1-17-2010

Shape Replication through Self-Assembly and RNase Enzymes

Zachary Abel

Nadia Benbernou

Mirela Damian

Erik D. Demaine

Martin L. Demaine

See next page for additional authors

Recommended Citation

Abel, Z., Benbernou, N., Damian, M., Demaine, E. D., Demaine, M. L., Flatland, R., Kominers, S. D., & Schwelle, R. (2010). Shape replication through self-assembly and RNase enzymes. Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, 1045–1064.

Shape Replication through Self-Assembly and RNase Enzymes

Zachary Abel[*] Nadia Benbernou[†] Mirela Damian[‡] Erik D. Demaine[†]
Martin L. Demaine[†] Robin Flatland[§] Scott D. Kominers[*] Robert Schweller[¶]

Abstract

We introduce the problem of *shape replication* in the Wang tile self-assembly model. Given an input shape, we consider the problem of designing a self-assembly system which will replicate that shape into either a specific number of copies, or an unbounded number of copies. Motivated by practical DNA implementations of Wang tiles, we consider a model in which tiles consisting of DNA or RNA can be dynamically added in a sequence of stages. We further permit the addition of RNase enzymes capable of disintegrating RNA tiles. Under this model, we show that arbitrary genus-0 shapes can be replicated infinitely many times using only $O(1)$ distinct tile types and $O(1)$ stages. Further, we show how to replicate precisely n copies of a shape using $O(\log n)$ stages and $O(1)$ tile types.

1 Introduction

Self-assembly can be seen as an approach to harnessing the power of (synthetic) biology to work with objects at the nanometer scale. Most theoretical models of self-assembly [Win98, Adl00, RW00a, DDF$^+$08] enable the building of structures by joining materials together, but forbid later splitting materials apart. Although such models are interesting and useful for manufacturing, biology offers substantially more power: biological structures may interact, change, and compute.

We introduce the *enzyme self-assembly model*, a slight strengthening of standard tile self-assembly models to enable partial disassembly of structures through the destruction of all tiles within a fixed tile class. This model is motivated by the existence of RNase enzymes [UE71], which can disintegrate RNA molecules without disturbing DNA molecules, and the work of Rothemund and Winfree [RW00a] which suggested the plausibility of such a technique. In enzyme self-assembly, tiles are divided into two classes—DNA and RNA—and a manufactured structure can be partially disassembled via the addition of RNase, which destroys all RNA tiles.

Although enzyme addition is a heavily restricted operation, its addition already enables more efficient construction of certain shapes. For example, constructing a $1 \times n$ rectangle requires n distinct tile types in the standard self-assembly model, but requires only $O(\log n/ \log\log n)$ tile types with enzymes.[1]

More interestingly, the enzyme self-assembly model enables manipulation beyond just assembly. For example, RNA tiles offer the ability to "probe" an existing, unknown DNA object without irreparably damaging that object. We show that this ability can be exploited to build a general-purpose *replicator*, which produces either a desired number or infinitely many copies of a given, unknown object, using a constant number of tile types and operations.

While algorithms for replicating shapes are a long way from answering Caption Jean-Luc Picard's "tea, Earl Grey, hot" [Rod94], our approach opens the door for a new genre of self-assembly problems. Ultimately, we might hope to build a general-purpose biological computer that can manipulate matter at nanometer scales.

Our results. We introduce enzyme self-assembly as an extension of the staged assembly model [DDF$^+$08], as defined formally in Section 2. In the *replication problem*, we are given a single copy of an unknown shape, with the small (necessary) restrictions that the shape's edges lie on the unit-square lattice (so that it exists within the model) and has a known glue type along its boundary (so that it can possibly interact with added tiles). In the *precise yield* version of the replication problem (Section 3), the goal is to produce exactly n copies of the given shape, while in the *infinite yield* version (Section 4), the goal is to produce infinitely many copies of the given shape. (The latter goal is natural, as self-assembly models normally

[*]Department of Mathematics, Harvard University, Cambridge, MA 02138, USA, {zabel,kominers}@fas.harvard.edu.

[†]MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA, {nbenbern,edemaine, mdemaine}@mit.edu. Partially supported by NSF CAREER award CCF-0347776 and DOE grant DE-FG02-04ER25647.

[‡]Department of Computer Science, Villanova University, PA 19085, USA, mirela.damian@villanova.edu.

[§]Department of Computer Science, Siena College, NY 12211, USA, flatland@siena.edu

[¶]Department of Computer Science, University of Texas-Pan American, Edinburg, TX 78539, USA, schwellerr@cs.panam.edu

[1]We can build an $n \times n$ square using $O(\log n/ \log\log n)$ tile types in the standard self-assembly model, and a simple modification makes all tiles RNA except for a single row of DNA.

assume that there are infinitely many copies of all objects, with the notable exception of the given shape.)

We solve both versions of the replication problem for all genus-0 shapes. For the precise yield of n replicas, we provide a solution using $O(1)$ tile types and $O(\log n)$ stages. For the infinite yield version, we provide a solution using $O(1)$ tile types and $O(1)$ stages. Both algorithms assume that the minimum feature size of the given shape is not too small: the feature size must be at least 5 for precise yield, and at least $\Omega(\log n)$ for infinite yield. In the special case of x-monotone shapes, the bound for infinite yield can be improved to 4. (Section 4.1).

Our results break the mold of previous self-assembly research, showing that a simple and heavily constrained destructive operation allows a wealth of new problems to be solved. On the one hand, our construction for precise yield is conceptually simple and may lead to practical shape replication. On the other hand, our construction for infinite yield is intricate, and seems necessarily so, illustrating the algorithmic depth of the enzyme self-assembly model and paving the way for further research.

Related work. In the area of replication, Schulman and Winfree considered the self-replication of DNA crystals in the tile assembly model and its relation to evolution [SW05]. In the direction of removing and breaking apart previous assembled structures, Aggarwal et al. [AGKS04] and Kao and Schweller [KS06] considered the multiple temperature model to add and remove assemblies by adjusting the temperature of the assembly system.

2 Staged Replication with RNA Removal: Model and Problem Formulation

In this section we define the Staged Replication Assembly Model (SRAM). We start with basic definitions common to most assembly models, then we describe the SRAM model.

2.1 Basic Definitions.

Tiles and glues. A *(Wang) tile type* t is a unit square defined by the ordered quadruple $\langle \mathrm{north}(t), \mathrm{east}(t), \mathrm{south}(t), \mathrm{west}(t)\rangle$ of glues on the four edges of the tile. Each *glue* is taken from a finite alphabet Σ, which includes a special "null" glue denoted null. For simplicity of bounds, we do not count the null glue in the *glue complexity* $g = |\Sigma| - 1$. Each glue type is assigned an integer strength from 0 to some given value τ by the *glue function* $G : \Sigma^2 \to \{0, 1, \ldots, \tau\}$. It is assumed that $G(x, y) = G(y, x)$ for all $x, y \in \Sigma$ and that $G(\mathrm{null}, x) = 0$ for all $x \in \Sigma$. Indeed, in all of our constructions, as in the original model of Adleman [Adl00], $G(x, y) = 0$ for all $x \neq y$[2], and each $G(x, x) \in \{1, 2, \ldots, \tau\}$. The *tile complexity* of the system is $|T|$.

Configurations. Define a *configuration* to be a function $C : \mathbb{Z}^2 \to T \cup \{\mathrm{empty}\}$, where empty is a special tile that has the null glue on each of its four edges. The *shape* of a configuration C is the set of positions (i, j) that do not map to the empty tile. For two configurations C and D that do not overlap, we define the union configuration $U = C \bigcup D$ to be the configuration such that $U(x, y) = C(x, y)$ if $C(x, y) \neq \mathrm{empty}$, and $U(x, y) = D(x, y)$ if $C(x, y) = \mathrm{empty}$.

Adjacency graphs. Define the *adjacency graph* G_C of a configuration C as the following weighted graph. The vertices are coordinates (i, j) such that $C(i, j) \neq \mathrm{empty}$. There is an edge between two vertices (x_1, y_1) and (x_2, y_2) if and only if $|x_1 - x_2| + |y_1 - y_2| = 1$. For two vertices at positions (x_1, y_1) and $(x_1 + 1, y_1)$, the weight of the edge connecting these vertices is $G(\mathrm{east}(C(x_1, y_1)), \mathrm{west}(C(x_1 + 1, y_1)))$. For vertices at positions (x_1, y_1) and $(x_1, y_1 + 1)$, the weight of the connecting edge is $G(\mathrm{south}(C(x_1, y_1)), \mathrm{north}(C(x_1, y_1 + 1)))$.

Supertiles and stability. Intuitively, a supertile tile is a connected pattern of tile types on the grid. However, the exact position on the grid is not important, so we define supertiles to be equal up to translation. For a connected configuration C, the set of all translations of C is referred to as a *supertile*. For each supertile S, let S_c denote some representative configuration of S. For a non-negative integer τ, a supertile S is said to be *stable* at temperature τ if the weight of the minimum cut of the adjacency graph G_{S_c} has weight at least τ.

Supertile combination. The self-assembly process is driven by the possibility for any two supertiles in solution to come together to form a new type of supertile. Formally, we define the set of all supertiles that can be assembled from a given pair of supertiles X and Y as the *combination* set of supertiles $C^{\tau}_{(X,Y)}$. For supertiles X and Y, if there exists a translation Y^r_c of Y_c such that X_c and Y^r_c do not overlap, and $X_c \bigcup Y^r_c$ is stable at temperature τ, then the supertile S corresponding to the set of all translations of $X_c \bigcup Y^r_c$ is in $C^{\tau}_{(X,Y)}$.

Breaking apart supertiles. In this paper we assume all tile types are labeled as either DNA or RNA tile types. For a given supertile Γ that is stable at temperature τ, we permit the operation of adding an RNase enzyme to the tile which removes all RNA tile types from the supertile configuration (all RNA tile positions are formally set to be the empty tile). For a given temperature the resultant supertile may no longer be stable at temperature τ, and

[2]With a typical implementation in DNA, glues actually attach to unique complements rather than to themselves. However, this depiction of the glue function is standard in the literature and does not affect the power of the model.

thus defines a multiset of subsupertiles consisting of the maximal stable subsupertiles of Γ at temperature τ, denoted as $BREAK_\tau(\Gamma)$.

2.2 Staged Replication with RNA removals.

Staged replication takes place over a number of stages. A stage replication system specifies each stage as either a *tile addition* stage, in which new tile types are added to the system, or an *enzyme* stage, in which assembled supertiles are broken into pieces by deleting occurrences of RNA tile types. In both cases, each stage consists of an initial set of preassembled supertiles from the previous stage, unioned with a new set of tile types in the case of a tile addition stage, or the current supertile set broken into subsupertiles in the case of an enzyme stage. From this initial set, the output of the stage is determined by the two-handed assembly model defined as follows.

Two-handed Assembly. For an initial set of supertiles S and a temperature τ, the set of produced supertiles $P'_{(S,\tau)}$ under the two-handed assembly model is defined to contain all supertiles in S, as well as any supertile that can be obtained by combining two produced supertiles at temperature τ. More formally, $P'_{(S,\tau)}$ is defined recursively as follows: (1) $S \subseteq P'_{(S,\tau)}$ and (2) for any $X, Y \in P'_{(S,\tau)}$, $C^\tau_{(X,Y)} \subseteq P'_{(S,\tau)}$. The set of *terminally* produced supertiles $P_{(S,\tau)}$ is the subset of produced supertiles that cannot grow any further. More formally, $P_{(S,\tau)} = \{X \in P' \mid \forall Y \in P', C^\tau_{(X,Y)} = \emptyset\}$. In the case that there exist arbitrarily large supertiles in $P'_{(S,\tau)}$, we say there exist infinite supertiles that are terminally produced. In such a case, for any (infinite) supertile Γ whose subsupertile specified by a dimension n box, centered at the origin, matches the corresponding subsupertile for some supertile in $P'_{(S,\tau)}$ (up to translation) for all $n > 0$, we say Γ is an element of $P_{(S,\tau)}$. In this paper we assume $\tau = 2$ and simply write P_S to denote a terminally produced supertile set derived from initial set S.

Staged replication assembly model. Formally, a *staged replication assembly system* (SRAM system) is a sequence $\langle OP_1, OP_2, \ldots, OP_r\rangle$ which denotes a list of operations to be performed at each of the r stages of the assembly process. The input to a SRAM system is some finite supertile s consisting of all DNA tile types. Further, s is assumed to have a known strength 1 glue type σ on all exposed surfaces.[3] Each operation OP_i consists of either the operation $ADD(T_i)$ for some set of singleton tile types T_i, or the operation $\mathtt{BREAK}$ in the case of an enzyme stage. At each stage i, a collection of supertiles is assembled under the two-handed assembly model and fed as input to the next stage. In particular, stage 1 starts with initial supertile set $s \bigcup T_1$ for an initial set of tile types T_1 specified by the SRAM system, and produces as output the set $\mathtt{OUTPUT}_1 = P_{(s \bigcup T_1)}$, the terminally produced set of supertiles derived from $s \bigcup T_1$. In general, the output of stage i is defined to be $\mathtt{OUTPUT}_i = P_{(\mathtt{OUTPUT}_{i-1} \bigcup T_i)}$ for tile addition stages, and $\mathtt{OUTPUT}_i = P_{BREAK(\mathtt{OUTPUT}_{i-1})}$ for enzyme stages. For an r stage system, the final output of the system is $\mathtt{OUTPUT}_r$.

Supertile multiplicity and problem formulation. We assume the input supertile s whose shape is to be replicated has initial multiplicity of 1. The goal is then to design an SRAM system that will either assemble exactly n supertiles with the shape of s, or infinitely many supertiles with the shape of s. For a given $\mathtt{ADD}(T_i)$ stage in an SRAM system, the multiplicities of each supertile in $\mathtt{OUTPUT}_i$ are determined by the multiplicities of the assembled supertiles from the previous stage assemblies $\mathtt{OUTPUT}_{i-1}$. In particular, if a supertile $x \in \mathtt{OUTPUT}_i$ can only be obtained by combining supertiles from T_i and $\mathtt{OUTPUT}_{i-1}$ with infinite multiplicities, then x has infinite multiplicity after stage i. On the other hand, if x can only be assembled by combining exactly one copy of an finite count supertile $y \in \mathtt{OUTPUT}_{i-1}$ with supertiles of infinite multiplicity from $\mathtt{OUTPUT}_{i-1} \bigcup T_i$, and x is the only supertile in $\mathtt{OUTPUT}_i$ that can be assembled from a copy of y, the multiplicity of x after stage i is the same as the multiplicity of y after stage $i-1$. If all assembled supertiles fall into the above two categories, the assembly for stage i is denoted as *reliable*. In the case that the i^{th} stage consists of a $BREAK(\mathtt{OUTPUT}_{i-1})$ operation, any supertile $y \in \mathtt{OUTPUT}_{i-1}$ with multiplicity m gives multiplicity m to each subsupertile in $BREAK(y)$, with multiplicities being summed in the case of multiple copies of the same supertile.

The goal of this paper is to design SRAM systems that will create as output copies of any given input shape s with either multiplicty precisely some value n (precise yield problem) or infinite multiplicity (infinite yield problem). We measure the efficiency of such SRAM systems in terms of the number of distinct tile types (tile complexity) and the number of stages used (stage complexity).

3 Precise Yield Replication

In this section we describe techniques for self-assembly of tiles that produce a specified number n of copies of a given input shape. We begin in Section 3.1 with a description of an efficient replication technique for rectangular input shapes, that achieves $O(\log n)$ tile complexity and constant stage complexity. We turn to arbitrary

[3]For clarity, in some of our constructions we assume the input shape has 4 distinct glues, one for each of the the possible directions north, south, east, and west. However, with some added tiles and stages, our constructions can be modified to satisfy the single glue version of the model while achieving the same asymptotic complexities.

input shapes in Section 3.2, where we explore tradeoffs between tile complexity and stage complexity.

3.1 Precise Yield for Rectangles. Given an integer $n > 0$ and a rectangle R, we show how to create n copies of R in one stage. The main idea is to use RNA-tiles to build a binary counter that increments precisely n times, and attach to each distinct number a copy of R built from DNA-tiles. The counter gets destroyed by an enzyme in a second phase, leaving the n independent copies of R undisturbed.

Several versions of binary counters exist [RW00b, ACGH01, CE03, AGKS04], differing in their tile and stage complexity. We start with a brief review of the tile set used in the counter implementation in [AGKS04], which we subsequently modify to support rectangle replication.

3.1.1 Binary Counter [AGKS04]. The binary counter is represented as a $k \times n$ rectangle, with $k = O(\log n)$. The leftmost column represents the initial number, and each successive column in the right direction encodes the next natural number. Five classes of tiles are used in this system (see Fig. 1):

1. Seed tiles(k). An initial column of $k = \log n$ distinct seed tiles are used to encode the initial number in binary, with the most significant bit represented by the topmost seed tile.

2. Chain tiles(2). Chain tiles encode the least significant bit of each number. The chain tile C_0 encodes a bit 0, which always increments in the next step by attaching to tile C_1 (encoding 1) by its strength-2 side. The chain tile C_1 becomes part of a tile column sequence that encodes a 1-bit string, which must rollover in the next step. The rollover is initiated by a hairpin tile (see the last two columns of Fig. 1f).

3. Probe tiles(2). A probe tile P is used to grow a column upwards, starting from the bottom chain tile, until the bit that requires incrementing in the next step is encountered (see the column encoding the bit string 0111 in Fig. 1f). In the next increment step, a return probe tile R attaches to each probe tile P, simulating a rollover of the 1-bit encoded by P.

4. Hairpin tiles(2). A hairpin tile H_0 in a column marks the bit that needs incrementing in the next column. Tile H_0 grows into a hairpin in the next increment step by attaching to a second hairpin tile H_1, which performs the actual incrementation. The column growing downwards from H_1 encodes 0-bits only (see last column in Fig. 1f).

5. Normal tiles (2). Normal 0-tiles and 1-tiles fill in a column upwards, starting from an H or a C tile.

The first eight increment steps of a counter based on this tile system are shown in Fig. 1f.

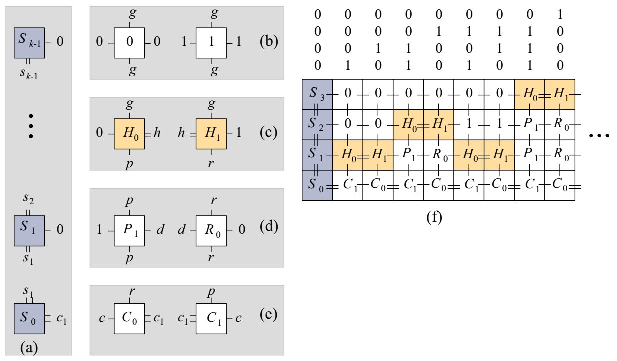


Figure 1: Assembling a binary counter [AGKS04]: (a) seed tiles (b) normal tiles (c) hairpin tiles (d) probe tiles (e) chain tiles (f) binary counter example (first 8 increment steps).

3.1.2 2-State Binary Counter. Our procedure to replicate a given rectangle R makes use of a binary counter that increments precisely n times, once for each copy of R. The key idea is to attach a copy of R to each distinct number represented by the counter.

This means that the counter cannot increment in each step, since it must leave room for a copy of R – call it R_1 – to attach to the current counter value. Instead, the counter must propagate its current value over the length of the top edge of R_1, at the end of which it can increment (if lower than the maximum value) to signal that a new copy of R must be assembled. This process requires a slight modification to the original counter, to enable the counter to operate in two states: an *increment (I)* state, in which the increment operation is enabled, and a *propagate (P)* state, in which the increment operation is disabled, and the current counter value is merely propagated forth to the next column.

Initially, the counter is in a P-state. The transition between the two states is implemented by means of a trigger consisting of two columns: a first column growing upwards enables the transition from a P-state to an I-state; the increment operation and the transition from an I-state to a P-state happen simultaneously in the next step, which assembles a new counter column attached to a second trigger column; this second trigger column grows downwards and, once it reaches the bottom of R, it triggers the assembly of a new copy of R. The trigger makes this assembly possible by encoding the two dimensions of R (horizontal length and vertical width) with different types of tiles.

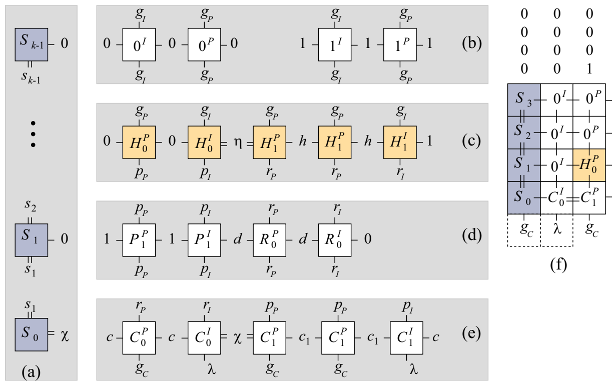


Figure 2: Tile system for the counter used in rectangle replication.

We now turn to describing the tile system used for the two-state (P, I) counter. Corresponding to each tile T in the original tile system from Fig. 1, our two-state counter uses two tiles T^I and T^P: tile T^I is used to indicate an I-state and tile T^P is used to indicate a P-state. See Fig. 2 for an illustration of this tile system. Tiles with bond strength 2 on one side (such as H_1 and C_1) constitute the only exception from the duplication rule. Corresponding to each such tile in the original system, we use three tiles in our two-state system: one I-tile and two P-tiles. One of the two P-tiles carries the bond strength 2 on one side, whereas the second P-tile has bond strength 1 on all sides (see the two H_1^P and the two C_1^P tiles from Figs. 2(c,e)). Two key ideas underlie this tile system design:

1. Replicas of a P-tile should enable the assembly of a horizontal chain of arbitrary length, to propagate the encoded bit. This renders necessary the condition that a P-tile should have matching glues on its vertical (left and right) sides.

2. A horizontal chain of P-tiles should not grow indefinitely at temperature $T = 2$. To prevent infinite growth, we use a different glue on a P-tile side with bond strength 2 (e.g., η for H_1^P in Fig. 2c and χ for C_1^P in Fig. 2e).

We also use different glues on the horizontal sides of P-tiles and I-tiles, to guarantee that a counter column contains only P-tiles or only I-tiles (never both).

By design, seed tiles are P-tiles; the transition from the P-state to an I-state will be triggered by an event outside of the counter. Let us assume for the moment that this event (indicated by the two dashed tiles in Fig. 2f) has occurred in the system, and that a new column of I-tiles encoding the original counter value δ has been assembled. In Fig. 2f, we use $\delta = 0$ as a running example. By design, the increment operation and the transition to a P-state will happen simultaneously in the next step. Note however that after the assembly step for the next column, the assembly process for the counter halts: since all unattached sides of the current counter assembly have strength 1, no tile can bond to it at temperature $T = 2$. This indicates that our two-state counter cannot function on its own, and is in fact designed to operate in conjunction with the rest of the replication system described next.

3.1.3 Rectangle Tile Sets. Let R be the input rectangle. We design a system that creates a replica R_1 of R under the following assumptions: (a) R is stable at temperature $T = 2$ (b) the upper right corner tile of R has bond strength 2 on its unattached (north and east) sides; the north side of this tile has glue g_N and the east side has glue g_E (c) the lower left corner tile of R has bond strength 2 and glue g_S on its south side. (d) any other tile of R has bond strength 1 on its unattached sides. East unattached sides have glue e, and south unattached sides of R have glue s. These ideas are illustrated in Fig. 3a. The north side of the upper right corner tile is used to initiate the counter; the east side of the same tile is used to initiate the trigger (see Fig. 3e); and the south side of the lower left corner tile is used to initiate the assembly of a square below R and alongside the trigger, so that the trigger can propagate both dimensions of R to R_1.

Square Tile Sets. Given a seed row (column), Rothemund et al. [RW00b] show how to assemble a square abutting that row (column) using four tiles only. We use their method to assemble a square below R; the bottom row of R will serve as the seed row. Label the four tiles A, B, F and G, as in Fig. 3b. With the help of tile B, tile A initiates a new row by its strength-2 side, which gets filled with F and G tiles: F tiles fill the gaps left of A, and G tiles fill the gaps to the right of B. The result is shown in Fig. 3c.

Later in our construction, we will be facing the task of assembling a square abutting a given column, in which case we will use a similar square tile set depicted in Fig. 3f.

Trigger Tile Set. The set of trigger tiles is depicted in Fig. 3d. The trigger begins with the tile I_1 bonding by its strength-2 side to the upper right tile of R, then continues with I_2 downwards alongside R and with I_3 alongside the square below. The transition between the two dimensions of R is marked by I_3. A special tile I_4 marks the bottom end of the trigger; its particular role will become clear later.

The presence of the tile I_1 triggers a change in the counter state from P(ropagate) to I(ncrement), thus enabling the counter to grow a new column representing the current counter value increased by one. The assembly of R_1 could start at this point. However, for reasons to be discussed later, we choose to grow the new counter

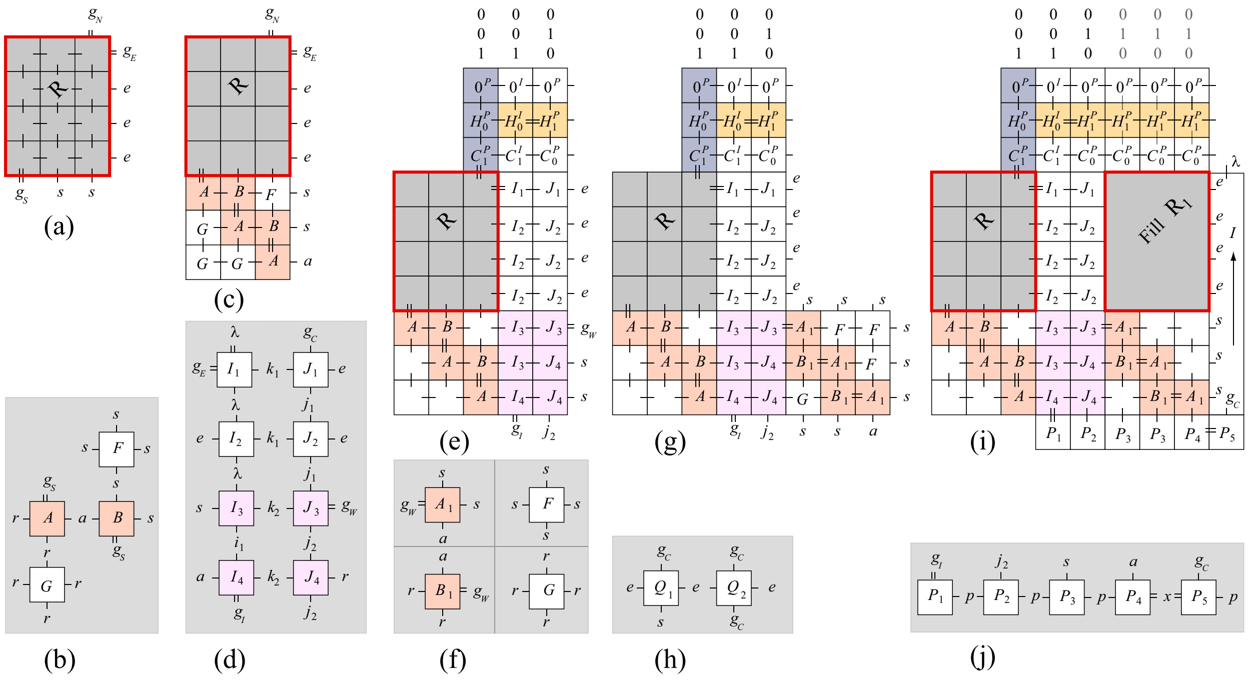


Figure 3: Tile system for rectangle replication. (a) Input rectangle R. (b) Square tiles (first set). (c) Square assembly below R. (d) Trigger tile set. (e) Trigger assembly to the right of R. (f) Square tiles (second set). (g) Square assembly to the right of trigger. (h) Rectangle tile set. (i) Rectangle assembly and progress line growth. (j) Progress tile set.

column downwards, until it reaches the bottom of the trigger (see Fig. 3e). The growth begins with the tile J_1 from Fig. 3d, then continues with J_2 alongside the I_2 tiles, marks the transition between the rectangle and the square below with a special tile J_3, then continues with J_4 until it reaches the bottom of the square. The tile J_3 is subsequently used to play the role of the lower left corner tile of R.Tiles A_1, B_1, F and G (depicted in Fig. 3f) cooperate in assembling a replica of the square below R, column by column; the result is shown in Fig. 3g.

Rectangle Tile Set. Two more tiles (labeled Q_1 and Q_2 in Fig. 3h) are needed to assemble R_1. An important characteristic of R_1 is that it has glue g_C on its top unattached sides, thus enabling the counter to propagate in its P-state alongside the length of R_1. Once the counter reaches the upper right corner of R_1 however, the assembly system halts; no nooks exist and, with the exception of I_4, unattached sides have bond strength 1.

Progress Tile Set. To enable further progress of the system, we need to initiate the assembly of a new trigger column to the right of R_1, whose task is to activates the counter. To this purpose, we design a set of "progress" tiles that assemble into a line segment running alongside the bottom of R_1.

The progress tiles are depicted in Fig. 3j. The strength-2 side of I_4 bonds to P_1, which serves as the seed tile for the line segment; P_1 then bonds to P_2, which continues with P_3 alongside the bottom of R_1, until it reaches the lower right corner tile A_1 of R_1; a special tile P_4 bonds to A_1 by its strength-1 top side, and to the end tile P_5 by its strength-2 right side. Refer to Fig. 3i. The tile P_5, which marks the end of the progress line, serves as the seed for a new trigger column. From this point on, the assembly process continues as described here and in section 3.1.4, until precisely n copies of R have been assembled.

3.1.4 Rectangle Replication Process.

Let k be the smallest integer such that $n < 2^k$, and let $\delta = 2^k - n - 1$. The value δ serves as the initial value for our counter. We start by attaching to the upper right corner tile of R a column of tiles representing δ in binary. This is the seed for our binary counter, which grows into a rectangle from left to right.

In the following we summarize the main steps of the self-assembly process that results in a new replica of R to be assembled. Consider a moment in time when a new trigger column composed of I-tiles is enabled to grow. This stage is initiated by the upper right corner tile in the original R for the first replica, and by the progress tile P_5 for subsequent (if any) replicas. In the first case, the trigger column grows downwards; in the latter case, the trigger column grows upwards, starting with I_3, which bonds to P_5. In either case, the following self-assembly

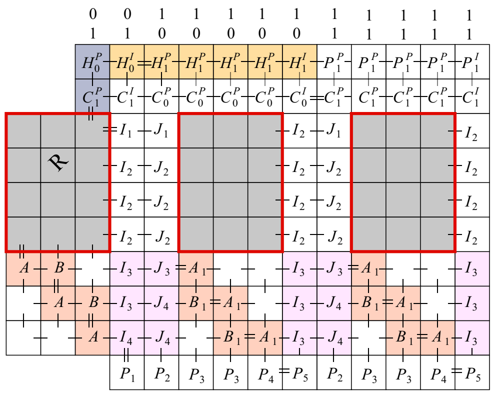


Figure 4: Rectangle replication example ($n = 2$).

steps succeed the growth of the trigger column:

1. The top I-tile of the trigger forces a change in the current state of the counter from P to I. If the counter has reached the maximum value of $2^k - 1$, the assembly process stops. Otherwise, a new counter column encoding the incremented counter value gets assembled.

2. The new counter column continues with J-tiles downwards, until it reaches the bottom of the I-tile column.

3. Next, the following four assembly processes happen somewhat in parallel: (i) The special tile J_3 initiates the assembly of a full square to the right and below J_3. (ii) A new rectangle replica is assembled on top of the square. (iii) The counter propagates its current value along the top of the rectangle replica. (iv) The progress line runs alongside the bottom of the rectangle replica, at the end of which it initiates the growth of a new trigger column. At this point, a new I-tile column starts growing and the process repeats.

As a running example, Fig. 4 illustrates a self-assembly system that creates two replicas of a given rectangle R. So we have the following result.

THEOREM 3.1. *For any given rectangle R and positive integer $n > 0$, R can be replicated exactly n times using $O(\log n)$ tiles in one stage only.*

3.2 Arbitrary Genus-0 Polygons. This section is concerned with assembling exactly n copies of a given tile structure, P, of genus-0 and some minimum feature size. We define the *feature size* of P as follows. Let $d_\infty(a, b) = \max(|a_x - b_x|, |a_y - b_y|)$ be the distance between two points a and b using the L_∞ metric. Then the feature size of P is the minimum $d_\infty(a, b)$ value such that a, b are points on two non-adjacent edges of P.

For ease of presentation, we assume that $n = 2^k$, for some integer value $k \geq 1$; we will later discuss how this assumption can be eliminated. Our construction is based on the following assumptions regarding P: (a) All glues on the boundary of P have a bond strength of 1 and are of the following types: northern tile edges have glue n southern edges have glue s, eastern edges have glue e, and western edges have glue w. Refer to Figure 6a. (b) The feature size of P is 5. The need for this requirement will become clear shortly. (c) P is non-rectangular (i.e., it has at least one reflex vertex.

The replication of P consists of two phases. In a first phase, we construct a frame for P, whose inner boundary follows the dents and turns of P. The frame consists of two layers, one RNA layer hugging P and one DNA layer hugging the RNA layer. The RNA layer gets disintegrated in a $BREAK$ operation, leaving the DNA frame.

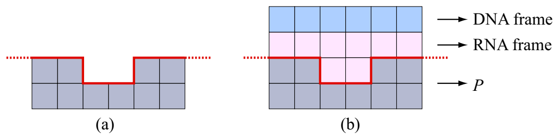


Figure 5: (a) Dent two units wide in P. (b) Dent leveled off in the interior of DNA[RNA[P]].

Observe that with too low a feature size, features of P may be smoothed over in the DNA frame. For example, the dent two units wide in Fig. 5a is leveled off in the surrounding DNA frame (as shown in Fig 5b), making it impossible to recreate the shape of P using the shape of the frame only. To prevent this, we require the RNA layer surrounding P to have a minimum feature size of 3, which means requiring a minimum feature size of 5 for P since the feature size can decrease by at most two when the RNA layer is added.

In a second phase, we use the DNA frame to create a copy of P. We do this by adding two layers of RNA tiles inside the DNA frame, to construct a frame for a smaller version of P, which we refer to as P_1. We then completely fill in this frame with DNA tiles to form P_1. A $BREAK$ operation destroys the two surrounding RNA layers, separating P_1 from its frame. We then add a DNA layer around P_1, turning it into a copy of P. As a last step, we line the interior of the DNA frame with DNA tiles to prevent it from participating in subsequent iterations of the replication process. By deactivating the used frame, we ensure that the number of copies exactly doubles in each iteration. An outline of these assembly steps is given in Table 1. A running example, is given in Fig. 6a.

3.2.1 Tile Set S_0 for RNA[P]. We start by defining the tile set S_0 used in assembling RNA[P], which is composed of P and an RNA-layer surrounding P. The

General Replication Process(P)

Phase 1: ADD tiles to construct a DNA frame for P:

1.1 Surround P with an RNA-layer. Call the result RNA[P]. (See Fig. 6b.)

1.2 Surround RNA[P] with a DNA-layer. Call the result DNA[RNA[P]]. (See Fig. 6c.)

1.3 Destroy all RNA tiles in a $BREAK$ operation. This step separates the outer DNA frame and P from DNA[RNA[P]]. (See the outer DNA frame in Fig. 6d.)

Phase 2: ADD tiles to fill in the DNA frame to create a copy of P:

2.1 Assemble an inner RNA-layer alongside the inner boundary of the DNA frame. Call the result DNA[RNA]. (See Fig. 7a.) (Note that the glues on the inner boundary of DNA[RNA] cannot match the glues on the outer boundary of P, because if the same glues were used, then the original P (which is floating in the tile soup) may bond to the interior of DNA[RNA] and fill the frame. It is because of this that we add a second RNA layer instead of immediately creating a copy of P using the DNA[RNA] frame.)

2.2 Assemble a second inner RNA-layer alongside the inner boundary of DNA[RNA]. Call the result DNA[RNA[RNA]]. (See Fig 7b.)

2.3 Fill in the interior of DNA[RNA[RNA]] with DNA tiles. Call the result DNA[RNA[RNA[P_1]]].

2.4 Destroy all RNA tiles in a $BREAK$ operation. This separates the outer DNA frame and P_1 from DNA[RNA[RNA[P_1]]]. (See Fig. 7c.)

2.5 Add an outer layer of DNA tiles around the boundary of P_1, thus turning P_1 into a replica of P. (See Fig. 7d.)

2.6 Deactivate the DNA frame by lining it with DNA tiles. (This step is designed so that the deactivation layer can never form on new DNA frames resulted from step 1.3 above.)

Table 1: Main steps of the replication process.

tiles in S_0 are divided into four subsets, depending on the role they play in the assembly of the RNA-layer: convex corner supertiles, reflex corner supertiles, horizontal edge tiles and vertical edge tiles.

Convex Corner Supertiles. S_0 contains four convex supertiles, one for each type of convex corner: north-west (C_{nw}), north-east (C_{ne}), south-east (C_{se}) and south-west (C_{sw}). The corner tile of each supertile has one of its outer edges marked with a special glue: α_1 for C_{nw}, α_2 for C_{ne}, α_3 for C_{se} and α_4 for C_{sw} (see Figure 8a).

Reflex Corner Supertiles. S_0 contains four reflex supertiles, one for each type of reflex corner: south-east (R_{se}), south-west (R_{sw}), north-west (R_{nw}) and east-north (R_{en}). Each reflex supertile has one of its inner

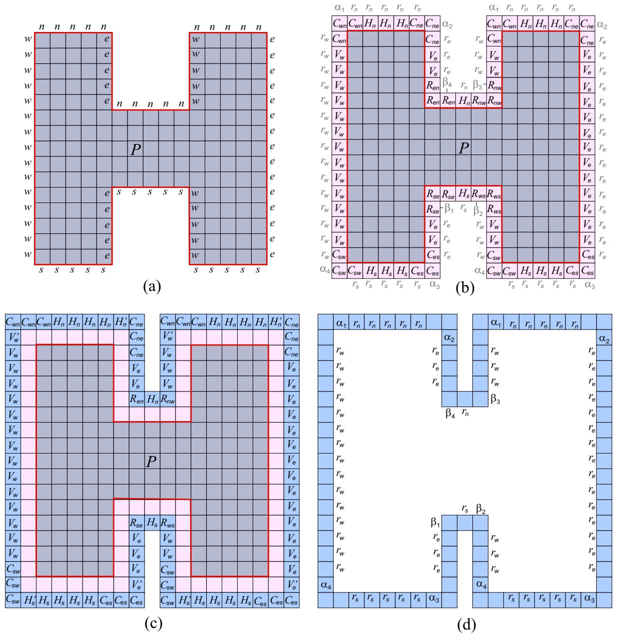


Figure 6: Phase 1: (a) P (b) RNA[P] (c) DNA[RNA[P]] (d) DNA Frame.

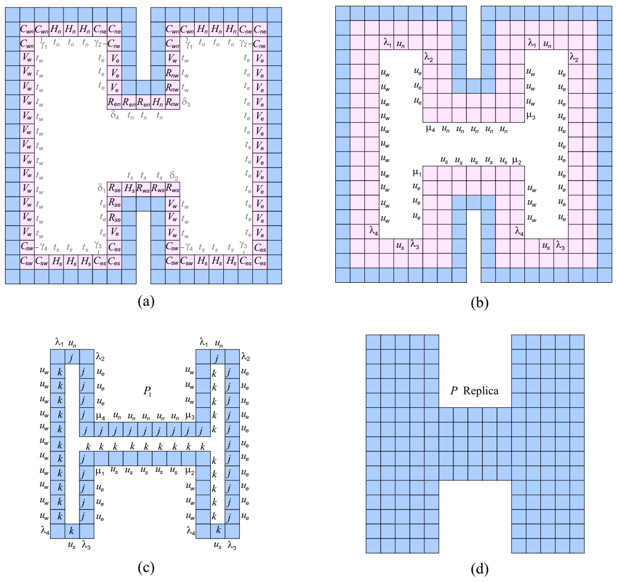


Figure 7: Phase 2: (a) DNA[RNA] (b) DNA[RNA[RNA]] (c) P_1, smaller replica of P (d) Exact copy of P.

tile edges marked with a special glue: β_1 for R_{se}, β_2 for R_{sw}, β_3 for R_{nw} and β_4 for R_{ne} (see Figure 8b). These special glues will mark the reflex corners of RNA[P] and facilitate the assembly of the surrounding DNA frame.

Horizontal Edge Tiles. S_0 contains two horizontal edge tiles, H_n and H_s, used to assemble horizontal runs alongside horizontal edges of P. Tiles H_n have south

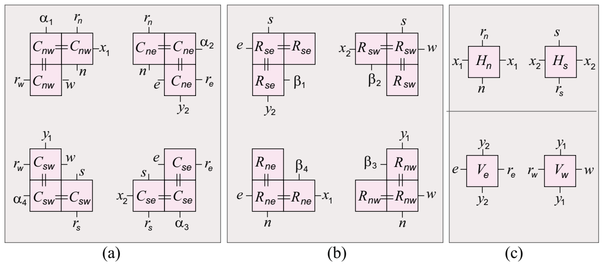


Figure 8: Tile set S_0 for RNA[P]. (a) Convex (C) corner tiles. (b) Reflex (R) corner tiles. (c) Horizontal (H) and vertical (V) tiles.

glue n and thus can only bond to north edges of P. Similarly, tiles H_s have north glue s and thus can only bond to south edges of P.

Vertical Edge Tiles. S_0 contains two vertical edge tiles, V_e and V_w, used to assemble vertical runs alongside vertical edges of P. Tiles V_e have west glue e and thus can only bond to east sides of P. Similarly, tiles V_w have east glue w and thus can only bond to west sides of P. Tiles $H, V \in S_0$ are depicted in Figure 8c. Note our convention that glue subscripts must match the orientation of the edges' outward-facing normals.

The surrounding RNA layer is assembled in two stages.

Stage 1: $\boldsymbol{ADD(C_i, R_i \in S_0)}$, with $i \in \{nw, ne, se, sw\}$. In the first stage, corner tiles are added to the mixture, and they attach to P at its convex and reflex corners. For example, the group labeled C_{nw} in Fig. 8a will bond to a northwest convex corner of P where w and n glues meet. The four reflex corner tiles in Fig. 8b each bond to one of the four types of reflex corners. For example, R_{se} will bond at a reflex corner where s and e glues meet.

Note that the requirement that the feature size of P be at least 5 guarantees enough room for corner supertiles, and ensures that no two corner supertiles are adjacent.

Stage 2: $\boldsymbol{ADD(H_i, V_j \in S_0)}$, $i \in \{n, s\}$ and $j \in \{e, w\}$. In the second stage, the horizontal (H) and vertical (V) tiles (depicted in Fig. 8c) are added to the mixture. From the corner tiles, the H and V tiles incrementally bond clockwise along the adjacent horizontal and vertical edges of P until the layer is complete. The clockwise assembly of the layer is enforced by the lack of glue on some tile edges. For instance, the lowest south tile edge of a C_{nw} supertile has no glue; the natural glue to place there is y_1, but doing so creates a situation where the C_{nw} supertile may bond in places where V_w is supposed to bond, using the combined strength of the r_w and the y_1 glues. Similar ideas hold for each of the other (convex) corner tiles. For a specific example of RNA[P] layer, with all RNA layer tiles labeled, see Fig. 6b.

Observe that the pattern of r glues on the outer boundary of RNA[P] is analogous to the pattern of glues on P, with the only exception at the corners of of RNA[P]. At each corner c of RNA[P], the incident tile edge clockwise from c is marked with a special glue (α or β); the other incident tile edge has the null glue. These special glues will help create and fill in the DNA frame in subsequent stages.

3.2.2 Tile Set S_1 for DNA[RNA[P]].

RNA tiles are destroyed through a $BREAK$ operation during each iteration of the replication process, and thus may be reintroduced to the mixture in subsequent replication iterations (see the algorithm description from Table 1). Unlike RNA tiles however, once DNA tiles are added to the mixture, they persist throughout all subsequent iterations of the replication process. For this reason, the two stage technique used to assemble RNA[P] cannot be used to assemble a DNA layer surrounding RNA[P]. Instead, the tile set S_1 shown in Fig. 9 is used, and the assembling of the DNA frame happens in one stage only: $ADD(S_1)$.

Similar to S_0, the tile set S_1 consists of four subsets: convex turn (C) supertiles, reflex turn (R) supertiles, horizontal edge (H) tiles and vertical edge (V) tiles. Unlike S_0 however, the tile set S_1 contains singleton reflex tiles in place of the reflex supertiles in S_0. The reason for this is twofold. First, there is no need to mark reflex corners of the DNA frame; subsequent stages will only attempt to fill in the DNA frame, and therefore glues on the outer boundary of the DNA frame are irrelevant to our replication process. Second, the reflex tiles are added to the mixture at the same time as H and V tiles, and therefore they must be designed so that no races to bond to one same tile occur. See Fig. 6(c,d) for an example of DNA frame.

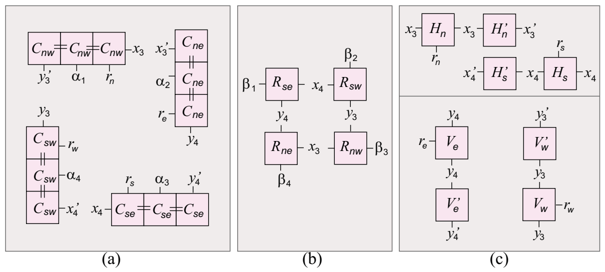


Figure 9: Tile set S_1 for DNA[RNA[P]]. (a) Horizontal (H) and vertical (V) tiles. (b) Convex (C) corner tiles. (c) Reflex (R) corner tiles.

We say that a layer of tiles satisfies the *bond-cycle property* if each tile in the layer is bonded to the tile that comes before and after it. It is important for our DNA frame to satisfy the bond-cycle property, since it must stand alone at temperature 2 after it gets separated from the lining RNA layer. Observe that each R tile from Figure 9b has one side with the null glue. The reason for this is to guarantee that R tiles do not bond to reflex corners of the RNA layer prematurely, which in turn will guarantee the bond-cycle property for the DNA layer. Also note the usage of the primed tiles $H', V' \in S_1$, intended to attach to convex supertiles at convex corners of RNA[P]. Their role is to ensure a null glue on the incident tile edge counterclockwise to each convex vertex of RNA[P], a property to be exploited in subsequent assembly steps. We now state without proof the following result:

LEMMA 3.1. *Any given shape A with feature size 3 and pattern of outer glues as in Figure 6b, can be surrounded at temperature 2 by a DNA layer that satisfies the bond-cycle property, using $O(1)$ tiles.*

Proof. The feature size of 3 ensures room for a layer surrounding A that preserves the shape of its boundary. The tile set used to assemble the DNA layer is depicted in Fig. 9.

The assembly starts when at least one convex corner supertile T bonds with strength 2 to a convex corner of A along the α, r glues. For ease of presentation, assume that $T = C_{nw} \in S_1$ bonds to a convex corner a of A. This event creates the conditions for H_n tiles to assemble incrementally alongside the horizontal edge of A incident to a (which extends clockwise from C_{nw}). The clockwise assembly direction is enforced by the lack of a glue on the vertical tile edge incident to a (see, for example, the top left convex corner in Figure 6b).

Let (a, b) be the horizontal edge of A incident to a. A run of H_n tiles assembles incrementally alongside the top of (a, b), using the combined strength of the x_3 and r_n glues, until the next corner b is reached. Two situations are possible:

1. b is a convex corner. Note that the horizontal tile edge incident to b has no glue; this forces the run of H_n tiles to stall if no convex supertile C_{ne} has yet bonded to the vertical edge incident to b. Once a C_{ne} bonds (or if a C_{ne} tile has bonded earlier), a tile red H'_n connects the horizontal run to C_{ne} using the combined strength of the opposing x_3, x'_3 glues.

2. b is a reflex corner. As in the case of convex corners, the horizontal tile edge incident to b has no glue, thus preventing the horizontal run of H_n tiles from extending further. The only tile that can bond in that position is a reflex turn tile – in our context, R_{nw}, along the x_3 and β_3 glues.

Observe that this layer forms clockwise starting from each convex corner. Each H, V and R tile bonds to the tile before it. The last H' or V' tile connecting a horizontal or vertical run to the next convex corner supertile bonds to both adjacent (super)tiles. It follows that the DNA layer satisfies the bond-cycle property, which allows it to exist independently.

We now show that, during the assembly of the DNA layer, a tile attaches to the current structure only in the conditions described above. In other words, C tiles only attach at convex corners, R tiles at reflex corners, and H and V tiles fill the gaps in between corner tiles.

First observe that H, V and R tiles share no pair of glues that would enable one to play the role of another. Similarly for R and C tiles. However, (super)tiles C_{nw} and H_n have corresponding glues (r_n, right x_3), which may potentially allow a C_{nw} tile to bond in places where H_n is intended to bond. We will show that this is not the case. Due to the clockwise nature of the assembly process, an H_n tile bonds using either of the glue pairs (left x_3, r_n) and (left x_3, right x_3). The glue pair (r_n, right x_3) is never used in bonding H_n to the DNA layer, since bonding with these two glues would imply that the layer forms in a counterclockwise direction. This guarantees that no conflict occurs between an H_n tile and a C_{nw} tile.

The only other pairs of glues on C_{nw} that may potentially bond it incorrectly during the forming of the DNA layer are (y_3, x_3) and (y_3, r_n). We show by contradiction that a (y_3, x_3) bonding is not possible; similar arguments hold for a (y_3, r_n) bonding. Assume without loss of generality that the *first* convex supertile that bonds in the wrong place is C_{nw} (otherwise the discussion would move symmetrically to another convex supertile). Consider a tile T with north glue y_3 that contributes to the bonding of C_{nw} along glues y_3 and x_3. Then the three tile positions immediately on top and to the right of T must be empty, so that C_{nw} can fit in. Looking at Figure 9, notice that the only candidates for T are C_{sw}, R_{nw} and V_w. We now consider each of these situations in turn:

1. $T = C_{sw}$ or $T = R_{nw}$. By our assumption that C_{nw} is the first supertile misplaced, T must have correctly bonded at a corner of RNA[P]. This along with the feature size of 3 guarantees that T bonded to a west edge of RNA[P] that extends at least one tile edge above T. Thus the tile space immediately above and to the right of T contains a tile of RNA[P], so C_{nw} cannot bond in that space.

2. $T = V_w$. The lack of glue on each tile edge counterclockwise to a north-west corner of RNA[P] guarantees that V_w does not touch a convex corner of A, at the time C_{nw} attempts to bond on top of T. But this means that the tile space immediately above and to the right of T contains a tile of RNA[P], so

again C_{nw} cannot bond in that space.

By symmetry, these ideas apply to all other convex supertiles, showing that the tile set S_1 properly constructs the DNA layer. □

3.2.3 Tile Set S_2 for DNA[RNA]. The tile set S_2 used in assembling an inner RNA layer is depicted in Fig. 10. We begin by defining some terminology for the inner boundary of a layer. We call a corner c on the inner boundary *reflex* if the angle at c interior to the region surrounded by the boundary exceeds π; otherwise, the corner is *convex*. We say that an inner reflex corner is a *north-west* corner if its two perpendicular tile edges have normals pointing into the layer that are oriented north and west. Thus, each north-west reflex corner on a layer's outer boundary corresponds to a north-west reflex corner on its inner boundary. Glue subscripts n, s, e, and w on the inner boundary tile edges refer to the direction of the edges' normals that point into the layer.

Recall that the clockwise tile edge incident to each inner convex and reflex corner of the DNA frame is marked with a special glue (α or β, as in Fig. 6d). The tile set S_2 is designed so that the assembled inner layer DNA[RNA] preserves this property: clockwise tile edges incident to inner corners of DNA[RNA] are marked with special glues. This will ensure that the pattern of glues along the inner boundary of DNA[RNA] is analogous to the pattern of glues along the inner boundary of the DNA frame, so that a second inner RNA layer can be built using a tile set similar to S_2. Observe from Fig. 10 that special glues α_i and β_i on the inner boundary of the DNA frame correspond to special glues γ_i and δ_i on the inner boundary of DNA[RNA].

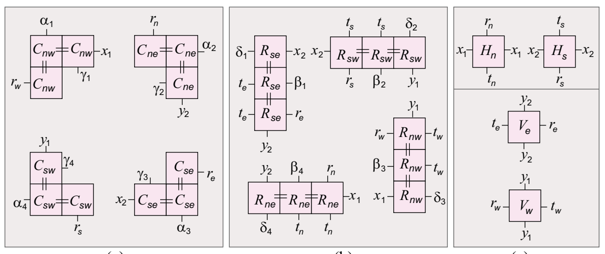


Figure 10: Tile set S_2 for DNA[RNA]. (a) Convex (C) corner tiles. (b) Reflex (R) corner tiles. (c) Horizontal (H) and vertical (V) tiles.

Similar to the assembly of an outer RNA layer, an inner RNA layer is assembled in two stages.

Stage 1: $\boldsymbol{ADD(C_i, R_i \in S_2)}$, with $i \in \{nw, ne, se, sw\}$. In a first stage, corner supertiles bond to convex and reflex interior corners.

Stage 2: $\boldsymbol{ADD(H_i, V_j \in S_2)}$, with $i \in \{n, s\}$ and $j \in \{e, w\}$. In a second stage, horizontal (H) and vertical (V) tiles are added to the mixture to fill in the gaps between corners of DNA[RNA].

Figure 7a shows an example of DNA[RNA] layer with labeled tiles, corresponding to the DNA frame from Figure 6d. So we have the following result.

LEMMA 3.2. *Any given shape A with feature size 3 and pattern of inner glues as in Figure 6d, can be lined at temperature 2 by an RNA layer, using $O(1)$ tiles.*

3.2.4 Tile Set S_2B for DNA[RNA[RNA]]. The structure DNA[RNA[RNA]] is formed by assembling a second inner RNA layer lining DNA[RNA]. The tile set for this second inner RNA layer can be generated from the tile set S_2 for the first RNA layer (depicted in Fig. 10) by replacing glue labels r_n, r_s, r_e, r_w by t_n, t_s, t_e, t_w respectively, t_n, t_s, t_e, t_w by u_n, u_s, u_e, u_w respectively, and special glues $\alpha, \beta, \gamma, \delta$ by $\gamma, \delta, \lambda, \mu$ respectively (with appropriate subscripts). Since the same tile structure (with different glues) is used in assembling both DNA[RNA] and DNA[RNA[RNA]], we refer to the tile set used in assembling DNA[RNA] as S_2A, and the tile set used in assembling DNA[RNA[RNA]] as S_2B. For a quick reference, we summarize the glues on the inner and outer boundaries of various layers in Table 2.

Layer	Tile Template	Glues on Inner Boundary	Glues on Outer Boundary
Outer RNA layer	S_0 (Fig. 8)	n, s, w, e	$r_n, r_s, r_w, r_e,$ $\alpha_1, \alpha_2, \alpha_3, \alpha_4,$ $\beta_1, \beta_2, \beta_3, \beta_4$
DNA frame	S_1A (Fig. 9)	$r_n, r_s, r_w, r_e,$ $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ $\beta_1, \beta_2, \beta_3, \beta_4$	null
First inner RNA layer	S_2A (Fig. 10)	$t_n, t_s, t_w, t_e,$ $\gamma_1, \gamma_2, \gamma_3, \gamma_4,$ $\delta_1, \delta_2, \delta_3, \delta_4$	$r_n, r_s, r_w, r_e,$ $\alpha_1, \alpha_2, \alpha_3, \alpha_4,$ $\beta_1, \beta_2, \beta_3, \beta_4$
Second inner RNA layer	S_2B (Fig. 10)	$u_n, u_s, u_w, u_e,$ $\lambda_1, \lambda_2, \lambda_3, \lambda_4,$ $\mu_1, \mu_2, \mu_3, \mu_4$	$t_n, t_s, t_w, t_e,$ $\gamma_1, \gamma_2, \gamma_3, \gamma_4,$ $\delta_1, \delta_2, \delta_3, \delta_4$
P_1's outer layer	S_3A (Fig. 11)	j, k	$u_n, u_s, u_w, u_e,$ $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ $\mu_1, \mu_2, \mu_3, \mu_4$
P's outer layer	S_1B (Fig. 9)	$u_n, u_s, u_w, u_e,$ $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ $\mu_1, \mu_2, \mu_3, \mu_4$	n, s, w, e
DNA frame deactivation layer	S_3B (Fig. 11)	null	$r_n, r_s, r_w, r_e,$ $\alpha_1, \alpha_2, \alpha_3, \alpha_4,$ $\beta_1, \beta_2, \beta_3, \beta_4$

Table 2: Summary of inner and outer boundary glues.

3.2.5 Tile Set S_3 for DNA[RNA[RNA[P_1]]]. To simplify the process of assembling P_1, we first assemble the outermost layer of P_1 by adding it as an inner layer of DNA[RNA[RNA]], and then we fill it in with DNA tiles.

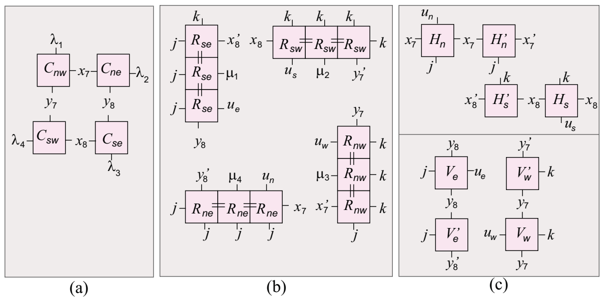


Figure 11: Tile set S_3 for DNA[RNA[RNA[P_1]]]. (a) Horizontal (H) and vertical (V) tiles. (b) Convex (C) corner tiles. (c) Reflex (R) corner tiles.

The tile set S_3 used in assembling the outer DNA layer for P_1 is similar to the tile set S_1 used in assembling the DNA frame, but with the roles of C and R tiles switched (since notches are formed at convex corners of the inner boundary of DNA[RNA[RNA]], as opposed to reflex corners of RNA[P]). The tile set S_3 used in assembling the outer DNA layer of P_1 is depicted in Fig. 11 (compare this set with the one depicted in Fig. 9). Note that the inner boundary of this DNA layer has two types of glue only, j and k; glue j is on north and east tile edges, and glue k is on south and west tile edges. This simple pattern makes it easy to fill the interior of this layer using two DNA tiles only, one with glue j and a second one with glue k on all its sides. The following lemma follows immediately from Lemma 3.1 and the similarity between the tile sets S_1 and S_3.

LEMMA 3.3. *The outer layer of P_1 satisfies the bond-cycle property.*

Note that the interior of P_1 has at least one corner, until it is completely filled in. We call the resulting structure DNA[RNA[RNA[P_1]]], to indicate its actual composition: the DNA frame, two inner RNA layers and P_1.

3.2.6 Completing the Replica of *P*.

In the last step of phase two, we dissolve the RNA tiles in a $BREAK$ operation, thus separating the DNA frame from P_1. The following lemma shows that P_1 can exist independently in the solution.

LEMMA 3.4. *P_1 is stable at temperature* 2.

Proof. By Lemma 3.3, the outer layer of P_1 satisfies the bond cycle property. We now show inductively that each tile in the interior of P_1 is connected by an orthogonal path of tiles to the outer layer of P_1, with each bond on the path of strength 2. Then the lemma follows.

For simplicity, let ∂P_1 refer to the outer layer of P_1. We assume that time is divided into rounds, and in each round one or more tiles bond to corners of the existing structure. The induction is on the number of rounds it takes to fill in ∂P_1.

The base case corresponds to round 1. In this round, a tile f bonds with strength 2 to a notch formed by two tiles $f_1, f_2 \in \partial P_1$. Then (f, f_1) is a path of length 1 and bond strength 2 connecting f to ∂P_1, and similarly for (f, f_2).

Assume that the argument holds for all rounds up to $i \geq 1$, and consider round $i+1$. The argument for this case is similar to the one for the base case: in round $i+1$, a tile f can only bond to a notch formed by two tiles f_1 and f_2, which must have bonded to the existing structure in round $j < i$. By the inductive hypothesis, there is a path p of bond strength 2 connecting f_1 to ∂P_1. Then $p \oplus f$ is a path of bond strength 2 connecting f to ∂P_1; here $\oplus$ is used to denote the concatenation operator. □

Note that the tiles in the interior of P_1 may not be bonded to each other on all four sides. Nevertheless, Lemma 3.4 guarantees that P_1 is stable at temperature 2. This property enables us to add an outer layer to P_1, turning it into a copy of P. To do so, first observe that the pattern of outer boundary glues on P_1 is identical to the pattern of outer boundary glues on RNA[P] (see Fig. 7c vs. Fig. 6b). Thus, in assembling a DNA layer around P_1, we can use the same tile set S_1 used earlier to assemble a DNA layer around RNA[P], with different types of glue: (a) replace glues r, α, β by u, λ, μ respectively (with appropriate subscripts), and (b) add glues n, s, e, w to the outer boundary of the new layer. We denote this tile set by S_1B in Table 2.

3.2.7 Deactivating the DNA Frame.

This final step is necessary to prevent the current set of DNA frames from making additional copies in subsequent iterations of the replication process. We deactivate each DNA frame by lining it with a layer of DNA tiles. The tile set used in assembling this inner DNA layer is identical to the tile set S_3 from Fig 11 (used in assembling the outer layer of P_1), after substituting glues u, λ, μ by r, α, β respectively (with appropriate subscripts), j, k by null, null, and x, y by new unique glues. This ensures that no glue appears on the inner boundary of the deactivation layer, and that H and V tiles in this tile set cannot play the role of H and V tiles used in any other tile set. In addition, we use RNA tiles as corner tiles and DNA tiles for all other tiles in the set, to enable the disintegration of all corner tiles in a subsequent $BREAK$ operation (after the inner layer is complete). By destroying all corner tiles, we ensure that new frames created in the next iteration of the replication process do not get deactivated until the end of that iteration (recall that the assembly of the deactivation layer is initiated by corner tiles).

3.2.8 Main Result. So far we have assumed that the input shape is non-rectangular (i.e., it has at least one reflex vertex). However, our tile set can be easily extended to handle rectangular shapes as well, at the cost of a few additional tiles.

Throughout this section we discussed the special case when n is a power of 2. We now show how to handle the general case when n is not a power of 2, using a variant of the double-and-add method. The double-and-add method converts a non-zero binary string $b = b_{\lfloor \log n \rfloor + 1} \dots b_1$ to base ten as follows. Start with an integer t whose value is initially 1; at the end of the method, t will have value equal to the base ten equivalent of b. For $i = \lfloor \log n \rfloor \dots 1$, double the value of t and then add b_i (a zero or a one) to t.

We use the same idea to create $n > 0$ copies of P. Start with a soup containing a single copy of P. Let $b = b_{\lfloor \log n \rfloor + 1} \dots b_1$ be the base two representation of n. For $i = \lfloor \log n \rfloor \dots 1$, we double the number of copies of P in the soup using the assembly process described in Table 1; if b_i is one, we add one additional copy of P to the soup. Thus we have the following result.

THEOREM 3.2. *For any given genus-0 polygonal tile structure, P, with feature size of 5 and positive integer $n > 0$, P can be replicated exactly n times using $O(1)$ tiles in $O(\log n)$ stages.*

4 Infinite Yield Replication

For the results that follow, we first show how to infinitely replicate x-monotone shapes with a feature size of at least 4 using $O(1)$ tile types and $O(1)$ stages. We then consider the more general problem of infinite replication of arbitrary genus 0 shapes. Our general construction is able to replicate any genus 0 shape with at least $\Omega(\log n)$ feature size in $O(1)$ tile types and $O(1)$ stages.

4.1 x-monotone Shapes. In this section we present our construction for the infinite replication of general vertically convex shapes. Our construction proves the following theorem:

THEOREM 4.1. *There exists a staged replication with RNA removal system with $O(1)$ tile type complexity and $O(1)$ stages that will infinitely replicate any vertically convex input shape whose faces have length at least 4.*

In the remainder of this section we present an assembly system that proves this theorem. For this construction, all RNA tile types are shaded pink.

Stages 1, 2 and 3. The tile sets added in the first 3 stages of the assembly process are given in Figure 12. The tiles added in stage 1 are all of type RNA and cover the convex and concave corners of the input shape. In stage 2, the four pink tiles in the left of Figure 12 finish a single layer coat of RNA tiles surrounding the surface of the input polygon. The blue DNA tiles and the remaining pink RNA tile create a rectangular encasing of the input shape, as shown in Figure 13. The rectangular encasing of the input shape is made of DNA tiles with the exception of the RNA tile used to fill the vertical columns corresponding to the edges of the faces of the input polygon.

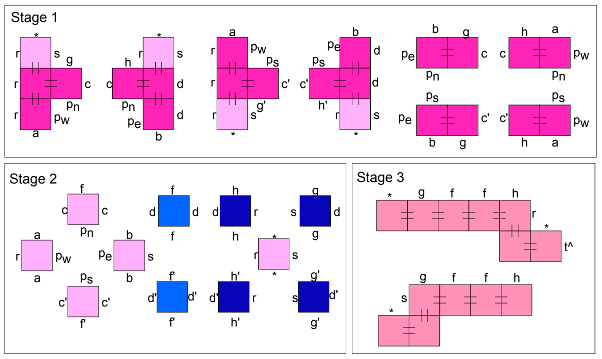


Figure 12: The above tile sets are added in stages 1, 2, and 3. The six blue tile types are of type DNA, and the rest are of type RNA.

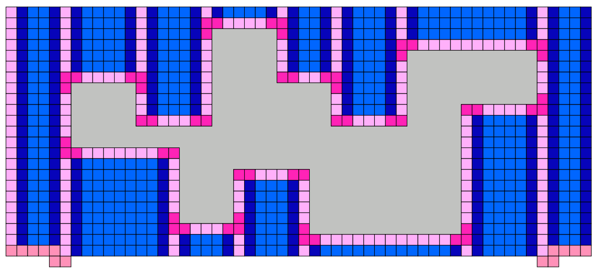


Figure 13: After the first 3 stages, the input shape is surrounded by a rectangular encasing of tiles, insulated with a layer of RNA tiles.

Stages 4. In stage 4 the binary counting tiles from Figure 14 grow across the top and bottom surfaces of the rectangle assembly, incrementing the counter whenever a star glue is encountered, which occurs at the edge of each face of the polygon. Thus, the binary counter maintains a fixed and distinct value for each of the different faces of the input polygon. Unlike counters which must grow to a set length, this counter is controlled by the surface of the polygon, and thus can be implemented with only $O(1)$ tile complexity. The final width of the counter is $2\lceil \log f \rceil$ where f is the number of faces of the input polygon. Additional tiles are provided to even out the shape of the counter into a rectangle. A conceptually identical set of counter tiles as those shown in Figure 14 is applied to the bottom face of the assembly as well. In stage 5, the RNase enzyme is added to the assembly mixture. For our assembly, the RNA tile types surround the shape and divide the constructed columns of the filled in rectangle at the points at which the level of the input shapes face changes. The result, as shown in Figure 15, is the breaking of the assembly into a collection of rectangles, each crowned with a specific, distinct binary number. While these rectangles are no longer connected, in some sense they encode a description of the input shape as the length and width of the rectangles denote the length and depth of a face from the input shape. Our approach is to now infinitely replicate each of these rectangles, and then reassemble an infinite number of templates of the input shape for replication by relying on the binary "barcode" to help reassemble the rectangles in the proper order.

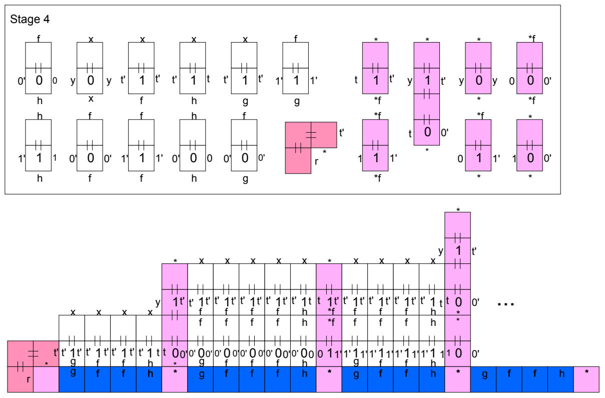


Figure 14: The above tile set constitutes a self growing set of binary counting tiles which increment once for each face of the input shape, denoted by the pink tiles with glue '*'. This tile set forms a counter on the north surface of the rectangular encasing of the input shape. A similar tile set is used to create a counter for the south side of the encasing as well. The x and y glues are used to allow 0 bits to fill all columns of the counter up to the height of the final column, thus making a rectangular shape.

Stages 6 and 7. In stage 6 we add a tileset designed to create an infinite number of copies of an input rectangle. The tile set is given in Figure 16, along with an example of the infinite pattern of the assembly in Figure 17. The tile set for this stage replicates any rectangle, and is specifically designed to copy the identifying binary strings for each rectangle as well. Thus, each of the rectangles for each of the faces of the input polygon causes the growth of an infinite supertile, each with infinitely many replica rectangles imbedded within. Stage 7 is an enzyme stage, which breaks each of the infinite supertiles into infinitely many copies of the binary rectangles.

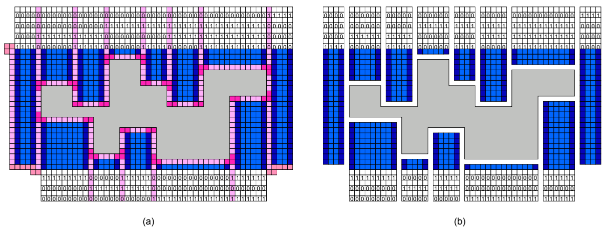


Figure 15: (a) After stage 4, the input shape is contained in a rectangle encasing with binary counters assembled across the top and bottom surfaces of the encasing, denoting the number of north and south faces of the shape respectively. (b) In stage 5, an RNase enzyme breaks the encasing into a collection of rectangles, each with a distinct binary *bar code*.

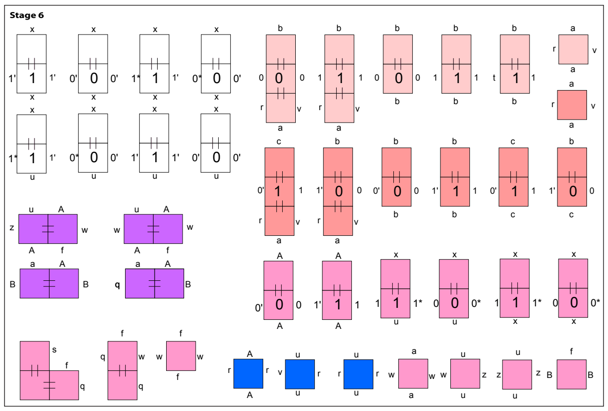


Figure 16: The above tileset is added in stage 6 of the construction. For each of the binary labeled rectangles, the above set will grow an infinite chain of copies of each rectangle, along with the correct binary label, with each rectangle insulated by a layer of RNA tile types. The 16 white tile types and 3 blue tile types are of type DNA, and the rest are of type RNA.

Stages 8, 9, and 10. Now that we have an infinite number of copies of each of the binary rectangles, the goal is to reassemble them all into infinitely many molds to replicate the original shape. The key is to get the rectangles to line up according to the order imposed by their binary labels. To accomplish this, we add the tiles from Figure 18. These tiles fill in the missing left and rightmost column of each binary rectangle, as these columns where intentionally removed by the previous enzyme stage. The leftmost column attaches a binary column that decrements the rectangles binary value by 1. The right column simply copies the value of the counter. Additionally, in the leftmost column of each binary label, for each bit of the counter a single tile juts out 1 extra column. For the '0' bits, this tile occurs one position

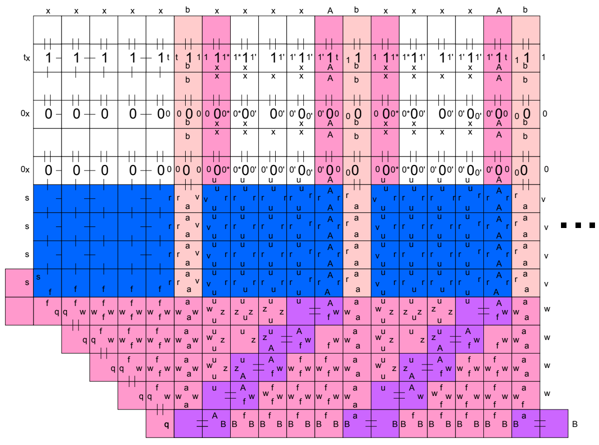


Figure 17: This is an example of the infinite replication of the binary labeled rectangles by the tile set from Figure 16 that occurs in stage 6. In stage 7, the RNase enzyme is added, breaking the infinite growth supertiles into infinite copies of each of the original binary labeled rectangles, modified such that the leftmost and rightmost binary columns are missing.

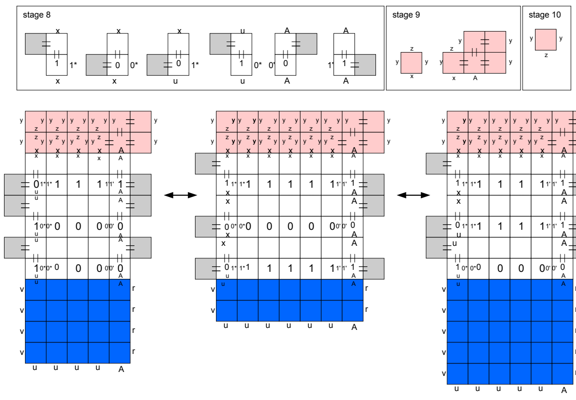


Figure 18: In stage 8, the above tile set is added to form a tooth like surface on the left and right side of each binary labeled rectangle. For each binary label, the tooth pattern is unique, ensuring that only the unique complement to the pattern will permit a rectangle to get close enough for attachment. The tiles added in stages 9 and 10 act as *toothpaste*, creating an affinity for binary labeled rectangles to attach at temperature 2, given that their teeth interlock correctly.

lower than for the '1' bits. A similar pattern occurs in the rightmost column, but the '0' bits have higher tiles than the '1' bits in contrast.

The effect of this tile growth is that each binary label now has a jagged set of teeth whose geometry uniquely identifies the blocks binary label. Further, each block has a left binary value that is one less that the right binary

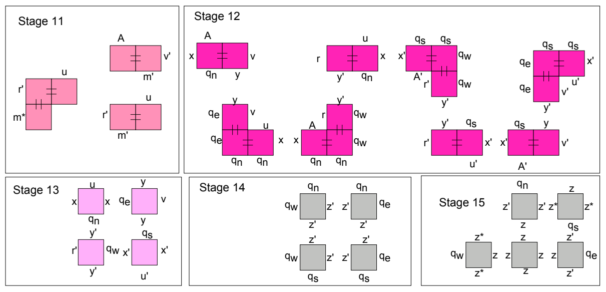


Figure 19: The tiles in stage 11 simply attach the top and bottom portions of the reassembled frame together. Once connected, the tiles in stage 12 are added to cover the internal corners of the assembled frame, followed by the stage 13 tiles to complete an inner coat of RNA tiles. The stage 14 and 15 tiles fill in the frame with DNA tiles to create a new copy of the original input shape. The glue types m' and $m*$ are not listed in the tile sets described as they are special glues that must occur on the teeth of binary labels from the bottom counters, whose tiles where not detailed.

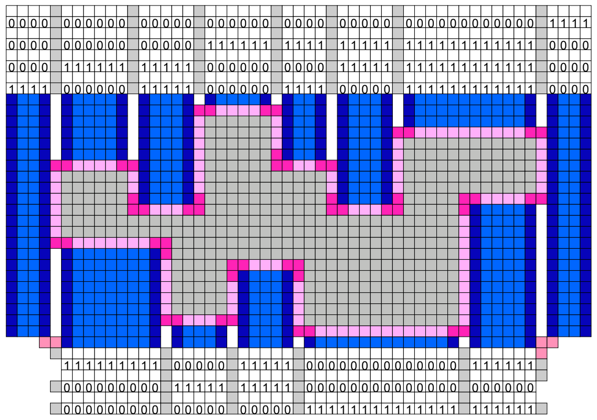


Figure 20: After stage 14 there are infinitely many assembled replicas of the input shape contained in frames as shown in this example. The final stage of the construction adds the RNase enzyme to release the infinitely many replicas of the input shape.

value. As the jagged pattern of teeth is opposite from left to right, the teeth of two binary blocks whose original rectangles where adjacent for the original assembly mold will have perfectly interlocking teeth. In contrast, any two blocks that do not represent binary numbers that are increments of one another are guaranteed to have at least one mismatching tooth, preventing the blocks from getting close to one another.

Now that we have the interlocking teeth in place, we add the tiles from Figure 18 in stages 10 and 11. These tiles simply attach glues to the binary rectangles giving any rectangle an affinity to attach adjacent to any other rectangle. However, due to the interlocking geometry of the previously attached teeth tiles, that affinity can only be realized by originally adjacent rectangles, ensuring that the unique, increasing binary counter order of block attachment is the only possible assembly. The final result is the assembly of an infinite number of copies of the original frame used to encase the input shape, broken into two pieces, a top and a bottom. All that remains is to connect the two pieces and fill in the infinite count molds.

Stages 11, 12, 13, 14, 15, and 16. Stage 11 adds the 7 RNA tiles from Figure 19 which reconnects the bottom and top halves of the reassembled frames of the input shape. Once these are assembled, stages 12 and 13 coat the inside of the assembled frame with a layer of RNA tiles. Stages 14 and 15 fill in the frame with DNA tile types. Finally, in the final stage 16 the RNase enzyme is added which breaks the infinitely many assembled replica shapes out of the assembled molds. An example of the final assembly after stage 15 is given in Figure 20.

4.2 Genus-0 Shapes.

THEOREM 4.2. *Any given genus-*0 *polygonal tile structure P, with n corners and feature size $\Omega(\log n)$, can be replicated infinitely using $O(1)$ tiles and $O(1)$ stages.*

The algorithm is complicated, with several high-level steps each consisting of several stages. The figures illustrate the simulated execution of some of the steps on a simple example, as computed automatically by a computer program. (Light color indicates RNA tiles; dark color indicates DNA tiles. Unlabeled edges denote the null glue.)

0. Suppose that the given shape has strength-1 glue o on all sides. The algorithm will operate entirely at temperature $\tau = 2$. Let n denote the number of corners in the given shape, and let $\ell = \lceil \log_2 n \rceil$. We assume that ℓ (or an upper bound thereof) is known and that the feature size is larger than $3\ell + 10$.

1. **Mark one tile edge**, namely, the top edge of the leftmost topmost tile; refer to Figure 21.

2. **Cover the boundary with identifying marks at corners**; refer to Figure 22. We add a single layer around our shape, distinguishing edges near the corners and in particular the leftmost topmost corner.

3. **Walk around the given shape to produce a multipart mold** by mixing in all the following tiles at once.

   (a) **Start walking at the leftmost topmost corner** by adding one RNA tile each at glues t_2 and t_3, with glue on the right side to start a growing binary counter at 0.

   (b) **Copy the bit string into DNA** in the next column.

   (c) **Increment the counter** in the next column, possibly growing by one bit. We use essentially the growing binary counter construction from the x-monotone case as shown in Figure 14, which writes a bit only every other row. We also mark the most-significant bit with a special type of glue (1^* instead of 1), and include a blank (non-bit) row below the least-significant bit.

   (d) **Repeatedly copy the counter to the outside and to the right along the edge**, continuing as long as we have the edge glue e. The construction, shown in Figure 23, scans alternately up and down to route the bits in turn to the outside of columns, while at all times transferring the bits along their own rows.

   (e) **Turn right at convex corners** upon detection of a c_1 or c_2 glue. Refer to Figure 24.

   (f) **Turn left at reflex corners**. In fact, the construction in Step 3d simply crashes into a wall at a reflex corner. We can detect this because the outside of the counter becomes adjacent to an edge glue e. This adjacency triggers the construction in Figure 25, which transfers the outside copy of the counter to the next edge of the shape. Though not shown in the figure, we also add tiles to fill in the bounding box of the construction, forming a rectangle of width $3\ell+O(1)$ and length roughly equal to the edge.

   (g) Immediately after every corner turn, **copy the bit string into RNA** in the next column, then as in Step 3b, **copy the bit string into DNA** in the following column, and **increment the bit string in DNA** in the next column.

4. **Break apart the multipiece mold and the original shape** by applying the enzyme. The result is a rectangle for every edge, with counter labels on two incident sides, which express a code for how to put the pieces back together.

5. **Infinitely replicate the rectangles** while preserving their two incident labeled sides. This construction is a modification of rectangle replication from the x-monotone case as shown in Figure 17.

6. **Re-assemble the molds**. This three-stage construction is similar to Figure 18 from the x-monotone

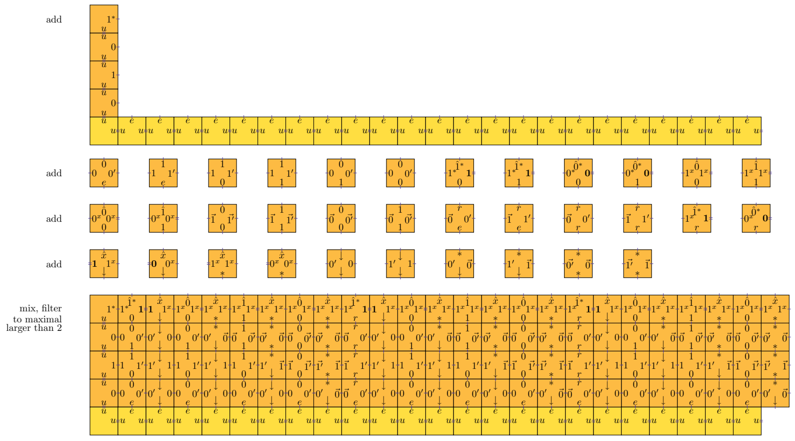


Figure 23: Simulation of Step 3d. For improved visibility, we do not display the blank rows in between bit rows.

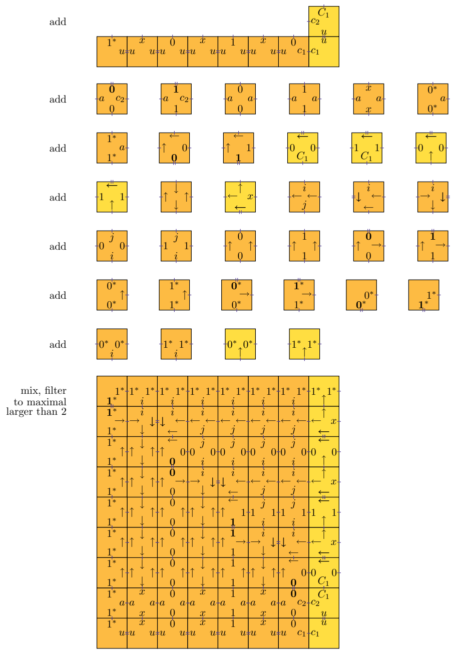


Figure 24: Simulation of Step 3e.

case, with the addition of "gums" to prevent teeth sequences of different lengths from attaching to each other.

(a) **Attach three layers of RNA gums to the most-significant bits** at the end of each edge. For reflex corners, we build a 1×3 column that attaches at the 1^* glue and the adjacent glue from the filled bounding box. For convex corners, we build a 2×2 box with a 1×1 tab that attaches to the 1^* glue and the incident side.

(b) **Attach RNA teeth to remaining bits so that pieces fit together.** Specifically, at the beginning of each edge, attach a tooth to each 0 bit and just above each 1 bit; and at the end of each edge (after the corner turn and increment), attach a tooth to each 1 bit and just above each 0 bit.

(c) **Attach two layers of RNA toothpaste to the least-significant bits** by building Ls that attach to the edge glue e and the adjacent blank (non-bit) row of the counter. The toothpaste sticks with collective strength of 2, causing the pieces of the mold to glue together, but only along matching labels.

Because at most two teeth cluster together at any point, they can never come in contact with the gums of another piece and compatibly join. Thus only equal-length labels can join.

7. **Fill the molds**:

(a) **Fill the remaining RNA inner layer** of each mold, including what would be the two mark tiles but now as RNA.

(b) **Fill the remaining mold space with DNA**.

(c) **Apply the enzyme** to break the molds back apart, releasing the infinitely many copies of the original shape and the rectangular pieces of the mold.

5 Future Work

Many future directions stem from this work. A few are as follows.

One interesting problem is how to magnify or miniaturize a given input shape by a specified magnification factor. In fact, the infinite yield constructions in our paper can be modified to scale up the replicated shape by a given factor k at the cost of an additional $O(k)$ tile complexity. The more general problem of magnifying shapes by given factors as efficiently as possible in terms of tile and stage complexity is a direction for future research.

Another direction is applying the staged RNA enzyme model to the assembly of shapes from scratch. A simple first result shows that this model can assemble thin lines more efficiently than the standard model can achieve in terms of tile complexity. Can more interesting shapes or computable patterns be built efficiently under this model?

In terms of the shape replication problem, are there

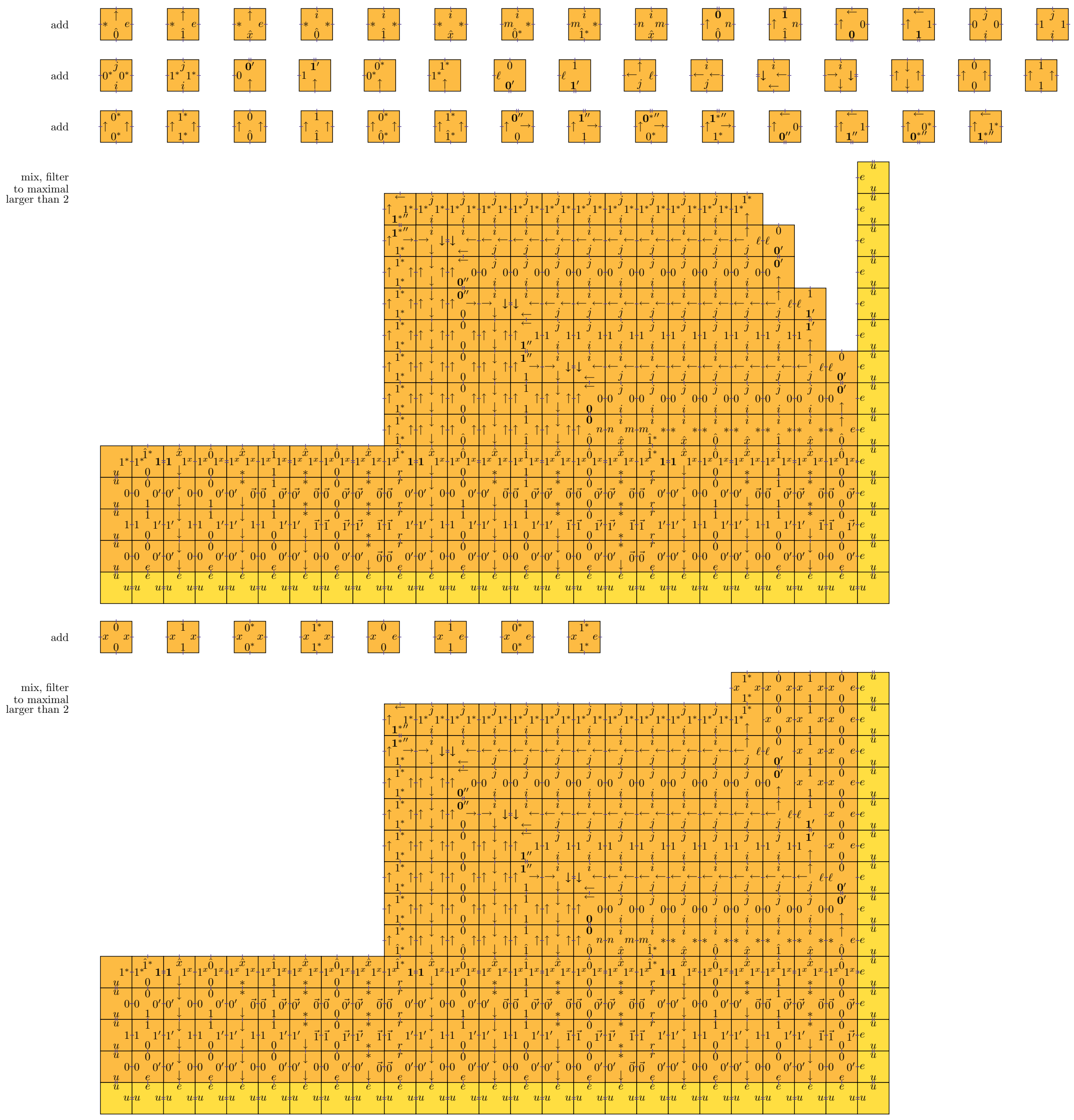


Figure 25: Simulation of Step 3f. For improved visibility, we do not display the blank rows in between bit rows.

other practically motivated self-assembly models that can permit replication? One potential example is the multiple temperature model [KS06] in which tiles can be broken off by increases to a systems temperature. Is it possible to simulate some RNA enzyme results under the multiple temperature model to perform shape replication?

In this paper we primarily focus on precise replication by using a non-constant number of stages to specify how many copies are created. An alternate approach is to maintain a constant number of stages but use a non-constant number of tile types. In this paper we show that both versions are achievable for the simple case of rectangles. More generally, work in progress has shown that the more general genus 0 shapes can be replicated with either logarithmic tile or stage complexity. Determining if some type of smooth tradeoff between tile and stage complexity is an interesting direction for both replication and staged assembly in general.

Acknowledgments

This work was initiated at the 24th Bellairs Winter Workshop on Computational Geometry, co-organized by Erik Demaine and Godfried Toussaint on February 6–13, 2009, in Holetown, Barbardos. We thank the other participants of the workshop—Greg Aloupis, Brad Ballinger, Prosenjit Bose, Jean Cardinal, Sébastien Collette, Vida Dujmović, Ferran Hurtado, John Iacono, Stefan Langerman, Godfried Toussaint, David Wood, and Stefanie Wuhrer—for providing helpful comments and a stimulating environment.

References

[ACGH01] Leonard Adleman, Qi Cheng, Ashish Goel, and Ming-Deh Huang. Running time and program size for self-assembled squares. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 740–748, New York, NY, USA, 2001. ACM.

[Adl00] Leonard M. Adleman. Toward a mathematical theory of self-assembly. Technical Report 00-722, Department of Computer Science, University of Southern California, January 2000.

[AGKS04] Gagan Aggarwal, Michael H. Goldwasser, Ming-Yang Kao, and Robert T. Schweller. Complexities for generalized models of self-assembly. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 880–889, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[CE03] Qi Cheng and Pablo Moisset Espanés. Resolving two open problems in the self-assembly of squares. Technical Report 03-793, University of Southern California, June 2003.

[DDF$^+$08] Erik D. Demaine, Martin L. Demaine, Sándor P. Fekete, Mashhood Ishaque, Eynat Rafalin, Robert T. Schweller, and Diane L. Souvaine. Staged self-assembly: Nanomanufacture of arbitrary shapes with $o(1)$ glues. *Natural Computing*, 7(3):347–370, September 2008.

[KS06] Ming-Yang Kao and Robert Schweller. Reducing tile complexity for self-assembly through temperature programming. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 571–580, 2006.

[Rod94] Gene Roddenberry. Star trek: The next generation. Television series, 1987–1994.

[RW00a] Paul W. K. Rothemund and Erik Winfree. The program-size complexity of self-assembled squares. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 459–468, 2000.

[RW00b] Paul W. K. Rothemund and Erik Winfree. The program-size complexity of self-assembled squares (extended abstract). In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 459–468, New York, NY, USA, 2000. ACM.

[SW05] Rebecca Schulman and Erik Winfree. Self-replication and evolution of dna crystals. In *Advances in Artificial Life, 8th European Conference*, pages 734–743, 2005.

[UE71] Tsuneko Uchida and Fujio Egami. Microbial ribonucleases with special reference to rnases t1, t2, n1, and u2. *The Enzymes*, 4:205–250, 1971.

[Win98] Erik Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, Pasadena, 1998.

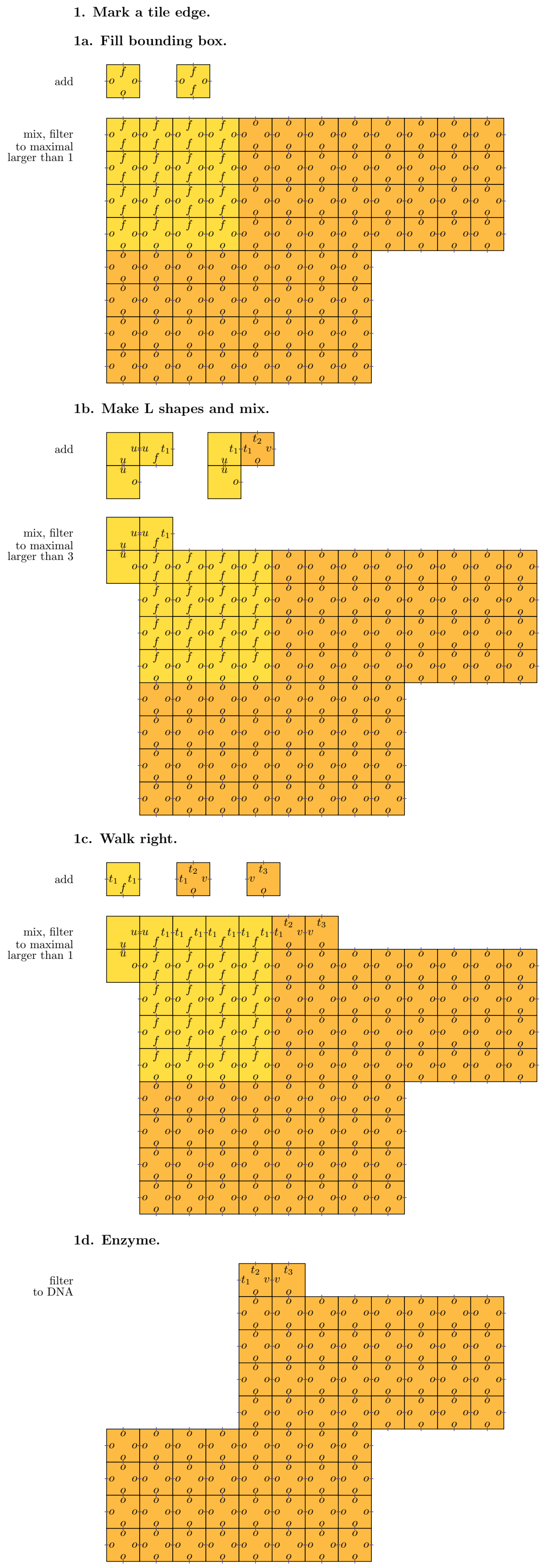
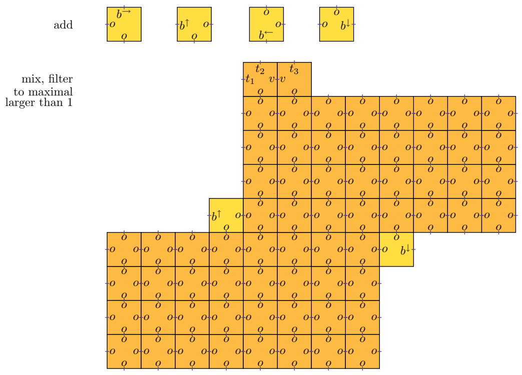

1. Mark a tile edge.
1a. Fill bounding box.
add
mix, filter to maximal larger than 1
1b. Make L shapes and mix.
add
mix, filter to maximal larger than 3
1c. Walk right.
add
mix, filter to maximal larger than 1
1d. Enzyme.
filter to DNA

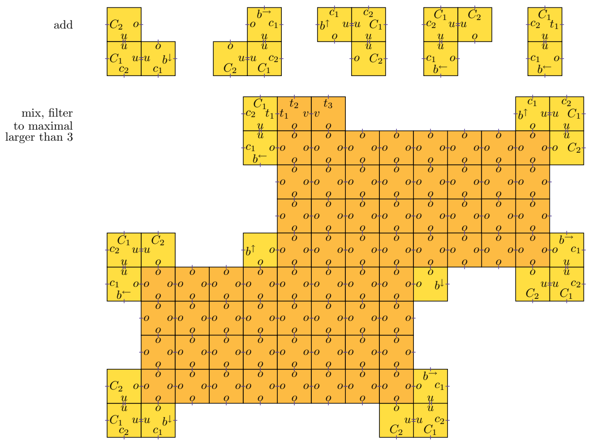
Figure 21: Simulation of Step 1.

2. Cover the boundary.

2a. Detect reflex corners.



2b. Detect convex corners.



2c. Fill in the sides.

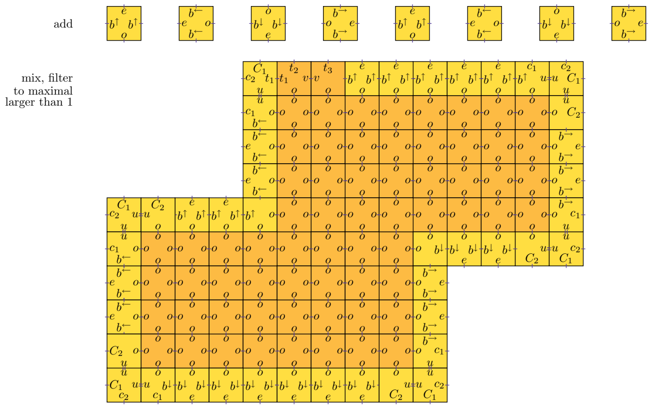


Figure 22: Simulation of Step 2.