

Strict Self-Assembly of Fractals using Multiple Hands

Cameron T. Chalk* Dominic A. Fernandez* Alejandro Huerta*
Mario A. Maldonado* Robert T. Schweller* Leslie Sweet*

Abstract

In this paper we consider the problem of the strict self-assembly of infinite fractals within tile self-assembly. In particular, we provide tile assembly algorithms for the assembly of the discrete Sierpinski triangle and the discrete Sierpinski carpet within a class of models we term the *h-handed assembly model* (*h*-HAM), which generalizes the 2-HAM to allow up to *h* assemblies to combine in a single assembly step. Despite substantial consideration, no purely growth self-assembly model has yet been shown to strictly assemble an infinite fractal without significant modification to the fractal shape. In this paper we not only achieve this, but in the case of the Sierpinski carpet are able to achieve it within the 2-HAM, one of the most well studied tile assembly models in the literature. Our specific results are as follows. We provide a 6-HAM construction for the Sierpinski triangle that works at scale factor 1, 30 tile types, and assembles the fractal in a *near perfect* way in which all intermediate assemblies are finite-sized iterations of the recursive fractal. We further assemble the Sierpinski triangle within the 3-HAM at scale factor 3 and 990 tile types. For the Sierpinski carpet, we present a 2-HAM result that works at scale factor 3 and uses 1,216 tile types. We further include analysis showing that the aTAM is incapable of strictly assembling the non-tree Sierpinski triangle considered in this paper, and that multiple hands are needed for the near-perfect assembly of the Sierpinski triangle and the Sierpinski carpet.

¹Department of Computer Science, University of Texas - Pan American, {ctchalk,dafernandez1,ahuerta5,mamaldonado6,rtschweller,lgsweet}@utpa.edu This author's research was supported in part by National Science Foundation Grant CCF-1117672.

1 Introduction

1.1 Self-Assembly.

Self-assembly is the process by which systems of simple objects autonomously organize themselves through local interactions into larger, more complex objects. Self-assembly processes are abundant in nature and serve as the basis for biological growth and replication. Understanding how to design molecular self-assembly systems that build complex, algorithmically specified shapes and patterns promises to be of fundamental importance for the future of nanotechnology. Some of the most interesting algorithmically specified shapes are discrete fractals such as the Sierpinski triangle. In this paper, we take on the classical problem of designing self-assembling systems for the strict assembly of discrete fractal patterns. We consider this problem in the context of *multiple handed* tile self-assembly, a natural generalization of two-handed tile self-assembly.

1.2 Tile Assembly Model: Seeds, Hands, and more Hands.

In the *tile assembly model* system components are 4-sided Wang tiles [25] which randomly agitate within the plane and stick together when bumping tile edges share glues with strong enough affinity. Tile attachment is random and based on tiles appearing in random places in the plane. This basic framework has been studied extensively and has also been developed into multiple different variants. In this paper we consider three such variants: the *abstract tile assembly model* (aTAM) [26], the two-handed tile assembly model (2-HAM) [1, 2, 4, 6–8, 10–13, 24], and a new generalization of the the 2-HAM termed the *h*-HAM. The aTAM is the original and most widely studied model and is characterized by a *seed* tile from which growth occurs by the attachment of singleton tiles given sufficient bonding strength. In contrast, the 2-HAM has no seed and allows any two already built and potentially large assemblies to translate together and stick given enough bonding strength. In the *h*-HAM presented in this paper, we generalize the 2-HAM to allow up to a given number *h* pre-built assemblies to come together to form a new producible assembly. Although the *h*-HAM is seemingly implausible in experimentation settings, it is interesting to explore the properties of such a natural, more parallelized generalization of the 2-HAM. Further, the *h*-HAM may prove useful for designing 2-HAM systems which are robust to multi-handed attachments as a sort of error-prevention mechanism for hierarchical self-assembly.

1.3 Strict Self-Assembly of Discrete Fractals.

We focus our consideration of the power of multiple handed self-assembly on a considerable problem in algorithmic self-assembly theory: the strict self-assembly of discrete fractals. The self-assembly of discrete fractals and infinite patterns has been a benchmark problem in self-assembly that has yielded both theoretical [5, 9, 15–18, 20–22, 27] and experimental work [23]. The interest in self-assembled fractals stems from their geometric aperiodicity, which forces the development of new fundamental self-assembly design techniques, along with their frequent occurrence in natural biological systems. However, much of the previous work on self-assembled fractals centers on the *weak* assembly (described in [9, 21, 22]) of fractal patterns in which the assembled shape is permitted to place tiles of a special color in locations that are not part of the fractal shape, or permits some level of substantial *warping* of the fractal shape in which tiles are placed outside of the fractal shape in some limited fashion [16–18, 21], or adds substantial power to the assembly model such as the ability to detach previously built subassemblies [20]. A challenging problem is the *strict* assembly of fractals in a purely growth (non-detaching) model in which the shape of the assembly is exactly the shape of the fractal. The difficulty stems from the intricate, thin features that characterize infinite fractals. Pure growth models must decide how to build a shape through a single mechanism, the placement of tiles, which inherently uses up geometric space. When that space is drastically constrained, typical approaches for self-assembly fail, and in many cases the failure is unavoidable [3, 14, 21]. In fact, no pure growth model of self-assembly has yet been shown to strictly self-assemble a fractal pattern such as the Sierpinski triangle or Sierpinski carpet without substantial modification to the fractal shape that goes beyond a simple scale factor increase.

Table 1: Our results for the non-tree Sierpinski triangle and the Sierpinski carpet are summarized here.

Infinite Fractals					
Fractal	Model	Scale	Tiles	Near Perfect?	Theorem
Sierpinski Triangle	3-HAM	3	990	No	3.3
Sierpinski Triangle	6-HAM	1	30	Yes	3.2
Sierpinski Carpet	2-HAM	3	1,216	Impossible for $h < 8$	4.1, 5.6
Sierpinski Triangle	aTAM	c		Impossible	5.5

1.4 Our Results.

We focus on two discrete fractals, a non-tree version of the discrete Sierpinski triangle, and the standard Sierpinski carpet. Note that the non-tree version of the discrete Sierpinski triangle considered in this paper is a different discretization than previous studies of fractal assembly. We present two systems that strictly self-assemble the non-tree Sierpinski triangle. The first system works within the 6-handed model, uses a surprisingly small number of tile types (30), achieves optimal scale factor of 1, and achieves a desirable property we term *near-perfect* assembly in which the intermediate assemblies of the system are almost exactly the finite-sized iterations that define the infinite fractal shape. The second system works within the 3-handed assembly model and achieves scale factor 3. Finally, we present our most intricate result which is the strict self-assembly of the Sierpinski carpet at scale factor 3 within the 2-HAM model. We accompany these positive results with some impossibility results which, along with the positive results, show provable increases in power with multiple handed assembly. In particular, we show that no aTAM system self-assembles our non-tree Sierpinski triangle, and we show that the non-tree Sierpinski triangle and the Sierpinski carpet cannot be near-perfectly assembled with fewer than 3 and 8 hands respectively. Our results are summarized in Table 1.

1.5 Outline.

The remainder of this paper is organized as follows. In Section 2 we define the aTAM, the 2-HAM, and the h -HAM tile assembly models, as well as the non-tree Sierpinski triangle and Sierpinski carpet fractals. In Sections 3 and 4 we present our positive results for the strict self-assembly of the non-tree Sierpinski triangle and the Sierpinski carpet respectively. In Section 5, we present our impossibility results. Finally, in Section 6, we present some directions for future work that arise from the study of fractal self-assembly and the h -HAM.

2 Definitions and Model

2.1 Tiles and Assemblies

Tiles. Consider some alphabet of glue types Π . A tile is a finite edge polygon with some finite subset of border points each assigned some glue type from Π . Further, each glue type $g \in \Pi$ has some non-negative integer strength $str(g)$. Finally, each tile may be assigned a finite length string *label*, e.g., “black”, “white”, “0”, or “1”. In this paper we consider a special class of tiles that are unit squares of the same orientation with at most one glue type per edge, with each glue being placed exactly in the center of the tile’s edge. Further, for simplicity, we assume each tile center is located at an integer coordinate in \mathbb{Z}^2 .

Assemblies. An assembly is a set of tiles at unique coordinates in \mathbb{Z}^2 . For a given assembly Υ , define the *bond graph* G_Υ to be the weighted graph in which each element of Υ is a vertex, and each edge weight

between tiles is the sum of the strengths of the overlapping, matching glue points of the two tiles. An assembly C is said to be τ -stable for positive integer τ if the bond graph G_C has min-cut at least τ , and τ -unstable otherwise. Note that if the set of border points of all tiles in an assembly is not a connected set, then the assembly cannot be τ -stable (for positive τ). Note that only overlapping glues that are the same type contribute a non-zero weight, whereas overlapping, non-equal glues always contribute zero weight to the bond graph. Relaxing this restriction has been considered as well [2]. Additionally, with the square tiles considered in this paper, stable assemblies will necessarily consist of tiles stacked face to face, forming a subset of the 2D grid denoted by the tiled locations in the assembly. We refer to any subset of an assembly A as a subassembly of A . For an assembly A and integer vector $\vec{v} = (v_1, v_2)$, let $A_{\vec{v}}$ denote the assembly obtained by translating each tile in A by vector \vec{v} . A *shape* is any subset of \mathbb{Z}^2 , and the shape of an assembly A is defined to be the shape consisting of the set of coordinate locations for the centers of each tile in A .

Combinable Assemblies. Informally, a set of h assemblies is said to be τ -combinable if the assemblies can be translated together to form a τ -stable assembly C . Formally, a set of stable assemblies $\{A_1, \dots, A_h\}$ is said to be τ -combinable into an assembly C if there exists a set of assemblies $\{A'_1, \dots, A'_h\}$ such that each A'_i is a translation of A_i , and $C = \bigcup_{i=1}^h A'_i$ is a τ -stable assembly.

2.2 Abstract Tile Assembly Model (aTAM)

A *tile system* in the Abstract Tile Assembly Model (aTAM) is an ordered triple $\Gamma = (T, \sigma, \tau)$ where T is a set of tiles called the *tile set*, σ is an assembly over T called the *seed*, and τ is a positive integer called the *temperature*. As notation we denote the set of all tiles that are a translation of some tile in T as set T^* . Assembly in the aTAM proceeds by growing from assembly σ by any sequence of single tile attachments from T so long as each tile attachment connects with strength at least τ . Formally, we define what can be built in this fashion as the set of producible assemblies:

Definition 2.1 (aTAM Producibility). For a given aTAM system $\Gamma = (T, \sigma, \tau)$, the set of *producible assemblies* for system Γ , PROD_Γ , is defined recursively:

- (Base) $\sigma \in \text{PROD}_\Gamma$
- (Recursion) For any $A \in \text{PROD}_\Gamma$ and $b \in T^*$ such that $C = A \cup \{b\}$ is τ -stable, then $C \in \text{PROD}_\Gamma$.

As additional notation, we say $A \rightarrow_1^\Gamma B$ if A may grow into B through a single tile attachment, and we say $A \rightarrow^\Gamma B$ if A can grow into B through 0 or more tile attachments. For a shape S , we say a system Γ *uniquely assembles* S if for all $A \in \text{PROD}_\Gamma$ there exists a $B \in \text{PROD}_\Gamma$ of shape S such that $A \rightarrow^\Gamma B$. An *assembly sequence* is a way to denote a particular possible sequence of growth for a given system and is formally defined to be any sequence (finite or infinite) of assemblies $\langle A_0, A_1, A_2, \dots \rangle$ such that $A_0 = \sigma$, and $A_i \rightarrow_1^\Gamma A_{i+1}$ for each i . For a more detailed description of the aTAM and related concepts and metrics, see [4].

2.3 Multiple Handed Tile Assembly Model (h -HAM)

The multiple handed tile assembly model generalizes the two-handed assembly model (2-HAM) to allow for potentially more than two hands. Formally, a multiple handed tile assembly system is an ordered triple (T, τ, h) where T is a set of tiles called the *tile set*, τ is a positive integer called the *temperature*, and h is a positive integer called the *number of hands*. In this paper we sometimes refer to a system (T, τ, h) as an h -HAM system (T, τ) . Assembly in an h -HAM system (T, τ) proceeds by repeatedly combining up to h assemblies at a time to form new τ -stable assemblies. Formally, we define what can be built in this fashion as the set of producible assemblies:

Definition 2.2 (h -HAM Producibility). For a given h -HAM system $\Gamma = (T, \tau)$, the set of *producible assemblies* for system Γ , PROD_Γ , is defined recursively:

- (Base) $T^* \subseteq \text{PROD}_\Gamma$
- (Recursion) For any set of r assemblies, $0 \leq r \leq h$, if a producible set of assemblies $A = \{A_1, \dots, A_r\} \subseteq \text{PROD}_\Gamma$ is τ -combinable into B , then $B \in \text{PROD}_\Gamma$.

Typically, we require that an initial assembly set T consist only of stable assemblies. In this paper, we further restrict T to consist of singleton tile assemblies and thus refer to T as the *tile set* of the system, and we refer to $|T|$ as the *tile complexity* of Γ . In the case of $h = 2$, we have the standard 2-HAM tile assembly model in which assembly proceeds by repeatedly combining any pair of combinable assemblies. We encourage the reader to see [4] for a number of 2-HAM related concepts and metrics not explicitly discussed here in this compact definition. One important concept we use in this paper is the assembly of infinite shapes. In this paper, we use the *finite assembly* [4] of infinite shapes as our definition of what the assembly of infinite shapes means for an h -HAM system. A system Γ is said to finitely assemble an infinite shape $X \subset \mathbb{Z}^2$ if every finite producible assembly of Γ has a possible way of growing into an assembly that places tiles exactly on those points in X (appropriately translated if necessary). For the remainder of this paper we will use the term “self-assembles” rather than “finitely self-assembles” as there is only one such concept under consideration in this paper.

2.4 Sierpinski Shapes

In this section we define the non-tree discrete Sierpinski triangle and the discrete Sierpinski carpet. The Sierpinski triangle we consider in this paper is not the shape considered in earlier work in tile self-assembly. The main distinction is that our Sierpinski triangle is not a *tree* shape. This is important since our positive results rely fundamentally on the non-tree shape of the fractal. In the case of the tree version of the Sierpinski triangle, [4] have shown that the shape cannot be strictly assembled at scale factor 1 in either the aTAM or the 2-HAM, and this proof applies more generally to the h -HAM. However, it remains unproven (but we conjecture) that the tree Sierpinski triangle cannot be strictly assembled at any scale within the h -HAM. Throughout the remainder of the paper, we refer to the non-tree Sierpinski triangle as the Sierpinski triangle.

Definition 2.3. (Discrete Sierpinski Triangle). Let $S_{i+1} = S_i \cup (S_i + 2^i V)$ where $V = \{(1,2), (-1,2)\}$, and $S_0 = \{(0,0), (-1,0), (0,1), (-1,1)\}$. The discrete Sierpinski triangle can be defined as the set $S_\infty = \bigcup_{i=0}^\infty S_i$. (Figure 1). S_∞^c and S_i^c are factor c scalings of S_∞ and S_i .

Definition 2.4. (Discrete Sierpinski Carpet). Let $C_{i+1} = C_i \cup (C_i + 3^i V)$, where $V = \{(1,0), (0,1), (0,2), (2,0), (1,2), (2,1), (2,2)\}$, and $C_0 = (0, 0)$. The discrete Sierpinski carpet will then be the set $C_\infty = \bigcup_{i=0}^\infty C_i$. (Figure 2) C_∞^c and C_i^c are factor c scalings of C_∞ and C_i .

3 Strict Self-Assembly of the Discrete Sierpinski Triangle

In this section, we present two constructions for the strict assembly of the non-tree discrete Sierpinski triangle. Note that the assembly of the standard discrete Sierpinski triangle was proven to be impossible in [4]. Our first construction is a 6-HAM system that strictly assembles the non-tree Sierpinski triangle at optimal scale 1 and 30 distinct tiles. Further, this construction achieves *near perfect assembly* (Def. 3.1). Our second construction uses only 3 hands and works at scale 3. We conjecture that 3 is the minimum number of hands required to assemble the non-tree Sierpinski triangle.

3.1 6-Handed Assembly of the Sierpinski Triangle

In this section, we provide a 6-HAM system that assembles the discrete Sierpinski triangle at scale 1, using 30 distinct tiles. The system also assembles the Sierpinski triangle in an idealized manner that we call *near perfect assembly* (Def. 3.1). When we say a system near perfectly assembles an infinite fractal, we mean that the system will only produce assemblies that differ from some finite stage (an index in sequence P) of the

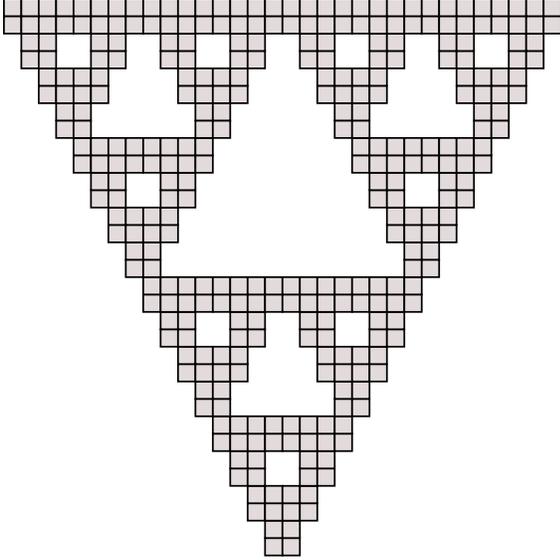


Figure 1: Scale 1 Sierpinski triangle. (S_i^1)

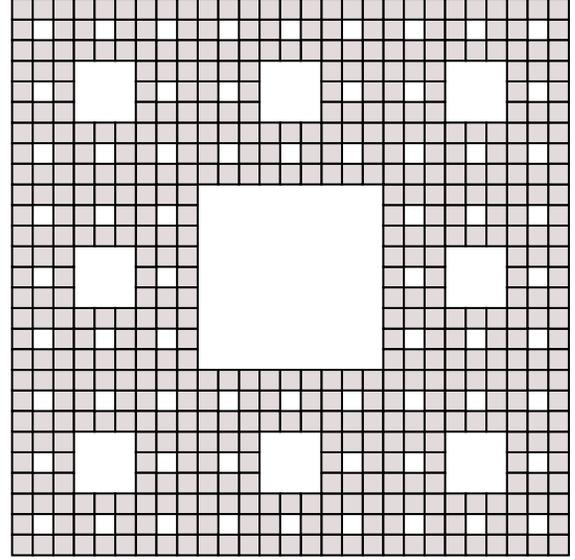


Figure 2: Scale 1 Sierpinski carpet. (C_i^1)

fractal by a constant number of points. In other words, if we take any producible assembly in the system, we will find one stage of the fractal that is almost equal to the shape of the chosen assembly with only a constant number of points missing. The inverse will also be true; if we pick any stage of the fractal, we will find one or more assemblies that are almost equal to that stage.

Definition 3.1. (Near Perfect Assembly). An infinite fractal pattern $P = \bigcup_{i=n}^{\infty} P_i$, where sets P_{i+1} are obtained by a function $f(P_i)$, is said to be *near perfectly assembled* by an h -HAM system $\Gamma = (T, \tau)$ if the system assembles P , and if:

1. There exists a c such that for all i , there exists an assembly $A \in \text{PROD}_{\Gamma}$ of shape S_A such that $|P_i \setminus S_A| \leq c$. In other words, for every P_i , there exists at least one producible assembly $A \in \text{PROD}_{\Gamma}$ whose shape is a subset of P_i smaller by at most a constant number of points independent from i .
2. For any producible assembly $A \in \text{PROD}_{\Gamma}$, the shape of some translation of A is a subset of some P_i , smaller by at most a constant number of points independent from i .

Theorem 3.2. There exists a 6-HAM system that near perfectly assembles the discrete Sierpinski triangle at scale factor 1, and has a tile complexity of 30.

Proof. The proof is by construction. We first show how condition 1 is met by presenting the tiles in the tileset, and describing how they will combine to produce assemblies with a shape that is 6 points away from an S_i , for all S_i . We then show that condition 2 is also satisfied by doing induction on the size (number of tiles) of the producible assemblies, and proving that the shape of all producible assemblies of size greater than 30 must be a subset of a stage of the Sierpinski triangle, smaller by only 6 points.

Consider 6-HAM system $\Gamma = (T, \tau)$ with tile types T shown in Figure 3, and $\tau = 2$. The tile set T (made up of 30 tiles) is designed to meet both conditions required for near perfect assembly (Def. 3.1) of the discrete Sierpinski triangle. The near perfect assembly feature of the system is achieved by taking advantage of the six hands and the temperature to prevent any other wrong assemblies from combining. Each of the large producible assemblies is made up of six pieces, which must be assembled before they combine into the next stage (S_i) of the Sierpinski triangle.

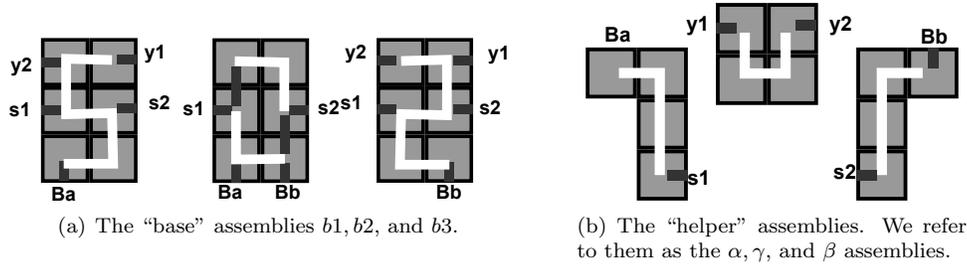


Figure 3: Tile set T used to near perfectly assemble a Sierpinski triangle. White lines show a necessary sequence in which the pieces must be constructed in order to achieve near perfect assembly.

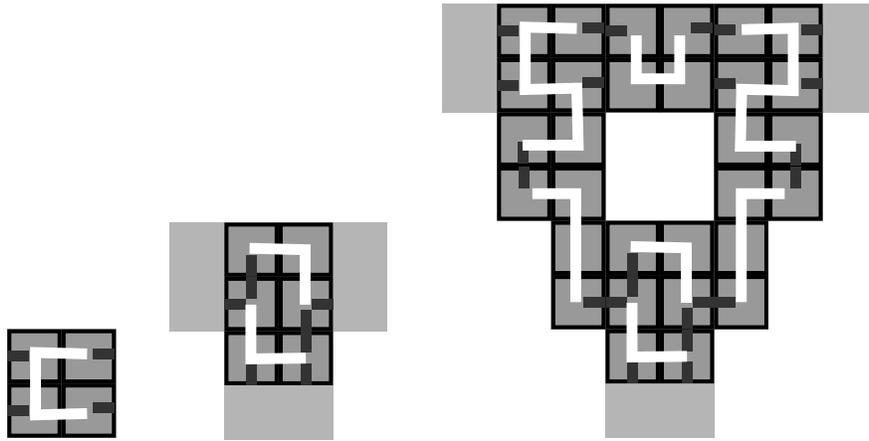


Figure 4: Assemblies used to satisfy condition 1 of near perfect assembly for $S_0, S_1,$ and S_2 .

Condition 1 To satisfy condition 1 of near perfect assembly, all stages (S_i) of the discrete Sierpinski triangle need to have at least one assembly with a shape that is almost, by a constant number of points, equal to that stage. Consider the first stage S_0 of the triangle, whose shape is defined to be a 2×2 square with its leftmost bottom point at $(-1, 0)$. An assembly of either b_1 or b_3 with the bottommost four tiles will form the squared shape exactly, with no points of difference (Figure 4). The next stage, S_1 , consists of three copies of the S_0 shape, one staying in the original's position, one translated up and to the left by 2 and 1 points respectively, and another one translated up and to the right by the same amount. The shape of $b_1, b_2,$ or b_3 is the closest to S_1 , and it is smaller than this stage by 6 tiles (it is missing the two leftmost, bottommost, and rightmost tiles) as shown in Figure 4. S_2 follows the same pattern, consisting of three copies of S_1 translated by the specified amount. The assembly with the biggest subset of S_2 will be composed of the three base assemblies, glued together by the set of $\alpha, \beta,$ and γ helper assemblies shown in Figure 3(b). These six assemblies will need to form a "loop" (Figure 4) joining them all together or the assembly will not be τ -stable.

The resulting assembly's shape is a subset of S_2 , smaller by 6 points. Condition 1 for all of the following stages will be met in the same way: three copies of the assembly used to satisfy the previous stage, all joined together by a new set of $\alpha, \beta,$ and γ assemblies. This process is shown in Figure 5 for a small part of the system, but it will be repeated for any S_i .

Note that these specific assemblies are smaller than an S_i for all S_i by 6 tiles, and therefore Γ satisfies condition 1 of near perfect assembly of the Sierpinski triangle.

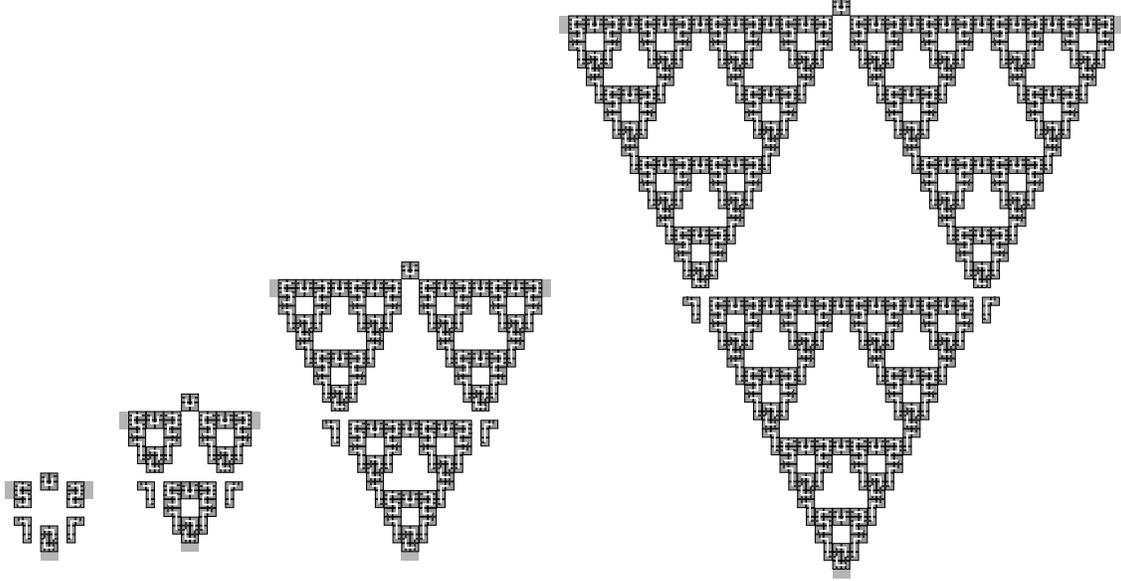


Figure 5: Stages of growth for 6-HAM system Γ . The floating pieces are separated to easily distinguish between the six different assemblies that combined to form a new iteration from the previous one.

Condition 2 To see how Γ satisfies condition 2, we first define *near-triangle assemblies*. Near-triangle assemblies are a class of assemblies whose shape is a subset of some stage of the Sierpinski triangle (S_i) with the same 6 corner points missing (the two leftmost, rightmost and bottommost points). Near-triangle assemblies must also have the same set of exposed glues at the same relative points. One example of a near-triangle assembly is the third assembly in Figure 4 that is close to stage S_2 . The shape of this assembly is missing the six corner points. Further, tiles at points adjacent to these missing points expose glues that allow only helper tiles to attach. We use notation A_i to denote the near-triangle assembly with shape close to S_i . Note that all near-triangle assemblies with shape close to S_i are of size $4 \times 3^i - 6$.

To show system Γ meets condition 2 of near perfect assembly we prove by induction, on the size of the assemblies, that all producible assemblies of size greater than 30 must be near-triangle assemblies. Note that assemblies smaller than 30 are base or helper assemblies, which can only combine to form near-triangle assembly A_2 . Consider the near-triangle assembly shown in Figure 4 (third assembly), of size $s = 30$ as the base case. Since the base case (A_2) is a near-triangle assembly we assume all producible assemblies of size $s > 30$, up to assembly A_n of size $4 \times 3^n - 6$, are near-triangle assemblies of some S_i for $i \leq n$. Assembly A_n is a near-triangle assembly, which means the six corner points in S_n are missing from A_n , and it also exposes the same glues as all other near-triangle assemblies.

Only helper tiles can attach to the exposed glues on near-triangle assemblies, and helper tiles attach only to near-triangle assemblies or base pieces, so the assembly of size $4 \times 3^n - 5$ (or any bigger producible assembly) has to follow the same process as the one shown in Figure 5, where 3 equal-sized near-triangle assemblies must be combined with the 3 helper assemblies, or it is not producible. The process must be followed because if the three near-triangle assemblies are not the same size, the distance between at least two assemblies is not filled and the loop described before is not closed, (as shown in Figure 6) resulting in an assembly that is not τ -stable. Any assembly of size $4 \times 3^n - 5$ is not producible since the same type of attachments must happen. In fact, we can be sure A_n can only grow into A_{n+1} which is a near-triangle assembly of size $4 \times 3^{n+1} - 6$. A_n cannot grow into an assembly that is not A_{n+1} , so Γ meets condition 2 of near perfect assembly.

To Summarize: The shape of any producible assembly smaller than size 30 is smaller than an S_i by at most 9 points (partial assembly of α and β) and the shape of anything larger than 30 is smaller than an S_i

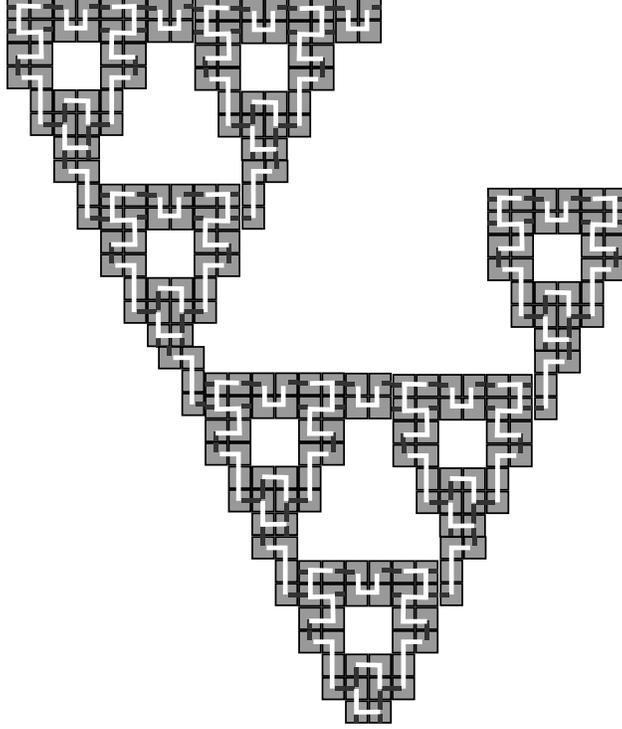


Figure 6: Near-triangle assemblies of different size will never be stable.

by 6 points. These are the only producible assemblies, so both conditions are met, and since we have shown and explained how the system assembles the Sierpinski triangle, we conclude that Γ near perfectly assembles the discrete Sierpinski triangle. \square

3.2 3-Handed Assembly of the Sierpinski Triangle

In this section, we provide a 3-HAM system that strictly self-assembles the discrete Sierpinski triangle at scale factor 3. Note that the system does not near perfectly assemble the Sierpinski triangle, since it relies on many “filler” assemblies that increase in number as other assemblies grow.

Theorem 3.3. There exists a 3-HAM system $\Gamma = (T, \tau)$ that self-assembles S_∞^3 with $|T| = 990$.

Proof.

Overview. An intuitive overview of our Sierpinski triangle construction follows. A 3-HAM system $\Gamma = (T, 2)$ with a tileset of size 990 is used to self-assemble the discrete Sierpinski triangle. 9 triangles that are nearly of size S_1^3 are assembled. These assemblies differ from the shape of S_1^3 in that they are missing tiles along two edges to allow for tiles to be placed along edges in later stages of the assembly process. An example is shown in Figure 9. These 9 triangles will attach in triplets to create 3 unique triangles that are nearly of size S_2^3 . These near- S_2^3 assemblies are designed to leave a perimeter such that their edges can be tiled. This process is shown in Figure 7.

Once a near- S_2^3 assembly is complete, it allows for the attachment of an assembly referred to as a *keystone*. A keystone initiates a growth of tiles along an edge of a near- S_i^3 assembly. Once the edge is tiled, the assembly will attach to 2 other near- S_i^3 assemblies, forming a near- S_{i+1}^3 assembly. The edges of the near- S_{i+1}^3 assembly are untiled, and one of the edges exposes glues that allow another attachment of

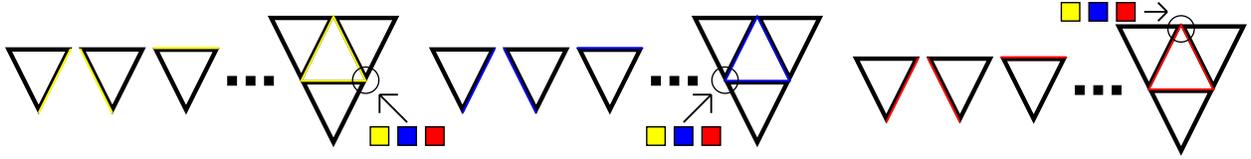


Figure 7: 3 sets of 3 unique S_1^3 assemblies attach to form 3 unique near- S_2^3 assemblies. Keystone attachment happens at the circled region.

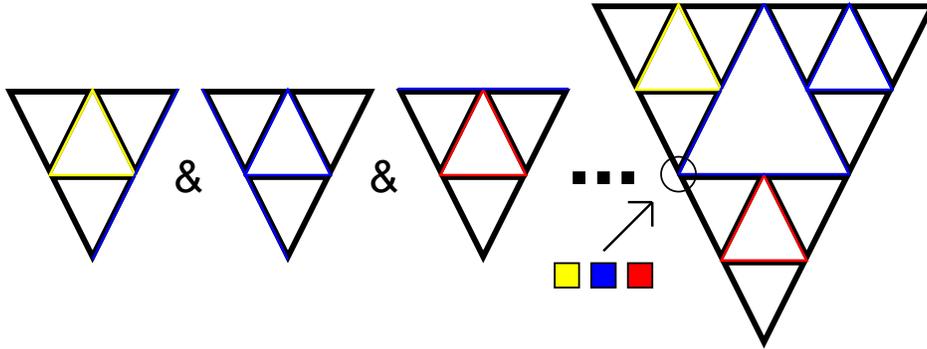


Figure 8: 3 near- S_2^3 assemblies, having attached the same blue colored keystone, join to form a near- S_3^3 assembly. Note that the region that a keystone attachment occurs is dependent on the color of the inner perimeter. The nondeterministic attachment of the keystone determines if the triangle assembled becomes the right, left, or bottom triangle in the next iteration of the assembly.

a keystone. One of 3 different keystones will nondeterministically attach. Once the keystone attaches, the keystone binding location is unavailable, therefore only one keystone type can attach to a near- S_i^3 assembly. The nondeterministic attachment of a keystone to a near- S_i^3 assembly determines on which edge tiles will be placed on a near- S_{i+1}^3 assembly composed of 3 near- S_i^3 assemblies that have attached that keystone. The 3 near- S_i^3 assemblies will only attach to form a near- S_{i+1}^3 assembly if they have attached the same keystone. This process is shown in Figure 8. The tiles that fill in edges of a near- S_i^3 assembly are of a constant number and are designed to fill in arbitrarily long edges, allowing for a constant-sized tileset.

Base Shapes. A *base shape* of S_i^3 is an assembly with shape S_i^3 that is missing tiles along the outer edges and the corners of the triangle as shown in Figure 10(a). A *combinable shape* of S_i^3 is an assembly with shape S_i^3 that is missing tiles along two outer edges of the triangle and has corner tiles where these edges meet as shown in Figure 10(b).

Formation of Base Pieces. The assembly process in the system begins by assembling the assemblies shown in Figure 9, which are combinable shapes of S_1^3 . There are three sets in total of these combinable shapes of S_1^3 , two of which are not pictured here. The white path traces the assembly process of these assemblies, where tiles connected by the path are connected by unique strength-2 glues. This pattern ensures that each combinable shape of S_1^3 is completely assembled before either can assemble into the base shape of S_2^3 . Once these combinable shapes of S_1^3 have assembled completely, they can attach to their two corresponding combinable shapes of S_1^3 . Each combinable shape of S_1^3 has two strength-1 glues that are used to construct a base shape of S_2^3 using cooperative binding. Each strength-1 glue attaches to one triangle each, forming a τ -stable assembly, as shown in Figure 11(a). The other two sets of combinable shapes of S_1^3 not shown here exhibit the same behavior. These three different base shapes of S_2^3 will be described as

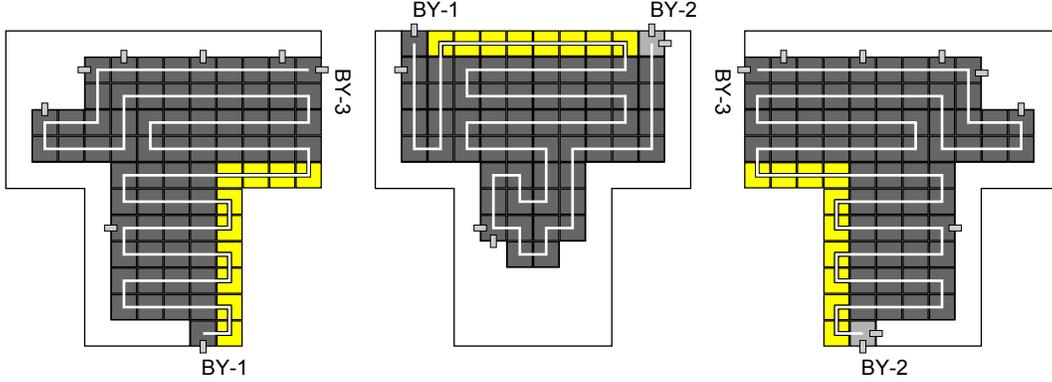


Figure 9: Yellow combinable shapes of S_1^3 . Once combined, they will become the left triangle in the next iteration. There are corresponding combinable shapes of size S_1^3 for the right and bottom triangles.

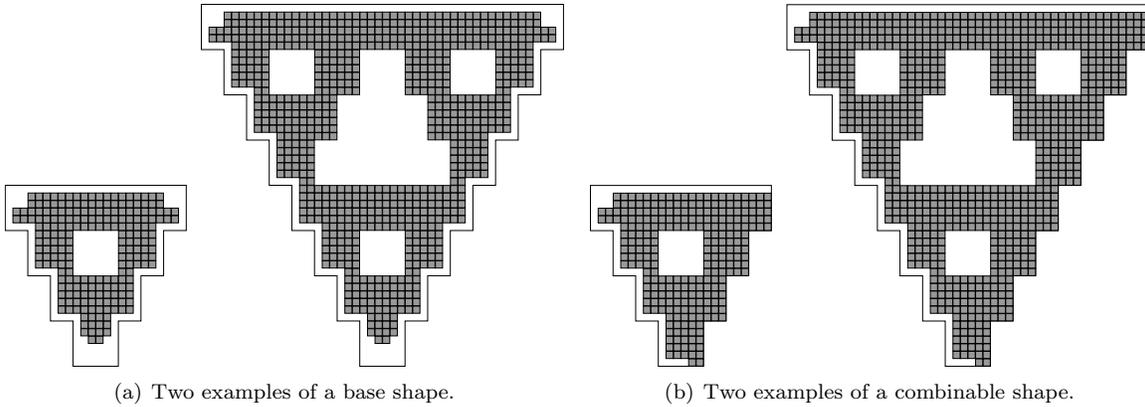


Figure 10: Examples of the two types of shapes defined.

either yellow, blue, or red. We show the yellow combinable shapes of S_1^3 that make up base shape of S_2^3 in Figure 9.

Keystone Attachments. *Keystone tiles* are assemblies that cooperatively attach to their corresponding exposed glues, in a nondeterministic manner. For each base shape of S_i^3 , one of three keystones will attach, followed by the filling of an edge of the assembly. This method of assembly is used to produce three different combinable shapes of S_i^3 . The assembled base shapes of S_2^3 , described in the previous paragraph, have two exposed glues on a single edge which are sites of keystone tile attachment. Each assembled base shape of S_2^3 has three keystone tiles that can attach at their exposed edges. We will describe these keystone tiles by the colors yellow, blue, and red. These three keystone tiles are each split into two sets, so that they must be assembled using 3 hands (two for the keystone tiles and one for the base shape of S_2^3) to be properly assembled, as shown in Figure 11(b). Each of the three different assembled base shapes of S_2^3 expose keystone glues on differing edges; a yellow base shape will attach their keystone tiles on the right edge, a blue base shape on the left edge, and a red base shape on the top edge.

Connector Glues. When the keystone tiles have bonded to a base shape of S_2^3 , they will place tiles along the edge of the assembly, matching the color of the keystone tiles that have attached. This process is shown

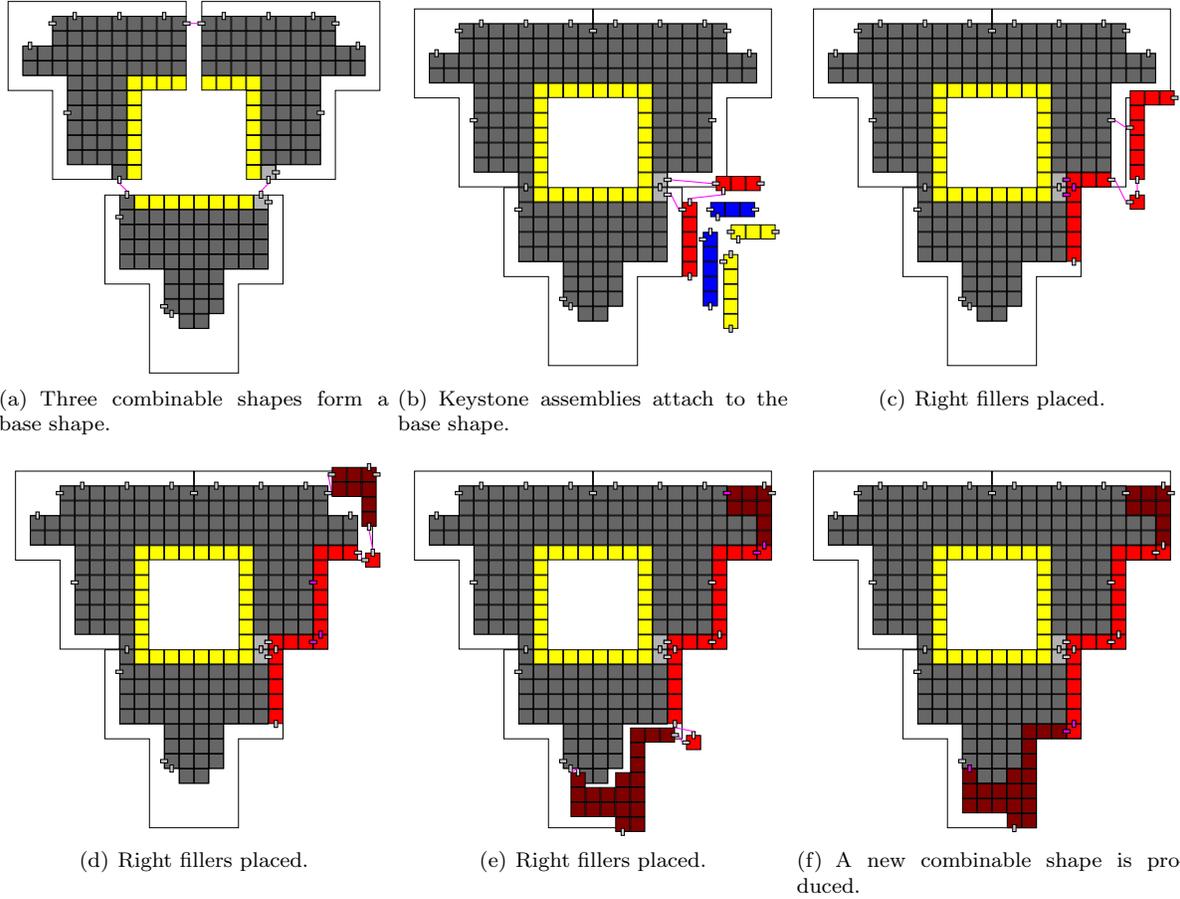


Figure 11: The process of keystone attachment and edge growth.

in Figure 11(b)-11(f). *Whitespace* is the area of the assembly that is part of the shape S_i^3 but is not covered by tiles. Each attachment of these tiles that fill the whitespace along the edge takes three hands to assemble, one for the base shape of S_2^3 (along with the already attached keystone and edge filling tiles) and two for the tiles that fill in the whitespace, which consists of a corner and the fitting assembly. The now combinable shape of S_2^3 will have exposed glues that will provide a method of attaching to the corresponding combinable shapes of S_2^3 . This process is shown for a yellow base shape of S_2^3 in Figure 11; in the case of the other base shapes of S_2^3 , the blue base shape has a process that is a mirror of the shown yellow attachment process. The process for a red base shape consists of having the keystone attach atop the assembly to grow filling tiles along the top edge. The produced combinable shapes of S_2^3 have exposed strength-1 glues that will allow them to attach to two other combinable shapes of S_2^3 , forming a base shape of S_3^3 . Three combinable shapes of S_2^3 will only attach to one another if they have attached the same type of keystone tile, as shown in Figure 12.

Infinite Assembly. A base shape of S_i^3 will form a combinable shape of S_i^3 , and three combinable shapes of S_i^3 will form a base shape of S_{i+1}^3 in order to self-assemble the infinite Sierpinski triangle. If a base shape of S_i^3 was designated yellow (using keystone assemblies in stage S_{i-1}^3), it will place tiles along its right edge; if blue, it will place tiles along its left edge; if red, it will place tiles along its top edge. The filling tiles are designed to fill in arbitrarily long edges using a constant tileset. The three resulting red, blue, and yellow

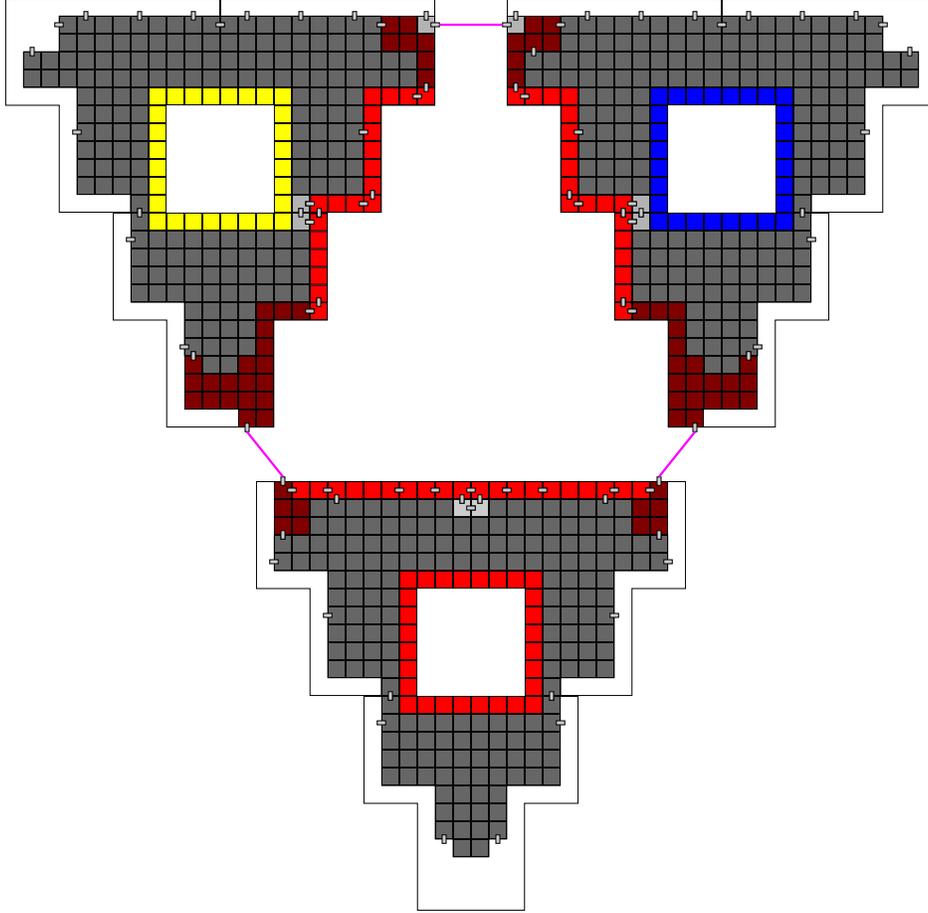


Figure 12: Different combinable shapes assemble into the next iteration of the Sierpinski triangle.

combinable shapes of S_i^3 will attach to form a base shape of S_{i+1}^3 if they have chosen the same keystone. The nondeterministic attachment of keystones allows for three combinable shapes of S_i^3 to produce three different base shapes of S_{i+1}^3 , which will become complementary combinable shapes of S_{i+1}^3 . These three combinable shapes of S_{i+1}^3 can produce base shapes of S_{i+2}^3 . The nondeterministic keystone attachment and arbitrary-length filling tiles allow us to self-assemble the discrete Sierpinski triangle using a 3-handed system at scale factor 3 with 990 tile types; 642 tiles for the base pieces and 348 tiles for the keystone assemblies and filler tiles. \square

4 Strict Self-Assembly of the Discrete Sierpinski Carpet

Weak self-assembly of the discrete Sierpinski carpet has been achieved previously [15, 16]. However, no purely-growth model of tile assembly has been shown to strictly self-assemble the fractal at any scale factor. In this section, we show a construction that strictly self-assembles the discrete Sierpinski carpet in the well-known 2-HAM.

Theorem 4.1. There exists a 2-HAM system $\Gamma = (T, \tau)$ that self-assembles C_∞^3 with $|T| = 1,216$.

Proof.

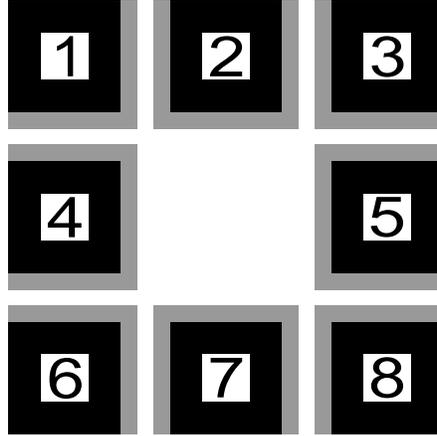


Figure 13: This figure shows the placement of tiles along the edges of the base pieces.

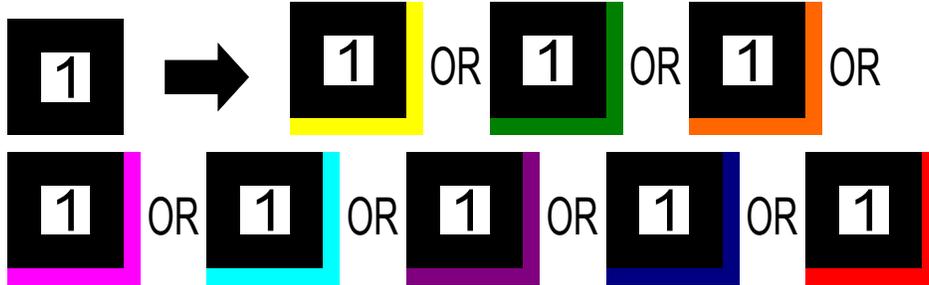


Figure 14: A base piece places tiles along specific edges, but will choose nondeterministically from 8 different tilesets to do so.

Overview. We show this by constructing a 2-HAM system $\Gamma = (T, 2)$. The tileset T of the system contains 1,216 tiles, and self-assembles the discrete Sierpinski carpet at a scale factor of 3. Intuitively, the construction works as follows. First, 8 base pieces are trivially assembled. Each base piece is nearly of shape C_1^3 except that each piece has a perimeter of width 1 missing from its outer edges so that tiles can be placed along the edges of the base pieces while staying consistent with the Sierpinski carpet shape. As seen in Figure 13, the base pieces will place tiles along specific edges that expose glues allowing the base pieces to attach to one another. However, these tiles will be placed such that when the 8 base pieces come together, they form an assembly that is nearly of shape C_2^3 except that it is missing a perimeter of width 1 so that tiles can again be placed along its edges, allowing it to attach to other C_2^3 sized assemblies. These C_2^3 sized assemblies will attach to 7 other C_2^3 sized assemblies in a similar manner as the base pieces, to form an assembly that is nearly of shape C_3^3 except that, again, the shape will be missing a perimeter of width 1. This process will repeat indefinitely, allowing the system to self-assemble the infinite shape of the discrete Sierpinski carpet.

Our construction fundamentally relies on the nondeterminism inherent to tile assembly in order to achieve our result. Each base piece nondeterministically attaches to one of 8 different assemblies, called *keystones*, that set off a chain of tile attachments that place glues in the appropriate positions. As shown in Figure 14, the base piece labeled 1 can grow into any one of the 8 different assemblies shown. The other base pieces act similarly. The purpose of nondeterminism is not yet seen in this step; while the 8 different assemblies are geometrically similar at this stage, they pass along information that determines the behavior of the assembly that consists of 8 attached base pieces. In other words, they determine the behavior of the next iteration of the Sierpinski carpet.

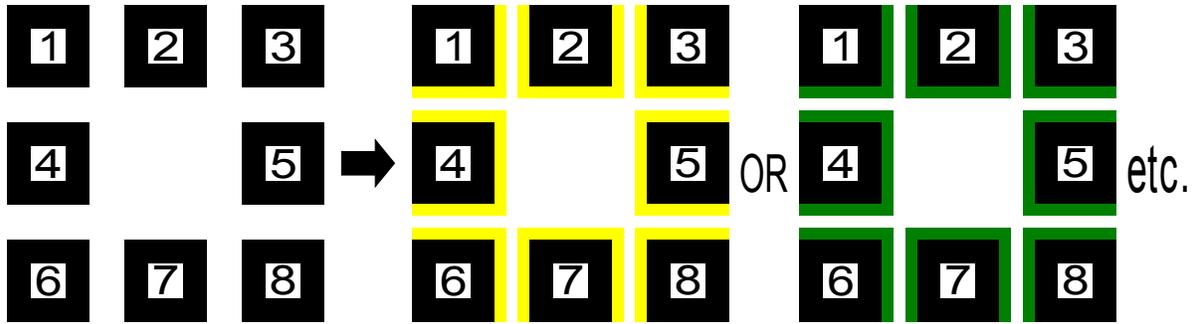


Figure 15: 8 base pieces will attach to one another if they all nondeterministically attach the same keystone.

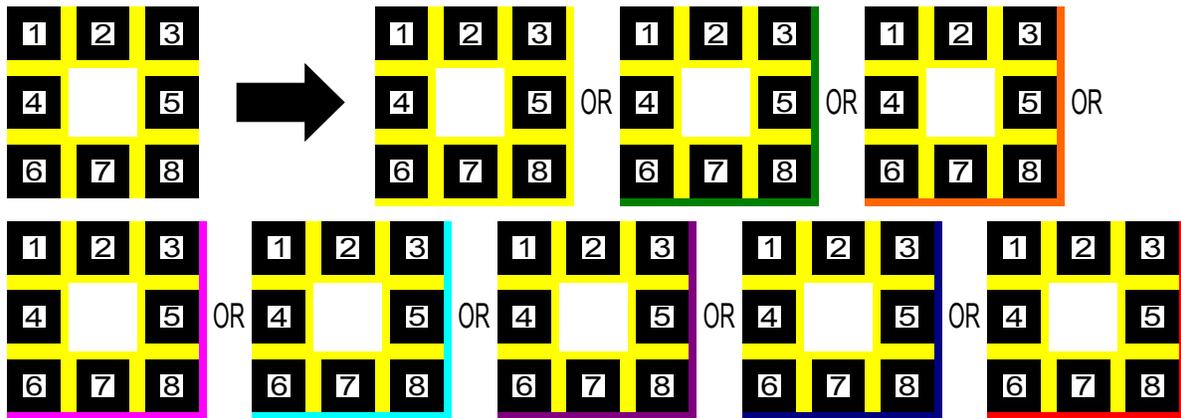


Figure 16: A set of 8 assemblies that have chosen the same keystone will attach and nondeterministically choose a keystone for the next iteration of the carpet.

As shown in Figure 15, 8 base pieces that have each nondeterministically attached the same *keystones* will be able to attach to one another. If two base pieces attach different keystones, they will not attach to one another. Once 8 base pieces that have nondeterministically chosen the same keystones attach to each other, they will place glues along their edges according to the keystone they chose.

The keystone determines along which edges tiles are placed and which edges are left open, similar to Figure 13. Figure 16 shows an assembly of 8 base pieces placing glues along their south and east edges as determined by their keystone. Note that the assembly can nondeterministically grow into one of 8 assemblies, again by the attachment of a keystone. The keystone assemblies attached at this step are the same as the ones in the base piece step, which allows our system a constant tile complexity. Figure 17 shows a different assembly of 8 base pieces, having attached different keystones than the base pieces in Figure 16, thus placing tiles along its west, east, and south edges.

8 sets of 8 base pieces are shown in Figure 18. In each set of 8 base pieces, each base piece has attached the same keystones at the first stage, but each set of 8 base pieces attached using different keystones than the other sets. These 8 sets will place tiles on the appropriate edges so that when the 8 sets attach, they will form the next iteration of the Sierpinski carpet with a missing perimeter of width 1. However, each set must choose the same keystone in order to attach to the other sets. 8 sets of 8 base pieces can grow nondeterministically into 8 different assemblies, again by the process of keystone attachment. The tile system is designed to allow the process of nondeterministic keystone attachment at any iteration of the discrete Sierpinski carpet, allowing for the finite assembly of the infinite shape.

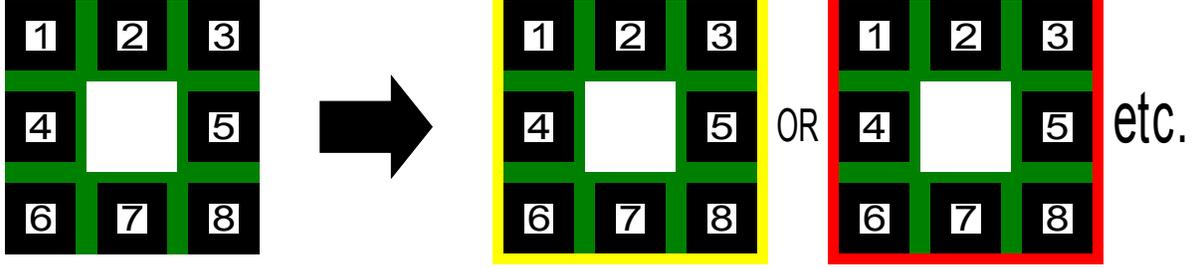


Figure 17: A group of 8 assemblies that have nondeterministically chosen to be the top-middle square in the Sierpinski carpet, placing their glues along the appropriate edges.

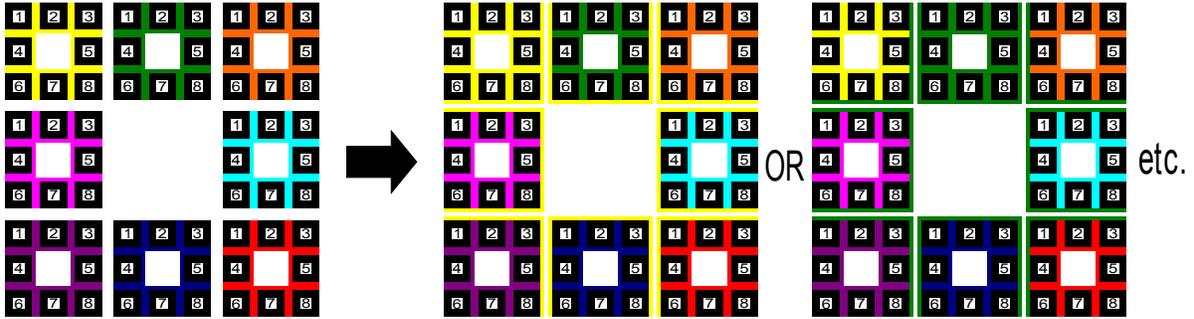


Figure 18: 8 sets of 8 base pieces nondeterministically choose which role they will play in the next iteration of the process.

Base Shapes. An assembly is considered to be a *base shape* of C_i^3 if it consists of only the points in C_i^3 minus an outer perimeter of width 1 with indented corners as shown in Figure 19. The outer perimeter of a base shape of C_i^3 must expose two *keystone glues* and otherwise expose *base glues*; both of these glue types will be described in following sections. Note that we require a scale factor of 3 in order to assemble these base shapes. We will construct base shapes of size C_1^3 , and then show that any base shape of C_i^3 can grow into a base shape of C_{i+1}^3 . Also, there will be 8 distinct assemblies of each base shape of C_i^3 , one for each *position*. A *position* is an integer 1 through 8 describing how one square-shaped assembly attaches to another, as shown in Figure 13. An assembly at position 1 will attach assemblies on its east edge and south edge, an assembly at position 2 on its west and east edge, etc.

Formation of Base Pieces. 8 distinct base pieces of base shape C_1^3 are formed using 36 tiles each. Each base piece consists of two assemblies called *base parts*. The base parts attach to one another using cooperative binding with glues at separate ends of the base parts, as shown in Figure 20. This ensures that each base part is completely assembled before it can attach to its corresponding base part to form a complete base piece. Along the edges of the base pieces are glues labeled with an N, W, S, or E followed by a B, corresponding to the cardinal directions. These glues will be referred to as *base glues*, and their purpose will be explained in the *Connector Glues* section.

Keystone Attachments. Each of the 8 base pieces has its own set of 8 keystones, for a total of 64, shown in Figure 21. A *keystone* is a two-tile assembly designed to attach to a base piece once the base piece is complete. Any one of the 8 keystones corresponding to a particular base piece is able to attach to the base assembly in a nondeterministic fashion.

Note that a base piece must complete before a keystone can attach, since each keystone binds coopera-

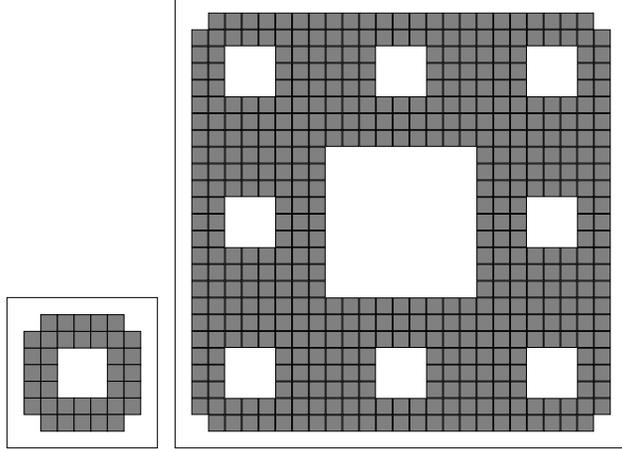


Figure 19: The points of a base shape of C_1^3 and base shape of C_2^3 , respectively. An outline is shown to indicate the actual shape of C_1^3 and C_2^3 .

tively to the base piece. On the keystone assemblies and the base pieces, there are glues labeled K followed by a number corresponding to that base piece's position, ensuring that the keystones attach to the appropriate base pieces. We will refer to these glues as the *keystone glues*. The keystone glues exposed on the base piece are not available for cooperative binding until the base piece completes. Similarly, the pair of tiles that make up the keystone assemblies must also bind to one another before they can attach to the base piece.

Connector Glues. The keystones initiate a growth of tiles, called *filler tiles*, along the edges of the base pieces. The filler tiles are designed with glues labeled with two numbers and a letter F, which will hereby be referred to as the *filler glue*. The first number of a filler glue refers to the position of the base piece that the filler tile attaches to. The second number refers to a *keystone type*. A keystone type determines the position of the assembly consisting of 8 base pieces that have attached corresponding keystone types; in other words, when a keystone attaches to a base shape of C_n^3 , the keystone type determines the position of the base shape of C_{n+1}^3 in the base shape of C_{n+2}^3 .

A filler tile is placed cooperatively using the base glue exposed on the base piece and the filler glue exposed on the keystone. Once a filler tile attaches to the keystone and the base piece, another filler tile can be cooperatively attached using the filler glue that is exposed by a filler tile that has already attached to the assembly and the base glue exposed on the base piece. This process, shown in Figure 22, can occur along arbitrarily long edges that expose the appropriate base glues. Note that the filler tiles can attach until they reach an indented corner of a base piece, since there is no base glue available for cooperative binding.

The *corner assemblies* are a set of assemblies geometrically designed to cooperatively attach using the filler glue exposed by a filler tile and the base glue exposed on the inside of an indented corner of a base piece, as shown in Figure 22. These *corner tiles*, once attached to a base piece, expose *connector glues* that allow for the base piece to attach to another base piece. A connector glue is a glue labeled with three numbers followed by a G. The first two numbers refer to the positions of the base pieces that are being attached to one another by the binding of the connector glues. The third number refers to the keystone type. Note that the base pieces will only attach to one another if they have attached the same keystone types, as the keystone type information is passed through the filler tiles to the corner tiles and their connector glues.

Assemblies that have attached sufficient connector glues attach to other assemblies that have been attached with the same *keystone type* and have also attached sufficient connector glues. The connector glues attach two assemblies using cooperative binding at opposite ends of the adjacent sides of the assemblies. See Figure 23.

For each keystone type, there is an assembly at a position or group of positions that is designed to attach

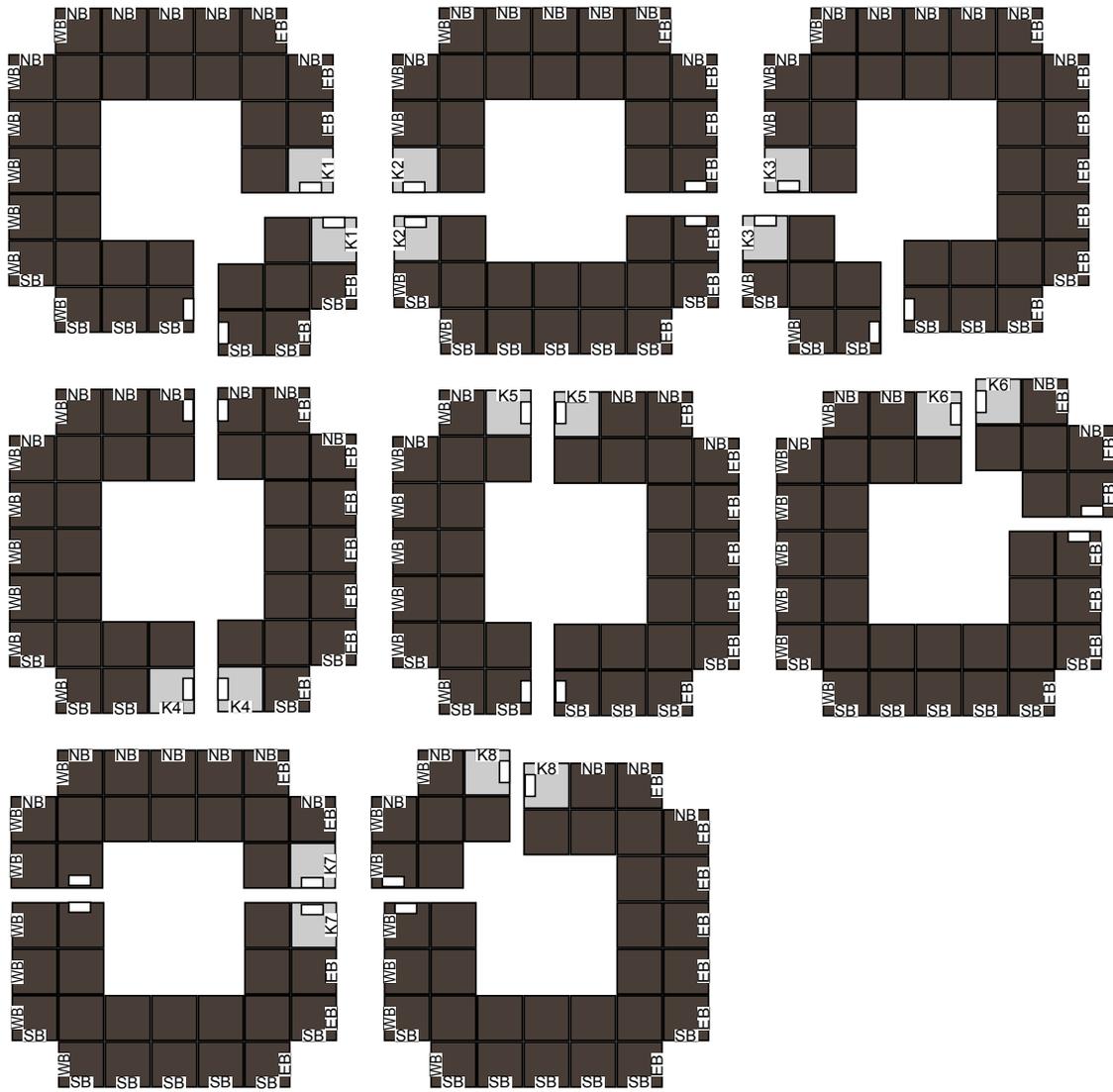


Figure 20: The distinct base pieces and their base part attachments.

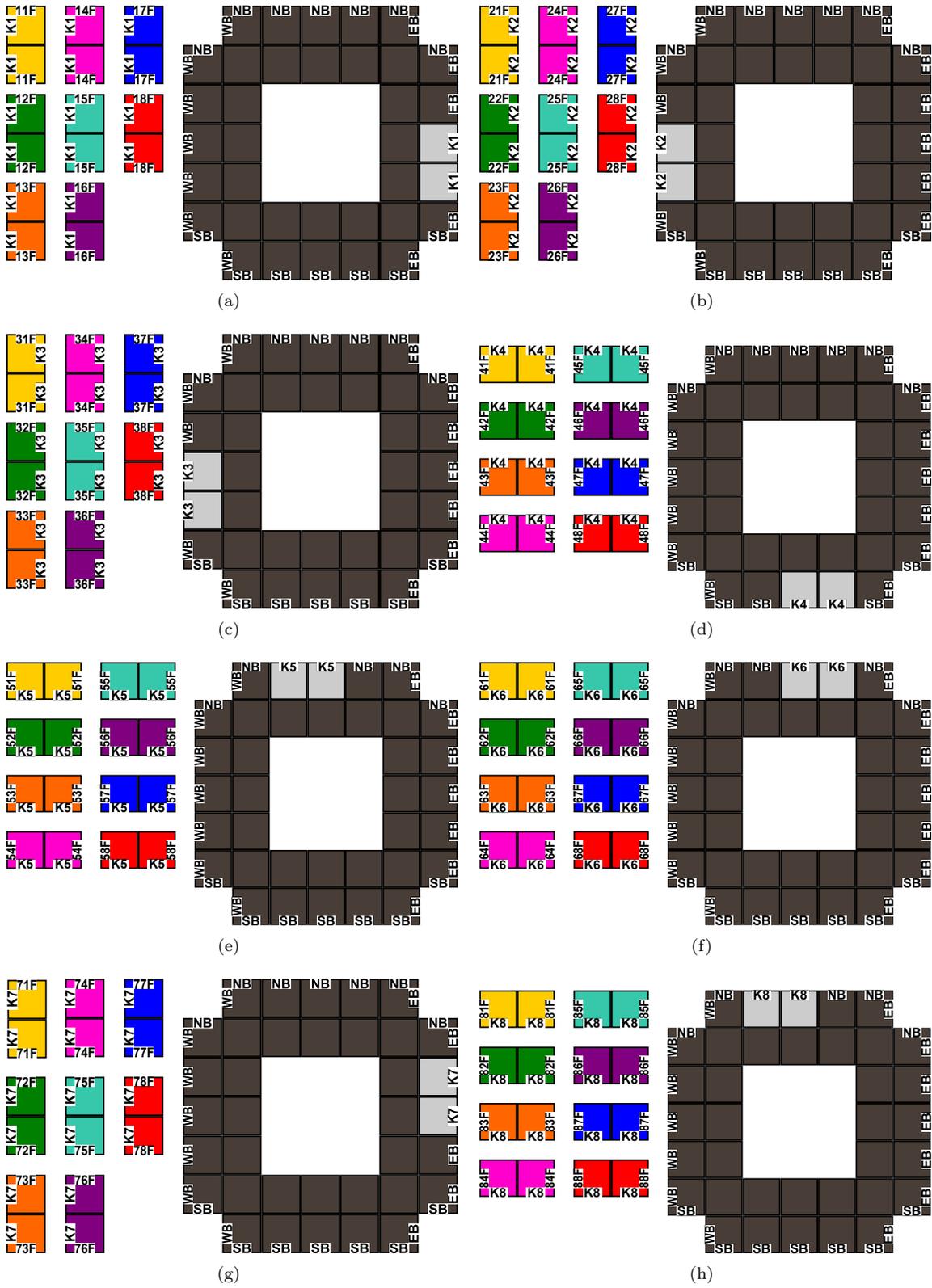


Figure 21: The base pieces and their corresponding keystone assemblies.

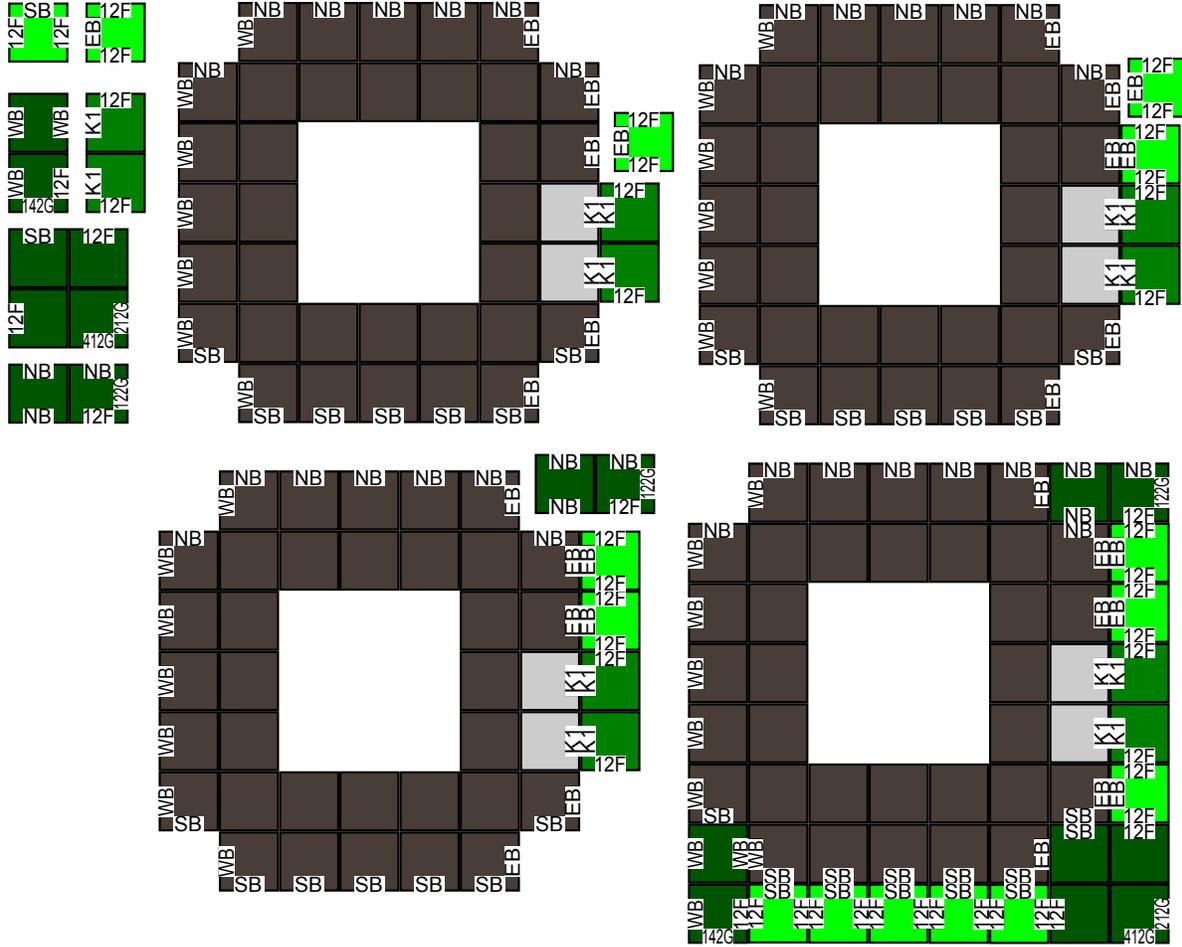


Figure 22: The process of filler tile attachment, leading to the attachment of corner assemblies.

cooperatively using two adjacent positioned assemblies, or in the case of positions 2, 4, 5, and 7, using three positioned assemblies. For example, an assembly with keystone type corresponding to position 2 will attach its subassembly corresponding to positions 6, 7, and 8 cooperatively, only after placing subassemblies at positions 4 and 5, as shown in Figure 24. Each of these assemblies imitates its base piece of corresponding position in terms of cooperative binding of the final subassembly. Note the similarities in the final attachment in the 2nd base piece in Figure 20 and Figure 24. Once these final subassemblies are attached, the resulting assemblies are base shapes of C_2^3 . At the point of these final cooperative attachments, two keystone glues are exposed, allowing for the attachment of another keystone, repeating the keystone attachment process, followed again by the filler tiles and then their corner assemblies. In the example shown in Figure 24, the assembly will be in position 2 in the creation of a base shape of C_3^3 . A finite set of tiles is used to repeat this process indefinitely. We use 288 tiles for the base pieces, 128 tiles for the keystone assemblies, and 800 tiles for the filler tiles and corner assemblies, for a total of 1,216 tiles. \square

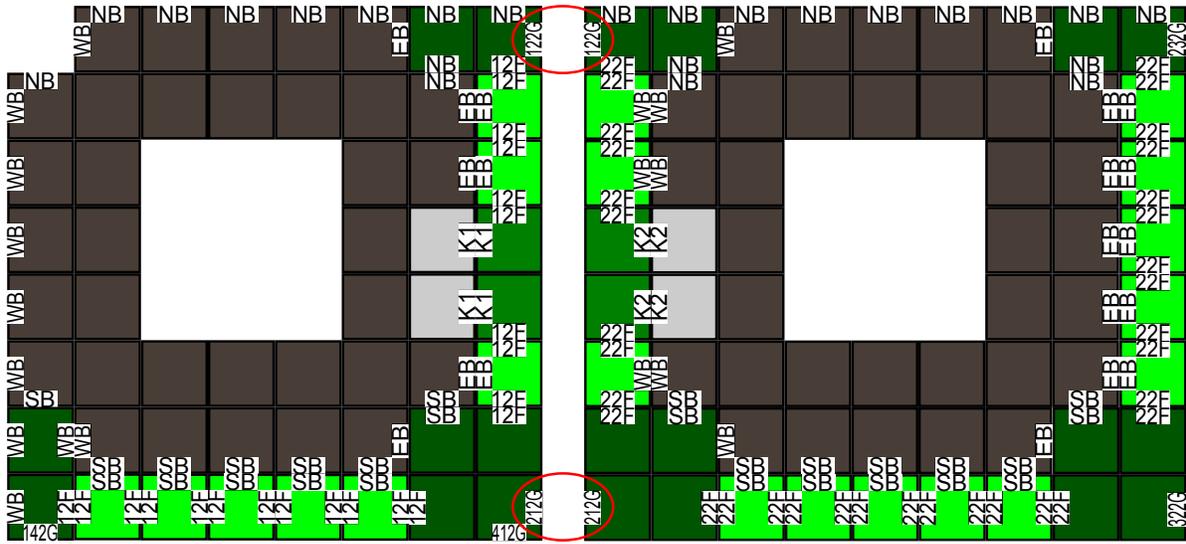


Figure 23: Two adjacent positioned assemblies attach using their connector glues.

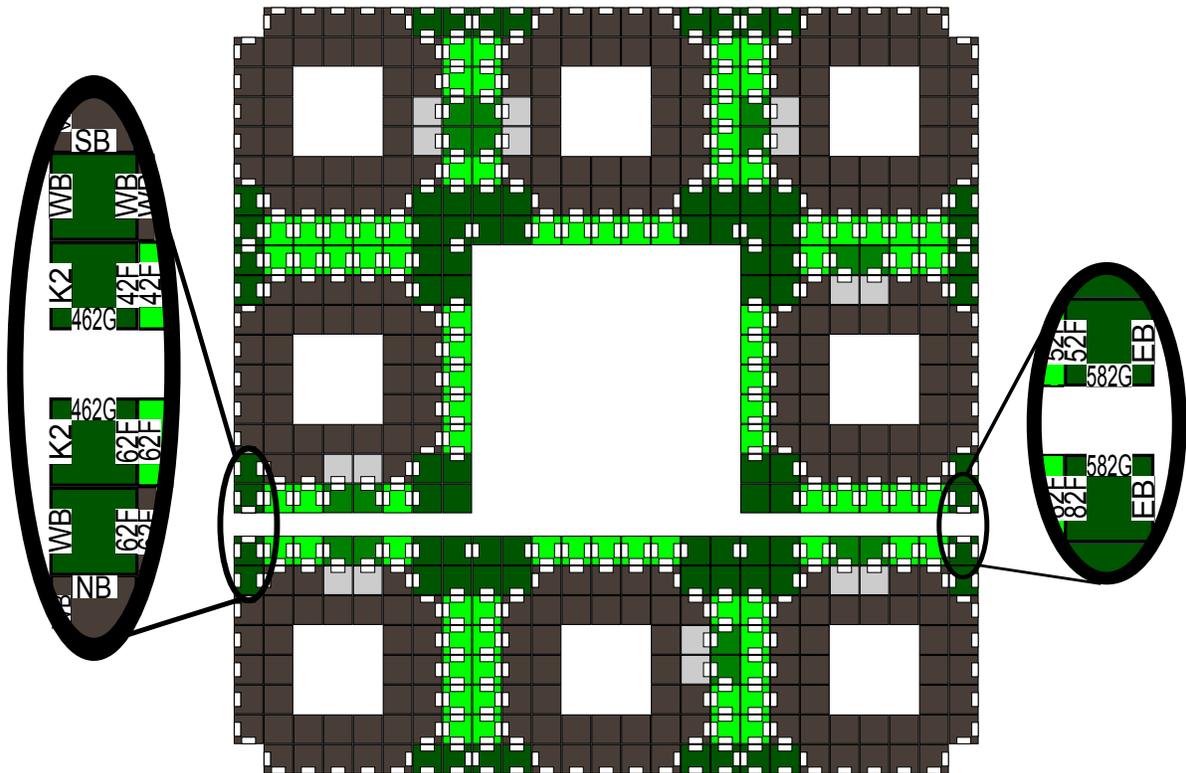


Figure 24: The creation of a base shape of C_2^3 , showing its newly exposed keystone glues.

5 Impossibility Results

5.1 The Sierpinski Triangle Does Not Strictly Assemble in the aTAM

In this section we show that S_∞^c is not strictly self-assembled by any aTAM system. Our argument is based on the constant width choke points that separate different iterations of the Sierpinski triangle. Our proof simply pigeon-holes over these choke points to generate a reoccurring tiling (with respect to tile types placed, order in which points are tiled, and even neighbors present when a tile is attached) to generate a contradiction. To prove our impossibility result, we first provide a formal definition of a reoccurring tiling called a *window-movie* (we borrow/abuse the terminology from [19]).

Our definitions of windows and window-movies are taken from [19]. However, our *windows* are slightly more general in that they may be any set of edges of the 2D lattice grid, even a finite set, as opposed to only allowing for sets that partition the 2D lattice into two disjoint regions.

Definition 5.1 (Window). A window is any subset of edges taken from the grid graph, which is defined by vertex set \mathbb{Z}^2 and edge set consisting of point pairs of exactly unit distance.

The next concept used in our result is the *window-movie*. A window-movie, with respect to a given window and an assembly sequence, is simply a specification of which glues appear along the edges of the window, and in what order, according to the given assembly sequence. Further, two window-movies are said to be *equivalent* if their respective windows are translations of each other, and if the glues placed on the window are the same in type and in order of placement. The formal definition as presented by Meunier, Patitz, Summers, Theyssier, Winslow, and Woods [19] is included below.

Definition 5.2 (Window-Movie [19]). Given an assembly sequence $\vec{\alpha}$ and a window w , the associated window movie is the maximal sequence

$$M_{\vec{\alpha},w} = (v_0, g_0), (v_1, g_1), (v_2, g_2), \dots \quad (1)$$

of pairs of grid graph vertices v_i and glues g_i , given by the order of the appearance of the glues along window w in the assembly sequence $\vec{\alpha}$. Furthermore, if k glues appear along w at the same instant (this happens upon placement of a tile which has multiple sides touching w), then these k glues appear contiguously and are listed in lexicographical order of the unit vectors describing their orientation in $M_{\vec{\alpha},w}$. Two window-movies are said to be equivalent if they are equal up to translation.

Our windows and window-movies will be focused on the following “choke point” sets of edges within the Sierpinski triangle which specify constant width- c regions that connect a given size iteration of the Sierpinski triangle to two equal size iterations through north/south bonding edges.

Definition 5.3. For a given scale factor c , define choke_i^l (left choke) to be the set of c edges of the 2D square lattice that connect the points of S_i^c with $S_i^c + (-c2^i, c2^{i+1})$, and choke_i^r (right choke) to be the set of c edges of the 2D lattice that connect the points of S_i^c with $S_i^c + (c2^i, c2^{i+1})$. See Figure 25 for an example.

Finally, our proof uses the following technical lemma which states that certain translations of pairs of points are such that they cannot both be contained within the Sierpinski triangle shape.

Lemma 5.4. Let c, i, j , and k be positive integers with $k, j > i \geq 1$, and let $\vec{v} = (c2^j - c2^i, c2^{j+1} - c2^{i+1})$. For any $p_y \in \{c2^{k+1}, c2^{k+1} + 1, \dots, c2^{k+1} + 2c - 1\}$, let $p^\ell = (-1, p_y)$ and $p^r = (0, p_y)$. Then the translations of p^ℓ and p^r by vector \vec{v} , p_v^ℓ and p_v^r , are such that either: 1) $p_v^\ell \notin S_\infty^c$ or 2) $p_v^r \notin S_\infty^c$.

Proof. First, consider the set of points defined by translating p^ℓ and p^r by any positive integer multiple of the vector $(-c, 2c)$, which includes the translation by vector \vec{v} . The only such translations in which both points lie within S_∞^c are those that move the point to a y position of height between $dc2^{k+1}$ and $dc2^{k+1} + 2c - 1$ for some positive integer d , i.e., any integer multiple of the 2^k high block of size $2c$. However, any translation of p^ℓ and p^r by $\vec{v} = (c2^j - c2^i, c2^{j+1} - c2^{i+1})$ will place the points at y -ordinate $p_y + c2^{j+1} - c2^{i+1}$. As $k > i$ and $j > i$, we know that $p_y + c2^{j+1} - c2^{i+1}$ cannot fall within the required range, i.e., it cannot sum to an integer multiple of $c2^{k+1}$ plus some Δ between 0 and $2c - 1$. \square

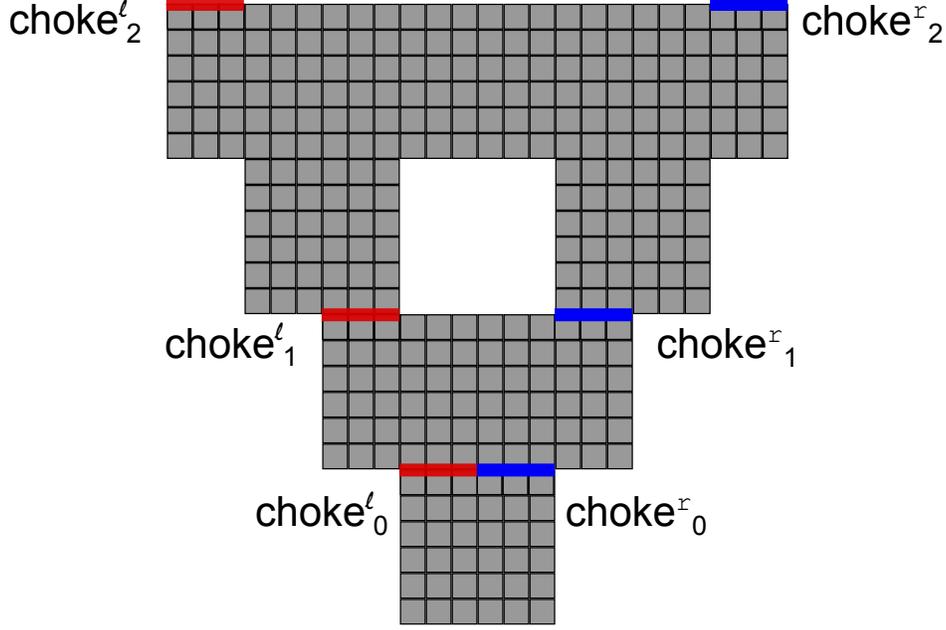


Figure 25: The left and right choke point edge sets for S_∞^3 are shown. Edges in choke_i^l are shown in red, while edges in choke_i^r are shown in blue.

Theorem 5.5. S_∞^c is not strictly assembled by any aTAM system.

Proof. An illustration of the following proof is provided in Figure 26. Towards a contradiction, suppose $\Gamma = (T, \sigma, \tau)$ strictly assembles S_∞^c for some scale factor c . Suppose σ occurs at point p_σ . Consider large enough integer **base** such that S_{base}^c contains point p_σ . Consider a second integer **big** \gg **base** along with an assembly sequence $\vec{\alpha}$ of $\Gamma = (T, \sigma, \tau)$ that tiles all points in S_{big}^c . Note that $\vec{\alpha}$ must exist for any such integer **big** as Γ strictly assembles S_∞^c .

We now consider the window-movies for $\vec{\alpha}$ at the choke points that lie between S_{base}^c and S_{big}^c . Define choke_i as follows: Consider the first point tiled by $\vec{\alpha}$ with x -ordinate value of -1 or 0 (the center two columns of the fractal) that lie north of the edges of choke_i^l and choke_i^r . If this first tiled center point has x -ordinate value -1 (i.e., the tile is placed “from the left”), define $\text{choke}_i \triangleq \text{choke}_i^l$, and define $\text{side}_i^{\vec{\alpha}} \triangleq 0$. Define $\text{choke}_i \triangleq \text{choke}_i^r$ symmetrically if the first tiled center column point has x -ordinate value 0 (i.e., the tile is placed “from the right”), and define $\text{side}_i^{\vec{\alpha}} \triangleq 1$.

Now assume **big** is large enough such that there exist i, j , $\text{big} > j > i > \text{base}$, such that 1) $\text{side}_i = \text{side}_j$, and 2) $M_{\vec{\alpha}, \text{choke}_i}$ is equivalent to $M_{\vec{\alpha}, \text{choke}_j}$. Note that such a **big** must exist by the pigeon-hole principle since there are an infinite number of chokes in S_∞^c .

We now construct a pair of points that must be placed by $\vec{\alpha}$. Further, we utilize the established equivalent window-movies and Lemma 5.4 to show that one of these two points must lie outside of S_∞^c , thus implying that $\vec{\alpha}$ does not self-assemble S_∞^c . The pair of tiles placed at these locations will be called **left** and **right**.

Without loss of generality, assume $\text{side}_i = \text{side}_j = 0$. Let $\vec{v} = (v_x, v_y)$ denote the translation difference from choke_i to choke_j . Consider the subassembly of $\vec{\alpha}$'s final produced assembly consisting of the tiled points north of the edge choke_j and west of the $x = 0$ column that are elements of the largest assembly in $\vec{\alpha}$ which contains exactly one tile in the $x = v_x$ column north of choke_j . Call this subassembly A , call the tile north of choke_j in the $x = v_x$ column **right**, and call the tile placed west of this tile **left**. Note that the translations of the **right** and **left** tiles by vector $-\vec{v}$, $\text{right}_{-\vec{v}}$ and $\text{left}_{-\vec{v}}$, lie in the center two respective columns $x = -1$ and $x = 0$ of the fractal. Let $A_{-\vec{v}}$ denote the translation of A by vector $-\vec{v}$. Examples of A and $A_{-\vec{v}}$ are shown in Figure 27.

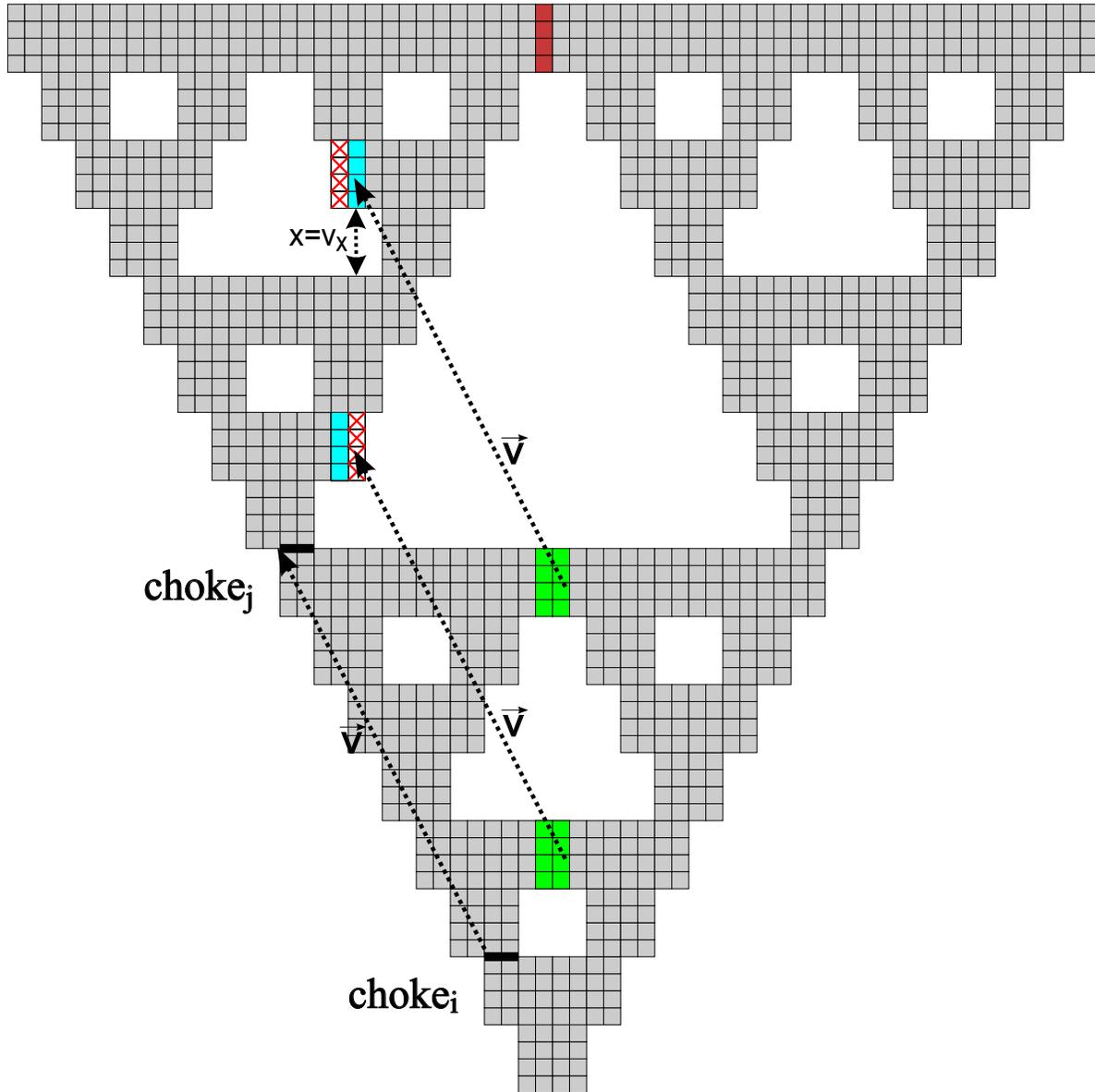


Figure 26: Simplified illustration of the proof. We consider the tiling of the $x = -1$ column of the fractal above $choke_j$ (shown in red). Tiling this column requires passing the $x = v_x$ column. As the chokes share equivalent window-movies, we can consider an alternate assembly sequence that tiles the region after $choke_i$ in the same manner as $choke_j$. If S_∞^c is self-assembled, it must be that the $x = -1$ and $x = 0$ columns above $choke_i$ (shown in green) are placed with specific y -ordinates. According to Lemma 5.4, when these tiles are translated back to $choke_j$ (shown in blue), either the left or right tiles lie outside of S_∞^c (shown by x'd tiles).

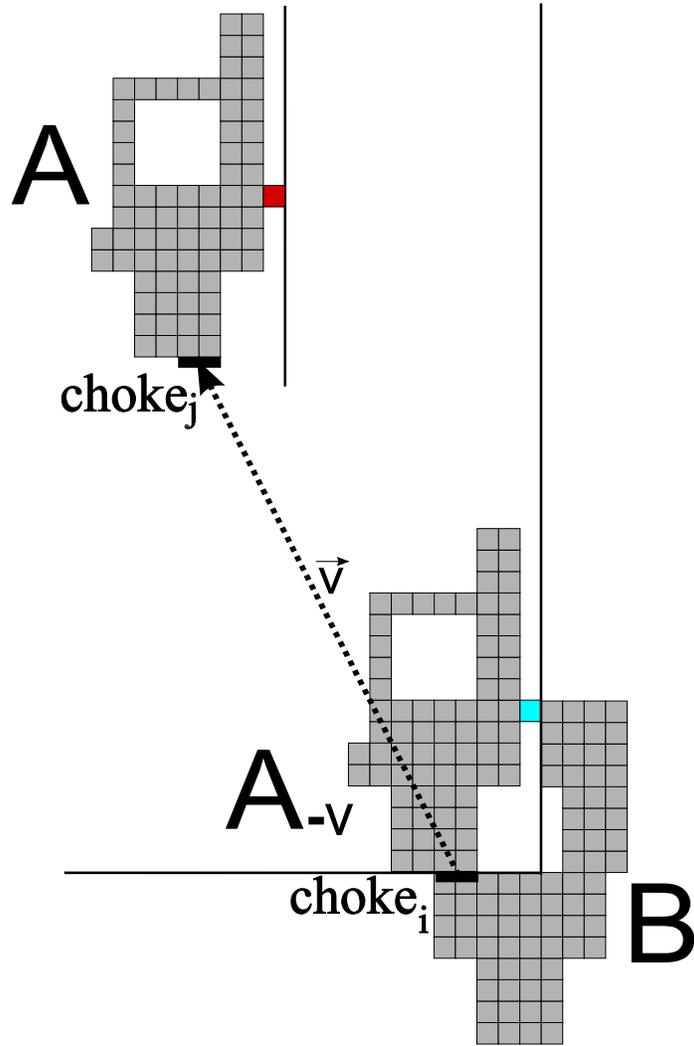


Figure 27: Illustrated here are examples of possible subassemblies A , $A_{-\vec{v}}$ and B . By definition, there are no tiles in A east of the $x = v_x$ column. Thus, when we translate A by vector $-\vec{v}$, there are no tiles east of the $x = 0$ column. B contains no tiles west of $x = 1$ above $choke_i$, therefore ensuring that $A_{-\vec{v}}$ and B contain no elements with the same coordinates. Thus, by also ensuring equivalent window-movies along $choke_i$ and $choke_j$, $A_{-\vec{v}} \cup B$ is producible. The single tile in this example of A in the $x = v_x$ column, **right**, is shown in red, and its translated copy **right_{-v}** is shown in blue.

Now consider the subassembly B defined to be the subassembly consisting of any tiles below choke_i or east of column $x = 0$ that are elements of the largest assembly in $\vec{\alpha}$ that does not contain any tiles in the $x = 0$ column north of choke_i . An example of B shown in Figure 27. Now observe that $A_{-\vec{v}} \cup B$ is producible. This is due to the equivalent window-movies that produced the original untranslated A , along with the constraint that B does not cross the middle columns of the fractal, which allows $A_{-\vec{v}}$ to grow in isolation in the same fashion A was assembled. As $A_{-\vec{v}} \cup B$ is producible by a system that allegedly assembles S_∞^c , all tiles in $A_{-\vec{v}} \cup B$ within the center $x = -1$ and $x = 0$ columns, which includes $\text{right}_{-\vec{v}}$ and $\text{left}_{-\vec{v}}$, must occur at y location between $c2^{k+1}$ and $c2^{k+1} + 2c - 1$ for a positive integer $k > i$. Note that if $\text{right}_{-\vec{v}}$ and $\text{left}_{-\vec{v}}$ fall outside of this region, they are inconsistent with S_∞^c . Thus, the location of $\text{right}_{-\vec{v}}$ and $\text{left}_{-\vec{v}}$ satisfies the constraints for points p^ℓ and p^r from Lemma 5.4, which in turn tells us that right and left , which are both placed by a valid assembly sequence, are not both contained in S_∞^c . Therefore, Γ does not self-assemble S_∞^c . \square

5.2 Perfection Requires Many Hands

In this section we show that achieving near-perfect assembly of the Sierpinski triangle and the Sierpinski carpet requires at least 3 and 8 hands respectively.

Theorem 5.6. Consider a multiple hand system (T, τ, h) .

- If $h < 3$, then Γ does not near-perfectly assemble S_∞^c for any scale factor c .
- If $h < 8$, then Γ does not near-perfectly assemble C_∞^c for any scale factor c .

Proof. We prove the result in the case of the Sierpinski triangle. The result for the Sierpinski carpet is realized in a similar fashion. Towards a contradiction, suppose $\Gamma = (T, \tau, 2)$ near-perfectly assembles S^c for some positive integer c . Let constant d be a constant at least as large as the constants Γ uses to satisfy constraints (1) and (2) of the near-perfect self-assembly definition. Consider S_i such that $|S_i| > d$. By constraint (1) of near-perfection, there exists an assembly $A \in \text{PROD}_\Gamma$ whose size is such that $|S_i| \geq |A| \geq |S_i| - d$. As $h = 2$, we know there exists $B_1, B_2 \in \text{PROD}_\Gamma$ such that $\{B_1, B_2\}$ is τ -combinable into A . Further, we know that there must exist such an A such that $|B_1| < |S_i| - d$ and $|B_2| < |S_i| - d$ (if this weren't the case, then grab the larger B assembly and try again, repeating until the constraint is satisfied). Finally, as B_1 and B_2 combine to make A , we know that $|B'| \triangleq \max(|B_1|, |B_2|) > 1/2|A| \geq 1/2|S_i| - 1/2d$. By considering large enough i with respect to constant d , we further get that $|B'| > 1/2|S_i| - 1/2d > 1/3|S_i| \geq |S_{i-1}|$, yielding $|S_i| - d > |B'| > |S_{i-1}|$. As $B \in \text{PROD}_\Gamma$, this contradicts property (2) of near-perfect assembly as B' is too small to be close to shape S_i , and too big to be a subset of shape S_{i-1} . With regard to the Sierpinski carpet, this contradiction is drawn in a similar fashion as above for all $2 \leq h \leq 7$. \square

6 Future Work

There are a number further research directions related to this work. First, knowledge of strictly assembled fractals within the aTAM is still incomplete. Impossibility results have shown that a large class of “pinch-point” fractals [21], “tree” fractals [3], and most generally “pier” fractals [14] cannot be self-assembled in the aTAM at any scale. Extending these impossibility proofs to essentially all non-trivial fractal shapes remains open. In particular, the Sierpinski carpet does not fall into these classes but is conjectured to not be self-assembled in the aTAM. When combined with our positive 2-HAM result, this proof would provide an interesting gap in power between the aTAM and the 2-HAM for a natural and well studied shape.

The strict self-assembly of fractals in the 2-HAM is still not well understood. Almost none of the impossibility results for the aTAM hold within the 2-HAM. As an example, while it is known that the scale 1 tree Sierpinski triangle cannot be self-assembled in the 2-HAM, it is only *conjectured* to be impossible at a constant scale. Beyond the 2-HAM, we can consider the *hand complexity* of an infinite shape S , which is the smallest number of hands needed for an h -HAM system to self-assemble S . Is the hand complexity of

the non-tree Sierpinski triangle 3? Moreover, we see that in the case of the non-tree Sierpinski triangle, an increase in hands from 3 to 6 allows for a much simpler construction with respect to tile complexity. Is there a more general connection between tile complexity, or other efficiency metrics, and hand complexity?

The consideration of multiple handed self-assembly introduces a number of additional directions beyond fractal assembly. For example, what finite shapes can be built more efficiently with more hands? Can a shape built with a given number of hands be transformed into an equivalent system with fewer hands, perhaps at the cost of scale factor? How are well studied computational complexity questions such as *producibility verification* and *unique assembly verification* [4] affected by the h parameter in an h -HAM systems? Another interesting direction is to consider h -HAM assembly as an error model in which an intended 2-HAM system must be designed be robust against assembly growth with up to h hands. Can general 2-HAM systems be made robust to such constraints at a reasonable cost to tile complexity and scale factor?

References

- [1] Zachary Abel, Nadia Benbernou, Mirela Damian, Erik D. Demaine, Martin L. Demaine, Robin Flatland, Scott D. Kominers, and Robert T. Schweller, *Shape replication through self-assembly and rnae enzymes*, Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, 2010, pp. 1045–1064.
- [2] Gagan Aggarwal, Qi Cheng, Michael H. Goldwasser, Ming-Yang Kao, Pablo Moisset de Espanés, and Robert T. Schweller, *Complexities for generalized models of self-assembly*, SIAM Journal on Computing **34** (2005), no. 6, 1493–1515.
- [3] Kimberly Barth, David Furcy, Scott M. Summers, and Paul Totzke, *Scaled tree fractals do not strictly self-assemble*, Unconventional Computation and Natural Computation - 13th International Conference, UCNC 2014, London, ON, Canada, July 14-18, 2014, Proceedings, 2014, pp. 27–39.
- [4] Sarah Cannon, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Matthew J. Patitz, Robert T. Schweller, Scott M. Summers, and Andrew Winslow, *Two hands are better than one (up to constant factors): Self-assembly in the 2ham vs. atom*, 30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany, 2013, pp. 172–184.
- [5] Alessandra Carbone and Nadrian C Seeman, *A route to fractal dna-assembly*, Natural Computing **1** (2002), no. 4, 469–480.
- [6] Ho-Lin Chen and David Doty, *Parallelism and time in hierarchical self-assembly*, Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012, 2012, pp. 1163–1182.
- [7] Erik D. Demaine, Martin L. Demaine, Sándor P. Fekete, Mashhood Ishaque, Eynat Rafalin, Robert T. Schweller, and Diane L. Souvaine, *Staged self-assembly: nanomanufacture of arbitrary shapes with O(1) glues*, Natural Computing **7** (2008), no. 3, 347–370.
- [8] Erik D. Demaine, Martin L. Demaine, Sándor P. Fekete, Matthew J. Patitz, Robert T. Schweller, Andrew Winslow, and Damien Woods, *One tile to rule them all: Simulating any tile assembly system with a single universal tile*, Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I, 2014, pp. 368–379.
- [9] Erik D. Demaine, Matthew J. Patitz, Robert T. Schweller, and Scott M. Summers, *Self-assembly of arbitrary shapes using rnae enzymes: Meeting the kolmogorov bound with small scale factor (extended abstract)*, 28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany, 2011, pp. 201–212.

- [10] Erik D. Demaine, Matthew J. Patitz, Trent A. Rogers, Robert T. Schweller, Scott M. Summers, and Damien Woods, *The two-handed tile assembly model is not intrinsically universal*, Automata, Languages, and Programming (Fedor V. Fomin, Rsi Freivalds, Marta Kwiatkowska, and David Peleg, eds.), Lecture Notes in Computer Science, vol. 7965, Springer Berlin Heidelberg, 2013, pp. 400–412 (English).
- [11] David Doty, *Producibility in hierarchical self-assembly*, Unconventional Computation and Natural Computation - 13th International Conference, UCNC 2014, London, ON, Canada, July 14-18, 2014, Proceedings, 2014, pp. 142–154.
- [12] David Doty, Matthew J. Patitz, Dustin Reishus, Robert T. Schweller, and Scott M. Summers, *Strong fault-tolerance for self-assembly with fuzzy temperature*, 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA, 2010, pp. 417–426.
- [13] Bin Fu, Matthew J. Patitz, Robert T. Schweller, and Robert Sheline, *Self-assembly with geometric tiles*, Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I, 2012, pp. 714–725.
- [14] David Furcy and Scott M. Summers, *Scaled pier fractals do not strictly self-assemble*, Computing Research Repository [abs/1406.4197](https://arxiv.org/abs/1406.4197) (2014).
- [15] Steven M. Kautz and James I. Lathrop, *Self-assembly of the discrete sierpinski carpet and related fractals*, DNA Computing and Molecular Programming, 15th International Conference, DNA 15, Fayetteville, AR, USA, June 8-11, 2009, Revised Selected Papers, 2009, pp. 78–87.
- [16] Steven M. Kautz and Brad Shuttters, *Self-assembling rulers for approximating generalized sierpinski carpets*, Algorithmica **67** (2013), no. 2, 207–233.
- [17] James I. Lathrop, Jack H. Lutz, and Scott M. Summers, *Strict self-assembly of discrete sierpinski triangles*, Theoretical Computer Science **410** (2009), no. 4-5, 384–405.
- [18] Jack H. Lutz and Brad Shuttters, *Approximate self-assembly of the sierpinski triangle*, Theory of Computing Systems **51** (2012), no. 3, 372–400.
- [19] Pierre-Etienne Meunier, Matthew J. Patitz, Scott M. Summers, Guillaume Theyssier, Andrew Winslow, and Damien Woods, *Intrinsic universality in tile self-assembly requires cooperation*, Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014, 2014, pp. 752–771.
- [20] Jennifer E. Padilla, Matthew J. Patitz, Robert T. Schweller, Nadrian C. Seeman, Scott M. Summers, and Kingsi Zhong, *Asynchronous signal passing for tile self-assembly: Fuel efficient computation and efficient assembly of shapes*, Int. J. Found. Comput. Sci. **25** (2014), no. 4, 459–488.
- [21] Matthew J. Patitz and Scott M. Summers, *Self-assembly of discrete self-similar fractals*, Natural Computing **9** (2010), no. 1, 135–172.
- [22] ———, *Self-assembly of decidable sets*, Natural Computing **10** (2011), no. 2, 853–877.
- [23] Paul WK Rothmund, Nick Papadakis, and Erik Winfree, *Algorithmic self-assembly of dna sierpinski triangles*, PLoS biology **2** (2004), no. 12, e424.
- [24] Robert T. Schweller and Michael Sherman, *Fuel efficient computation in passive self-assembly*, Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013, 2013, pp. 1513–1525.
- [25] Hao Wang, *Dominoes and the aea case of the decision problem*, Computation, Logic, Philosophy, Springer, 1990, pp. 218–245.

- [26] Erik Winfree, *Algorithmic self-assembly of dna*, Ph.D. thesis, California Institute of Technology, 1998.
- [27] Erik Winfree and Renat Bekbolatov, *Proofreading tile sets: Error correction for algorithmic self-assembly*, DNA Computing, 9th International Workshop on DNA Based Computers, DNA9, Madison, WI, USA, June 1-3, 2003, revised Papers, 2003, pp. 126–144.