Signal Passing Self-Assembly Simulates Tile Automata

Angel A. Cantu
Department of Computer Science, University of Texas – Rio Grande Valley, TX, USA
angel.cantu01@utrgv.edu

Austin Luchsinger
Department of Electrical and Computer Engineering, University of Texas at Austin, TX, USA
amluchsinger@utexas.edu

Robert Schweller
Department of Computer Science, University of Texas – Rio Grande Valley, TX, USA
robert.schweller@utrgv.edu

Tim Wylie
Department of Computer Science, University of Texas – Rio Grande Valley, TX, USA
timothy.wylie@utrgv.edu

Abstract

The natural process of self-assembly has been studied through various abstract models due to the abundant applications that benefit from self-assembly. Many of these different models emerged in an effort to capture and understand the fundamental properties of different physical systems and the mechanisms by which assembly may occur. A newly proposed model, known as Tile Automata, offers an abstract toolkit to analyze and compare the algorithmic properties of different self-assembly systems. In this paper, we show that for every Tile Automata system, there exists a Signal-passing Tile Assembly system that can simulate it. Finally, we connect our result with a recent discovery showing that Tile Automata can simulate Amoebot programmable matter systems, thus showing that the Signal-passing Tile Assembly can simulate any Amoebot system.

2012 ACM Subject Classification Theory of computation → Models of computation; Applied computing → Computational biology

Keywords and phrases self-assembly, signal-passing tile assembly model, tile automata, cellular automata, simulation

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.53

Supplementary Material Supplemental videos of example STAM signal tools used in this paper may be found at `https://asarg.hackresearch.com/main/isaac-2020/`.

Funding This research was supported in part by National Science Foundation Grant CCF-1817602.

1 Introduction

In this paper we explore the connection between two previously studied models of active self-assembly: the Signal-Passing Tile Assembly Model (STAM) [5, 6, 8, 10], a tile self-assembly model in which signals are passed based on a DNA strand-displacement mechanism [11], and the Tile Automata (TA) model [1, 2, 3], a recently proposed mathematical abstraction of active self-assembly that merges tile self-assembly and asynchronous Cellular Automata [7]. We show that any TA system can be simulated by a corresponding STAM system.

Tile Automata and the Signal-Passing Tile Assembly model are models of active self-assembly that serve two different purposes. The STAM provides a method for tiles within a self-assembly system to turn glues on and off based on glue attachments, which are motivated by a simple DNA strand-displacement mechanism. Due to its direct tie to a

DNA implementation, along with successful experimental implementation [11], the STAM provides a direct path to implement active tile self-assembly constructions with DNA strand-replacement methods. Tile Automata, on the other hand, is an intentional mathematical abstraction designed to implement the key features of active algorithmic self-assembly while avoiding specifics tied to any one particular implementation (using state change rules and tile attachments/detachments based on local affinities between states). By abstracting away implementation details, TA strives to serve as a proving ground for exploring the power of active algorithmic self-assembly, along with providing a central *hub* model through which various disparate models of self-assembly can be related by way of comparison to TA. One recent example of this type of application includes [1] in which TA is shown capable of simulating the *Amoebots* model [4] of programmable matter.

As Tile Automata seeks to serve as a model for examining the intrinsic power of active self-assembly systems, it is crucial that the features of the model are based on an experimentally plausible foundation. Obtaining this foundation is exactly the focus of this paper: we show that any TA system may be implemented with a STAM system, which has a direct connection to a DNA strand-replacement implementation. Since the features of TA are quite natural and may have numerous potential implementations, ours may not be the only or simplest. However, this connection gives one such path for implementation explicitly. Further, this provides a new way to program for STAM systems by allowing a programmer to solve a problem with the simpler and more powerful rules of a TA system, and then *compile* the system into a STAM system. As more models are connected to TA, this expands the *programming languages* available for STAM systems. For example, with this work and the work of [1], we now have a proposed DNA implementation of Amoebots [4], as well as a new method for which to program the STAM through a powerful model of programmable matter.

To show that STAM simulates TA, our approach has three key steps. First, we define a limited subset of TA based on the key features of TA that are particularly difficult to simulate within STAM. Second, we prove that this limited version of TA can still simulate regular TA at scale. Third, we use macro-tiles in the STAM to implement this limited version of TA at scale. With these components, we get STAM simulating TA.

Limited Tile Automata. The limited version of TA addresses the following features.

- STAM tiles send signals based on DNA strand-displacement mechanisms. However, based on the motivating implementation, these signals are used up after each firing, implying that each STAM tile has a limited number of signals it may fire before becoming inert. In contrast, TA tiles may cycle through a given state arbitrarily many times without becoming used up. To bring TA closer to STAM, we focus on *freezing* TA systems [3] which limit state transition rules so that a tile may not revisit a state, implying that each TA tile will eventually become inert (or *freeze*).
- General TA includes the ability to flip the state of a pair of tiles in a single step. In contrast, STAM system signal passing must adjust each of a pair of STAM tiles one at a time, in some order, causing an inaccurate simulation, as well as race conditions. To address this we consider TA systems with limited rule sets that never change both states of an adjacent pair within a single rule.
- Motivated by the DNA strand-displacement implementation, STAM signals fire as a result of a bonding between two glues, implying signals are only fired between two tiles which are stuck together by some positive strength force. In response, we consider limited TA rules which only induce state changes between state pairs with positive strength affinities.

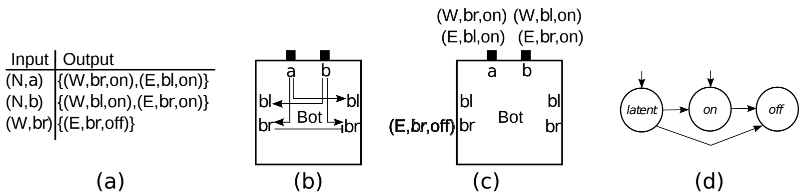


Figure 1 (a)–(c) An example signal tile. (d) The acyclic graph which represents the transitioning of glue labels.

- Preventing race conditions among the sides of a STAM tile is difficult when attempting to adjust the state of a STAM tile in a single conceptual step. To address this, we consider limited TA systems for which a TA state only changes state based on adjacency from one of the four cardinal directions.

Together, we refer to the model of TA obtained by applying each of these limitations as the *Freezing, Single-Rule, Bonded, Same-sided* Tile Automata model (FRBS).

We begin with a high-level overview of the definitions and models in Section 2. In Section 3, we discuss the key techniques used to allow each limited variant of TA to individually simulate general TA, and then show in Section 4 that FRBS simulates all of TA. We then show in Section 5 that STAM simulates FRBS, to get our main result that STAM simulates TA. Finally, we look at future work in Section 6.

2 The Models

This work focuses on two models of self-assembly, the signal-passing tile assembly model and the Tile Automata model. Here, we provide a brief description of the two models. We refer the reader to [10] for formal STAM definitions and [3] for formal TA definitions.

2.1 Signal Tile Model

The *signal-passing tile assembly model* (STAM) [5, 6, 8, 9, 10] is a model of self-assembly which considers semi-intelligent monomers. This form of self-assembly, known as *active* self-assembly, allows for system monomers to react to their environment. In the STAM, system monomers are tiles which have sets of glues on each edge (as opposed to only one glue per side as in the 2HAM). The glues in these sets are either **on**, **off** or **latent**. Only glues in the **on** state may be used for tile attachment. STAM tiles (Figure 1) also each implement a *transition function* which produces *output actions* that may change the state of a particular glue. Upon the binding of two tiles, the output actions of each tile's corresponding transition functions are all instantly queued into a set of pending actions. This allows STAM tiles to "pass signals" through a series of these transition functions.

2.2 Tile Automata

The Tile Automata model is a marriage between cellular automata and 2-handed self-assembly. TA systems consist of a set of monomer states, along with local affinities between states denoting the strength of attraction between adjacent monomers in those states. A set of local state change rules are included for pairs of adjacent states. Assemblies in the model are created from an initial set of starting assemblies by combining previously built assemblies given sufficient binding strength from the affinity function (binding strength that meets or

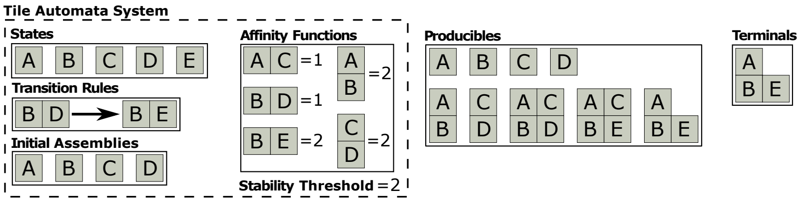
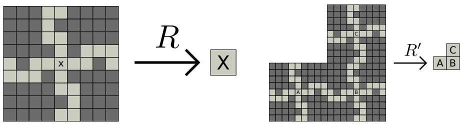


Figure 2 An example TA system. Recursively applying the transition rules and affinity functions to the initial assemblies of a system yields a set of producible assemblies. Any producibles that cannot combine with, break into, or transition to another assembly are considered to be terminal.



$$\text{STAM} \xrightarrow[\text{Lem. 3}]{} \text{FRBS} \xrightarrow[\text{Lem. 1}]{} \text{F} \xrightarrow[\text{Thm. 1 in [3]}]{} \text{TA}$$

(a) **(b)** **(c)**

Figure 3 (a) Examples of m-block representation and mapping. The partial function R takes a macro-block and maps it to a state in the state space of some other system. (b) The function R' takes a positioned assembly, containing m-blocks, and maps it to a positioned assembly over the state space of the other system using the m-block representation function to perform the mapping. (c) Visual depiction of the simulation roadmap. For clarity, the results are presented in reverse order with the first result coming from previous work [3], Lem. 1 in Sec. 4, and finally Lem. 3 in Sec. 5.

exceeds a given parameter known as a stability threshold). Further, existing assemblies may change states of internal monomer tiles according to any applicable state change rules. An example system is shown in Figure 2.

In order to compare the capability of various models, we formulate our notion of simulation. This is an oft-argued upon concept (what it means to simulate), but the goal is to capture a reasonable idea of simulation while allowing for small amount of flexibility.

Simulation

We use the same notion of simulation as was presented in [3]. At a high-level, for system B to simulate system A means there is a surjective mapping from B to A so that every producible in B uniquely maps to a producible of A. At a constant scale, we call each mapped tile an m-block representation. Figure 3a shows this for a 9-block representation, and Figure 3b shows the mapping with a 3-tile producible. We require that every producible in A has a producible in B that maps to it. This definition of simulation guarantees that the systems have the same producibles and terminals (at scale in system B).

3 High-Level Simulation Roadmap

In this section we discuss a set of limitations we impose on the general TA model to allow for the simplest possible simulation within the STAM. Each imposed limitation addresses a particular aspect of TA that does not have a clear corresponding aspect within STAM. After simulating this limited version of TA with STAM, we then show how even this highly limited version of TA is still able to simulate general TA.

Freezing Tile Automata (F) are TA systems whose rules are restricted so that a tile can never revisit any state twice. *Single-Rule Transition (R)* TA systems only change one state of a given state-pair of tiles, as opposed to simultaneously changing both of the state-pairs.

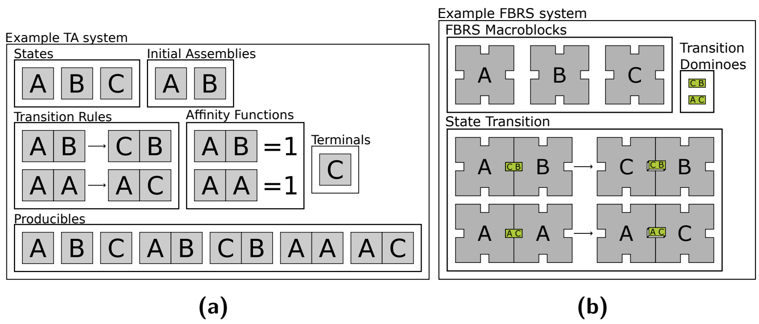


Figure 4 An example TA system and the corresponding FBRS macroblocks and transition dominoes.

Bonded Transition Rule (B) TA systems only utilize transition rules between state-pairs with positive bonding affinity. *Same-sided Transition Rule (S)* TA systems require that all rules which change a given state to a new state utilize the same relative orientation (e.g., state A only transitions to a new state due to state-pair rules with A *on the right*); there do not exist any transition rules with A transitioning except horizontal transitions with A on the right.

Simulation Chain Through TA Variants. In order to connect the STAM simulation to general TA, we chain simulation results through the collection of TA variants. It is important to note that simply connecting each variant to general TA is not enough. We must show that a system with a *combination* of these variants may also simulate general TA.

4 FRBS → F: Freezing Single-Rule Bonded Same-Sided Tile Automata simulates Freezing Tile Automata

This simulation utilizes the result from [3], which proves that freezing TA systems can simulate non-freezing systems. We show that a limited TA variant, freezing single-rule bonded same-sided (FRBS) Tile Automata, is capable of simulating freezing TA, and thus general TA. This section is separated into three parts. First, we present a high-level overview of different aspects of the simulation. We then introduce several preliminary concepts that are crucial tools used throughout the simulation. Lastly, we present the construction details for designing FRBS TA systems which are capable of simulating freezing TA systems.

4.1 High-Level Concepts

Macroblock Simulation. This simulation uses constant-scaled macroblocks. Figure 4 sketches what a FRBS macroblock representation for a given general TA system might look like. Each macroblock encodes a state from the original system. Notice the macroblock boundaries have a particular geometry which allows dominoes to cooperatively attach to adjacent macroblocks. This property will be used throughout the simulation construction.

Machinery for Simulating Each Variant Limitation. The simulation construction is presented in sections, each of which details which mechanics are used to overcome the limitations of a particular TA variant. The next paragraphs provide high-level descriptions of the tools used to accomplish this.

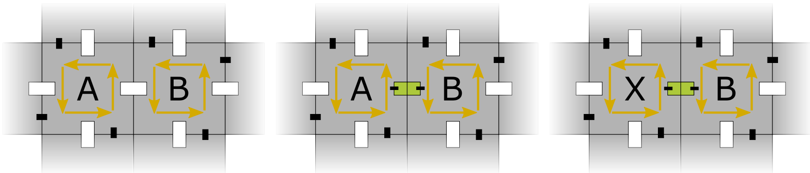
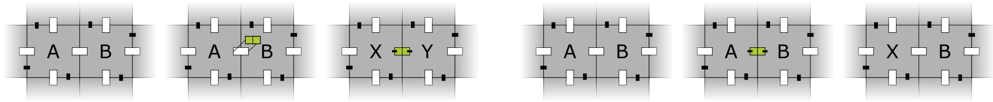


Figure 5 The primary idea behind the same-sided portion of the simulation is that individual signals cannot be bidirectional. To overcome this, we ensure that all internal macroblock signals propagate in the same counterclockwise direction, and all inter-macroblock communications use temporary transition dominoes.



(a) **(b)**

Figure 6 (a) In order for a single-transition system to simulate a double-transition system, we use the attachment of a transition domino. The mapping of both macroblocks can be changed simultaneously with the attachment of such a domino. (b) The key concept behind the bonded portion of the simulation is the transition domino. Adjacent macroblocks (bonded or not) allow for the attachment of a transition domino which may allow communication of transition information.

Same-sided Transitions. Figure 5 presents the key idea for solving the problem of same-sided transitions. Since this limitation restricts the direction from which a state can be changed, we present a directional communication scheme. Signals within macroblocks always travel counterclockwise while we utilize the temporary attachment of transition dominoes to allow bidirectional communication between macroblocks.

Single-transitions. The transition domino is utilized in a different way to overcome the problem of single transitions (Figure 6a). Since this variant limits transition rules within a system to only change one of the two states involved, simultaneous double-transitions cannot be achieved via state changes alone. Thus, we allow the cooperative attachment of the transition domino to change the mapping of both macroblocks the moment it attaches.

Bonded-transitions. When trying to handle the bonded-transition restriction, we turn to the transition domino yet again. With this TA variant, non-bonded state pairs are not allowed to have transition rules. This can be simulated with the scaled macroblock simulation, as shown in Figure 6b. Two adjacent macroblocks (that are part of the same assembly) allow the cooperative attachment of a transition domino regardless of whether or not the macroblocks themselves are bonded. We can then use the domino bonds to communicate state change information between the macroblocks.

4.2 Construction Preliminaries

Motivation for Scale Factor. First, we provide some brief motivation for why this is not done at scale-1. The primary motivation comes from trying to simulate a double-transition system with a single-transition system. Figure 7a shows an example system with a double-transition that can not be simulated at scale-1 by a single-transition system. The system only uses states B and B' after a double rule transition. Thus, neither single tile with states B nor B' are producibles; only the combined assembly of both is producible. Similarly, there is no producible with an $A(A')$ and $B(B')$ state combined. Any single-transition system would have to allow one of these cases to occur.

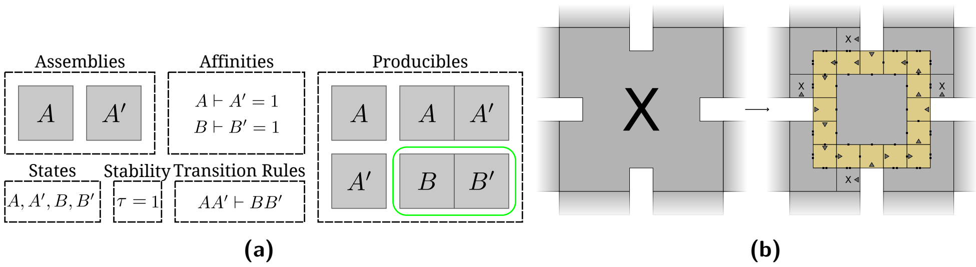


Figure 7 (a) Motivation for scale factor. (b) Details of macroblock sections. Glue tiles represent the current state of the macroblock X and are placed along the edges to bond with other macroblocks. The wire is the circular network inside the macroblock (in yellow), encompassing and encompassed by filler tiles. Strength-1 and strength-2 affinities are depicted by single and double squares, respectively.

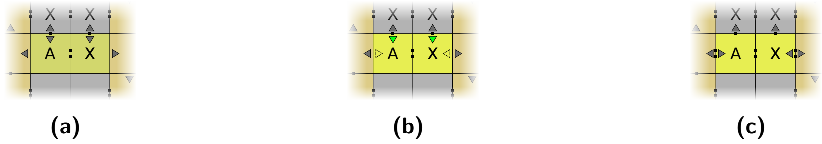


Figure 8 When a transition domino attaches cooperatively with a pair of glue tiles, the transition directions can be redirected with the series of state changes shown.

Macroblock. For each state tile in a TA system, we construct and map a scale-7 macroblock composed of three sections: *glue* tiles, *filler* tiles, and *wire* tiles. Filler tiles are used to maintain the structure and connectivity of the macroblock, especially when some tiles detach from the macroblock. Also, these tiles never change their states except at the beginning when macroblocks are generated using the method from [3]. Glue tiles are a way to have macroblocks bond with other macroblocks according to the affinity function of the TA system it simulates. For example, if state tiles A' and B' in a TA system bond s.t $A' \vdash B' \geq \tau$, then the west and east glue tile of A and B, respectively, bond the same way (thus allowing macroblocks to bond and mimic the affinity function of the TA system).

Wires. To have macroblock-level state transitions, we use the *wire* system of each macroblock to communicate the activity of its four edges, looking out for possible state transitions. Wires are interconnected series of state tiles that can cascade a series of state changes throughout its structure. For instance, a state s inside the wire is said to "propagate" if every state tile of the wire reacts to it by copying the state s onto itself, thus moving the s state across the wire. These states that propagate are called *signals*. The state tiles are defined by the description of the task they achieve and the manner that they propagate through the wire. Each section of the macroblock is detailed in Figure 7b.

Transition Dominoes. The macroblock's structure is designed to induce a 1×2 or 2×1 empty area next to the glue tiles when two of them bond together, allowing a *transition* domino to cooperatively attach with the glue tiles. Transition dominoes are two state tiles with a τ strength bond that represent the transition that a macroblock can undergo. For example, the transition domino illustrated in Figure 8 represents the transition rule $(X, X, A, X, \vdash)$ that can be realized with the transition domino. When the transition directions point outwards, the wire tiles will be able to initiate the processes necessary for macroblock-state transitions.

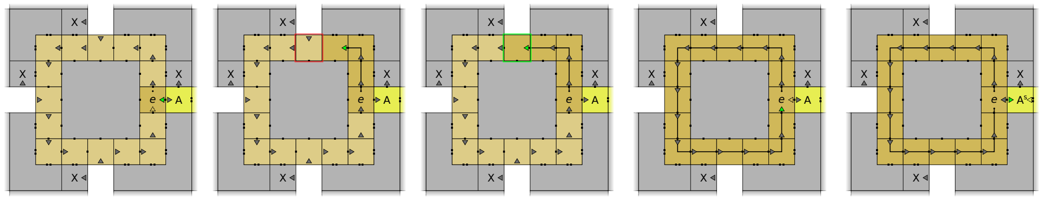
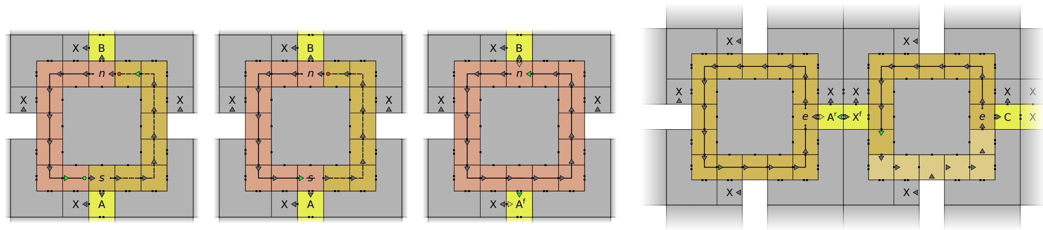


Figure 9 Depiction of a transition signal label propagating from state e. A red outline around a tile indicates tile detachment, whereas green indicates tile attachment. Cycle completion is represented by the *success* state A^s.



(a) **(b)**

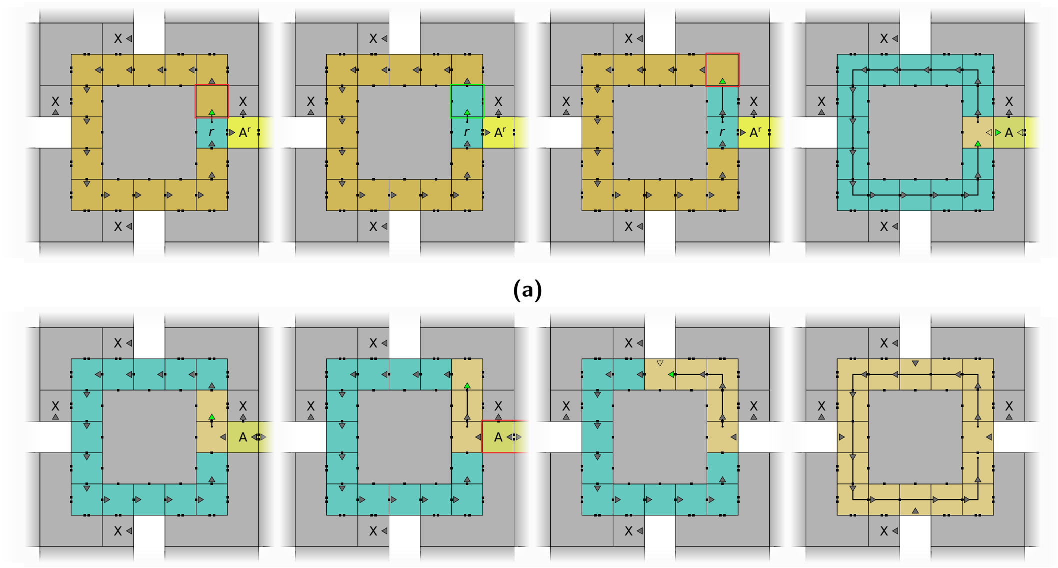
Figure 10 (a) Overriding illustration of the transition signal originating from the s state by the signal originating from the n state. This event is then detected by the failed transitioned domino with the *fail* state A^f. (b) An example of two transition dominoes competing to cycle a macroblock's wire. Given that the middle transition domino's right state tile is the failed state X^f, the left state tile proceeds by representing the *replace* state A^r.

4.3 FRBS $\rightarrow$ F: Freezing Single-Rule Bonded Same-Sided Tile Automata simulates Freezing Tile Automata Construction Details

Here, we present the details for each part of the simulation construction. We depict the *transition direction* (i.e., the edge of a state tile subject to state transitions) using the ▲, ▼, ►, ◄ symbols for the *south*, *north*, *west*, *east* edge of state tiles, respectively.

Same-Sided. The primary mechanic used in this portion of the simulation is the same as the high-level idea shown in Figure 5. As mentioned, macroblock state transitions are initiated by transition dominoes that bond cooperatively to a pair of glue tiles. This attachment "activates"' the state tiles of the transition domino, making their transition directions point towards the adjacent wire tile. The domino's state tile then transitions the adjacent wire tile to a cardinal direction state $d \in \{n, s, e, w\}$ (i.e., the location where the transition domino attached to a macroblock), which initiates the propagation of a *transition* signal.

Transition signals are used by transition dominoes in order to control a macroblock's wire and prevent other transition signals from initiating. For example, if a transition signal completes a cycle through a macroblock's wire, then it is "decided" that the macroblock must transition according to the transition domino. This is depicted in Figure 9 where, w.l.o.g., the transition domino transitions the adjacent wire tile to the e state, initiating the transition signal that successfully cycles through the macroblock's wire. The state tile in the transition domino then receives notion of the cycle completion from the e state when it transitions with it to represent the *success* state.



(a)

(b)

Figure 11 (a) State tile A^r initiates the replacement signal starting from state r. (b) After the first stage is complete, the wire tiles' transition directions are reinitialized.

Given that many transitions signals may exist within a macroblock simultaneously (race condition), the hierarchy $n > e > s > w$ is induced when transition signals originating from "greater" states are allowed to override the propagation of "lesser" states. For example, applying the hierarchy on the propagation of both transition signals from the n and s states in Figure 10a makes the transition signal from s stop upon reaching the n state, whereas the transition signal from n overrides the other signal's propagation, thus allowing it to complete the cycle. After the transition signal overrides the s state, the state tile in the transition domino transitions to represent the *fail* state (Figure 10a).

A macroblock state transition is said to be possible only if the state tiles in a transition domino have a *success-success* state combination. For *fail-fail* state combinations, the transition domino can simply detach from the assembly. Figure 10b depicts a *success-fail* combination where the middle transition domino controlled the left wire but failed to control the right wire. For *success-fail* combinations, the wire tiles in the *success* side are first replaced with new tiles before having the transition domino detach from the assembly. This replacement scheme begins when the *success* state transitions with the *fail* state to represent the *replace* state, which initiates the propagation of the *replacement* signal in the wire.

The tile replacement scheme is performed in two stages, starting with replacing wire tiles with new "unused" ones (Figure 11a). First, the replacement signal initiated from the domino propagates by having wire tiles detach and replaced with new tiles. In the second stage, the signal performs one final cycle through the wire while redirecting the transition directions (Figure 11b). If another transition domino attaches while the second stage is in progress, it can not propagate its transition signal beyond the tiles placed in the first stage.

For a *success-success* state combination scenario, the transition domino's state tiles first transition with each other to activate a τ-strength affinity with the adjacent macroblocks, followed by the detachment and replacement of the adjacent glue tile with one representing the new state of the macroblock. Afterwards, the state tile in the transition domino representing the new macroblock's state transitions with the new glue tile and points its transition

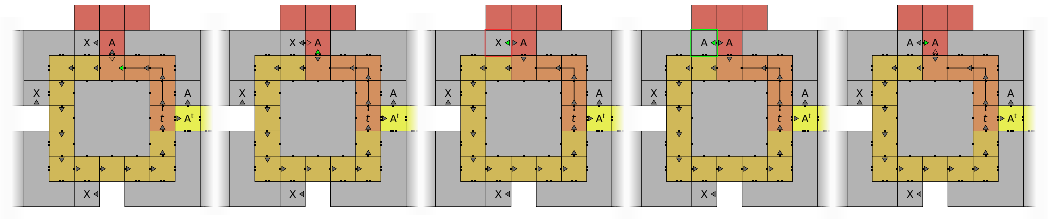


Figure 12 Illustration of a t-shape dummy polyomino attaching to the north side of a macroblock, replacing the X glue tile with A glue tile.

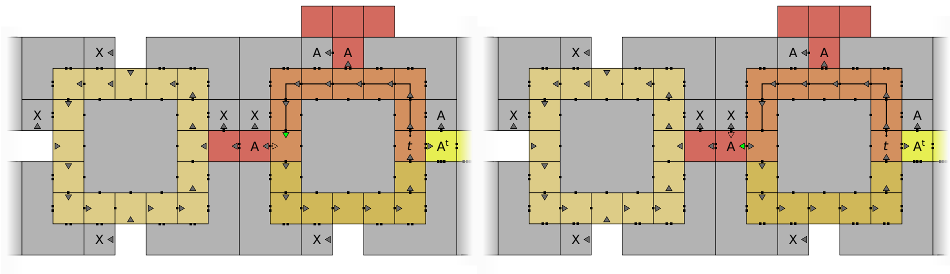


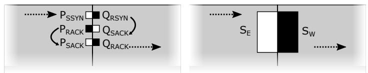
Figure 13 In the case that a transitioning macroblock is adjacent to another, the domino-shaped dummy polyomino attaches and allow the transition process to continue.

directions towards to the wire, initiating the *transition* signal. This signal propagates in the manner shown in Figure 12, allowing t-shape *dummy* polyominoes to attach in the location where transition dominoes bond. These dummy polyominoes allow the signal to continue propagating within it in order to detach and replace the macroblock's glue tile with the new appropriate one. In the case that a t-shaped dummy polyomino can not be used due to the presence of another macroblock, a domino-shaped dummy polyomino attaches cooperatively with the macroblocks wire tiles (Figure 13).

When the transition signal completes a cycle, all of the macroblock's glue tiles will have been replaced with the glue tiles of the new state. When the signal returns to the transition domino, it changes the transition directions of the domino state tile towards the other domino state tile, signaling to it that the transition process is finished. When this occurs, both domino state tiles begin the tile replacement scheme on both macroblocks, which cause the dummy polyominoes to detach. After both replacement schemes are completed, both state tiles in the transition domino remove their τ-strength bond with the adjacent macroblocks and detach from the assembly, thus successfully performing macroblock transitions. With this simulation scheme, the FRBS model can simulate any FRB system.

Single-Transition. We extend the previous simulation scheme to eliminate single-transition restrictions and perform double-transitions by modifying the transition domino. Instead of having only one side of the transition domino send the transition signals, we can have both sides initiate a transition signal for *success*-*success* state combinations. When both transition dominoes represent the *success* state, both can change their transition directions towards the wire tiles adjacent to them and start the *transition* process. After the transition, the tile replacement scheme occurs on both sides of the transition domino, and only until both of the tile replacement schemes finish does the transition domino detach (simultaneously changing both macroblock mappings similar to Figure 6a). This extension shows that a Freezing Single-Rule Bonded Same-Sided system can simulate any Freezing Bonded system.

(a)



(b)

Figure 14 (a) Standard signal passing within the STAM. A simple cascade of signal firings and glue activations. (b) A standard 3-way handshake setup between two STAM tiles, and our abbreviated representation of a 3-way handshake. We use this depiction to emphasize each handshake's role in the larger communication scheme.

Bonded. To tackle this last restriction, we allow transition dominoes to exist for macroblocks that do not necessarily have affinity with one another (as shown in Figure 6b). Since transition dominoes do not require the affinity of two glue tiles to exist for them to perform their function, they can simply bond in-between two macroblocks and perform the same state transitions. Thus, we can eliminate the bonded requirement by simulating a Freezing system using a Freezing Single-Rule Bonded Same-Sided system.

Simulation Results.

▶ **Lemma 1** (FRBS → F). *For any freezing Tile Automata system $\Gamma = (\Sigma, \Lambda, \Pi, \Delta, \tau)$, there exist a freezing single-rule bonded same-sided Tile Automata system $\Gamma' = (\Sigma', \Lambda', \Pi', \Delta', \tau')$ which simulates Γ using a scale-7 macroblock mapping function and $\mathcal{O}(|\Sigma|^2)$ states with a stability threshold of 2.*

Due to space constraints, the detailed proof for this lemma has been omitted.

▶ **Corollary 2** (FRBS → TA). *For any general Tile Automata system Γ, there exist a Freezing Single-Rule Bonded Same-Sided Tile Automata system Γ' which simulates Γ using a 7-block mapping function.*

Proof. This follows from Lemma 1 and Theorem 1 from [3]. ◀

5 STAM simulates Tile Automata/Amoebots

This section is presented in two parts. The first subsection covers several preliminary concepts and provides the STAM details for various tools used throughout the simulation. The second subsection presents the complete idea for the simulation construction, relying on the detailed tools introduced in the first subsection.

5.1 Construction Preliminaries

Here, we cover some key concepts used throughout our simulation method. We refer the reader to https://asarg.hackresearch.com/main/isaac-2020/, where we provide supplemental time-step videos for the details of each of the tools described here.

STAM signal passing. Standard signal-passing is straightforward in the STAM. Tile attachment can begin a cascade of signal firings and glue activations (as depicted in Figure 14a). This is useful when a simple signal is required to travel to a particular location. A drawback of this simple scheme is that an individual STAM tile does not have the information of whether or not a signal has been passed to its neighbor. To remedy this, we show how to execute a simple handshaking scheme with signal tiles in Figure 14b.

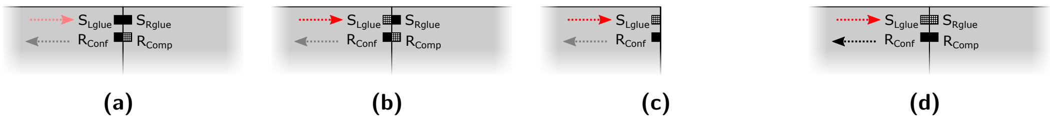


Figure 15 An example which verifies completion of an "off" signal. (a) An off signal is fired from another glue in the STAM tile. Notice that an off signal is denoted by a red arrow. (b) Upon signal execution, the S_{Lglue} turns off. (c) This causes the right STAM tile to detach. (d) A new tile may now attach, sending a signal which confirms that the S_{Lglue} has successfully been turned off.

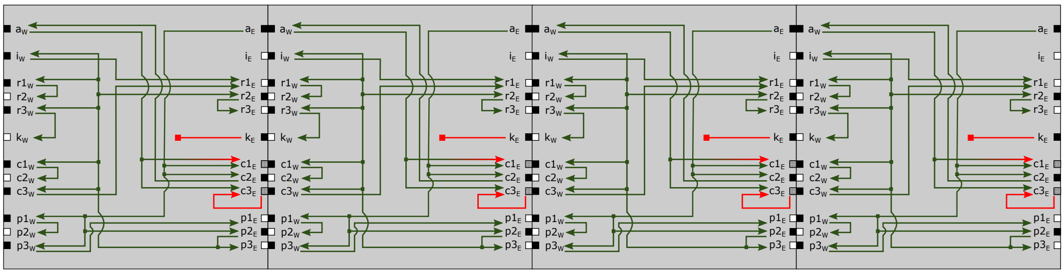


Figure 16 STAM wire replacement scheme details. This method uses standard STAM signals, handshaking signals, and detachment verification to achieve a wire replacement signal. Green arrows represent "on" signals, red arrows represent "off" signals, and the red square represents a signal which turns off many glues on the STAM tile so that detachment may occur.

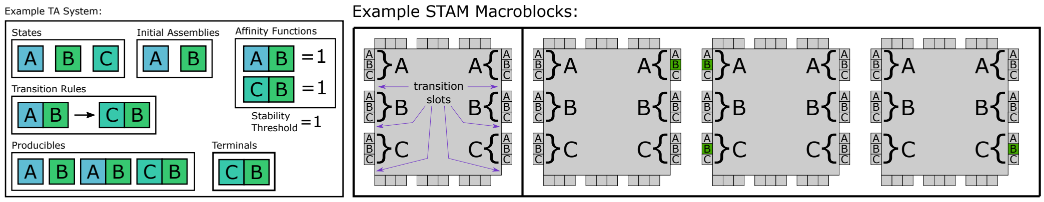
Verified Glue Deactivation. In the STAM, the natural signal-passing method involves turning glues on. Another type of signal relies on turning glues off. A similar straightforward signal may be used to deactivate glues in the STAM; however, it is not immediately clear how to communicate whether or not glue deactivation has occurred. We show a tile detachment/attachment sequence to *verify* when a glue deactivation signal has been executed. This concept is demonstrated in Figure 15, where confirmation of the S_{Lglue} "off" signal is achieved. This is a two-step process in which the tile previously bonded to S_{Lglue} detaches, and a new tile takes its place by attaching to the R_{Conf} glue and sending the confirmation signal.

Macroblocks and Wires. Similar to the simulations of Lemma 1 and [3], this construction also employs a macroblock replacement method. In both of these previous simulations, macroblocks communicated with one another via signals sent along "wires." The tools presented above – standard signal passing, handshaking, and completion of "off" signals – allow us to implement similar wires in STAM macroblocks. We consider the scaled macroblock scheme, in part, because it allows us the ability to detach "expended" STAM tiles and attach "fresh" tiles. We call this method *wire-replacement*.

Wire Replacement. The signal-passing techniques presented allow for non-trivial communication between STAM tiles/assemblies. Since the STAM is naturally a *freezing* model (signals may not be reused), we give a *wire-replacement* scheme allowing us to pass a signal which "refreshes" a communication wire. An example of such a scheme is in Figure 16.

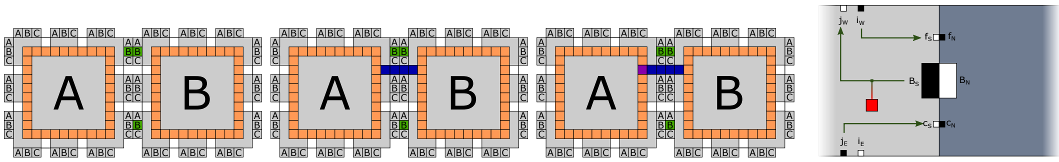
5.2 STAM → FRBS: STAM simulates Freezing Single-Rule Bonded Same-Sided TA

Macroblock Mapping. We use a mapping similar to that shown in Figure 17 in our STAM simulates TA result. The idea is that different TA states are represented by unique glue-state combinations among tiles in a respective STAM macroblock. Furthermore, state transitions



(a) (b)

Figure 17 An example STAM simulation of TA. (a) A freezing single-rule bonded same-sided TA system. (b) A STAM macroblock which mimics the behavior of the states from the simulated TA system. The three macroblocks on the right correspond to states A, B, and C, respectively. Different combinations of active/inactive glues on each of the inner tiles change the state which the macroblock represents.



(a) (b)

Figure 18 (a) A sketch of the transition polyomino triggering STAM signals. (b) The initialization of the signal from Figure 18a. This depicts what happens when the transition polyomino attaches to the macroblock pair. After a brief handshake, two signals are queued. A standard positive signal sent counterclockwise by the j glues, and a detachment signal which will allow a newly attaching tile to begin the wire replacement signal sent by the i glues.

in the given TA model would be represented/executed by corresponding signal firings in the representative STAM macroblocks. The example in Figure 17 shows how a simple TA system can be mapped to a set of STAM macroblocks. All TA affinity rules are represented by the geometric teeth on the edge of the macroblocks. Also, this geometry creates gaps between adjacent macroblocks. These gaps are used in macroblock state changes.

Macroblock State Changes. Since a single TA state is represented by a macroblock, we must ensure that state changes occur in the same manner. Namely, we must ensure that a transition is complete before exposing glues which represent new attachments or transitions that may occur. To accomplish this, we must carefully engineer these macroblocks to perform state changes in a specific way. Below are the steps for simulating the "state change" process:

1. Transition polyomino attaches and initiates signals (Figure 18a).
2. Two opposing signals race to determine if a state change occurs (Figure 19).
3. Attach dummy polyominoes between any adjacent macroblocks or along any empty edge (Phase 1 in Figure 20).
4. Turn on all glues representing new macroblock affinity (Phase 2 in Figure 20).
5. Turn off all glues representing old macroblock affinity (Phase 3 in Figure 21).
6. Refresh signal wire and detach dummy polyominoes (Phase 4 in Figure 21).

Step 1 of the process attempts to initiate a state change by sending two opposing signals around an internal state wire within the macroblock. Details for this initiation are shown in Figure 18b. It should be noted that, since we are simulating same-sided TA systems, we can guarantee that multiple state transitions cannot be queued on the same macroblock.

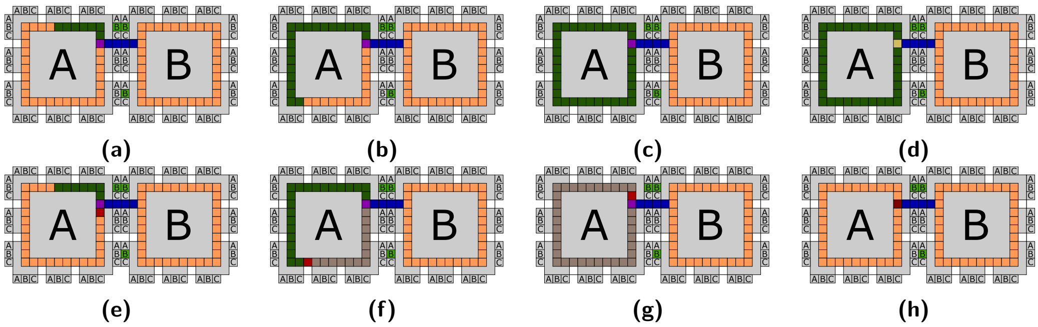


Figure 19 (a-d) A sketch of the transition signal succeeding. (e-h) A sketch of the transition signal failing.

Step 2 of the "state change" process involves clockwise and counterclockwise signals being sent around the internal state wire of the macroblock. The counterclockwise signal is a straightforward STAM signal such as that shown in Figure 14a. This signal halts if it ever encounters another transition polyomino attached to the macroblock. Should this signal finish a complete cycle around the state wire, a state transition is imminent. The clockwise signal is a reset signal, similar to wires from Figure 16. If this signal is ever started, the counterclockwise signal can never complete a cycle around the state wire. This race prevents deadlocking by adding a nondeterministic reset to the state change initiation signal.

Step 3 ensures that the outer edges of a macroblock cannot be involved in any attachments while undergoing a state change. If there are any adjacent macroblocks, dummy "l"-shaped polyominoes attach. Otherwise, dummy "T"-shaped polyominoes attach.

Step 4 sends a signal which activates all macroblock edge glues that represent the corresponding new TA state's affinities. This signal is a straightforward STAM signal like Figure 14a, but the verification of glue activations involves a tile attachment/detachment sequence similar to that of Figure 15.

Step 5 sends a signal similar to the one from step 4. The difference is that this signal is responsible for deactivating all macroblock edge glues which correspond to the old TA state's affinities. This signal is also a combination of a straightforward STAM signal (Figure 14a) and an attachment/detachment verification signal (Figure 15).

Step 6 in the transition process "refreshes" the internal state wire, preparing the proper signals which correspond to the newly represented TA state. This signal is a wire replacement signal very similar to the one shown in Figure 16.

Macroblock Construction. Macroblocks in this system are constructed in the same method as was presented in [3]. This process involves building a *frame* (via a sequence of deterministic single-tile attachments), and then constructs the various initial macroblocks of a system within that frame. The purpose for this is to ensure macroblock construction is complete before exposing any edges that may cause attachments/transitions with other macroblocks within the system. The TA process from [3] involves simple attachments, a single state change per tile, and one detachment phase; thus, it can easily be implemented with straightforward STAM signals (Figure 14a) and handshaking (Figure 14b). No detachment verification is needed, as that is the last step of the macroblock construction process.

▶ **Lemma 3** (STAM → FRBS). *For any freezing single-rule bonded same-sided TA system $\Gamma = (\Sigma, \Lambda, \Pi, \Delta, \tau)$, there exists a STAM system $\Gamma' = (T', \tau')$ which simulates it under the 2-fuzz rule via a $\mathcal{O}(|\Sigma|^2)$-block replacement function.*

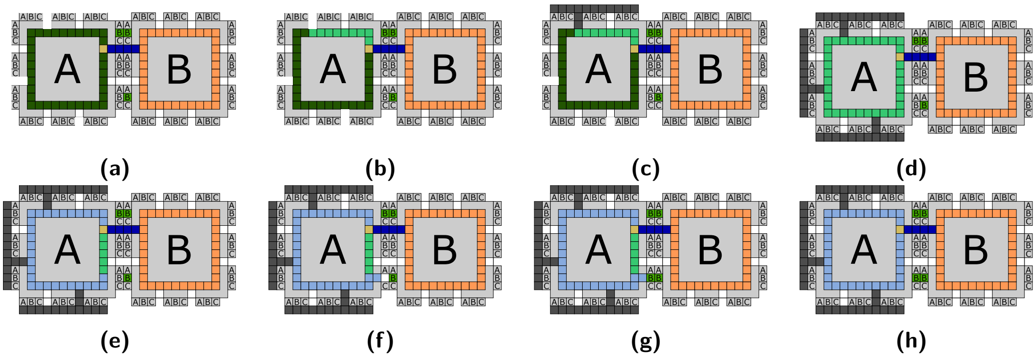


Figure 20 (a-d) Phase 1 of the transition process is responsible for attaching dummy polyominoes to ensure no neighbors are currently transitioning. (e-h) Phase 2 of the transition process sends a counterclockwise signal which activates all glues of the state being transitioned to.

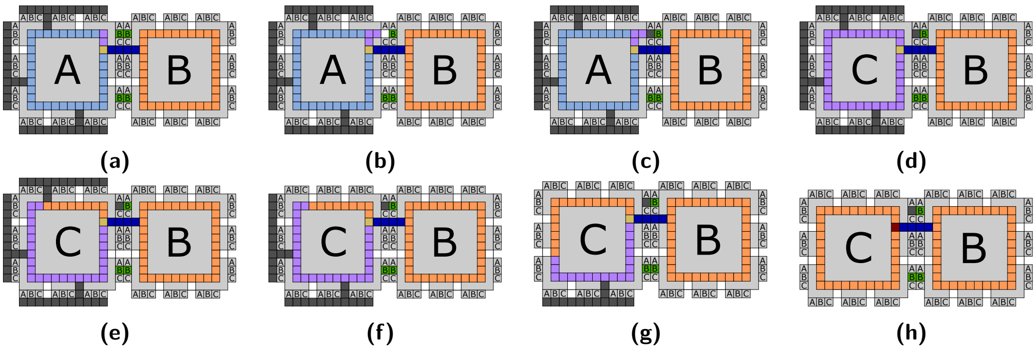


Figure 21 (a-d) Phase 3 of the transition process sends a counterclockwise signal which deactivates all glues of the previous state and replaces them with dummy tiles. (e-h) Phase 4 refreshes the wire for the new state, and detaches all dummy blocks as well as the transition polyomino.

Due to space constraints, the detailed proof for this lemma has been omitted.

▶ **Theorem 4.** *For any general TA system* Γ*, there exists a Signal-passing Tile Assembly system* Γ' *which simulates it.*

Proof. This result follows from Corollary 2, Lemma 3, and Theorem 1 from [3]. ◀

▶ **Corollary 5.** *For any arbitrary amoebot system* Γ*, there exists a Signal-passing Tile Assembly system* Γ' *which simulates it.*

Proof. This result follows from Theorem 4 and Theorem 1 from [1] which shows that general TA is capable of simulating any arbitrary amoebot system. ◀

6 Towards STAM/TA Equivalence

We consider the possibility of simulating STAM within TA as a direction for future work. The STAM, as formulated in [10], was intended to provide a highly asynchronous framework which allows robust behavior independent of physical system parameters. However, their method of instantly queuing the output actions of transition functions allows a subtle, but strong, synchronous operation. With this in mind, we propose an important alteration to the definition of Active Tiles and Transition Functions to address this subtle synchronicity issue.

1. *Double-sided Firing (Queue All).* The transition functions of t_1 and t_2 fire simultaneously.
2. *Single-sided Firing (Queue One Side).* The transition functions of t_1 and t_2 are fired independently.
3. *Single-action Firing (Queue One Action).* The output actions of the transition functions of t_1 and t_2 are fired independently.

Notice that the synchronicity issue discussed earlier is resolved by removing the instantaneous firing of transition functions. The three firing variations presented are just natural ways to consider how the firing could be performed.

▶ **Conjecture.** *For any STAM variant system* Γ*, there exists a simulating TA system* Γ'*.*

We believe the asynchronous STAM variations are not only motivated, but also required in order for TA to simulate the STAM. Furthermore, the concepts introduced within this work may be helpful in resolving this conjecture. For example, from this work we know that the single-rule transition restriction does not inherently limit the power of TA. Thus, we can plausibly use this to show the single-sided variant of STAM can simulate all of TA.

References

1 John Calvin Alumbaugh, Joshua J. Daymude, Erik D. Demaine, Matthew J. Patitz, and Andréa W. Richa. Simulation of programmable matter systems using active tile-based self-assembly. In Chris Thachuk and Yan Liu, editors, *DNA Computing and Molecular Programming*, pages 140–158, Cham, 2019. Springer International Publishing.

2 David Caballero, Timothy Gomez, Robert Schweller, and Tim Wylie. Verification and Computation in Restricted Tile Automata. In Cody Geary and Matthew J. Patitz, editors, *26th International Conference on DNA Computing and Molecular Programming (DNA 26)*, volume 174 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

3 Cameron Chalk, Austin Luchsinger, Eric Martinez, Robert Schweller, Andrew Winslow, and Tim Wylie. Freezing simulates non-freezing tile automata. In David Doty and Hendrik Dietz, editors, *DNA Computing and Molecular Programming*, pages 155–172, Cham, 2018. Springer International Publishing.

4 Joshua J. Daymude, Kristian Hinnenthal, Andréa W. Richa, and Christian Scheideler. *Computing by Programmable Particles*, pages 615–681. Springer International Publishing, Cham, 2019.

5 Tyler Fochtman, Jacob Hendricks, Jennifer E. Padilla, Matthew J. Patitz, and Trent A. Rogers. Signal transmission across tile assemblies: 3d static tiles simulate active self-assembly by 2d signal-passing tiles. *Natural Computing*, 14(2):251–264, June 2015.

6 Jacob Hendricks, Matthew J. Patitz, and Trent A. Rogers. Replication of arbitrary hole-free shapes via self-assembly with signal-passing tiles. In Cristian S. Calude and Michael J. Dinneen, editors, *Unconventional Computation and Natural Computation*, pages 202–214, Cham, 2015. Springer International Publishing.

7 Jarkko Kari. Theory of cellular automata: A survey. *Theoretical Computer Science*, 334(1):3–33, 2005.

8 Alexandra Keenan, Robert Schweller, and Xingsi Zhong. Exponential replication of patterns in the signal tile assembly model. *Natural Computing*, 14(2):265–278, June 2015.

9 Jennifer E. Padilla, Wenyan Liu, and Nadrian C. Seeman. Hierarchical self assembly of patterns from the robinson tilings: Dna tile design in an enhanced tile assembly model. *Natural Computing*, 11(2):323–338, June 2012.

10 Jennifer E. Padilla, Matthew J. Patitz, Robert T. Schweller, Nadrian C. Seeman, Scott M. Summers, and Xingsi Zhong. Asynchronous signal passing for tile self-assembly: fuel efficient computation and efficient assembly of shapes. *International Journal of Foundations of Computer Science*, 25(04):459–488, 2014.

11 Jennifer E. Padilla, Ruojie Sha, Martin Kristiansen, Junghuei Chen, Natasha Jonoska, and Nadrian C. Seeman. A signal-passing dna-strand-exchange mechanism for active self-assembly of dna nanostructures. *Angewandte Chemie International Edition*, 54(20):5939–5942, 2015.