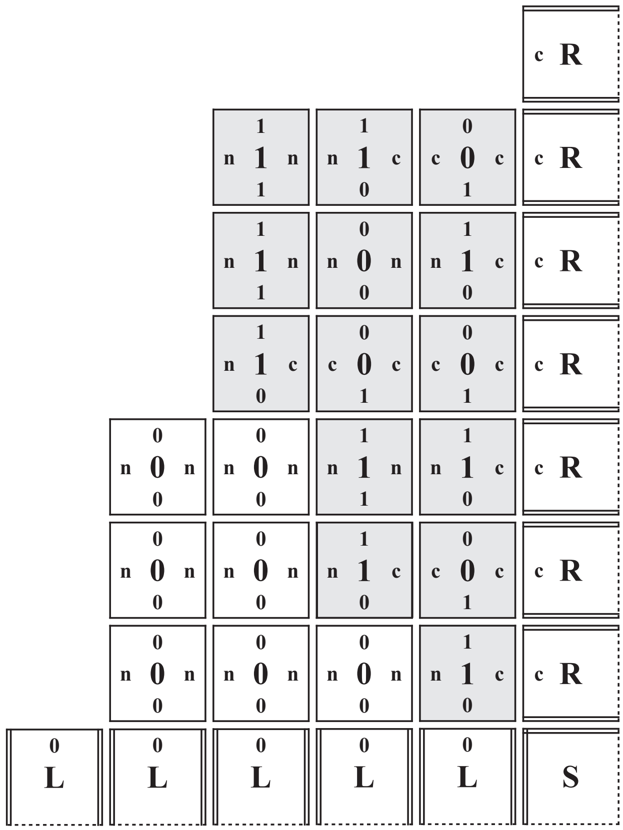


Figure 1.2 Self-assembly of an infinite binary counter in the TAM.

Despite its simplicity, the TAM turns out to be a useful theoretical model (at least in the sense of universal computation). In his Ph.D. thesis, Winfree [18] not only proved that the TAM is computationally universal (in two or more dimensions) but also showed that it is possible to self-assemble the discrete Sierpinski triangle **S**, which is illustrated in Figure 6.1. Specifically, Winfree exhibited a set of seven tile types that self-assembles the set **S** via a simple XOR-like algorithm that places a "black" tile type at every point $(x, y) \in \mathbf{S}$ and a "white" tile type elsewhere. Winfree's construction for **S** essentially "paints" the first quadrant of the discrete Euclidean space $\mathbb{Z}^2$ with a picture of the discrete Sierpinski triangle. Furthermore, Papadakis, Rothemund, and Winfree [14] confirmed the practicality of the TAM in 2004 where they experimentally implemented Winfree's seven tile type construction of **S** using DNA double-crossover molecules. In their experiments, they were able to achieve the correct placement of between 100 and 200 tiles.

It is clear from the previous paragraph that the process of self-assembly can be directed algorithmically in theory *and* in practice. However, one can also regard self-assembly *itself* as a kind of computational process where the input (a finite set of tile types) is transformed (self-assembled) into some kind of output (a particular shape). This naturally leads to the following question.

Question 1.1. What kind of shapes can self-assemble in the Tile Assembly Model?

It is certainly the case that all finite shapes self-assemble in the TAM but in general Question 1.1 is undecidable and is only interesting when infinite shapes are under consideration. Note that the natural "optimization" analog of Question 1.1 (with respect to the number of tile types) is a well-studied (as well as an undecidable) problem, and applies to both finite and infinite shapes. See [3, 6, 2, 13, 12] for results on the minimum number of tile types required to self-assemble finite shapes such as trees, squares, and rectangles.

Before we can tackle Question 1.1, we must first specify what it means for a shape to self-assemble in the TAM. Recall that Winfree [18] proved that **S** self-assembles in the TAM by essentially painting the first quadrant with a picture of **S**. However, in terms of the resulting shape, this is a less than satisfactory notion of the self-assembly of a shape because self-assembling infinite squares (along with other infinite periodic shapes) in the TAM is about as exciting as building a single state deterministic finite automaton that accepts the set Σ^*. Thus, we introduce the notion of the *strict* self-assembly of a shape. In order for a shape X to strictly self-assemble in the TAM, there must exist a finite set of tile types that self-assemble the set X by placing a tile at all the locations in X and never placing a tile at any location not in X. To say that a particular shape strictly self-assembles in the TAM is to say that the *shape* itself self-assembles and not simply an infinite square onto which the shape is painted.

In this thesis, we study the strict self-assembly of fractal structures in the TAM. Note that an important characteristic of fractals is that their dimension is less than that of the space which they occupy. Thus, (natural or engineered) fractal structures offer advantages for materials transport, heat exchange, information processing and are generally robust structures. We specifically investigate the strict self-assembly of the discrete Sierpinski triangles. In doing so, we present two results: one of which is positive while the other is negative.

Our negative result is that the standard discrete Sierpinski triangle **S** does not strictly self-assemble in the Tile Assembly Model. We prove this using a result by Adleman, Cheng, Goel, Huang, Kempe, Moisset de Espanés, and Rothemund [2] that specifies a lower bound on the number of tile types required to strictly self-assemble a finite tree shape.

Our positive result is the construction of a modified version of **S** that we will refer to as the fibered Sierpinski triangle **T**. We prove that **T** and **S** share the same fractal dimension, but **T**, unlike **S**, strictly self-assembles in the TAM. Our tile set that self-assembles **T** consists of 51 tile types, and we prove that our construction is correct using the method of local determinism due to Soloveichik and Winfree [15].

The remainder of this thesis is organized as follows. We first establish notation in Chapter 2. In Chapter 3 we develop our own flavor of the TAM, which not only incorporates the notion of strict self-assembly but also treats infinite and finite assemblies equally. In Chapters 4, 5, and 6, we briefly define local determinism, zeta-dimension, and the standard discrete Sierpinski triangle respectively. In Chapter 7 we prove that **S** does not strictly self-assemble in the TAM. In Chapter 8, we construct the fibered Sierpinski triangle and verify that its fractal dimension agrees with that of **S**. Finally, in Chapter 9 we prove that **T** strictly self-assembles in the TAM, and also prove the correctness of our construction.

This thesis is based on joint work with James I. Lathrop and Jack H. Lutz. Chapter 3 and 4 are based on Jack Lutz's lecture notes in the course "Computational Models of Nanoscale Self-Assembly", which he and Jim Lathrop taught at Iowa State University in the fall of 2006. Chapters 5 through 9 are based on the research paper [10].

2 NOTATION AND TERMINOLOGY

Throughout this thesis we work in the n-dimensional discrete Euclidean space $\mathbb{Z}^n$, where n is a positive integer. (In fact, we are primarily concerned with the discrete Euclidean plane $\mathbb{Z}^2$.) We write U_n for the set of all *unit vectors*, i.e., vectors of length 1, in $\mathbb{Z}^n$. We regard the $2n$ elements of U_n as (names of the cardinal) *directions* in $\mathbb{Z}^n$.

We write $[X]^2$ for the set of all 2-element subsets of a set X. All *graphs* in this thesis are undirected graphs, i.e., ordered pairs $G = (V, E)$, where V is the set of *vertices* and $E \subseteq [V]^2$ is the set of *edges*. A *cut* of a graph $G = (V, E)$ is a partition $C = (C_0, C_1)$ of V into two nonempty, disjoint subsets C_0 and C_1.

A *binding function* on a graph $G = (V, E)$ is a function $\beta : E \to \mathbb{N}$. (Intuitively, if $\{u, v\} \in E$, then $\beta(\{u, v\})$ is the strength with which u is bound to v by $\{u, v\}$ according to β. If β is a binding function on a graph $G = (V, E)$ and $C = (C_0, C_1)$ is a cut of G, then the *binding strength* of β on C is

$$\beta_C = \sum \{\beta(e) \,|e \in E, e \cap C_0 \neq \emptyset, \text{ and } e \cap C_1 \neq \emptyset\}.$$

The *binding strength* of β on the graph G is then

$$\beta(G) = \min \{\beta_C \,|C \text{ is a cut of } G\}.$$

A *binding graph* is an ordered triple $G = (V, E, \beta)$, where (V, E) is a graph and β is a binding function on (V, E). If $\tau \in \mathbb{N}$, then a binding graph $G = (V, E, \beta)$ is τ-stable if $\beta(V, E) \geq \tau$.

An n-dimensional *grid graph* is a graph $G = (V, E)$ in which $V \subseteq \mathbb{Z}^n$ and every edge $\{\vec{a}, \vec{b}\} \in E$ has the property that $\vec{a} - \vec{b} \in U_n$. The *full grid graph* on a set $V \subseteq \mathbb{Z}^n$ is the graph $G^{\#}_V = (V, E)$ in which E contains *every* $\{\vec{a}, \vec{b}\} \in [V]^2$ such that $\vec{a} - \vec{b} \in U_n$.

3 THE TILE ASSEMBLY MODEL

We review the basic ideas of the Tile Assembly Model. Our development largely follows that of [13, 12], but some of our terminology and notation are specifically tailored to our objectives. In particular, our version of the model only uses nonnegative "glue strengths", and it bestows equal status on finite and infinite assemblies.

Definition 1. An n-dimensional *tile type* is a function $t : U_n \to \mathbb{N} \times \mathbb{N}$. We write $t = (\mathrm{col}_t, \mathrm{str}_t)$, where $\mathrm{col}_t, \mathrm{str}_t : U_n \to \mathbb{N}$ are defined by $t(\vec{u}) = (\mathrm{col}_t(\vec{u}), \mathrm{str}_t(\vec{u}))$ for all $\vec{u} \in U_n$.

Intuitively, a tile of type t is an n-dimensional unit cube. It can be translated but not rotated, so it has a well-defined "side $\vec{u}$ " for each $\vec{u} \in U_n$. Each side $\vec{u}$ of the tile is covered with a "glue" of *color* $\mathrm{col}_t(\vec{u})$ and *strength* $\mathrm{str}_t(\vec{u})$. If tiles of types t and t' are placed with their centers at $\vec{a}$ and $\vec{a} + \vec{u}$, respectively, where $\vec{a} \in \mathbb{Z}^n$ and $\vec{u} \in U_n$, then they will *bind* with strength $\mathrm{str}_t(\vec{u}) \cdot [\![t(\vec{u}) = t'(-\vec{u})]\!]$ where $[\![\phi]\!]$ is the *Boolean* value of the statement ϕ. Note that this binding strength is 0 unless the adjoining sides have glues of both the same color and the same strength.

For the remainder of this chapter, unless otherwise specified, T is an arbitrary set of n-dimensional tile types, and $\tau \in \mathbb{N}$ is the "temperature."

Definition 2. A T-*configuration* is a partial function $\alpha : \mathbb{Z}^n \dashrightarrow T$.

Intuitively, a configuration is an assignment α in which a tile of type $\alpha(\vec{a})$ has been placed (with its center) at each point $\vec{a} \in \mathrm{dom}\ \alpha$. The following data structure characterizes how these tiles are bound to one another.

Definition 3. The *binding graph of* a T-configuration $\alpha : \mathbb{Z}^n \dashrightarrow T$ is the binding graph

$G_\alpha = (V, E, \beta)$, where (V, E) is the grid graph given by

$$V = \text{dom } \alpha,$$
$$E = \left\{ \vec{a}, \vec{b} \in [V]^2 \,\middle|\, \vec{a} - \vec{b} \in U_n, \text{col}_{\alpha(\vec{a})}\left(\vec{b} - \vec{a}\right) = \text{col}_{\alpha(\vec{b})}\left(\vec{a} - \vec{b}\right), \text{and str}_{\alpha(\vec{a})}\left(\vec{b} - \vec{a}\right) > 0 \right\},$$

and the binding function $\beta : E \to \mathbb{Z}^+$ is given by

$$\beta(\{\vec{a}, \vec{b}\}) = \text{str}_{\alpha(\vec{a})}(\vec{b} - \vec{a})$$

for all $\{\vec{a}, \vec{b}\} \in E$.

Definition 4.

1. A T-configuration α is *τ-stable* if its binding graph G_α is τ-stable.

2. A *τ-T-assembly* is a T-configuration that is τ-stable. We write $\mathcal{A}_T^\tau$ for the set of all τ-T-assemblies.

Note that, if $\tau > 0$, every τ-T-assembly $\alpha \in \mathcal{A}_T^\tau$ has a binding graph G_α that is connected.

Definition 5. Let α and α' be T-configurations.

1. α is a *subconfiguration* of α', and we write $\alpha \sqsubseteq \alpha'$, if dom $\alpha \subseteq$ dom α' and, for all $\vec{a} \in$ dom α, $\alpha(\vec{a}) = \alpha'(\vec{a})$.

2. α' is a *single-tile extension* of α if $\alpha \sqsubseteq \alpha'$ and dom α' − dom α is a singleton set. In this case, we write $\alpha' = \alpha + (\vec{a} \mapsto t)$, where $\{\vec{a}\} =$ dom α' − dom α and $t = \alpha'(\vec{a})$.

Note that the expression $\alpha + (\vec{a} \mapsto t)$ is only defined when $\vec{a} \in \mathbb{Z}^n$ − dom α.

We next define the "τ-t-frontier" of a τ-T-assembly α to be the set of all positions at which a tile of type t can be "τ-stably added" to the assembly α.

Definition 6. Let $\alpha \in \mathcal{A}_T^\tau$.

1. For each $t \in T$, the *τ-t-frontier* of α is the set

$$\partial_t^\tau \alpha = \left\{ \vec{a} \in \mathbb{Z}^n - \text{dom } \alpha \,\middle|\, \sum_{\vec{u} \in U_n} \text{str}_t(\vec{u}) \cdot [\![\alpha(\vec{a} + \vec{u})(-\vec{u}) = t(\vec{u})]\!] \geq \tau \right\}.$$

2. The τ-*frontier* of α is the set

$$\partial^\tau \alpha = \bigcup_{t \in T} \partial^\tau_t \alpha. \tag{3.1}$$

Remark. We note that the union (3.1) is not in general a disjoint union.

The following lemma shows that the definition of $\partial^\tau_t \alpha$ achieves the desired effect.

Lemma 3.1. Let $\alpha \in \mathcal{A}^\tau_T$, $\vec{a} \in \mathbb{Z}^n - \text{dom } \alpha$, and $t \in T$. Then $\alpha + (\vec{a} \mapsto t) \in \mathcal{A}^\tau_T$ if and only if $\vec{a} \in \partial^\tau_t \alpha$.

Proof. Assume the hypothesis, let $\alpha' = \alpha + (\vec{a} \mapsto t)$, and let $G_{\alpha'} = (V, E, \beta)$ be the binding graph of α'.

($\Rightarrow$): Assume that α' is a τ-T-assembly. Let $C = (C_0, C_1)$, where $C_0 = \text{dom } \alpha$ and $C_1 = \{\vec{a}\}$. Then $G_{\alpha'}$ is τ-stable, and C is a cut of $G_{\alpha'}$, so

$$\begin{aligned}
\tau &\leq \beta_C \\
&= \sum \{\beta(e) \,|\, e \in E, e \cap C_0 \neq \emptyset, \text{ and } e \cap C_1 \neq \emptyset\} \\
&= \sum \left\{\text{str}_t(\vec{u}) \,\middle|\, \vec{u} \in U_n, \vec{a} + \vec{u} \in \text{dom } \alpha, \alpha'(\vec{a})(\vec{u}) = \alpha'(\vec{a} + \vec{u})(-\vec{u}), \text{ and } \text{str}_{\alpha'(\vec{a})}(\vec{u}) > 0\right\} \\
&= \sum_{\vec{u} \in U_n} \text{str}_t(\vec{u}) [\![\alpha(\vec{a} + \vec{u})(-\vec{u}) = t(\vec{u})]\!],
\end{aligned}$$

so $\vec{a} \in \partial^\tau_t \alpha$.

($\Leftarrow$): Assume that $\vec{a} \in \partial^\tau_t \alpha$. To see that α' is a τ-T-assembly, let $C' = (C_0, C_1)$ be a cut of $G_{\alpha'}$. Without loss of generality, assume that $\vec{a} \in C_1$. If $C_1 = \{\vec{a}\}$ then we are done because $\vec{a} \in \partial^\tau_t \alpha$, so assume otherwise. Then $C = (C_0, C_1 - \{\vec{a}\})$ is a cut of G_α, and

$$\beta_{C'} \geq \beta_C \geq \tau$$

since G_α is τ-stable and $\vec{a} \in \partial^\tau_t \alpha$. $\square$

Notation. We write $\alpha \xrightarrow[\tau, T]{1} \alpha'$ (or, when τ and T are clear from context, $\alpha \xrightarrow{1} \alpha'$) to indicate that $\alpha, \alpha' \in \mathcal{A}^\tau_T$ and α' is a single-tile extension of α.

In general, self-assembly occurs with tiles adsorbing nondeterministically and asynchronously to a growing assembly. We now define assembly sequences, which are particular “execution traces” of how this might occur.

Definition 7. A *τ-T-assembly sequence* is a sequence $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ in $\mathcal{A}^\tau_T$, where $k \in \mathbb{Z}^+ \cup \{\infty\}$ and, for each i with $1 \le i+1 < k$, $\alpha_i \xrightarrow[\tau,T]{1} \alpha_{i+1}$.

Note that assembly sequences may be finite or infinite in length. Note also that, in any τ-T-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$, we have $\alpha_i \sqsubseteq \alpha_j$ for all $0 \le i \le j < k$.

Definition 8. The *result* of a τ-T-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ is the unique T-configuration $\alpha = \mathrm{res}(\vec{\alpha})$ satisfying $\mathrm{dom}\,\alpha = \bigcup_{0 \le i < k} \mathrm{dom}\,\alpha_i$ and $\alpha_i \sqsubseteq \alpha$ for each $0 \le i < k$.

It is clear that $\mathrm{res}(\vec{\alpha}) \in \mathcal{A}^\tau_T$ for every τ-T-assembly sequence $\vec{\alpha}$.

Definition 9. Let $\alpha, \alpha \in \mathcal{A}^\tau_T$.

1. A *τ-T-assembly sequence from α to α'* is a τ-T-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ such that $\alpha_0 = \alpha$ and $\mathrm{res}(\vec{\alpha}) = \alpha$.

2. We write $\alpha \xrightarrow[\tau,T]{} \alpha'$ (or, when τ and T are clear from context, $\alpha \longrightarrow \alpha'$) to indicate that there exists a τ-T-assembly sequence from α to α'.

Theorem 3.2. The binary relation $\xrightarrow[\tau,T]{}$ is a partial ordering of $\mathcal{A}^\tau_T$.

Proof. We write $\longrightarrow$ for $\xrightarrow[\tau,T]{}$. It is clear that $\longrightarrow$ is reflexive and antisymmetric. To see that $\longrightarrow$ is transitive, let $\alpha, \alpha', \alpha'' \in \mathcal{A}^\tau_T$ satisfy $\alpha \longrightarrow \alpha'$ and $\alpha' \longrightarrow \alpha''$. Then there exist a τ-T-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ from α to α' and a τ-T-assembly sequence $\vec{\alpha}' = (\alpha'_j \mid 0 \le j < k)$ from α' to α''. We cannot merely concatenate these assembly sequences, because k may be ∞. Instead, we "dovetail" $\vec{\alpha}$ and $\vec{\alpha}'$ in the following manner.

For each i and j with $1 \le i+1 < k$ and $1 \le j+1 < l$, let $\vec{a}_i, \vec{a}'_j \in \mathbb{Z}^n$ and $t_i, t'_j \in T$ be the locations and tile types such that $\alpha_{i+1} = \alpha_i + (\vec{a}_i \mapsto t_i)$ and $\alpha'_{j+1} = \alpha'_j + (\vec{a}'_j \mapsto t'_j)$. Define a sequence $\hat{\vec{\alpha}} = (\hat{\alpha}_m \mid 1 \le m+1 < k+l)$ via the procedure

$\hat{\alpha}_0 := \alpha_0;$

$i, j, m := 0, 0, 0;$

while $m + 2 < k + l$ **do**

    $P; Q$

end while,

where P is the macro

> **if** $i+1<k$ **then**
>
> $\quad \hat{\alpha}_{m+1} := \hat{\alpha}_m + (\vec{a}_i \mapsto t_i)$;
>
> $\quad i, m := i+1, m+1$
>
> **end if**

and Q is the macro

> **if** $j+1<l$ **then**
>
> $\quad$ **while** $\vec{a}'_j \notin \partial^{\tau}_{t'_j} \hat{\alpha}_m$ **do**
>
> $\quad\quad \hat{\alpha}_{m+1} := \hat{\alpha}_m + (\vec{a}_i \mapsto t_i)$;
>
> $\quad\quad i, j := i+1, m+1$
>
> $\quad$ **end while**
>
> $\quad \hat{\alpha}_{m+1} := \hat{\alpha}_m + (\vec{a}'_j \mapsto t'_j)$;
>
> $\quad j, m := j+1, m+1$
>
> **end if**

It is routine to verify that, regardless of the values of k and l, $\hat{\vec{\alpha}}$ is a τ-T-assembly sequence from α to α'', whence $\alpha \longrightarrow \alpha''$. □

Definition 10. An assembly $\alpha \in \mathcal{A}^{\tau}_{T}$ is *terminal* if it is a $\underset{\tau,T}{\longrightarrow}$-maximal element of $\mathcal{A}^{\tau}_{T}$.

It is clear that an assembly α is terminal if and only if $\partial^{\tau}\alpha = \emptyset$.

We now define an assembly sequence to be fair if no location remains forever in its frontier.

Definition 11. A τ-T-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ is *fair* if, for every $\vec{a} \in \mathbb{Z}^n$,

$$|\{i \,|0 \le i < k \text{ and } \vec{a} \in \partial^{\tau}\alpha_i\}| < \infty.$$

Note that every finite-length assembly sequence is fair. An obvious but useful property of infinite-length assembly sequences is that they are fair exactly when their results are terminal.

Observation 3.3. Let $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ be a τ-T-assembly sequence. If $k = \infty$ then $\mathrm{res}(\vec{\alpha})$ is terminal if and only if $\vec{\alpha}$ is fair.

We now show that every assembly is $\underset{\tau,T}{\longrightarrow}$-bounded by (i.e., can lead to) a terminal assembly.

Lemma 3.4. For each $\alpha \in \mathcal{A}_T^\tau$, there exists a $\alpha' \in \mathcal{A}_T^\tau$ such that $\alpha \underset{\tau,T}{\longrightarrow} \alpha'$ and α' is terminal.

Proof. Let $\alpha \in \mathcal{A}_T^\tau$. Fix an enumeration $\vec{a}_0, \vec{a}_1, \vec{a}_2, \ldots$ of $\mathbb{Z}^n$, and define a τ-T-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \leq i < k)$ via the procedure

$\alpha_0 := \alpha;$

$i := 0;$

while $\partial^\tau \alpha_i \neq \emptyset$ **do**

choose the least $j \in \mathbb{N}$ such that $\vec{a}_j \in \partial^\tau \alpha_i$;

choose $t \in T$ such that $\vec{a}_j \in \partial^\tau \alpha_i$;

$\alpha_{i+1} := \alpha_i + (\vec{a}_j \mapsto t);$

$i := i + 1$

end while

Note that our choice of the *least* j in each iteration of the while-loop ensures that $\vec{\alpha}$ is fair.

Let $\alpha' = \text{res}(\vec{\alpha})$. Then $\alpha \underset{\tau,T}{\longrightarrow} \alpha'$ is clear. If the while-loop terminates, then α' is terminal because $\partial^\tau \alpha_i = \emptyset$. If not, then α' is terminal by Observation 2.3. □

We now define tile assembly systems.

Definition 12.

1. An n-dimensional *generalized tile assembly system* (n*-GTAS*, or *GTAS*) is an ordered triple

$$\mathcal{T} = (T, \sigma, \tau),$$

where T is a set of n-dimensional tile types, $\sigma \in \mathcal{A}_T^\tau$ is the *seed assembly*, and $\tau \in \mathbb{N}$ is the *temperature*.

2. An n-dimensional *tile assembly system* (n*-TAS*, or *TAS*) is an n-GTAS $\mathcal{T} = (T, \sigma, \tau)$ in which the sets T and dom σ are finite.

Definition 13. A GTAS $\mathcal{T} = (T, \sigma, \tau)$ is *singly seeded* if σ is of the form

$$\sigma(\vec{a}) = \begin{cases} t \text{ if } \vec{a} = \vec{0} \\ \uparrow \text{otherwise} \end{cases}$$

for some $t \in T$.

Intuitively, a "run" of an n-GTAS $\mathcal{T} = (T, \sigma, \tau)$ is any τ-T-assembly sequence $\vec{\alpha} = (\alpha_i \mid 0 \le i < k)$ that begins with $\alpha_0 = \sigma$. Accordingly, we define the following sets.

Definition 14. Let $\mathcal{T} = (T, \sigma, \tau)$ be an n-GTAS.

1. The *set of assemblies produced by* $\mathcal{T}$ is

$$\mathcal{A}[\mathcal{T}] = \left\{ \alpha \in \mathcal{A}_T^\tau \middle| \sigma \underset{\tau,T}{\longrightarrow} \alpha \right\}.$$

2. The *set of terminal assemblies produced by* $\mathcal{T}$ is

$$\mathcal{A}_\Box[\mathcal{T}] = \{ \alpha \in \mathcal{A}[\mathcal{T}] \, | \alpha \text{ is terminal} \}.$$

Note that $\mathcal{A}_\Box[\mathcal{T}]$ is always nonempty by Lemma 2.4.

We are often interested in tile assembly systems that produced unique assemblies in the following sense.

Definition 15. An n-GTAS $\mathcal{T}$ is *definitive* if $|\mathcal{A}_\Box[\mathcal{T}]| = 1$. In this case, we say that $\mathcal{T}$ *produces the unique assembly* α, where $\mathcal{A}_\Box[\mathcal{T}] = \{\alpha\}$.

The following theorem characterizes definitive tile assembly systems in terms of the assembly sequence relation $\underset{\tau,T}{\longrightarrow}$.

Theorem 3.5. An n-GTAS $\mathcal{T}$ is definitive if and only if the partial ordering $\underset{\tau,T}{\longrightarrow}$ directs the set $\mathcal{A}[\mathcal{T}]$.

Proof. Let $\mathcal{T} = (T, \sigma, \tau)$ be an n-GTAS. Assume that $\mathcal{T}$ is definitive. To see that $\underset{\tau,T}{\longrightarrow}$ directs $\mathcal{A}[\mathcal{T}]$, let $\alpha', \alpha'' \in \mathcal{A}[\mathcal{T}]$. By Lemma 3.4, there exist $\hat{\alpha}', \hat{\alpha}'' \in \mathcal{A}_\Box[\mathcal{T}]$ such that $\alpha' \underset{\tau,T}{\longrightarrow} \hat{\alpha}'$ and $\alpha'' \underset{\tau,T}{\longrightarrow} \hat{\alpha}''$. Since $\mathcal{T}$ is definitive, we must have $\hat{\alpha}' = \hat{\alpha}''$, whence $\alpha' \underset{\tau,T}{\longrightarrow} \hat{\alpha}'$ and $\alpha'' \underset{\tau,T}{\longrightarrow} \hat{\alpha}'$. This shows that $\underset{\tau,T}{\longrightarrow}$ directs $\mathcal{A}[\mathcal{T}]$.

Conversely, assume that $\mathcal{T}$ is not definitive. Then there exist $\alpha', \alpha'' \in \mathcal{A}_\Box[\mathcal{T}]$ with $\alpha' \neq \alpha''$. Since α' and α'' are terminal, there is no assembly $\alpha \in \mathcal{A}[\mathcal{T}]$ with $\alpha' \underset{\tau,T}{\longrightarrow} \alpha$ and $\alpha'' \underset{\tau,T}{\longrightarrow} \alpha$. Hence, $\underset{\tau,T}{\longrightarrow}$ does not direct $\mathcal{A}[\mathcal{T}]$. ∎

In the present paper, we are primarily interested in the self-assembly of sets.

Definition 16. Let $\mathcal{T} = (T, \sigma, \tau)$ be an n-GTAS, and let $X \subseteq \mathbb{Z}^n$.

1. The set X *weakly self-assembles* in $\mathcal{T}$ if there is a set $B \subseteq T$ such that, for all $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, $\alpha^{-1}(B) = X$.

2. The set X *strictly self-assembles* in $\mathcal{T}$ if, for all $\alpha \in \mathcal{A}_\Box[\mathcal{T}]$, dom $\alpha = X$.

Intuitively, a set X weakly self-assembles in $\mathcal{T}$ if there is a designated set B of "black" tile types such that every terminal assembly of $\mathcal{T}$ "paints the set X - and only the set X - black". In contrast, a set X strictly self-assembles in $\mathcal{T}$ if every terminal assembly of $\mathcal{T}$ has tiles on the set X and only on the set X. Clearly, every set that strictly self-assembles in a GTAS $\mathcal{T}$ also weakly self-assembles in $\mathcal{T}$.

We now have the machinery to say what it means for a set in discrete Euclidean space to self-assemble in either the weak or the strict sense.

Definition 17. Let $X \subseteq \mathbb{Z}^n$.

1. The set X *weakly self-assembles* if there is an n-TAS $\mathcal{T}$ such that X weakly self-assembles in $\mathcal{T}$.

2. The set X *strictly self-assembles* if there is an n-TAS $\mathcal{T}$ such that X strictly self-assembles in $\mathcal{T}$.

Note that $\mathcal{T}$ is required to be a TAS, i.e., finite, in both parts of the above definition.

4 LOCAL DETERMINISM

In this chapter, we review the concept of *local determinism* invented by Soloveichik and Winfree [15]. Local determinism is a method for proving the correctness of tile assembly systems in which "irregular" shapes self-assemble. Namely, if an n-GTAS $\mathcal{T} = (T, \sigma, \tau)$ is locally deterministic, then one is assured that every τ-T-assembly sequence will lead to the correct (a.k.a., unique) result.

Notation. For each n-dimensional T-configuration α, each $\vec{m} \in \mathbb{Z}^n$, and each $\vec{u} \in U_n$,

$$\mathrm{str}_\alpha(\vec{m}, \vec{u}) = \mathrm{str}_{\alpha(\vec{m})}(\vec{u})[\![\alpha(\vec{m})(\vec{u}) = \alpha(\vec{m} + \vec{u})(-\vec{u})]\!].$$

(The Boolean value on the right is 0 if $\{\vec{m}, \vec{m} + \vec{u}\} \nsubseteq \mathrm{dom}\ \alpha$.)

Notation. If $\vec{\alpha} = (\alpha_i | 0 \le i < k)$ is a τ-T-assembly sequence and $\vec{m} \in \mathbb{Z}^n$, then the $\vec{\alpha}$-*index* of $\vec{m}$ is

$$i_{\vec{\alpha}}(\vec{m}) = \min\{ i \in \mathbb{N} \,|\, \vec{m} \in \mathrm{dom}\ \alpha_i \}.$$

Observation 4.1. If dom α_0 is finite, then

$$\vec{m} \in \mathrm{dom}\ \mathrm{res}(\vec{\alpha}) \Leftrightarrow i_{\vec{\alpha}}(\vec{m}) < \infty.$$

Notation. If $\vec{\alpha} = (\alpha_i | 0 \le i < k)$ is a τ-T-assembly sequence, then, for $\vec{m}, \vec{m}' \in \mathbb{Z}^n$,

$$\vec{m} \prec_{\vec{\alpha}} \vec{m}' \Leftrightarrow i_{\vec{\alpha}}(\vec{m}) < i_{\vec{\alpha}}(\vec{m}').$$

Definition 18. (Soloveichik and Winfree 2004 [15]) Let $\vec{\alpha} = (\alpha_i | 0 \le i < k)$ be a τ-T-assembly sequence, and let $\alpha = \mathrm{res}(\vec{\alpha})$. For each location $\vec{m} \in \mathrm{dom}\ \alpha$, define the following sets of directions.

1. $\mathrm{IN}^{\vec{\alpha}}(\vec{m}) = \left\{ \vec{u} \in U_n \,\middle|\, \vec{m} + \vec{u} \prec_{\vec{\alpha}} \vec{m} \text{ and } \mathrm{str}_{\alpha_{i_{\vec{\alpha}}(\vec{m})}}(\vec{m}, \vec{u}) > 0 \right\}.$

2. $\mathrm{OUT}^{\vec{\alpha}}(\vec{m}) = \left\{ \vec{u} \in U_n \mid -\vec{u} \in \mathrm{IN}^{\vec{\alpha}}(\vec{m}+\vec{u}) \right\}$.

3. $\mathrm{TERM}^{\vec{\alpha}}(\vec{m}) = U_n - \mathrm{IN}^{\vec{\alpha}}(\vec{m}) - \mathrm{OUT}^{\vec{\alpha}}(\vec{m})$.

Intuitively, $\mathrm{IN}^{\vec{\alpha}}(\vec{m})$ is the set of sides on which the tile at $\vec{m}$ initially binds in the assembly sequence $\vec{\alpha}$, and $\mathrm{OUT}^{\vec{\alpha}}(\vec{m})$ is the set of sides on which this tile propagates information to future tiles. $\mathrm{TERM}^{\vec{\alpha}}(\vec{m})$ is the set of sides which are neither input nor output sides.

Note that $\mathrm{IN}^{\vec{\alpha}}(\vec{m}) = \emptyset$ for all $\vec{m} \in \alpha_0$.

Notation. If $\vec{\alpha} = (\alpha_i | 0 \leq i < k)$ is a τ-T-assembly sequence, $\alpha = \mathrm{res}(\vec{\alpha})$, and $\vec{m} \in \mathrm{dom}\ \alpha - \mathrm{dom}\ \alpha_0$, then

$$\vec{\alpha} \setminus \vec{m}\,(\vec{x}) = \begin{cases} \alpha\,(\vec{x}) \text{ if } \vec{x} \notin \mathrm{dom}\ \alpha - \{\vec{m}\} - \left(\vec{m} + \mathrm{OUT}^{\vec{\alpha}}(\vec{m})\right) \\ \qquad \uparrow \text{ otherwise} \end{cases}$$

(Note that $\vec{\alpha} \setminus \vec{m}$ is a T-configuration that may or may not be a τ-T-assembly.

Definition 19. (Soloveichik and Winfree 2004 [15]). A τ-T-assembly sequence $\vec{\alpha} = (\alpha_i | 0 \leq i < k)$ with $\alpha = \mathrm{res}(\vec{\alpha})$ is *locally deterministic* if it has the following three properties.

1. For all $\vec{m} \in \mathrm{dom}\ \alpha - \mathrm{dom}\ \alpha_0$,

$$\sum_{\vec{u} \in \mathrm{IN}^{\vec{\alpha}}(\vec{m})} \mathrm{str}_{\alpha_{i_{\vec{\alpha}}(\vec{m})}}(\vec{m}, \vec{u}) = \tau.$$

2. For all $\vec{m} \in \mathrm{dom}\ \alpha - \mathrm{dom}\ \alpha_0$ and all $t \in T - \{\alpha(\vec{m})\}$, $\vec{m} \notin \partial_t^\tau \vec{\alpha} \setminus \vec{m}$.

3. $\partial^\tau \alpha = \emptyset$.

That is, $\vec{\alpha}$ is locally deterministic if (1) each tile added in $\vec{\alpha}$ "just barely" binds to the assembly; (2) if a tile of type t_0 at a location $\vec{m}$ and its immediate "OUT-neighbors" are deleted from the *result* of $\vec{\alpha}$, then no tile of type $t \neq t_0$ will attach itself to the thus-obtained configuration at location $\vec{m}$; and (3) the result of $\vec{\alpha}$ is terminal.

Definition 20. An n-dimensional GTAS $\mathcal{T} = (T, \sigma, \tau)$ is *locally deterministic* if there exists a locally determinstic τ-T-assembly sequence $\vec{\alpha} = (\alpha_i | 0 \leq i < k)$ with $\alpha_0 = \sigma$.

Theorem 4.2. (Soloveichik and Winfree 2004 [15]) Every locally deterministic n-GTAS is definitive.

5 ZETA-DIMENSION

The most commonly used dimension for continuous fractals is Hausdorff dimension while the most commonly used dimension for discrete fractals is zeta-dimension. In this chapter, we give a brief overview of zeta-dimension.

Zeta-dimension has been re-discovered several times by researchers in various fields over the past few decades, but its origins actually lie in Euler's (real-valued predecessor of the Riemann) zeta-function [8] and Dirichlet series. For each set $A \subseteq \mathbb{Z}^2$, define the *A-zeta-function* $\zeta_A : [0,\infty) \to [0,\infty]$ by $\zeta_A(s) = \sum_{(0,0)\neq(m,n)\in A}(|m|+|n|)^{-s}$ for all $s \in [0,\infty)$. Then the *zeta-dimension* of A is

$$\mathrm{Dim}_\zeta(A) = \inf\{s | \zeta_A(s) < \infty\}.$$

It is clear that $0 \leq \mathrm{Dim}_\zeta(A) \leq 2$ for all $A \subseteq \mathbb{Z}^2$. It is also easy to see (and was proven by Cahen in 1894; see also [4, 9]) that zeta-dimension admits the "entropy characterization"

$$\mathrm{Dim}_\zeta(A) = \limsup_{n\to\infty} \frac{\log |A_{\leq n}|}{\log n} \tag{5.1}$$

where $A_{\leq n} = \{(m,n) \in A \mid |m| + |n| \leq n\}$. Various properties of zeta-dimension, along with extensive historical citations, appear in the recent paper [7], but our technical arguments here can be followed without reference to this material. We use the fact that (5.1) can be transformed by changes of variable up to exponential, e.g.,

$$\mathrm{Dim}_\zeta(A) = \limsup_{n\to\infty} \frac{\log |A_{[0,2^n]}|}{n}$$

also holds.

6 THE STANDARD DISCRETE SIERPINSKI TRIANGLE S

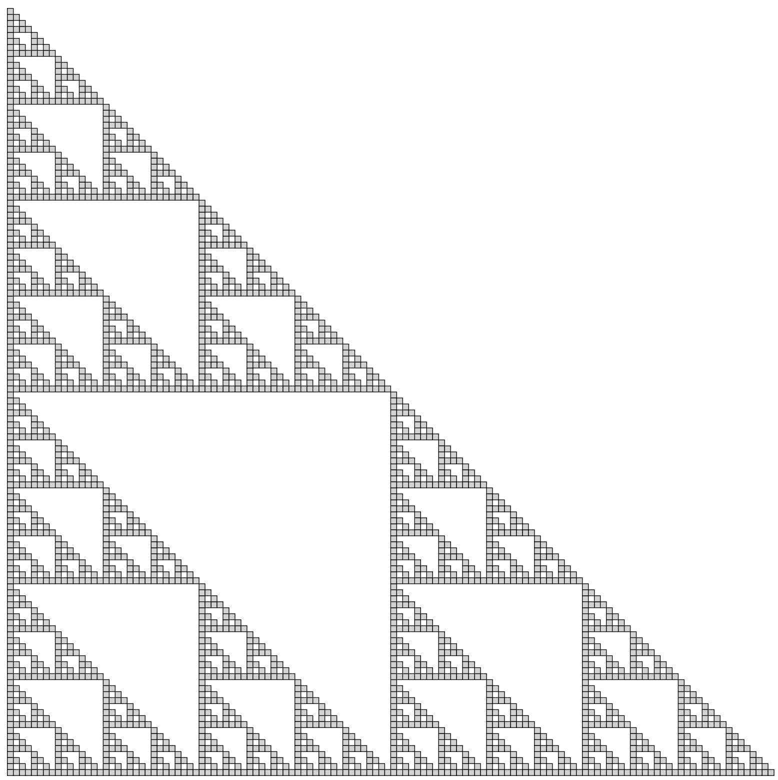


Figure 6.1 Standard Discrete Sierpinski Triangle **S**

In this chapter, we briefly review the standard discrete Sierpinski triangle and the calculation of its zeta-dimension.

Let $V = \{(1,0),(0,1)\}$. Define the sets $S_0, S_1, S_2, \cdots \subseteq \mathbb{Z}^2$ by the recursion

$$S_0 = \{(0,0)\}, \tag{6.1}$$

$$S_{i+1} = S_i \cup \left(S_i + 2^i V\right),$$

where $A + cB = \{\vec{a} + c\vec{b} | \vec{a} \in A \text{ and } \vec{b} \in B\}$. Then the *standard discrete Sierpinski triangle* is the set

$$\mathbf{S} = \bigcup_{i=0}^{\infty} S_i,$$

which is illustrated in Figure 1. Note that there is an obvious resemblance between the discrete Sierpinski triangle and its continuous counterpart despite the fact that the latter is a compact subset of real numbers while the former is an infinite subset of $\mathbb{Z}^2$. Moreover, the discrete and continuous Sierpinski triangles not only look similar but they also share the same fractal dimension. This is not a mere coincidence but rather an instance of a more general relationship [7] between the Hausdorff and zeta-dimension of certain types of self-similar fractals.

It is well known that $\mathbf{S}$ is the set of all $(k, l) \in \mathbb{N}^2$ such that the binomial coefficient $\binom{k+l}{k}$ is odd. For this reason, the set $\mathbf{S}$ is also called *Pascal's triangle modulo 2*. It is clear from the recursion (6.1) that $|S_i| = 3^i$ for all $i \in \mathbb{N}$. The zeta-dimension of $\mathbf{S}$ is thus

$$\begin{aligned}
\mathrm{Dim}_\zeta(\mathbf{S}) &= \limsup_{n\to\infty} \frac{\log \left|\mathbf{S}_{[0,2^n]}\right|}{n} \\
&= \limsup_{n\to\infty} \frac{\log |S_n|}{n} \\
&= \log 3 \\
&\approx 1.585.
\end{aligned}$$

7 IMPOSSIBILITY OF STRICT SELF-ASSEMBLY OF S

This chapter presents our first main theorem, which says that the standard discrete Sierpinski triangle **S** does not strictly self-assemble in the Tile Assembly Model. In order to prove this theorem, we first develop a lower bound on the number of tile types required for the self-assembly of a set X in terms of the depths of finite trees that occur in a certain way as subtrees of the full grid graph $G^\#_X$ of X.

Intuitively, given a set D of vertices of $G^\#_X$ (which is in practice the domain of the seed assembly), we define a D-subtree of $G^\#_X$ to be any rooted tree in $G^\#_X$ that consists of *all* vertices of $G^\#_X$ that lie at or on the far side of the root from D. For simplicity, we state the definition in an arbitrary graph G.

Definition 21. Let $G = (V, E)$ be a graph, and let $D \subseteq V$.

1. For each $r \in V$, the *D-r-rooted subgraph* of G is the graph $G_{D,r} = (V_{D,r}, E_{D,r})$, where

   $$V_{D,r} = \{v \in V \mid \text{every path from } v \text{ to (any vertex in) } D \text{ in } G \text{ goes through } r\}$$

   and

   $$E_{D,r} = E \cap [V_{D,r}]^2 .$$

   (Note that $r \in V_{D,r}$ in any case.)

2. A *D-subtree* of G is a rooted tree B with root $r \in V$ such that $B = G_{D,r}$.

3. A *branch* of a D-subtree B of G is a simple path $\pi = (v_0, v_1, \ldots)$ that starts at the root of B and either ends at a leaf of B or is infinitely long.

We use the following quantity in our lower bound theorem.

Definition 22. Let $G = (V, E)$ be a graph and let $D \subseteq V$. The *finite-tree depth* of G *relative to* D is

$$\text{ft-depth}_D\,(G) = \sup\left\{\text{depth(B)} | B \text{ is a finite } D\text{-subtree of } G\right\}.$$

We emphasize that the above supremum is only taken over *finite* D-subtrees. It is easy to construct an example in which G has a D-subtree of infinite depth, but $\text{ft-depth}_D\,(G) < \infty$.

Example 7.1. Let $X = \{(m, 0)\,| m \in \mathbb{N}\} \cup \{(2n+1, 1)\,| n \in \mathbb{N}\}$ and note that $G^{\#}_X$ is an infinite tree. Now let T be subtree rooted at $\vec{r} = (0, 0)$. It is clear that T is a D-subtree rooted at $\vec{r}$ of infinite depth. However, $\text{ft-depth}_D\,(T) < \infty$ because every finite D-subtree must be rooted at some point in the set $\{(n, 1)| n \geq 1 \text{ and } n \text{ is odd}\}$, which clearly results in a constant depth.

To prove our lower bound result, we use the following theorem from [2].

Theorem 7.2. (Adleman, Cheng, Goel, Huang, Kempe, Moisset de Espanés, and Rothemund 2002 [2]) Let $X \subseteq \mathbb{Z}^2$ with $|X| < \infty$ such that $G^{\#}_X$ is a tree rooted at the origin. If X strictly self-assembles in the singly seeded 2-TAS $\mathcal{T} = (T, \sigma, 2)$ then $|T| \geq \text{depth}\left(G^{\#}_X\right)$.

We now prove our lower bound result, which is the following.

Theorem 7.3. Let $X \subseteq \mathbb{Z}^2$. If X strictly self-assembles in a 2-GTAS $\mathcal{T} = (T, \sigma, \tau)$, then $|T| \geq \text{ft-depth}_{\text{dom }\sigma}\left(G^{\#}_X\right)$.

Proof. Assume the hypothesis, and let B be a finite dom σ-subtree of $G^{\#}_X$. If suffices to prove that $|T| \geq \text{depth}(B)$.

Let $\alpha \in \mathcal{A}_{\Box}[\mathcal{T}]$, and let $\vec{r}$ be the root of B. Let σ' be the assembly with dom $\sigma' = \{\vec{r}\}$ and $\vec{u}, \vec{v} \in U_2$. We define $\sigma'(\vec{r})$ as follows.

$$\sigma'(\vec{r}) = \left(\text{col}_{\sigma(\vec{r})}(\vec{u}), \text{str}_{\sigma(\vec{r})}(\vec{u})\right) = \begin{cases} \left(\text{col}_{\alpha(\vec{r})}(\vec{u}), \text{str}_{\alpha(\vec{r})}(\vec{u})\right) \text{ if } \vec{u} + \vec{v} \in B \\ \qquad\qquad (-\infty, 0) \text{ otherwise} \end{cases}$$

Then $\mathcal{T}' = (T, \sigma', \tau)$ is an n-GTAS in which B self-assembles. By Theorem 7.2, this implies that $|T| \geq \text{depth}(B)$. $\square$

We note that Adleman, Cheng, Goel, Huang, Kempe, Moisset de Espanés, and Rothemund [2] proved the special case of Theorem 7.3 in which $G^{\#}_X$ is itself a finite tree and dom $\sigma = \{\vec{r}\}$, where $\vec{r}$ is the root of $G^{\#}_X$.

We now show that the standard discrete Sierpinski triangle $\mathbf{S}$ has infinite finite-tree depth.

Lemma 7.4. For every finite set $D \subseteq S$, $\text{ft-depth}_D\left(G^{\#}_{\mathbf{S}}\right) = \infty$.

Proof. Let $D \subseteq \mathbf{S}$ be finite, and let m be a positive integer. It suffices to show that $\text{ft-depth}_D\left(G^{\#}_{\mathbf{S}}\right) > m$. Choose $k \in \mathbb{N}$ large enough to satisfy the following two conditions.

(i) $2^k > \max\{a \in \mathbb{N} | (\exists b \in \mathbb{N})(a,b) \in D\}$,

(ii) $2^k > m$.

Let $\vec{r}_k = (2^{k+1}, 2^k)$, and let

$$B_k = \left\{(a,b) \in \mathbf{S} \,\middle|\, a \geq 2^{k+1}, b \geq 2^k \text{ and } a+b \leq 2^{k+2}-1\right\}.$$

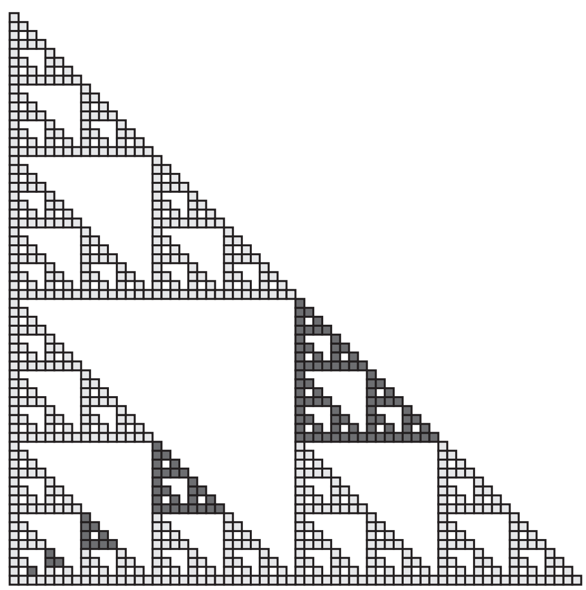


Figure 7.1 The set B_k for $k = 0, 1, 2, 3, 4$.

It is routine to verify that B_k is a finite D-subtree of $G^{\#}_{\mathbf{S}}$ with root at $\vec{r}$ and depth 2^k. It follows that

$$\text{ft-depth}_D\left(G^{\#}_{\mathbf{S}}\right) \geq \text{depth}\left(B_k\right) = 2^k > m.$$

□

We now have the machinery to prove our first main theorem.

Theorem 7.5. **S** does not strictly self-assemble in the Tile Assembly Model.

Proof. Let $\mathcal{T} = (T, \sigma, \tau)$ be an 2-GTAS in which **S** strictly self-assembles. It suffices to show that $\mathcal{T}$ is not an 2-TAS. If dom σ is infinite, this is clear, so assume that dom σ is finite. Then Theorem 7.3 and Lemma 7.4 tell us that $|T| = \infty$, whence $\mathcal{T}$ is not an 2-TAS. □

Note that the results of this chapter easily extend to other infinite "tree shapes" in which finite subtrees of arbitrary depth exist.

8 THE FIBERED SIERPINSKI TRIANGLE T

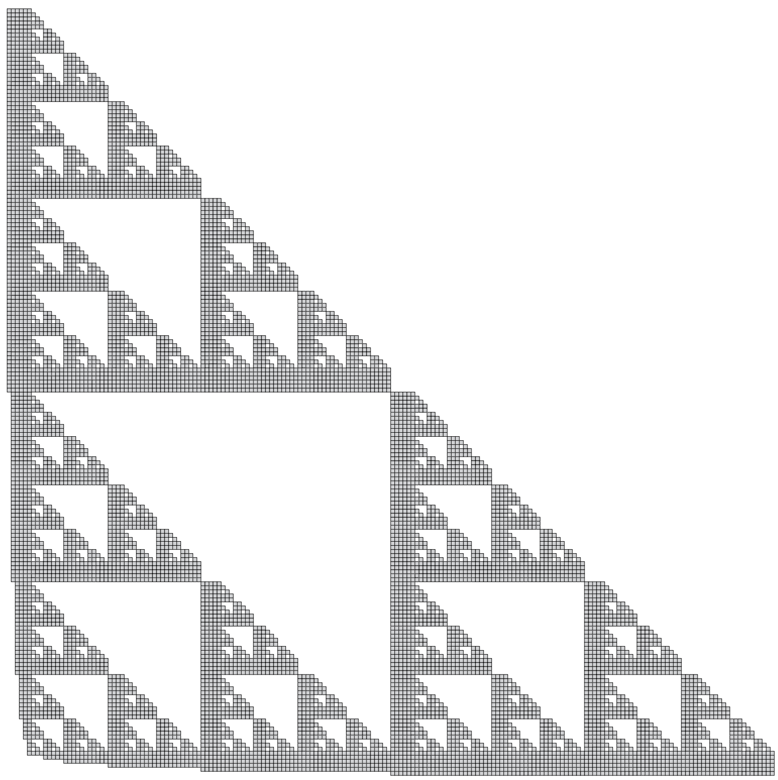


Figure 8.1 Fibered Sierpinski Triangle **T**

We now define the fibered Sierpinski triangle and show that it has the same zeta-dimension as the standard discrete Sierpinski triangle.

As in Section 2, let $V = \{(1,0),(0,1)\}$. Our objective is to define sets $T_0, T_1, T_2, \dots \subseteq \mathbb{Z}^2$, sets $F_0, F_1, F_2, \dots \subseteq \mathbb{Z}^2$, and functions $l, f, t : \mathbb{N} \to \mathbb{N}$ with the following intuitive meanings.

1. T_i is the i^{th} stage of our construction of the fibered Sierpinski triangle.

2. F_i is the *fiber* associated with T_i, a thin strip of tiles along which data moves in the self-

assembly process of Section 5. It is the smallest set whose union with T_i has a vertical left edge and a horizontal bottom edge, together with one additional layer added to these two now-straight edges.

3. $l(i)$ is the length (number of tiles in) the left (or bottom) edge of $T_i \cup F_i$.

4. $f(i) = |F_i|$.

5. $t(i) = |T_i|$.

These five entities are defined recursively by the equations

$$
\begin{aligned}
&T_0 = S_2 \text{ (stage 2 in the construction of } S\text{)},\\
&F_0 = (\{-1\} \times \{-1, 0, 1, 2, 3\}) \cup (\{-1, 0, 1, 2, 3\} \times \{-1\}),\\
&l(0) = 5,\\
&f(0) = 9,\\
&t(0) = 9,\\
&T_{i+1} = T_i \cup ((T_i \cup F_i) + l(i)V), \qquad (4.1)\\
&F_{i+1} = F_i \cup (\{-i-1\} \times \{-i-1, -i, \cdots, l(i+1) - i - 2\})\\
&\cup (\{-i-1, -i, \cdots, l(i+1) - i - 2\} \times \{-i-1\}),\\
&l(i+1) = 2l(i) + 1,\\
&f(i+1) = f(i) + 2l(i+1) - 1,\\
&t(i+1) = 3t(i) + 2f(i).
\end{aligned}
$$

Comparing the recursions (2.1) and (4.1) shows that the sets $T_0, T_1, T_2, \cdots$ are constructed exactly like the sets $S_0, S_1, S_2, \cdots$, except that the fibers F_i are inserted into the construction of the sets T_i. A routine induction verifies that this recursion achieves conditions 2, 3, 4, and 5 above. The *fibered Sierpinski triangle* is the set

$$\mathbf{T} = \bigcup_{i=0}^{\infty} T_i,$$

which is illustrated in Figure 2. The resemblance between **S** and **T** is clear from the illustrations. We now verify that **S** and **T** have the same zeta-dimension.

Lemma 8.1. $\mathrm{Dim}_\zeta(\mathbf{T}) = \mathrm{Dim}_\zeta(\mathbf{S})$.

Proof. Solving the recurrences for l, f, and t, in that order, gives the formulas

$$\begin{aligned} l(i) &= 3 \cdot 2^{i+1} - 1, \\ f(i) &= 3(2^{i+3} - i - 5), \\ t(i) &= \frac{3}{2}\left(3^{i+3} - 2^{i+5} + 2i + 11\right), \end{aligned}$$

which can be routinely verified by induction. It follows readily that

$$\mathrm{Dim}_\zeta\left(\mathbf{T}\right) = \limsup_{n\to\infty} \frac{\log t(n)}{\log l(n)} = \log 3 = \mathrm{Dim}_\zeta\left(\mathbf{S}\right).$$

□

We note that the thickness $i+1$ of a fiber F_i is $O(\log l(i))$, i.e., logarithmic in the side length of T_i. Hence the difference between S_i and T_i is asymptotically negligible as $i \to \infty$. Nevertheless, we show in the next chapter that $\mathbf{T}$, unlike $\mathbf{S}$, strictly self-assembles in the Tile Assembly Model.

9 STRICT SELF-ASSEMBLY OF T

This chapter is devoted to proving our second main theorem, which is the following.

Theorem 9.1. T strictly self-assembles in the Tile Assembly Model.

To prove Theorem 9.1, we present a modified version of an optimal binary counter that contains a "shift" element in tandem with the counter. This results in the placement of output ports (points at which structures may attach to the modified counter) which are used to grow additional modified counters at right angles. Recursively applying this procedure to the new counters yields the fibered Sierpinski triangle, as shown in Figure 8.1. Our singly-seeded tile set that produces the fibered Sierpinski triangle in this manner contains 51 tile types.

The remainder of this chapter gives a detailed description on how these modified counters are constructed and joined together to form the fibered Sierpinski triangle along with a proof of correctness. We first review the optimal counter from [6].

The Standard Optimal Counter

In this subsection we review the optimal binary counter due to Cheng, Goel and Moisett de Espanés [6]. The optimality of the counter refers to its running time, which is a stochastic process and thus beyond the scope of this thesis. However, it is perhaps the simplest construction of a binary counter in the TAM, and is an ideal candidate for use in more complicated structures such as the fibered Sierpinski triangle. We give the complete tile set for the optimal binary counter in the following figure.

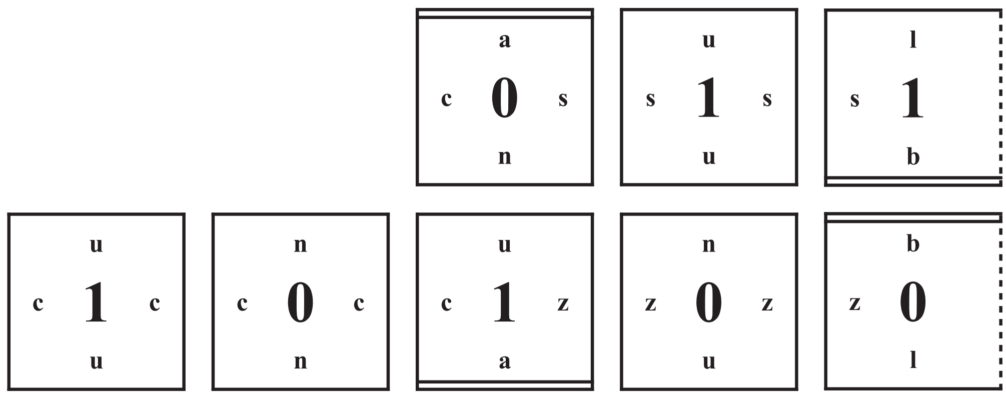


Figure 9.1 The tile set for the optimal binary counter.

Self-assembly of the optimal counter proceeds as follows. We first assume the existence of an initial row of k tiles, which actually requires k additional tile types to self-assemble from a single seed tile. The right most tile in the initial row will begin the self-assembly of the counter. In general, the first tile type to attach in any row will do so above the right most 0 bit in the previous row. Note that there is a special tile type designed to attach in the least-significant bit position. The self-assembly of a row (i.e., an increment operation) emanates outward from the initial tile that attaches in both directions. The bits in the previous row that are located to the left of the initial tile are simply copied up to the current row, and then the tiles to the right of the initial tile are simply set to 0. This process continues until a right most 0 bit does not exist. We give an example of the self-assembly of the optimal binary counter in Figure 9.2. (See [6] for a complete discussion of the optimal binary counter.)

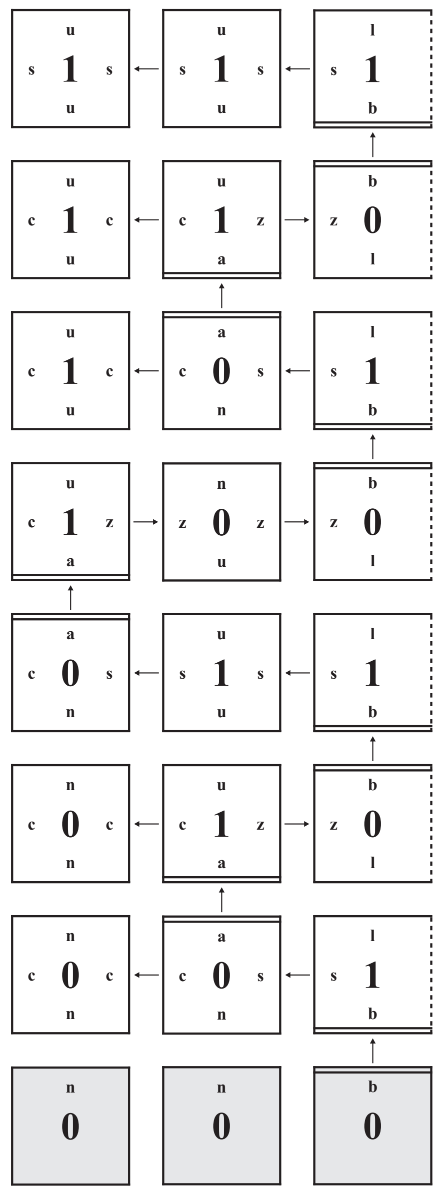


Figure 9.2 Optimal binary counter.

A Modified Optimal Counter

In this subsection, we will construct a vertically oriented, "log-width" binary counter, based on the optimal binary counter presented in the previous subsection. We modify our counter to have the property that each number is counted twice and then is repeated as many times as there are 0 bits to the right of its right most 1 bit. This results in our modified counter having a recursive structure similar to that of the tick marks on a ruler.

Figures 9.3, 9.4, and 9.5 collectively represent a set of tile types that we will use to construct a self-contained, singly-seeded 2-TAS in which our modified vertical log-width counter strictly self-assembles.

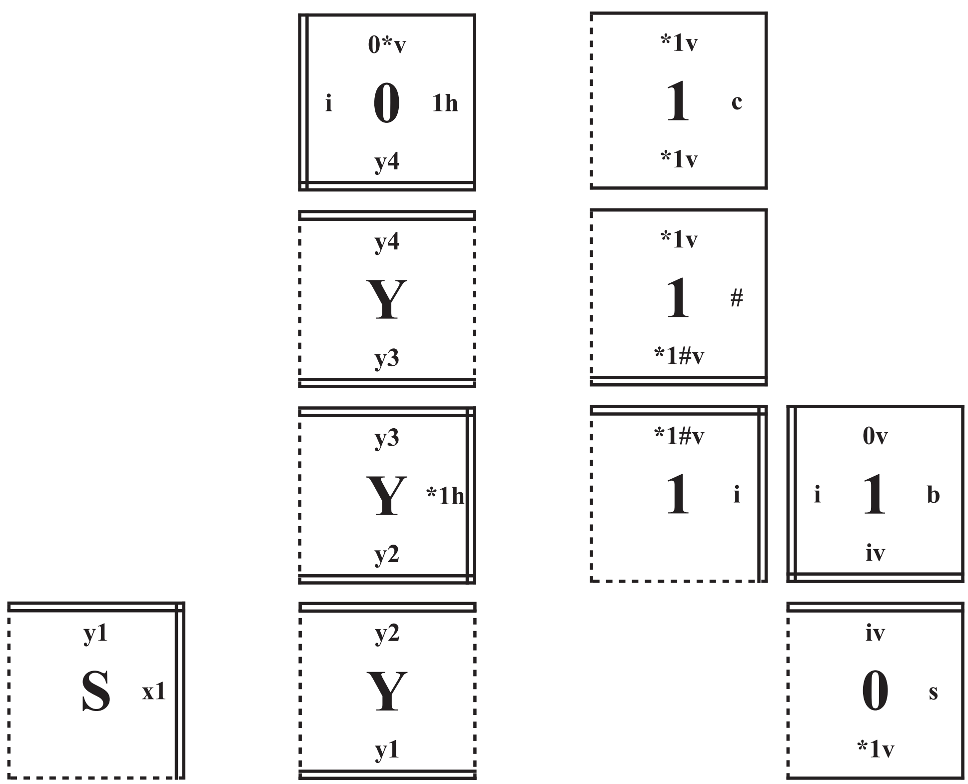


Figure 9.3 The seed, initialization, and transition tile types.

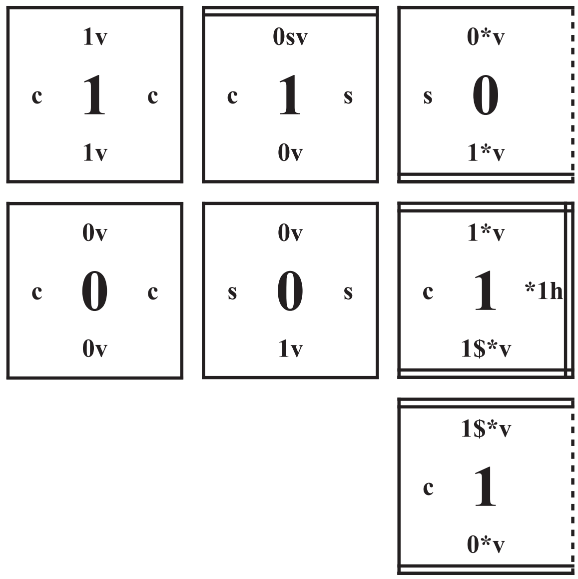


Figure 9.4 Tile types used to increment the counter.

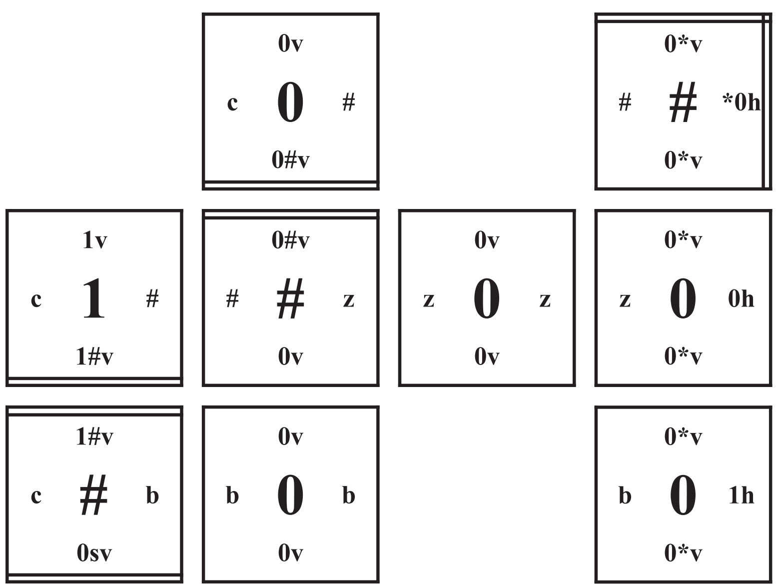


Figure 9.5 Tile types used to self-assemble spacing rows and provide output ports.

Construction 9.2. Let $\mathcal{T} = (T_{\mathrm{V}}, \sigma, 2)$ be the 2-TAS where

- T_{V} is the set of tile types given in Figures 9.3, 9.4, and 9.5, and
- σ is the seed assembly that places the seed tile at the origin and is undefined at all other locations.

Self-assembly of $\mathcal{T}$ starts with a single seed tile, which is the tile labeled 'S' in Figure 9.3. The initial eight tile types that attach in a modified log-width counter do so according to the unique assembly sequence illustrated below.

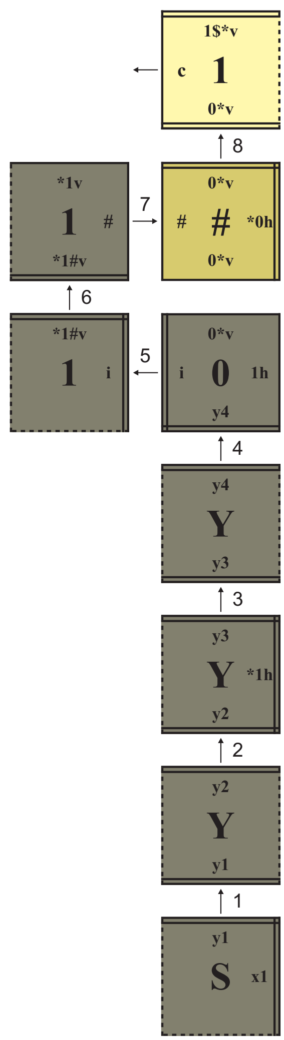


Figure 9.6 Initialization of a vertical log-width optimal counter.

In general, self-assembly proceeds in *stages*, where a *stage* of width w of a log-width counter is defined as any contiguous sequence of rows having width w. A *transition* from the current stage to the next stage in a log-width counter is when the tile type whose south edge is colored with the character 'i' binds. Note that there is one, and only one tile type to which the first tile of the next stage can bind.

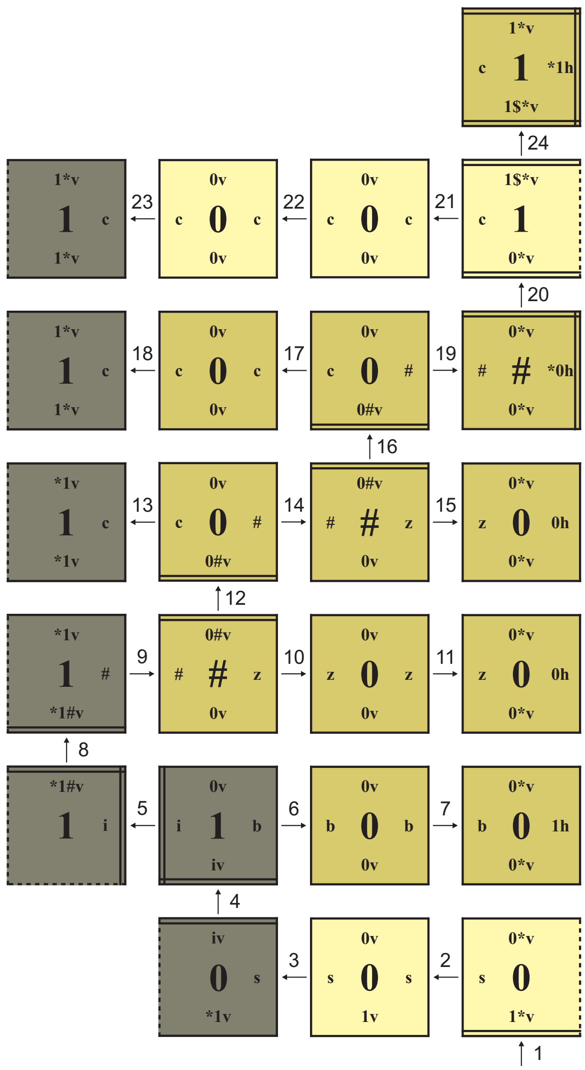


Figure 9.7 Vertical log-width modified optimal counter transition from T_2 to T_3.

As a result of a transition, the width of the counter increases by 1 to the left. This means that the right borders of every stage in the log-width counter are flush. The above figure shows a detailed example of a transition.

Each row in a modified log-width counter is either a *count* row or a *spacing* row. Count rows are the rows in which the counter is incremented by 1. We force the most significant bit in every count row, excluding the final count row, in our modified counter to always be 1. Figure 9.8a shows the relationship between the counting rows and the spacing rows for a particular stage (of width 4) of a vertical log-width counter. As shown in this figure, the darker shaded, un-numbered tiles depict the spacing rows, which are used to "delay" the counter by an appropriate amount. The height (number of rows) in the spacing element is determined by the position of the right most 1 digit in the binary counter. A 1 in the least significant bit (position 0) generates a single spacing row above the count. In general, if position i is the location of the right most 1 digit in the binary counter, then the number of spacing rows after that count is $i + 1$. Note that for a stage of width w, the initial w rows that attach are all spacing rows.

Figure 9.8b shows the locations of the output ports, depicted in dark green on the right edge. It is to these locations (and only these locations) that oppositely oriented fixed-width modified counters ultimately attach.

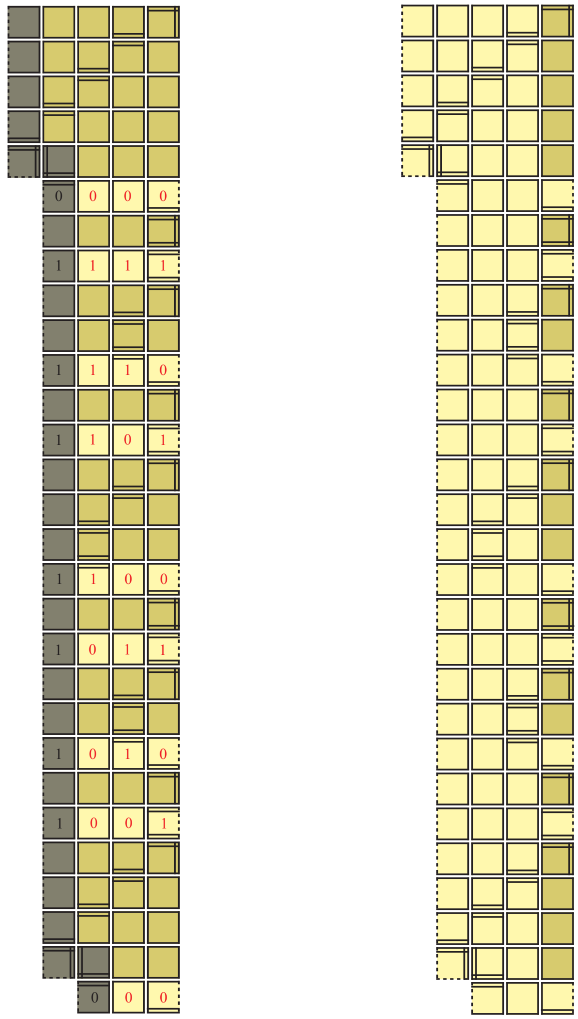


Figure 9.8a. Spacing rows   Figure 9.8b. Output ports

Figure 9.8 Vertical log-width modified optimal counter.

The internal structure of the modified counter simply contains a fixed-width optimal binary counter (see [6] for a detailed discussion), interweaved with spacing rows, along with a special set of left border tiles that allow it to make transitions (i.e., the darkest shaded, left most tiles in Figure 9.8a). To the north edge of each count row, a spacing row attaches. The number of spacing rows that follow the count row are determined by the position of the right most 1 in the previous count row. A "token" tile attaches to this special tile and additional spacing rows attach but with the token tile shifted to the right one tile. The spacing rows are terminated when the token tile reaches the right edge of the counter. The spacing rows retain the current count of the counter by passing this information through the glue labels of the spacing tiles vertically. In this way, the last spacing row provides the correct glue labels to produce the next counting row. This process can be seen in the following figure with the '#' indicating this special token tile. Here, we can see that the number 1100 (represented by the first row of tiles) is counted once, followed by exactly three spacing rows because the number 1100 has two zero bits following the right most one bit.

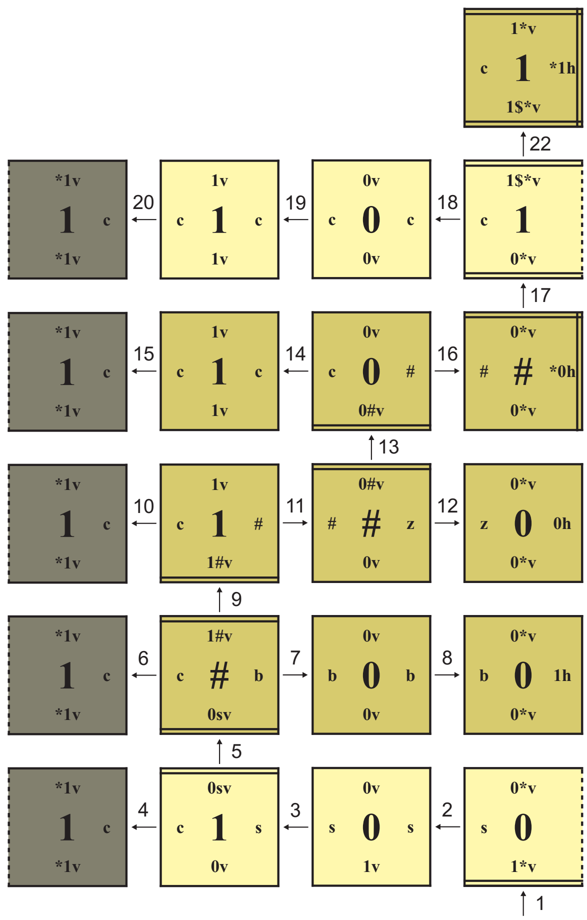


Figure 9.9 Shifting element in a vertical modified log-width optimal counter.

Note that the final row of each stage is a count row in which the north label of left most tile type is colored with the string 'i' and hence initiates a transition from the current stage to the next stage (see Figure 9.7).

The following observation testifies to the fact that there exists a direct relationship between the fibered Sierpinski triangle and our modified optimal counter.

Observation 9.3. The number of rows in a stage of width $w + 2$ for a log-width modified counter is $l(w) = 3 \cdot 2^{w+1} - 1$.

Proof. Note that the total number of rows in a particular stage of width $w + 2$ is the number of count rows plus the number of spacing rows. It is clear that there are exactly 2^{w+1} count rows, and the number of spacing rows is

$$(w + 2) + \sum_{i=0}^{w} (i + 1) \cdot 2^{w-i} = 2^{w+2} - 1.$$

□

A Tile Assembly System for T

We will now present a 2-TAS in which the set **T** strictly self-assembles. However, before we can do so, we must first construct three more types of modified optimal counters: a vertical fixed-width counter and both a horizontal log and fixed-width counter.

First, note that Construction 9.2 actually gives us two modified counters for the price of one. From the 2-TAS $\mathcal{T}$, we can easily extract a set $T'_{\mathrm{V}} \subset T_{\mathrm{V}}$ of tile types that self-assemble into a vertical fixed-width modified counter, but do so *only* in the presence of the output ports that are provided by some horizontal modified counter. We simply take T'_{V} to be the set of tile types given in Construction 9.2 that include all but the two left most columns of tile types (i.e., all non-left border tile types).

Furthermore (and with hardly any additional effort), we can squeeze two more modified counters out of Construction 9.2! To see this, simply take the set T_{V} from Construction 9.2 and "reflect" each tile type, other than the seed tile, about the line $y = x$. Call this new set of tile types T_{H}. We must also change all the vertical indicator symbols to horizontal indicators and vice versa. This will not affect the behavior of T_{H}, but it will prevent the inadvertent binding between tile types that belong to oppositely oriented modified counters. It is clear that this simple construction gives us a new 2-TAS in which a horizontal log-width counter, having the same fundamental properties as its vertical counterpart, self-assembles. And of course, doing so also gives us a horizontal fixed-width modified counter that is able to attach to the output ports provided by modified vertical counters.

We will forego a detailed exposition of the aforementioned counters since their behavior can all be understood in terms of the vertical log-width modified optimal counter of Construction 9.2.

The following construction yields a 2-TAS in which $\mathbf{T}$ strictly self-assembles.

Construction 9.4. Let $\mathcal{T}_{\mathbf{T}} = (T, \sigma, 2)$, where

- $T = T_{\mathrm{V}} \cup T_{\mathrm{H}}$, and
- σ is the same as in Construction 9.2.

Proof of Correctness

We are now ready to prove that $\mathcal{T}_{\mathbf{T}}$ is a 2-TAS that uniquely produces the fibered Sierpinski triangle. Note that the following lemma implies Theorem 9.1.

Lemma 9.5. The 2-TAS $\mathcal{T}_{\mathbf{T}} = (T, \sigma, 2)$, given in Construction 9.4, is definitive.

Proof. To prove the claim, it suffices to exhibit a locally deterministic 2-T-assembly sequence.

Let $\vec{\alpha} = (\alpha_i | 0 \leq i < k)$ be the infinite 2-T-assembly sequence that self-assembles $\mathbf{T}$ one stage at a time, where each stage is self-assembled one modified counter at a time, and the modified counters are self-assembled in the order of increasing distance from the origin. Note that if two modified counters share the same distance from the origin, we assume that $\vec{\alpha}$ self-assembles the counter having the greater x-coordinate first. This implies that $\vec{\alpha}$ will self-assemble a modified counter if and only if the output ports to which it can attach exist. Further, assume that each modified counter is self-assembled one row at a time according to the assembly sequence that is implicit from Figures 9.6, 9.7, and 9.9. It is clear from Sections 5.1 and 5.2 that such an assembly sequence not only exists, but is in fact unique, whence dom res$(\vec{\alpha}) = \mathbf{T}$. A *finite* snapshot of the *infinite* assembly sequence $\vec{\alpha}$ is illustrated in the following figure, where T_1 is shown fully assembled in the upper left corner, and then modified counters are attached one at a time by $\vec{\alpha}$ to yield T_2.

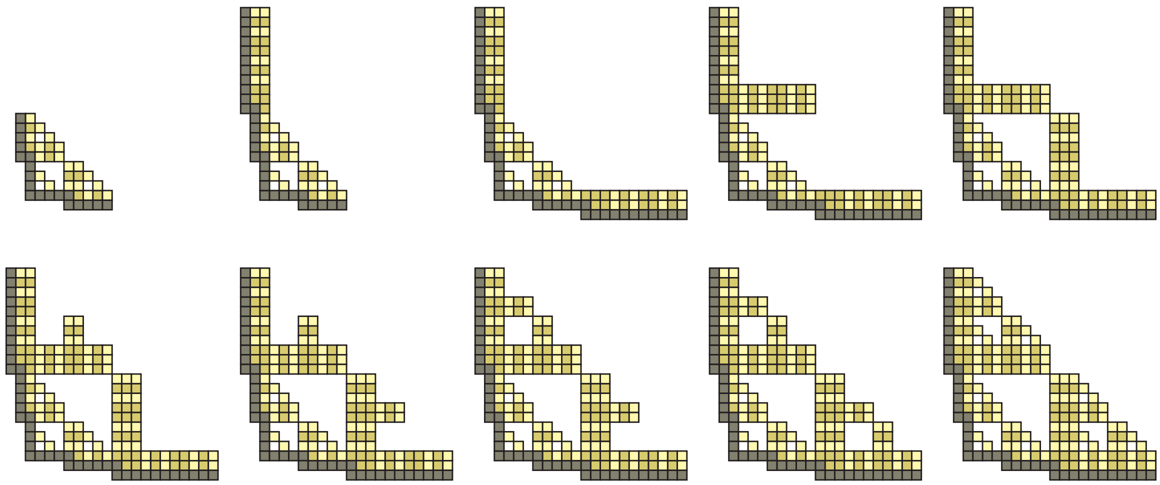
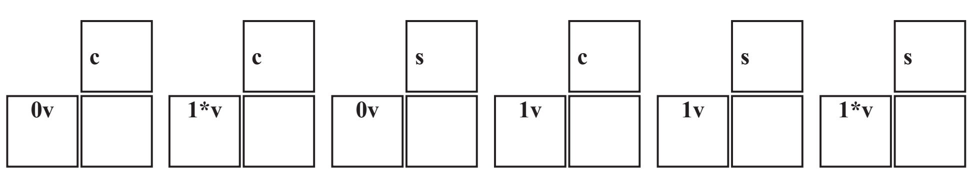


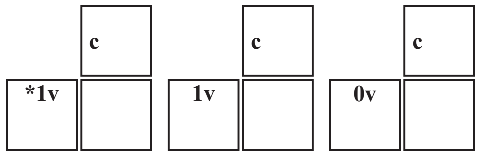
Figure 9.10 A snapshot of $\vec{\alpha}$.

We will first show that every tile that binds through $\vec{\alpha}$ does so deterministically via its input sides, and with exactly strength 2. In doing so, we will restrict our attention to vertical modified counters and note that our analysis can be easily extended to handle horizontal modified counters as well.

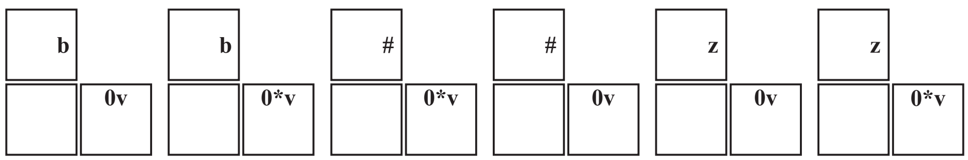
Recall that $\vec{\alpha}$ self-assembles each modified counter one row at a time. This means that the first tile type to attach in every row does so via a double strength bond. It is routine to verify that for all $t \in T$, if there exists $\vec{u} \in U_2$ where $\mathrm{str}_t(\vec{u}) = 2$, then there is a unique $t' \in T$ such that $\mathrm{col}_{t'}(-\vec{u}) = \mathrm{col}_t(\vec{u})$. Now consider the self-assembly of an increment row. In such a row, self-assembly proceeds from right to left in the natural way and, therefore it suffices to verify that following single tile extensions are unique.



In a spacing row, the first tile type attaches via a double strength bond and in general there will be two directions in which the row can self-assemble. In our definition of $\vec{\alpha}$, we assume that self-assembly first proceeds to the left and then to the right. For the former case, it suffices to verify that the following three single tile extensions are unique.



For the latter case, in which self-assembly proceeds to the right, there are six single tile extensions to verify.



Observe that all of the above single tile extensions are in fact unique.

To prove that $\vec{\alpha}$ is locally deterministic, we must show that $\text{res}(\vec{\alpha}) \in \mathcal{A}_{\Box}[\mathcal{T}_{\mathbf{T}}]$. In other words, we must argue that after $\vec{\alpha}$ has strictly self-assembled the set $\mathbf{T}$, it is impossible for any tile type to bind at any location in the set $\mathbb{Z}^2 - \mathbf{T}$. We will show that for all $\vec{m} \in \mathbb{Z}^2 - \mathbf{T}$, $\vec{m} \notin \partial^{\tau}\alpha$

Let $\vec{m} \in \mathbb{Z}^2 - \mathbf{T}$. If $\vec{m} + \vec{u} \notin \mathbf{T}$ for every $\vec{u} \in U_2$, then $\vec{m}$ is not adjacent to $\mathbf{T}$ and $\vec{m} \notin \partial^{\tau}\alpha$ because $\vec{\alpha}$ is a 2-T-assembly sequence. However, if $\vec{m}$ is adjacent to $\mathbf{T}$ then $|\{\vec{u} \in U_2 | \vec{m} + \vec{u} \in \mathbf{T}\}| \in \{1, 2, 3, 4\}$. Suppose that $|\{\vec{u} \in U_2 | \vec{m} + \vec{u} \in \mathbf{T}\}| = 1$. Then it is easy to see that $\vec{m} \notin \partial^{\tau}\alpha$ due to a lack of binding strength along the border of $\mathbf{T}$. On the other hand, if $|\{\vec{u} \in U_2 | \vec{m} + \vec{u} \in \mathbf{T}\}| \geq 2$ then $\vec{m}$ will be in a "corner" formed when two oppositely oriented modified counters meet. This implies that if there exists $t \in T$ such that $\vec{m} \in \partial^{\tau}_t\alpha$, then it must be the case that t has a vertical indicator symbol on its bottom edge and a horizontal indicator symbol on its left edge. However, this situation is impossible by the way we constructed $\mathcal{T}_{\mathbf{T}}$. In either case, $\vec{m} \notin \partial^{\tau}\alpha$, whence $\text{res}(\vec{\alpha}) \in \mathcal{A}_{\Box}[\mathcal{T}_{\mathbf{T}}]$. $\square$

10 CONCLUSION

This thesis investigated the strict self-assembly of discrete Sierpinski triangles.

We first developed a version of the standard Tile Assembly Model which (1) differentiates between weak and strict self-assembly and (2) does not discriminate against infinite assemblies.

We showed that the standard discrete Sierpinski triangle does not strictly self-assemble in the Tile Assembly Model using a lower bound for the number of tile types required to self-assemble tree shapes [2]. We then went on to show that a "fibered" version of the standard discrete Sierpinski triangle, having the same fractal dimension as its non-fibered counterpart, does in fact strictly self-assemble in the Tile Assembly Model. We proved that the fibered Sierpinski triangle strictly self-assembles by using a modified version of the optimal binary counter presented in [6], and then showed that our tile assembly system was definitive using local determinism from [15].

A goal for future research will be to answer the following question, which at the time of this writing, remains open:

Question 10.1. Does there exist a discrete self-similar fractal $F \subseteq \mathbb{Z}^2$ that strictly self-assembles in the Tile Assembly Model?

BIBLIOGRAPHY

[1] L. Adleman. Towards a mathematical theory of self-assembly. Technical report, 1999.

[2] Leonard M. Adleman, Qi Cheng, Ashish Goel, Ming-Deh A. Huang, David Kempe, Pablo Moisset de Espans, and Paul W. K. Rothemund. Combinatorial optimization problems in self-assembly. pages 23–32, 2002.

[3] Gagan Aggarwal, Michael H. Goldwasser, Ming-Yang Kau, and Robert T. Schweller. Complexities for generalized models of self-assembly. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, 2004.

[4] T. M. Apostol. *Modular Functions and Dirichlet Series in Number Theory*, volume 41 of *Graduate Texts in Mathematics*. Springer-Verlag, 1997.

[5] Robert Berger. *Undecidability of the Domino problem.* Memoirs of the American Mathematicl Society, 1965.

[6] Qi Cheng, Ashish Goel, and Pablo Moisset de Espans. Optimal self-assembly of counters at temperature two. In *Proceedings of the first conference on Foundations of nanoscience: self-assembled architectures and devices*, 2004.

[7] D. Doty, X. Gu, J.H. Lutz, E. Mayordomo, and P. Moser. Zeta dimension. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, pages 283–294. Springer-Verlag, 2005.

[8] L. Euler. Variae observationes circa series infinitas. *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, 9:160–188, 1737.

[9] G. Hardy and E. Wright. *An Introduction to the Theory of Numbers.* Clarendon Press, 5th edition, 1979.

[10] James I. Lathrop, Jack H. Lutz, and Scott M. Summers. Strict self-assembly of discrete sierpinski triangles. In *Proceedings of The Third Conference on Computability in Europe (Siena, Italy, June 18-23, 2007). To appear*, 2007.

[11] John H. Reif. Molecular assembly and computation: From theory to experimental demonstrations. In *International Colloquium on Automata, Languages and Programming*, pages 1–21, 2002.

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

[13] Paul W. K. Rothemund and Erik Winfree. The program-size complexity of self-assembled squares (extended abstract). In *STOC*, pages 459–468, 2000.

[14] Paul W.K. Rothemund, Nick Papadakis, and Erik Winfree. Algorithmic self-assembly of dna sierpinski triangles. *PLoS Biology*, 2(12), 2004.

[15] David Soloveichik and Erik Winfree. Complexity of self-assembled shapes. In *Proceedings of DNA 10*, pages 344–354. Springer, 2004.

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

[17] Hao Wang. Dominoes and the AEA case of the decision problem. In *Proceedings of the Symposium on Mathematical Theory of Automata (New York, 1962)*, pages 23–55. Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., 1963.

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