Fundamenta Informaticae 154 (2017) 239–260
DOI 10.3233/FI-2017-1564
IOS Press

Smart Tile Self-Assembly and Replication

Lila Kari*
School of Computer Science, University of Waterloo &
Department of Computer Science
University of Western Ontario, Canada
lila.kari@uwaterloo.ca

Amirhossein Simjour
Department of Computer Science
University of Western Ontario, Canada
asimjour@csd.uwo.ca

Abstract. We propose the concept of self-assembly of smart tiles, i.e., tiles which possess a local computational device in addition to having edge glues that can be activated or deactivated by signals. The local tile computational device can range from its being absent, to being a counter, a simple look-up table, a finite state machine, all the way to being a Turing machine. Thus, this model offers a general framework to discuss and compare various tile self-assembly systems. We demonstrate the potential of self-assembly with smart tiles to efficiently perform robotic tasks such as the replication of convex shapes. The smart tile assembly system that we propose for convex shape replication does not make any assumption on the glues and signals of the interior tiles of the input supertile, and uses a scaffold to assemble a replica adjacent to the input supertile.

Keywords: DNA Self-Assembly Systems, Signal Tile Assembly Model, Smart Tiles, Robotics

1. Introduction

Self-assembly in nature is one the main inspirations for multi-robot systems [1, 2]. In such multi-robot systems, robots can cooperate in order to move blocks and build structures [3], assemble complex two-dimensional shapes [4, 5], or replicate given two-dimensional shapes [6]. Robots in many multi-robot

*Address for correspondence: School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada

systems do not use any central control, and use instead their limited local computational capabilities and communication with nearby robots to accomplish these tasks. Indeed, robots in multi-robot systems often do not need to have extensive computational power [7].

In the context of DNA computing and molecular programming, self-assembly of DNA tiles has been used extensively for either computational tasks or for the assembly of complex nanostructures [8]. Traditionally, these tiles have been "static" in the sense that they are unchangeable entities that can attach to their neighbouring tiles based on pre-programmed local attachment rules. However, recently, an "active" dimension has been added to tiles in the form of, e.g., the addition of signal transmission that can activate or deactivate glues on the tile edges [9, 10, 11]. On the other hand, several DNA-based computational devices have been implemented, such as finite state automata [12], computational state transitions [13], and digital circuits [14], to name just a few.

In this paper, we bring together these three ideas, multi-robot systems with local computing capabilities, static DNA tiles, and DNA-based computational devices, by proposing a dynamic self-assembly model based on *smart tiles*, that is, dynamic tiles which are equipped with local computational devices in addition to being able to transmit signals that can activate or deactivate glues.

Smart tiles are based on the classical tiles as defined in the abstract tile self-assembly model, aTAM [15], but they are enhanced with the ability to transmit signals that can activate or deactivate glues, as proposed in the signal tile assembly model STAM [10, 9]. The novel feature of smart tiles, which we propose in this paper, is the fact that each is equipped with a local computational device.

We show the potential of smart tile self-assembly (smart-TAM) systems to perform complex tasks by proving their ability to replicate arbitrary convex shapes, in a manner that is similar to the 2D shape replication by smart sand robotic systems [6]. Note that, in contrast to previous methods for pattern replication by self-assembly [16], the smart tile self-assembly systems we use for shape replication do not make any special assumptions about the tiles located in the interior of the shape. Moreover, in contrast to other methods for shape replication by self-assembly systems [17] [18] [19], our proposed smart tile assembly replication system is not in-place. This means that our construction assembles a replica separately from the original object, rather than cutting out the original object to leave an imprint to be filled in by the replica. We also note that [17] uses negative glues and is based on the 2HAM attachment model, wherein two supertiles can attach to each other during an attachment transition, while our model allows attachment of one tile only to the existing structure, during each transition. Another replication system, [19], is based on the staged tile assembly model, which allows changes of the tile set at different stages of the assembly process.

Overall, the model we propose offers a general framework to discuss and compare various tile self-assembly systems. For example, STAM with an one-tile-at-a-time attachment model becomes a particular case of smart-TAM wherein the local tile computational device is a simple look-up table, and activation of deactivated glues is not allowed. The local tile computational device that we use in the smart-TAM system presented in this paper, for two-dimensional shape replication, is a generalized sequential machine (a deterministic finite state automaton with output). At the other end of the spectrum, the local tile computational device can be as complex as a Turing Machine.

The paper is organized as follows. Section 2 introduces basic definitions and notations, as well as formally defines a smart tile assembly system. Section 3 constructs a smart-TAM system that can replicate a supertile of any L-convex shape, and indicates how this construction can be easily

modified to replicate any convex shape. Section 4 discusses complexity issues and illustrates the idea for a construction of a smart-TAM system with lower tile and space complexity, which could replicate a supertile of any shape.

2. Smart tile assembly

In this section we will introduce the Smart Tile Assembly Model (smart-TAM), a tile self-assembly model with tiles that are equipped with identical local computational devices. The local computational device may be as simple as a look-up table or as complex as a Turing machine. The only constraint placed on the computational devices is that the input and the output must be defined based on the signals and the glues on the edges of the tile.

2.1. Basic definitions

The following basic definitions will be used in the formal description of the smart tile assembly model.

Given a set A, its cardinality is denoted by $|A|$, and the set of all subsets of A is denoted by $\mathcal{P}(\mathcal{A})$. A *multiset* is a generalized set, the elements of which can appear more than once. The multiplicity of an element a from a multiset S is the number of times that a appears in S. The cardinality of a multiset is the sum of the multiplicities of its elements. For example, if $S = \{a, a, b, c, d, c\}$ is a multiset, the multiplicity of a in S is 2 and the cardinality of S is 6.

The set of *directions* is defined as $D = \{N, E, S, W\}$, and the elements in D represent the directions north, east, south and west respectively.

In this paper we will define and investigate smart tile assembly systems, which use smart tiles, that is, tiles endowed with a local computational device. The local computational device that we will use to illustrate the capabilities of such self-assembly systems is the generalized sequential machine. A *generalized sequential machine* (GSM) is similar to a finite nondeterministic automaton but which can output a word for each input letter it reads. Formally, a GSM is a sextuple $g = (S, V_I, V_O, s_0, S_f, P)$ where S is the set of *states*, V_I is the *input* alphabet, V_O is the *output* alphabet, with $S \cap (V_I \cup V_O) = \emptyset$, and s_0 is an element in S called the *start state*, $S_f \subseteq S$ is the set of *final states*, and the *productions* in P are of the form

$$s_i a \longrightarrow w s_j, \qquad s_i, s_j \in S, \quad a \in V_I, \quad \text{and } w \in V_O^*.$$

For a generalized sequential machine g and a word u over its input alphabet V_I, we denote

$$g(u) = \{w | s_0 u \Longrightarrow^* w s_1, \text{ for some } s_1 \in S_f\}$$

where $\Longrightarrow^*$ is derivation relation induced by $\longrightarrow$. If L is a language over V_I, then the GSM *translates*, or *maps*, L into the language

$$g(L) = \{w | w \in g(u) \text{ for some } u \in L\}.$$

A generalized sequential machine is deterministic iff, for every $s_i \in S$ and $a \in V_I$, there is exactly one production in P. A deterministic GSM is a *Meally machine* iff all words w appearing in productions in P consists of only one letter in V_O.

The smart tile assembly model that we will define is a generalization of the *Signal Tile Assembly Model (STAM)*, which was introduced by Padilla et al. [10], which itself is an extension of the Abstract Tile Assembly Model (aTAM) introduced by Winfree in [15]. More precisely, STAM is a tile assembly model based on 2HAM [20, 21], wherein each tile possesses a set of glues on each edge (instead of one glue per edge, like in aTAM and 2HAM), and glues can be activated or deactivated by signals. In addition, unlike aTAM, where the growth of an assembled structure happens one tile at a time, in STAM (as in 2HAM), a whole multi-tile structure that was assembled separately can attach to an existing structure in one attachment step.

In the STAM model, the status of each glue on an edge can be *latent*, *on*, or *off*. Only a glue whose status is *on* is active and can contribute to attaching the tile to another tile with an identical glue with *on* status on the abutting edge. If the status of a glue is *off* or *latent*, the glue is inactive and it does not have any attachment capabilities. In order to change the status of glues in STAM, signals are used. Intuitively, a signal is a mapping associated to a given tile that assigns to a glue on an edge a set of changes in the status of the glues on the other edges. For example, assume that tile t has glue g_e on its east edge, glue g_s on its south edge, and glue g_n on its north edge. Also assume that all these glues are *on*, and assume that there is a signal on the east side of the tile t that assigns a change of the status of the glue g_s to *off*. If that is the case, and if the tile t attaches to another tile via its east edge, the signal deactivates the glue g_s, that is, it changes its status to *off*. Signals can change the status of a glue from *latent* to *on* or to *off*, or from *on* to *off*. Note that, once a glue is in the *off* state, its status cannot be changed anymore. A tile can send a signal to its neighbour tile by activating a glue on the edge that they have in common. Signals can change the status of the glues, therefore signals can activate new glues and thus initiate a signal in the next tile. Moreover, signals can activate glues on a free (unattached) edge and make new attachments possible. In addition to the activation, signals can deactivate glues and, as a result, an existing structure might become unstable. In the STAM model, if the deactivation of a glue makes a structure unstable, the structure will break apart into stable components.

More formally, if Γ is a set of glues, Σ is a set of tile labels, $Q = \{on, off, latent\}$, and D is the set of directions, then an STAM tile over the alphabet $\Gamma \times \Sigma$ is a quadruple $t = (G, L, \Pi, \delta)$, where $G : D \to \mathcal{P}(\Gamma \times Q)$ denotes the sets of glues on the edges of t, and $L \subseteq \Sigma$ is a set of tile labels. The transition function $\delta : D \times \Gamma \to \mathcal{P}((D \times \Gamma \times \{on, off, latent\}) \cup (\Sigma \times \{on \times off\}))$ defines the set of actions that result as a consequence of an attachment through a glue on one of the edges of the tile t. More precisely, the transition function of the tile t associates to a glue on one of its edges changes of the status of glues on other edges, or a change of a tile label, or both. These outputs are called actions. During a self-assembly process, when a transition is applied, the outputs of the transition function are added to Π, the multiset of pending actions. For a detailed formal definition of STAM, the reader is referred to [10].

2.2. The smart tile assembly model

Informally, in the smart tile assembly model, a local computational device is added to each tile which, if it is in a certain state and receives as input a glue and a direction, it changes its state and outputs an action that amounts to the activation or deactivation of some other glues and tile labels. This is

a generalization of the STAM feature wherein the mapping between the glues and the actions was defined by the *transition function*. In the smart tile assembly model, this role is played by the local computational device, which may be as simple as a look-up table or as complex as a Turing machine.

Smart tiles: A *smart tile* can be viewed as a unit square with a tile label, as well as glues on each of its four edges. A *glue* over the glue alphabet Γ is a triplet (γ, d, q) where $\gamma \in \Gamma$, the alphabet of glues, $d \in D$ is one of the directions, and $q \in \{on, off\}$. A *tile label* over the tile label alphabet Σ is a pair (σ, q) where $\sigma \in \Sigma$ and $q \in \{on, off\}$. Moreover, a smart tile is endowed with a computational device C, that can change the status q of glues and tile labels.

Formally, a *smart tile* over the alphabet $\Gamma \times \Sigma$ is a quadruple (G, L, Π, C) that has a set of glues G over the glue alphabet Γ, a set of labels L over the tile label alphabet Σ, a multiset of a *pending actions* Π with elements from $(\Gamma \times D \times \{on, off\}) \cup (\Sigma \times \{on, off\})$, and a *tile computational device* C. A computational device C in the context of smart-tile assembly is a rewriting system with input alphabet $I \subseteq \Gamma \times D$, and output alphabet $O \subseteq \mathcal{P}((\Gamma \times D \times \{on, off\}) \cup (\Sigma \times \{on, off\}))$. Generalized sequential machines (finite automata with output) and Turing machines are examples of valid computational devices.

Definition 2.1. A *smart tile assembly system (smart-TAM system)*, over the alphabet $\Gamma \times \Sigma$ is a quadruple $(\Theta, g, \tau, s_{seed})$, where Θ is a set of smart tiles over the same alphabet, $g : \Gamma \to \mathbb{N}$ is the *glue-strength function* that defines the binding strength of each glue, $\tau \in \mathbb{N}$ is the *temperature* which defines the minimum total strength required for an attachment of a tile to occur, and $s_{seed} \in \Theta$ is the *seed* smart tile, which is the tile from which the self-assembly process starts.

A *configuration* over the set of smart tiles Θ is a mapping $c : \mathbb{Z}^2 \longrightarrow \Theta$. Informally, a configuration is a placement of tiles on the rectangular $\mathbb{Z} \times \mathbb{Z}$ grid, with the centers of the tiles placed on the integer coordinate nodes of the grid. If for a position (x, y) we have that $c(x, y)$ is undefined, we will say that $c(x, y) =$ *null*. Two smart tiles in a configuration are called *adjacent* if the Euclidean distance between their centers is equal to 1. In other words, two smart tiles are adjacent if they have a common edge.

The weighted graph $B = (N, E, W)$ with the set of nodes N, set of edges E and the weight function $W : E \to \mathbb{N}$ is the binding graph of a configuration $c : \mathbb{Z}^2 \longrightarrow \Theta$ if there exists a bijective mapping between the set of nodes N and $\mathrm{dom}(c)$, such that two nodes in N are connected by an edge in the graph if and only if the corresponding smart tiles in the configuration are adjacent and, moreover, the strength of the attachment between the tiles equals the weight of the edge connecting the corresponding nodes in the graph. A configuration is called a *supertile* iff its binding graph is connected. A weighted graph G has a *cut* with weight w if one can partition the nodes of G into two disjoint sets such as the sum of the weights of the edges between these two sets is w. A supertile is called τ-*stable* if its binding graph does not have any *cut* with weight less than τ. If its binding graph G has at least one cut with the weight $w < \tau$, then the supertile is called τ-unstable.

The self-assembly in a smart-TAM system proceeds through transitions: A supertile V is obtained from the supertile U in one transition, iff the supertile V is the result of one of the following three types of transitions applied to the supertile U: attachment transitions, signal transitions, and detachment transitions.

Attachment transition: The $\tau-$stable supertile V is the result of the attachment of a new smart tile t to the $\tau-$stable supertile U, in position (x, y), iff $U(x, y) = null$, $V(x, y) = t$, and all other smart tiles in position (i, j) in U are the same as the tiles in position (i, j) in V, with the exception of the tiles in positions adjacent to (x, y), where the following hold:

- The sum of the strengths of the glues on the common edges between tile t and all its adjacent tiles t' from U is greater than or equal to the temperature τ. In addition, for all such t', the glues on the edges common with the newly attached smart tile t become inputs for the computational devices in both t and t'.
- For each of the inputs (on all newly attached edges) to the computational device of t and all its adjacent t' in U, one computational step is performed, and the output is added to the multiset of pending actions of that smart tile. In case there are multiple inputs, the order in which they are processed is non-deterministic.

Example 1. The smart tile (G_1, L_1, Π_1, C_1) with tile label T' in Figure 1 is defined such that $G_1 = \{(a, E, on), (b, E, on), (g, W, off) , (h, N, on)\}$, $L_1 = \{T'\}$, $\Pi_1 = \emptyset$, and the computational device C_1 is defined so that it never changes its state and outputs the empty set for every input. The smart tile (G_2, L_2, Π_2, C_2) with tile label T'' is defined such that $G_2 = \{(a, W, on), (b, W, on), (c, W, on), (d, E, on) , (e, E, off), (f, E, on)\}$, $L_2 = \{T''\}$, $\Pi_2 = \emptyset$, and the computational device C_2 is the GSM with set of states $\{s_1, s_2\}$, the input alphabet $\{a, b, c, d, e, f, g, h\} \times D$, the output alphabet $\{(d, E, \text{off}), (e, E, \text{on})\}$, and transition function $(s_1, (a, E)) \longrightarrow \{s_2\} \times \{(d, E, off), (e, E, on)\}$.

Figure 1(i) shows the smart tile T'' in the process of attaching to a a supertile containing the smart tile T'. Assume that the temperature of the system is $\tau = 2$ and that the strength of the glues a and b is equal to 1 (therefore the smart tiles T' and T'' attach via the glues a and b on the west edge of T'').

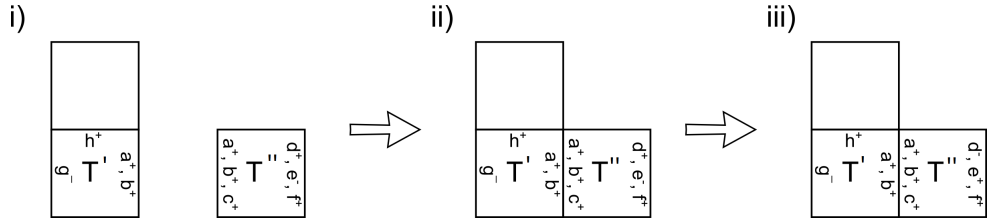

Figure 1. The attachment transition from Example 1. Note that superscripts are used to denote the status of glues and labels, with "+" indicating *on*, and "-" indicating *off*. *(i)* A smart supertile and a smart tile that can attach to it. *(ii)* The result of an attachment transition of the smart tile labelled T'' to the smart supertile containing the tile labelled T'. The glues of T'' after the attachment are changed due to the actions of the computational device on T''. The pending actions that are added to the set of pending actions of T'' are $\{(d, E, \text{off}), (e, E, \text{on})\}$. *(iii)* The outcome of a signal transition applied to the smart tile T'' with pending actions $\{(d, E, \text{off}), (e, E, \text{on})\}$, in the supertile from *(ii)*.

As a result of an attachment transition, the supertile in Figure 1(ii) is formed. During this attachment transition, the multiset Π_2 of the smart tile T'' will change from the empty set to $\Pi_2 = \{(d, E, \text{off}), (e, E, \text{on})\}$. Note that the attachment of the smart tiles T' and T'' changes the state of the computational device C_2 in T'' to s_2, and generates the output $\{(d, E, \text{off}), (e, E, \text{on})\}$, which will be

added to the set of pending actions which becomes $\Pi_2 = \{(d, E, \text{off}), (e, E, \text{on})\}$. Note also that this output only changes the set of pending actions Π_2 and it does not have any effect on the status of the glues. The status of the glues will be changed during a subsequent *signal transition*.

Signal transition: The supertile V is the result of *signal transition* in the smart tile $t = (G, L, \Pi, C)$ from the position (x, y) of the $\tau-$stable supertile U, iff t has at least one pending action in the set of its pending actions Π (activate or deactivate a glue or a label) and the following conditions hold. For all $(i, j) \in \mathbb{Z}^2$ all the smart tiles in the positions (i, j) in V are the same as the smart tiles in the positions (i, j) in U, except the smart tile t and the smart tiles t' that are adjacent to t which are changed as follows:

- The smart tile t removes one of the pending actions from its multiset Π and applies it to the indicated glue or tile label. The application of an action means changing the status of the corresponding glue or tile label accordingly.
- In addition, if the pending action is to activate a glue (change its status to *on*), then the computational device of the smart tile t' that is adjacent to t, on the same edge as the glue, performs a computational step with this glue as the input, and adds the output of this computational step to the set of its own pending actions. Subsequently, the used pending action is removed from the set Π of the pending actions of t.

Figure 1(iii) shows the outcome of the two signal transitions that result from applying the two pending actions $\{(d, E, \text{off}), (e, E, \text{on})\}$ of the smart tile T'' from Figure 1(ii).

Detachment transition: If a supertile U consists of two parts whose connection strength is lower than the temperature, then U can break into two supertiles. Formally, supertiles U_1 and U_2 are the result of a detachment transition of a τ-unstable supertile U if the following conditions hold: $G_V = (N, E, W)$ is the associated assembly-graph of the supertile U, and there exist two disjoint sets $N_1, N_2 \subset N, N_1 \cup N_2 = N$ such that the weight of the cut (N_1, N_2) is smaller than τ, and $G|_{N_1}$ is the assembly-graph associated to U_1 and $G|_{N_2}$ is the assembly-graph associated to U_2.

A computation in a smart-TAM system starts from the seed smart tile and proceeds by nondeterministic applications of attachment transitions, signal transitions, and detachment transitions. A configuration is called final iff it is a τ-stable configuration, the computational devices on all its tiles are in a final state, and no more attachment or signal transitions are possible.

The following example shows a smart-TAM system at temperature 1, with a single tile type, that can assemble a linear supertile consisting of a row of three tiles. Without the local tile computational devices, such a tile system in the abstract tile assembly model, aTAM, could only generate an infinite linear supertile.

Example 2. Consider a smart-TAM system at temperature 1 that consists of a single smart tile, which is also its seed tile, $t_1 = (G_1, L_1, \Pi_1, C_1)$, see Figure 2 (1). The tile t_1 is defined such that $G_1 = \{(a, E, \textit{on}), (a, W, \textit{on}), (s, E, \textit{on}), (s, W, \textit{off})\}$, the strength of glue a is 1, the strength of

glue s is 0, $L_1 = \{T\}$, $\Pi_1 = \emptyset$, and the computational device C_1 is the GSM with set of states $\{q_0, q_R, q_C, q_L, q_F\}$, the initial state q_0, the set of final states $\{q_R, q_L, q_F\}$, input alphabet $\{a, s\} \times D$, output alphabet $\{(a, E, \text{off}), (s, W, \text{on}), (a, W, \text{off})\}$, and transition function defined as follows:

$$
\begin{aligned}
(q_0, (a, W)) &\to \{q_R\} \times \{(a, E, \text{off})\} \\
(q_0, (a, E)) &\to \{q_C\} \times \{(s, W, \text{on})\} \\
(q_0, (s, E)) &\to \{q_L\} \times \{(a, W, \text{off})\} \\
(q_C, (s, E)) &\to \{q_L\} \times \{(a, W, \text{off})\} \\
(q_C, (a, W)) &\to \{q_F\} \times \emptyset.
\end{aligned}
$$

The seed tile t_1 can attach to two more tiles of same tile type, one at its left and one at its right, to assemble a final configuration consisting of a row of three tiles. In the beginning, the computational devices in all individual smart tiles are in their start state, q_0. Figure 2 (2) illustrates an attachment transition of one tile to the right of the seed tile. The seed tile changes its state from q_0 to q_C, indicating that it will become the center tile in the final configuration. The newly attached tile changes its state from q_0 to q_R, indicating that it will become the rightmost tile in the final configuration. Figure 2 (3)

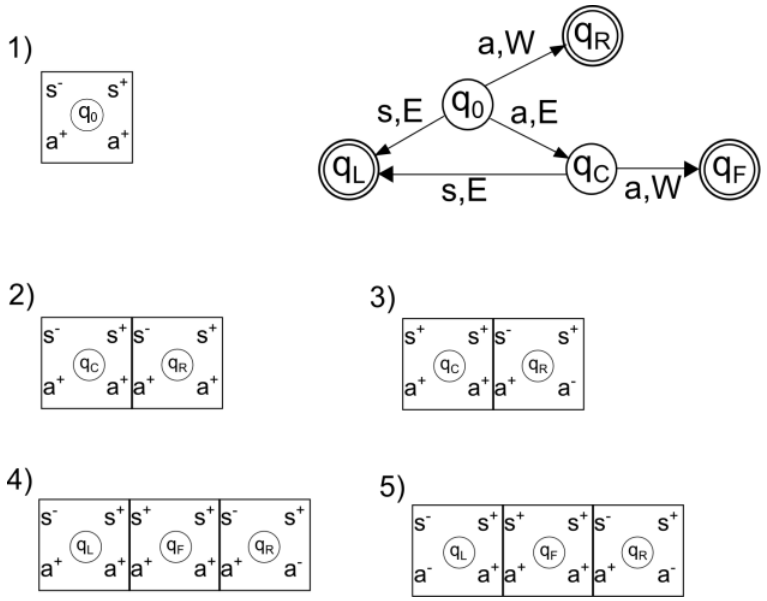

Figure 2. A smart-TAM system at temperature 1 with a single tile type that can assemble a final configuration consisting of a row of three tiles (Example 2). The state of the GSM of a smart tile is shown inside a circle at the center of the tile. The tile labels are not shown. The final states of the GSM are indicated by double circles around the states. (1) Left: The only smart tile type in the system; Right: The GSM of the smart tile. (2) The result of an attachment transition of one tile to the right of the seed tile. (3): The result of applying a signal transition to the supertile in (2), that is, of applying all the pending actions. (4): The result of an attachment transition of a tile to the left of the supertile in (3). (5): The final configuration obtained after applying a signal transition to the supertile in (4).

shows the result of applying a signal transition to the supertile in Figure 2 (2), that is, of applying the pending actions to that supertile. Note that glue a on the east edge of the rightmost tile is deactivated, and thus no tile can attach to the east side of this supertile. Figure 2 (4) shows the result of an attachment transition to the left of the supertile in Figure 2 (3). The newly attached tile changes its state from q_0 to q_L, indicating that this tile will be leftmost tile in the final configuration. Figure 2 (5) shows the result of applying a signal transition to the supertile in Figure 2 (4), that is, of applying all pending actions. This supertile is the final configuration since no more attachment or signal transitions are possible, and the GSMs of all tiles are in a final state.

There are three main differences between smart-TAM and STAM. First, smart tiles in smart-TAM are endowed with computational devices which allow signal reuse as well as having additional control over the self-assembly due to their internal states. Second, smart-TAM allows both activation and deactivation of glues, while STAM does not allow activation of deactivated glues. Third, smart-TAM as defined in this paper only allows the attachment of one tile at a time in each attachment transition, while STAM allows the attachment of two supertiles to each other during one attachment transition.

Informally, if STAM were restricted to the attachment of one tile at a time, then STAM would be a particular case of smart-TAM, with a "look-up table" as the local tile computational device, and no activation of deactivated glues. Alternatively, if one were to generalize smart-TAM to 2smart-TAM, that is, a smart tile assembly model exactly like the one in this paper except that it would use 2HAM-style attachments instead of only one-tile-at-a-time attachments, then STAM would be strictly weaker than such a 2smart-TAM.

3. A smart tile assembly system that replicates L-convex shapes

In this section, we construct a smart tile assembly system that can replicate any L-convex shape, and indicate how it can be easily modified to perform arbitrary convex shape replications. We note that the replication is at a one-to-one scale, and that the constructed smart-TAM system (and implicitly its tile-complexity) is independent of the size and particular shape that it is replicating. The only requirement is the existence of an initial supertile of the desired shape, whose exterior edges contain minimal information about their being a north, east, south or west edge, or their respectively being edges of one of the four "corner tiles" (similar assumptions have been made in, e.g., [17]). We start by recalling some basic notions about 2D grids.

A *cell* on a 2D grid is an ordered pair (x, y), where $x, y \in \mathbb{Z}$. The vectors $(0, 1), (0, -1), (-1, 0)$, and $(1, 0)$ are the unit vectors. Two cells $c_1 = (x_1, y_1)$ and $c_2 = (x_2, y_2)$ are adjacent if $(x_1, y_1) = (x_2, y_2) + \vec{u}$ where $\vec{u}$ is one of the unit vectors. A *shape* is, informally, a set of connected cells. Formally, a set of cells $C = \{(x, y)|x, y \in \mathbb{Z}\}$ is called a *shape* if for all pairs of distinct cells $c_1 \in C$ and $c_2 \in C$ there exists a sequence of unit vectors $p = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n)$, $n \geq 1$, such that $c_2 = c_1 + \sum_{i=1}^{n} u_i$ and, for all $1 \leq k \leq n$ we have $(c_1 + \sum_{i=1}^{k} u_i) \in C$. A sequence of unit vectors $p = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n)$ satisfying the conditions above is called a path from cell c_1 to cell c_2, inside the shape C, and n is called the length of p.

Definition 3.1. A shape C is called a convex shape if, for all pairs of cells $c_1, c_2 \in C$, there exists a path $p = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n)$ inside C from the cell c_1 to the cell c_2 and, moreover, the cardinality of the set $\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n\}$ is 2. In other words, p contains at most two distinct types of unit vectors.

A convex shape C is called L-convex if, for all pairs of cells $c_1, c_2 \in C$, there exists a path p inside of C with no more than one change of direction.

Definition 3.2. A convex shape C is called L-convex if, for all pairs of cells $c_1, c_2 \in C$, there exists a path $p = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n)$, $n \geq 1$, inside the shape C, from cell c_1 to c_2, such that either the length of the path is $n \leq 2$ or there exists a $1 \leq k < n$ such that $\vec{u}_1 = \ldots = \vec{u}_k$ and $\vec{u}_{k+1} = \ldots = \vec{u}_n$.

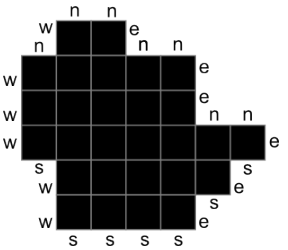

Figure 3. An example of an L-convex supertile. From every tile in the above supertile, all other tiles inside the supertile can be reached via a path inside the supertile that has maximum one change in direction. The edges labelled with n are north-free, the edges labelled with w are west-free, the ones labelled with e are east-free, and the ones labelled with s are south-free.

Given a supertile S, since the binding graph of S is connected, we have that $\text{dom}(S)$ is a shape. The set $\text{dom}(S)$ will be called the shape of S, and a supertile will be called *L-convex* if its shape is L-convex. Figure 3 shows an example of an L-convex supertile.

Conversely, given a shape $A \subseteq \mathbb{Z}^2$, a supertile S is said to have shape A iff $\text{dom}(S)$ either equals A, or can be obtained from A via a translation, rotation by a multiple of $90°$, or reflection. When it is clear from the context, the mapping between a supertile domain and its shape will not be mentioned, and the same path definition will be used for both supertiles and shapes.

The north edge of a tile t from the supertile S is called north-free if there is no tile from S attached to the north edge of t. The east-free, west-free, and south-free edges of a tile t from the supertile S are similarly defined. Figure 3 illustrates the north-free, east-free, west-free, and south-free edges of an L-convex supertile.

A north-free edge of the tile t from the L-convex supertile S is called the *north-west edge* of S, if the tile t is the tile located on the west-most column of the north-most row of the supertile S. Formally, the north-free edge of a tile t from the L-convex supertile S, with coordinates (x, y), is called the *north-west* edge of S if, for all tiles t' with coordinates (x', y') from S, we have that either $y' < y$ or $y' = y$ and $x' > x$. Moreover, the tile t that has the north-west edge of S is called the *north-west tile* of S. Similarly, the *east-north tile* is the north-most tile of the east-most column, the *south-east tile* of S is the east-most tile of its south-most row, the *west-south tile* is the south-most tile of west-most column. These tiles will be called the *corner* tiles of an L-convex supertile.

Lemma 3.3. Every L-convex supertile S has at most four corner tiles: one north-west, one east-north, one south-east, and one west-south tile.

Lemma 3.4. Let tile c_1 with coordinates (x_1, y_1) be the north-west tile of the L-convex supertile A, and tile c_2 with coordinates (x_2, y_2) be the east-north tile of A. Then, for all $x, y \in \mathbb{Z}$, if $x_1 \leq x \leq x_2$ and $y_1 \leq y \leq y_2$ then the tiles with coordinates (x, y_2) and (x_1, y) are in A.

Proof:
Because the supertile A is L-convex, there exists a path between c_2 and c_1 that lies inside A and that has only one change of direction. Since c_2 is the east-north tile and c_1 is the north-west tile of the supertile A, the directions on this path from c_2 to c_1 can only be north and west. Moreover, this path that starts from c_2 cannot start with the north direction, so it has to start in a west direction, move west until it reaches the x-coordinate x_2, then turn north and move north until it reaches the y-coordinate y_2. As a result, for all $x, y \in \mathbb{Z}$, if $x_1 \leq x \leq x_2$ and $y_1 \leq y \leq y_2$ the tiles with coordinates (x, y_2) form the horizontal part of the path, while the tiles with coordinates (x_1, y) form the vertical part of the path between c_2 and c_1, and all these lie within A because A is L-convex. □

Corollary 3.5. Statements analogous to Lemma 3.4 hold for the pairs of tiles east-north with south-east, south-east with west-south, and west-south with north-west, respectively.

Our goal is to design a smart-tile assembly system at temperature $\tau = 2$ that can replicate any L-convex shape. That is, we construct a smart-tile assembly system that, given as input an L-convex supertile of the required shape, produces a second supertile that is identical to the original one, modulo a translation and a reflection. In contrast with [16], our construction makes no assumptions on the interior tiles of the input supertile. In particular, our construction does not require any specific glues, or signal transitions associated with the tiles from the interior of the input supertile. The only requirement is that the border tiles of the input supertile satisfy the following conditions: All the free edges (edges without attachment and belonging to a tile on the border of the supertile) on the north, east, south and west must have glues n, e, s, and w respectively. The exceptions are the north-west, east-north, south-east, and west-south edges of the supertile which must have the glues S_n, S_e, S_s, and S_w respectively. Figure 4 (left) shows an example of an L-convex supertile and the expected glues on its borders.

Theorem 3.6. There exists a smart-TAM system $(\Theta, g, \tau, s_{seed})$ at temperature $\tau = 2$ with the following property: For any L-convex shape A, there exists an L-convex supertile A' with shape A such that, given A' as an input, the smart-TAM system self-assembles a second supertile with shape A.

Proof:
Figure 5 shows the general idea of the replication process.

The self-assembly process starts from an L-convex supertile A' of the required shape A, with predefined glues on the border edges as described in the preamble of this theorem, and the seed tile s_{seed} (the tile marked in grey in Figure 4). The replication process starts with *Step 1*, filling in the areas on the four sides of the input supertile A' to form a rectangle that surrounds it. This is accomplished using the tile set $\Theta_{fill} = \Theta_{fill\text{-}ne} \cup \Theta_{fill\text{-}nw} \cup \Theta_{fill\text{-}se} \cup \Theta_{fill\text{-}sw}$. The borders of the rectangle that surrounds the input supertile encode information about the external border of the supertile itself.

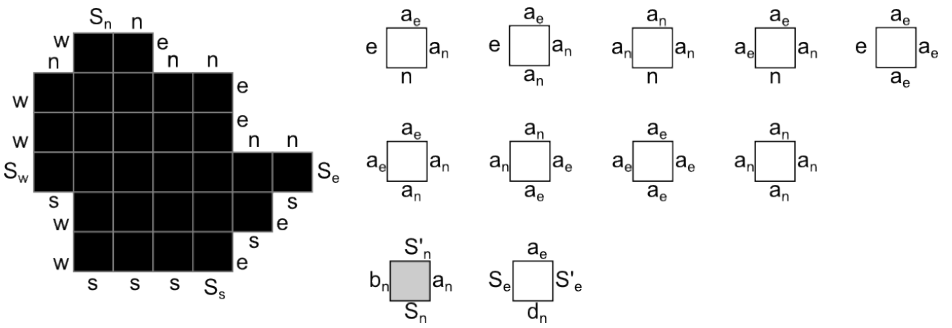

Figure 4. *Left*: An example of an L-convex supertile with the border glues needed for its replication by the smart-TAM system from Theorem 3.6. *Right:* The tile set $\Theta_{fill-ne}$, which fills the area at the north-east of the L-convex supertile, up to a rectangle. The tile marked in grey is the seed tile, s_{seed}. Glues S_n (on the north-east edge of the supertile), S_e (on the east-south edge of the supertile), S'_s (on the south-west edge of the supertile) and S'_w (on the west-north edge of the supertile) have strength equal to the temperature, $\tau = 2$, and the remaining glues have strength 1.

In *Step 2*, a different set of tiles, $\Theta_{transfer}$, is used to transfer the information from the borders of this rectangle to construct an identical rectangular hole underneath it. Following this, in *Step 3*, the set of tiles Θ_{border} fills in this rectangular hole, modulo a supertile-shaped hole in the middle. Finally, in *Step 4*, the set of tiles $\Theta_{replica}$ fills in the remaining supertile-shaped hole with a replica of the input supertile, and detaches this replica from the entire scaffold structure. Note that *Step 3* and *Step 4* are the only ones requiring the active use of the local computational devices on tiles, herein generalized sequential machines. The details of the construction are described below.

Step 1: Figure 4 (right) shows the tiles in $\Theta_{fill\text{-}ne}$ that fill the north-east corner of the rectangle that will surround the input supertile. The first row consists of the tiles that attach to the border of the supertile, while the second and the third row are the tiles that fill the remaining area of the rectangle. All glues of the tiles in $\Theta_{fill\text{-}ne}$ have temperature 1, except glues of the north-west edge S_n and that of the east-north edge, S_e, which have temperature 2.

The role of the tiles in $\Theta_{fill\text{-}ne}$ is to transmit the information about the border of the supertile towards the edges of the rectangle. They transmit the information from their south edge to the east, and the information from their west edge to the north. For example, the first tile on the first row in Figure 4 (right) has glue n on its south edge and glue e on its west edge, which means it attaches to the supertile as illustrated in Figure 6 (left) (the grey tile). To transmit the information n coming from its south edge, the east edge of this tile has glue a_n. Similarly, to transmit information e coming from its west edge, the north edge of the grey tile has glue a_e. Figure 6 (left) shows the result of the attachment of the tile set $\Theta_{fill\text{-}ne}$ to the L-convex supertile from Figure 4. Figure 6 (right) shows the mapping between the glues on the border of the supertile and the glues on the edges of the tiles of the filled rectangle. The result of *Step 1* is shown in Figure 7.

Step 2: The arrows in Figure 5 show the general directions in which the self-assembly proceeds, that results in the transfer of the information from the borders of the rectangle that surrounds the

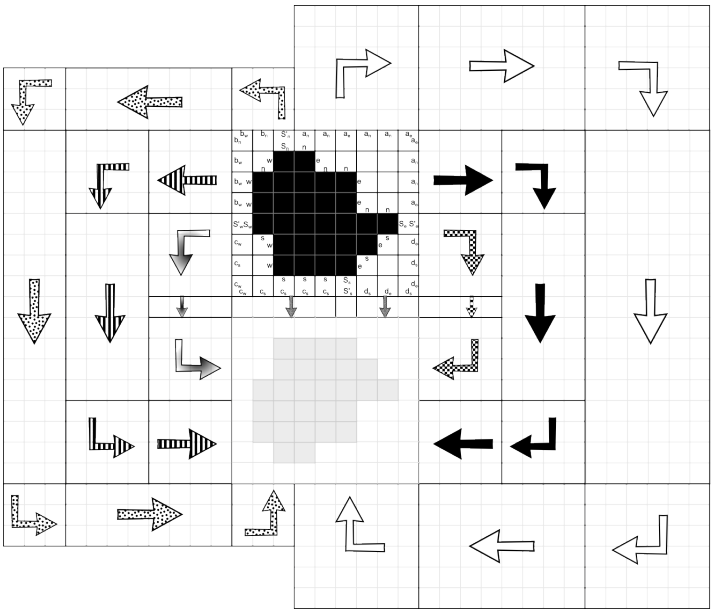

Figure 5. The general idea of the replication process of an L-convex supertile of a given shape, by a smart tile self-assembly system. The original shape is shown in black, and the constructed replica is in light grey. After tiles are used to fill in a rectangle that surrounds the original supertile (*Step 1*), information about the borders of the supertile (now encoded on the edges of the rectangle) is transported to form an identical rectangular hole underneath it (*Step 2*).The flow of this information is indicated by arrows. Afterwards, the rectangular hole is filled in, modulo a supertile-shaped hole (*Step 3*). Finally, the supertile-shaped hole is filled in with a replica of the input supertile, and the replica then detaches from the scaffold structure (*Step 4*).

supertile to the borders of a rectangular "hole", underneath it. The tiles on the inside border of this rectangular hole encode information about the edges of the input supertile, and this hole will be the place where the replica of the input supertile will be assembled. The strength of all the glues on this inner border of the rectangular hole is 1. The construction of the tile set $\Theta_{transfer}$, that assembles squares and rectangles which transfer information about the input supertile's border glues horizontally, vertically, or at 90° (Figure 5), is standard, see, e.g., [22] or [23], and will be omitted. Note that all the squares that transfer information at 90° (see Figure 5), that is, "turn corners", can be built using a constant number of tiles, similar to [22]. The rectangles that transfer information vertically or horizontally (marked with horizontal or vertical thick arrows in Figure 5) have sizes determined by the adjacent squares and the border of the original supertile, and thus the tile set that constructs these rectangles does not need any prior information regarding their size. The result of this step is illustrated in Figure 8.

Step 3: The purpose of this step is to fill in the rectangular hole obtained in *Step 2*, until only a smaller hole remains in the middle, shaped exactly like the input supertile. The smart tiles in the tile set used in this step, $\Theta_{border\text{-}ne}$ and $\Theta_{replica}$ are listed in Figure 9, and each of them is equipped with a GSM as a computational device.

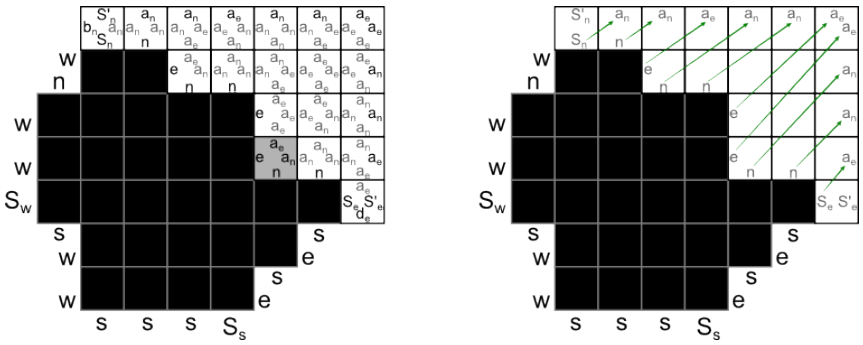

Figure 6. *Left:* The tile set $\Theta_{\textit{fill-ne}}$ is used to fill the area at the north-east of the L-convex supertile (in black). The inside border of the assembled quarter-rectangle, read from top-left to bottom-right, starts with S'_n and ends with S'_e. The glues on the free edges of the white quarter-rectangle contain information about the "north-east" border of the supertile. *Right:* The arrows show the bijective mapping between the glues on the edges on the "north-east" border of the original supertile and the glues on the free edges of the tiles on the border of the quarter-rectangle.

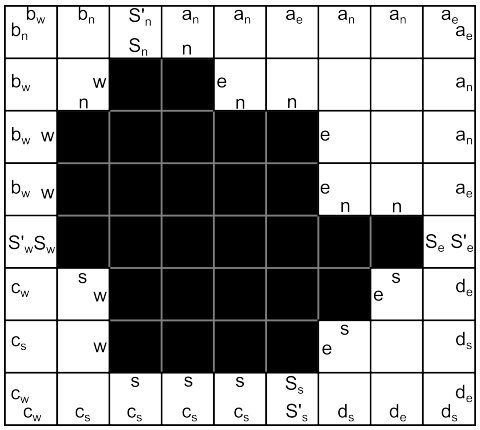

Figure 7. The result of *Step 1*: Filling all four quarter-rectangles around the original supertile from Figure 4. The glues on the border of the resulting rectangle correspond to the glues on the free edges of the original supertile.

The filling in of the north-east part of the rectangular hole starts from its north-east corner (since all glues on the tiles of its border have strength 1), and it continues through attachments of tiles through their north and east edges. The tiles labelled T_1, T_2, T_3, T_4 account for all possible combinations of north and east encoding glues (a'_n and a'_e) on the two attaching edges of each tile. Tiles T_5 and T_6 (see Figure 8) are used to attach to the tiles that contain information about the special north-west and east-north tiles of the input supertile, represented through glues S''_n and S''_e.

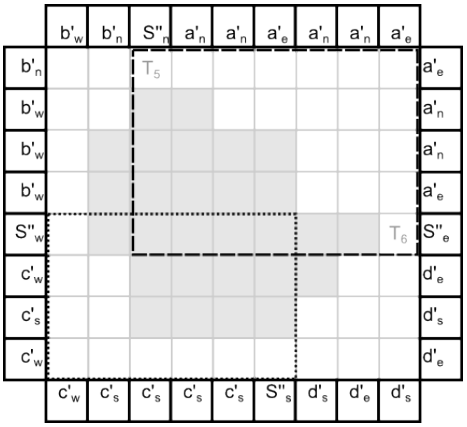

Figure 8. The rectangular hole that is obtained at the end of *Step 2*. The tiles that border the rectangular hole are outlined in thick lines: The inside glues of these tiles are identical in information content to the glues on the free edges of the tiles on the outside border of the rectangle that surrounded the input supertile. The thin lines inside the rectangular hole are a preview of the subsequent steps, which will first (*Step 3*) fill up the rectangular hole, modulo a supertile-shaped hole (light grey). This will be followed (*Step 4*) by filling the supertile-shaped hole with a replica. Tiles T_5 and T_6 will guide this process in the north-east quarter-rectangle. Note that the north-east quarter-rectangle (dashed lines) can overlap with, e.g., the south-west quarter-rectangle (dotted lines).

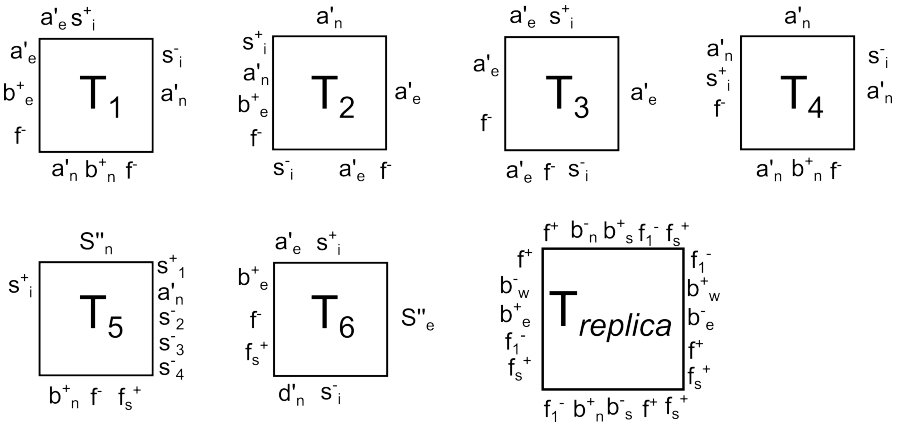

Figure 9. The tile set $\Theta_{\textit{border-ne}} = \{T_1, \ldots, T_6\}$, and $\Theta_{\textit{replica}} = \{T_{\textit{replica}}\}$.

While this is a non-deterministic process, in principle, the tiles T_1 through T_6 first fill in the north-east quarter rectangle of the hole. Afterwards, the GSM's implement the carving of a supertile-shaped hole from this rectangle. The GSM will start from its start state, q_{start} and then branch in one of four different computations, depending on whether the tile it belongs to is located in the north-east, south-east, south-west or north-west part of the border of the rectangular hole. The set of states of the north-east component of the GSM is $\{q_{start}, q_{north}, q_{east}, q_{west}, q_{south}\} \cup \{q_0^j, q_1^j, q_2^j, q_3^j, q_4^j | \ 1 \leq j \leq$

$6\} \cup \{q_3'^1\} \cup \{q_i^{replica} | \ 1 \leq i \leq 4\}$, the input alphabet is a set included in $\Gamma \times \{N, S, E, W\}$ where Γ is the set of glues, the output alphabet is the set of subsets of Π, the set of pending actions, and the final state set is $\{q_4^j | \ 1 \leq j \leq 6\} \cup \{q_4^{replica}\}$. The transitions of the GSM that direct the computations for its north-east component are illustrated in Figure 10 and listed in the sequel.

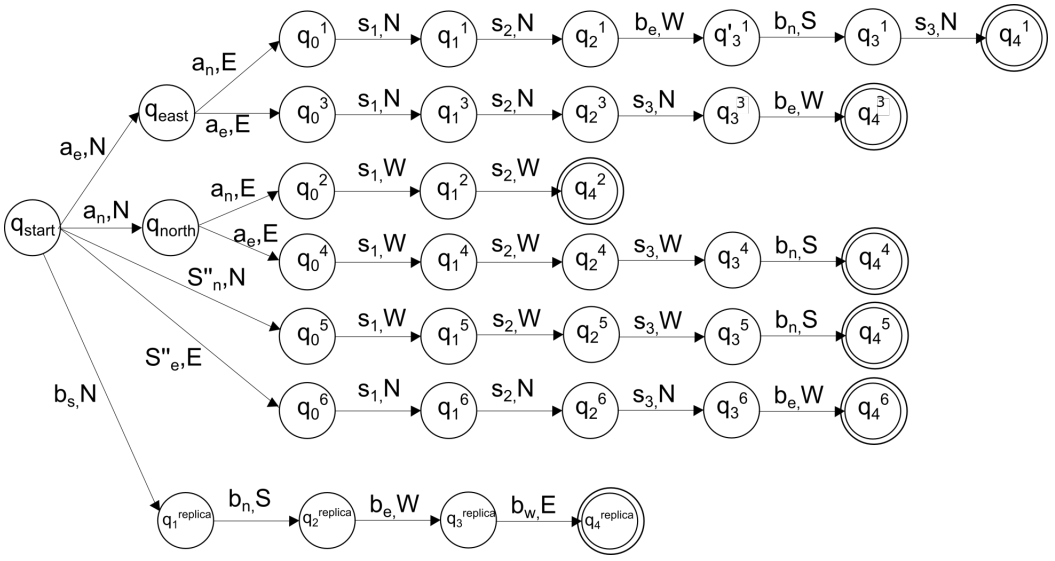

Figure 10. The local tile computational device (GSM). Only the component that deals with the north-east rectangle is shown. The complete input and outputs for each transition are listed in the text.

$$
\begin{aligned}
\text{Initial transitions}: \ & (q_{\text{start}}, (a_e, N)) \to \{q_{\text{east}}\} \times \emptyset \\
& (q_{\text{start}}, (a_n, N)) \to \{q_{\text{north}}\} \times \emptyset \\
& (q_{\text{east}}, (a_n, E)) \to \{q_0^1\} \times \emptyset \\
& (q_{\text{east}}, (a_e, E)) \to \{q_0^3\} \times \emptyset \\
& (q_{\text{north}}, (a_n, E)) \to \{q_0^2\} \times \emptyset \\
& (q_{\text{north}}, (a_e, E)) \to \{q_0^4\} \times \emptyset \\
T_1: \ & (q_0^1, (s_1, N)) \to \{q_1^1\} \times \{(a_e', W, \text{off}), (a_n', S, \text{off}), (s_1, E, \text{on})\} \\
& (q_1^1, (s_2, N)) \to \{q_2^1\} \times \{(f, W, \text{on}), (f, S, \text{on}), (s_2, E, \text{on})\} \\
& (q_2^1, (b_e, W)) \to \{q'^1_3\} \times \emptyset \\
& (q'^1_3, (b_n, S)) \to \{q_3^1\} \times \emptyset \\
& (q_3^1, (s_3, N)) \to \{q_4^1\} \times \{(s_4, W, \text{on}), (s_4, S, \text{on}), (s_3, E, \text{on})\} \\
T_2: \ & (q_0^2, (s_1, W)) \to \{q_1\} \times \{(s_1, S, \text{on})\} \\
& (q_1, (s_2, W)) \to \{q_4^2\} \times \{(s_2, S, \text{on})\}
\end{aligned}
$$

$$
\begin{aligned}
T_3 :&(q_0^3,(s_1,N)) \to \{q_1^3\} \times \{(a'_e,W,\text{off}),(s_1,S,\text{on})\}\\
&(q_1^3,(s_2,N)) \to \{q_2^3\} \times \{(f,W,\text{on}),(s_2,S,\text{on})\}\\
&(q_2^3,(s_3,N)) \to \{q_3^3\} \times \emptyset\\
&(q_3^3,(b_e,W)) \to \{q_4^3\} \times \{(s_4,W,\text{on}),(s_3,S,\text{on})\}\\
T_4 :&(q_0^4,(s_1,W)) \to \{q_1^4\} \times \{(a'_n,S,\text{off}),(s_1,E,\text{on})\}\\
&(q_1^4,(s_2,W)) \to \{q_2^4\} \times \{(f,S,\text{on}),(s_2,E,\text{on})\}\\
&(q_2^4,(s_3,W)) \to \{q_3^4\} \times \emptyset\\
&(q_3^4,(b_n,S)) \to \{q_4^4\} \times \{(s_4,S,\text{on}),(s_3,E,\text{on})\}\\
T_5 :&(q_{\text{start}},(S''_n,N)) \to \{q_0^5\} \times \{(s_1,E,\text{on})\}\\
&(q_0^5,(s_1,W)) \to \{q_1^5\} \times \{(f,S,\text{on}),(s_2,E,\text{on})\}\\
&(q_1^5,(s_2,W)) \to \{q_2^5\} \times \{(s_3,E,\text{on})\}\\
&(q_2^5,(s_3,W)) \to \{q_3^5\} \times \{(s_3,E,\text{on})\}\\
&(q_3^5,(b_n,S)) \to \{q_4^5\} \times \{(s_4,E,\text{on})\}\\
T_6 :&(q_{\text{start}},(S''_e,E)) \to \{q_0^6\} \times \emptyset\\
&(q_0^6,(s_1,N)) \to \{q_1^6\} \times \{(f,W,\text{on}),(s_1,S,\text{on})\}\\
&(q_1^6,(s_2,N)) \to \{q_2^6\} \times \{(s_2,S,\text{on})\}\\
&(q_2^6,(s_3,N)) \to \{q_3^6\} \times \{(s_3,E,\text{on})\}\\
&(q_3^6,(b_n,W)) \to \{q_4^6\} \times \{(s_4,E,\text{on})\}
\end{aligned}
$$

$$
\begin{aligned}
T_{\text{replica}} :&(q_{\text{start}},(f_s,N)) \to \{q_{\text{start}}\} \times \{(f_1,S,\text{on})\}\\
&(q_{\text{start}},(b_s,N)) \to \{q_1^{\text{replica}}\} \times \{(b_s,S,\text{on})\}\\
&(q_1^{\text{replica}},(b_n,S)) \to \{q_2^{\text{replica}}\} \times \{(b_n,N,\text{on})\}\\
&(q_2^{\text{replica}},(b_e,W)) \to \{q_3^{\text{replica}}\} \times \{(b_e,E,\text{on})\}\\
&(q_3^{\text{replica}},(b_w,E)) \to \{q_4^{\text{replica}}\} \times \{(b_e,E,\text{on})\}\\
&(q_4^{\text{replica}},(s_4,E)) \to \{q_4^{\text{replica}}\} \times \{(f,E,\text{off})\}\\
&(q_4^{\text{replica}},(s_4,W)) \to \{q_4^{\text{replica}}\} \times \{(f,W,\text{off})\}\\
&(q_4^{\text{replica}},(s_4,N)) \to \{q_4^{\text{replica}}\} \times \{(f,N,\text{off})\}\\
&(q_4^{\text{replica}},(s_4,S)) \to \{q_4^{\text{replica}}\} \times \{(f,S,\text{off})\}
\end{aligned}
$$

Note that the first two transitions of the GSM in tiles $T_i, 1 \leq i \leq 4$ are applied during the two attachment transitions that attach the tile to the assembly and that, after this attachment stage, the GSM's in all tiles in the north-east quarter-rectangle are in a state q_0^j, $1 \leq j \leq 4$.

The role of the "signal" s_1 (a signal is a glue with strength 0) is to carve a supertile-shaped hole from the rectangle. It starts in tile T_5 (see Figure 8) and travels clockwise through the tiles right outside of the replica, deactivating the glues on the replica border tiles, and changing their GSM states to q_1^j, $1 \leq j \leq 6$. After arriving in T_6, this signal continues travelling through the other three quarter-rectangles, deactivating the glues of the replica border tiles on its way, until it arrives back to T_5. At this moment, the supertile-shaped middle of the rectangle detaches completely, leaving a supertile-shaped hole.

At this point, tile T_5 activates the "signal" s_2 (glue with strength 0). This signal then travels clockwise through the same path as s_1, along the tiles right outside the supertile-shaped hole, making their inside edges "sticky", and changing the states of their GSM's to q_2^j, $1 \leq j \leq 6$. This is in preparation for the next step (the assembly of the supertile-shaped replica). Making the inside edges of the tiles sticky is achieved by the use of a single new glue, f, which is activated by s_2. At this moment, all computational devices in the tiles T_1, T_3 ,T_4 in the north-east quarter-rectangle are in state $q_2^j, j \in \{1, 2, 4\}$. Moreover all tiles T_2 in the north-east quarter rectangle are in state q_4^2.

At the end of *Step 3*, we have a rectangle underneath that input supertile, which has a supertile-shaped hole with sticky edges in the middle.

The fact that the input supertile is L-convex is essential for *Step 3*. Due to the L-convexity of the shape, Lemma 3.4 guarantees that the attachment of the tiles T_1 through T_4 from $\Theta_{\textit{border-ne}}$ does not interfere with the construction of the border of the supertile-shaped hole in the other quarter-rectangles. Indeed, the one problem that could have arisen in this step that, because the system is non-deterministic, the four quarter-rectangles can assemble independently and potentially overlap (see Figure 8) and this could interfere with signal transmission. However, we note that the inside "straight" borders of these quarter-rectangles coincide with the path between two of the "corner" tiles of the replica that was mentioned in Lemma 3.4, augmented with two end-tiles. Due to Lemma 3.4, this entire path lies on the inside of the replica. Together, these guarantee that any intersection between two quarter-rectangles lies inside the replica, and since both signals s_1 and s_2 travel right outside the replica, this implies that the paths of the signals never fall inside an overlap. Thus, any such overlap is unimportant, since it does not interfere with signal transmission and, moreover, any overlap tiles will anyway be all detached in the process of carving the supertile-shaped hole.

Step 4. At the start of this step, tile T_5 activates signal s_3 (which will later be used for starting the detachment of the constructed replica). In this last step, the one tile from the tile set $\Theta_{\textit{replica}}$, see Figure 9, fills in the supertile-shaped hole as follows.

Tile $T_{\textit{replica}}$ has glue f^+ and f_1^- as well as glues b_n, b_e, b_s, b_w, on all of its edges. Glue f has strength 1 and is used to attach each tile inside the replica to its neighbours in the replica. Glue f_1 has strength 1 and is used to make sure that the replica is τ-stable, by strengthening the power of the attachment of the corner tile to the replica (see transitions corresponding to $T_{\textit{replica}}$). Glues b_n, b_e, b_s, b_w have strength 0 (are signals), and are used to guarantee that the replica is hole-free. Indeed, signal b_n is sent by each tile from the south edge of the replica to the north, and b_s is sent in the opposite direction. Similarly, signal b_e is sent from each tile on the west edge of the replica to the east edge, and b_w is sent in the opposite direction. The fact that these signals have all reached the opposite edge means that the supertile-shaped hole is completely filled in. This fact is recorded by the GSM's of the tiles situated immediately outside the replica, on its outside border, changing their state

from q_2^j to q_3^j, $1 \leq j \leq 6$ (via state q'^1_3 in the case of T_1). In addition, when each tile in the replica has passed all four signals, b_n, b_e, b_s, b_w, the state of its GSM changes to $q_4^{replica}$, a final state.

When signal s_3, after having started in tile T_5, comes back to T_5, after having travelled clockwise, right outside the border of the replica, if the GSM of T_5 is in state q_3^5 (indicating that the supertile-shaped hole has been filled with the replica) then tile T_5 initiates signal s_4 which travels along the outside border of the replica and deactivates the outside glues of the tiles on the border of the replica. As result of this second part of *Step 4* the replica, whose tiles all have their GSM's in a final state, detaches from the scaffold structure.

Note that the replica of the input supertile is τ-stable. Indeed, all the tiles in the replica are from the tile set Θ_{replica} and have at least two common edges with the tiles from the same tile set, with maximum four exceptions. These exceptions are the tiles that are attached to the four special "border" tiles (representatives of the north-west, north-east, south-east and south-west edges), as these tiles might have only one common edge with tiles from the replica. However, the attachments between these special tiles and the replica have been strengthened by the activation of the glue f_1, and now suffice to keep such special tiles attached to the replica. □

The smart-TAM system in Theorem 3.6 can be used, with slight modifications, to replicate any convex shape (not necessarily L-convex). Indeed, the only part in the construction that relied on the input supertile being L-convex was *Step 3*, that fills in the rectangular hole in which the replication will take place. Here, the problem was the potential overlap between the north-east, south-east, south-west and nort-west quarter-rectangles (Figure 8) could potentially interfere with the signal transmission. In our construction, this problem was avoided by the assumption of L-convexity and Lemma 3.4 and Corollary 3.5. However, this situation can also be handled in a different way, by using (additional) different signals in each of the quarter rectangles. We did not use this solution, and preferred adding the L-convex assumption instead, because using more signals would have lead to the GSM's having more states and being more complex.

4. Discussion and future work

Note that the tile complexity (number of tiles) and the space complexity (the "area" of the computation) of the construction of the smart-TAM replicating systems in Theorem 3.6 can be further reduced if we use a more complex tile computational device. The main idea is illustrated in Figure 11.

Basically, in order to replicate the original supertile, the smart-TAM system should be able to read the edges and send signals accordingly. During each phase of the replication, the system attaches a new tile (labelled with "A") to the border of the original supertile that reads the glue on one of the free edges of a tile on the border of the original supertile, and sends a signal to the position where this edge will be replicated. Unlike the construction in this paper, this signal has to travel back to the next free-edge on the original supertile, and be reused to repeat the process. The complexity of the GSM will increase accordingly, as it has to keep track of the position of the last tile from where the signal was sent. When the border structure (labelled with "B") is complete, the interior is filled in, forming a replica of the original shape. The tiles that fill the inside of the supertile-shaped hole can be similar to the tile T_{replica}. This smart-TAM system for the replication of arbitrary shapes has lower tile and space

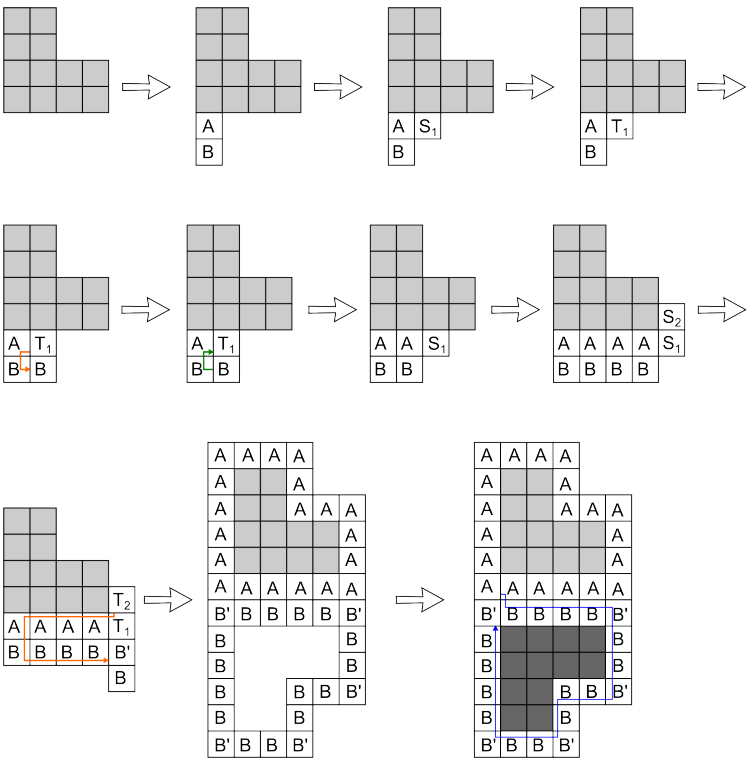

Figure 11. High-level design of another smart tile assembly system that replicates arbitrary shapes, and which uses a method similar to the smart sand robotic system replication of shapes [6].

complexity than the smart-TAM system constructed in Theorem 3.6, but this comes at the expense of an increased complexity of the computational devices on the tiles.

Acknowledgements:

This research was supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant to L.K.

References

[1] Oh H, Shirazi AR, Sun C, Jin Y. Bio-inspired self-organising multi-robot pattern formation: A review. Robotics and Autonomous Systems, 2017;91:83–100. URL `https://doi.org/10.1016/j.robot.2016.12.006`.

[2] Werfel J. Building blocks for multi-robot construction. In: Alami R, Chatila R, Asama H (eds.), Distributed Autonomous Robotic Systems 6. Springer Japan, Tokyo, 2007 pp. 285–294. doi:10.1007/978-4-431-35873-2_28.

[3] Werfel J. Collective construction with robot swarms. In: Doursat R, Sayama H, Michel O (eds.), Morphogenetic Engineering, Toward Programmable Complex Systems, pp. 115–140. Springer, 2012. doi:10.1007/978-3-642-33902-8_5

[4] Naz A, Piranda B, Bourgeois J, Goldstein SC. A distributed self-reconfiguration algorithm for cylindrical lattice-based modular robots. In: Pellegrini A, Gkoulalas-Divanis A, di Sanzo P, Avresky DR (eds.), 15th IEEE International Symposium on Network Computing and Applications, NCA 2016. IEEE Computer Society, 2016 pp. 254–263. doi:10.1109/NCA.2016.7778628.

[5] Rubenstein M, Cornejo A, Nagpal R. Programmable self-assembly in a thousand-robot swarm. Science, 2014;345(6198):795–799. doi:10.1126/science.1254295.

[6] Gilpin K, Rus D. A distributed algorithm for 2D shape duplication with smart pebble robots. In: Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2012. IEEE, 2012 pp. 3285–3292. doi:10.1109/ICRA.2012.6225227.

[7] Derakhshandeh Z, Gmyr R, Richa AW, Scheideler C, Strothmann T. Universal shape formation for programmable matter. In: Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA, 2016 pp. 289–299. doi:10.1145/2935764.2935784.

[8] Patitz MJ. An introduction to tile-based self-assembly and a survey of recent results. Natural Computing, 2014;13(2):195–224. doi:10.1007/s11047-013-9379-4.

[9] Padilla JE, Liu W, Seeman NC. Hierarchical self assembly of patterns from the Robinson tilings: DNA tile design in an enhanced Tile Assembly Model. Natural Computing, 2012;11(2):323–338. doi:10.1007/s11047-011-9268-7.

[10] Padilla JE, Patitz MJ, Pena R, Schweller RT, Seeman NC, Sheline R, Summers SM, Zhong X. Asynchronous signal passing for tile self-assembly: fuel efficient computation and efficient assembly of shapes. In: Proceedings UCNC 2013. 2013 pp. 174–185.

[11] Jonoska N, Karpenko D. Active tile self-assembly, Part 1: universality at temperature 1. International Journal of Foundations of Computer Science, 2014;25(2):141–164. URL https://doi.org/10.1142/S0129054114500087.

[12] Benenson Y, Gil B, Ben-Dor U, Adar R, Shapiro E. An autonomous molecular computer for logical control of gene expression. Nature, 2004;429(6990):423–429.

[13] Komiya K, Sakamoto K, Gouzu H, Yokoyama S, Arita M, Nishikawa A, Hagiya M. Successive state transitions with I/O interface by molecules. In: Condon A, Rozenberg G (eds.), 6th International Workshop on DNA-Based Computers, DNA 2000, volume 2054 of Lecture Notes in Computer Science. Springer, 2000 pp. 17–26. doi:10.1007/3-540-44992-2_2.

[14] Qian L, Winfree E. Scaling up digital circuit computation with DNA strand displacement cascades. Science, 2011;332(6034):1196–1201. doi:10.1126/science.1200520.

[15] Winfree E. Algorithmic Self-Assembly of DNA. Ph.D. thesis, 1998.

[16] Keenan A, Schweller RT, Zhong X. Exponential replication of patterns in the signal tile assembly model. Natural Computing, 2015;14(2):265–278. doi:10.1007/s11047-014-9431-z.

[17] Chalk C, Demaine ED, Demaine ML, Martinez E, Schweller R, Vega L, Wylie T. Universal shape replicators via self-assembly with attractive and repulsive forces. In: Proceedings of ACM-SIAM Symposium on Discrete Algorithms, SODA '17. Philadelphia, PA, USA, 2017 pp. 225–238.

[18] Hendricks J, Patitz MJ, Rogers TA. Replication of arbitrary hole-free shapes via self-assembly with signal-passing tiles. In: Calude CS, Dinneen MJ (eds.), UCNC 2015, Proceedings, volume 9252 of Lecture Notes in Computer Science. Springer, 2015 pp. 202–214. doi:10.1007/978-3-319-21819-9_15.

[19] Abel Z, Benbernou N, Damian M, Demaine ED, Demaine ML, Flatland RY, Kominers SD, Schweller RT. Shape replication through self-assembly and RNase enzymes. In: Charikar M (ed.), Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA. SIAM, 2010 pp. 1045–1064. URL https://doi.org/10.1137/1.9781611973075.85.

[20] Cheng Q, Aggarwal G, Goldwasser MH, Kao MY, Schweller RT, de Espanés PM. Complexities for generalized models of self-Assembly. SIAM Journal on Computing, 2005;34:1493–1515. URL https://doi.org/10.1137/S0097539704445202.

[21] Demaine ED, Demaine ML, Fekete SP, Ishaque M, Rafalin E, Schweller RT, Souvaine DL. Staged self-assembly: nanomanufacture of arbitrary shapes with O(1) glues. Natural Computing, 2008;7(3):347–370. doi:10.1007/s11047-008-9073-0.

[22] Brun Y. Improving efficiency of 3-SAT-solving tile systems. In: Sakakibara Y, Mi Y (eds.), DNA Computing and Molecular Programming, DNA 16, volume 6518 of Lecture Notes in Computer Science. Springer, 2010 pp. 1–12. doi:10.1007/978-3-642-18305-8_1.

[23] Seki S. Combinatorial optimization in pattern assembly - (Extended Abstract). In: Mauri G, Dennunzio A, Manzoni L, Porreca AE (eds.), UCNC 2013, Proceedings, volume 7956 of Lecture Notes in Computer Science. Springer, 2013 pp. 220–231. doi:10.1007/978-3-642-39074-6_21.