

University of Memphis

University of Memphis Digital Commons

Electronic Theses and Dissertations

2021

Probabilistic Analysis of Self-assembly

Tyler Garrett Moore

Follow this and additional works at: <https://digitalcommons.memphis.edu/etd>

Recommended Citation

Moore, Tyler Garrett, "Probabilistic Analysis of Self-assembly" (2021). *Electronic Theses and Dissertations*. 2687.

<https://digitalcommons.memphis.edu/etd/2687>

This Dissertation is brought to you for free and open access by University of Memphis Digital Commons. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of University of Memphis Digital Commons. For more information, please contact khhgerty@memphis.edu.

PROBABILISTIC ANALYSIS OF SELF-ASSEMBLY

by

Tyler G Moore

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Major: Computer Science

The University of Memphis

August 2021

Copyright© Tyler G Moore
All rights reserved

ACKNOWLEDGEMENTS

I would like to thank my partner Alison Brown for, among innumerable other things, supportive and insightful conversations. Without her, I would be lost.

ABSTRACT

Self-assembly is a pervasive process in natural phenomena that builds complex structures from very simple components. These processes are central to nanoscale manufacturing and crystallography and rely on stochastic self-assembly to bring order out of molecular chaos. In particular, the algorithmic models of self-assembly (for example, the aTAM) employ highly nondeterministic dynamics in an attempt to emulate the massively parallel physical feats effected by DNA *in vivo* to support life. In this work, we define a new object in algorithmic self-assembly called the synoptic pattern to explore structural properties and the long-term behavior of algorithmic self-assembly systems. The synoptic pattern is analogous to the computation tree of a nondeterministic Turing machine, which are known to be computational universal and hence unpredictable algorithmically. We show that synoptic patterns are a robust and useful framework for self-assembly processes at large and enable analysis of their behavior. In particular, they afford a useful analytical tool to understand structural properties of and a nontrivial probabilistic analysis of the long-term behavior of self-assembly processes in large families of 1D and some 2D aTAM self-assembly systems, including systems generating infinite families of patterns describable by regular and linear languages in the Chomsky hierarchy. Finally, we also introduce probabilistic definitions of the efficiency of assembly processes and expand on previous work to show how one assembler can efficiently probabilistically approximate a given target assembler by mostly producing patterns that match closely patterns produced by the target assembler, even in the presence of assembly errors.

TABLE OF CONTENTS

Chapter	Page
List of Figures	vi
Index Of Terms	ix
1 TILE ASSEMBLY SYSTEMS	1
Introduction	1
General Definitions	4
Probability Theory	11
2 SYNOPTIC PATTERNS	16
Synoptic Patterns	16
Structure of Synoptic Patterns	33
Probabilistic Synoptic Patterns	46
Probabilistic Analysis of 1D Rulers	52
3 PROBABILISTIC ANALYSIS OF LONG-TERM BEHAVIOR	59
1D Language Assemblers	60
Stochastic Processes for 1D Assemblers	62
Markov Chains	66
ϵ -Approximability of Linear Languages	78
4 APPROXIMATION BY SELF-ASSEMBLY	83
Probabilistically Approximable Assemblers	84
Conclusions	91
Bibliography	93

LIST OF FIGURES

Figure		Page
1	[Left] A tile diagram using a common convention for centered tile labels along with glue labels and prongs to show binding strength. Glue and tile labels may also be denoted by colors and/or shapes. [Right] A representation of a tile assembly system that uniquely assembles a 3x3 square, defined in Example 1.	7
2	An assembly sequence produced by the TAS shown in Figure 1. Open binding sites where a tile from the tile set can attach by forming a temperature stable bond are shown in a dotted outline. The rightmost assembly is the single terminal produced by the TAS from seed tile s through successive stable attachments at open binding sites.	7
3	A graph of the immediate successor relation on spatial patterns assemblable by the 3x3 square TAS defined in Example 1.	10
4	The tile set T belonging to TAS $\mathbb{T}_{a^n b^n}$. Grey tile types map to character ‘a’ and black tile types map to character ‘b’.	11
5	Spatial pattern z is the overlap of x and y relative to the yellow seed label. Although x' has labels in identical cells as y and the same seed label, x and x' do not form a valid overlap.	18
6	A spatial pattern with local contexts w_1, w_2, w_3 identified in red. A local context may appear in a spatial pattern many times.	20
7	Each node in a claw is a local context. Sites with λ labels are left blank. At the root, the region surrounding the site in the dotted outline (the <i>center</i>) is the precondition. That same region in a leaf is a postcondition with a tile label in the center. A single precondition may lead to many postconditions in a nondeterministic assembly system.	21
8	[Left] A set of two claws where precondition u dominates u' . It is not complete. [Right] We can add a claw with precondition u'' to make the set complete under domination.	24
9	A complete set of claws representing the TAS that assembles a 3×3 square defined in Example 1.	25
10	Precondition u dominates each other precondition in the set of claws. u is barely sufficient for ua as ua does not dominate any other postcondition.	26
11	The set of all disjunctions of local contexts u^1, u^2 with common center c and labels from alphabet Σ including distinct labels $a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c$, one of which may be λ . The size of the set of disjunctions is 2^r where r is the number of positions at which the local contexts differ. If $a_1 = b_1$, then the set of disjunctions would be a subset of the above consisting of only 8 local contexts.	27

- 12 The local contexts shown here are a subset of $\delta(u^1, u^2)$ where u^1 shares only a center label with u^2 . If the depicted local contexts appear as postconditions in some set of claws K and K has no other postconditions sufficient for c , then $W_{bsuf}^c = \{u^1, u^2\}$ and $W = W_{suf}^c = \delta_{suf}^{S,c}(W_{bsuf}^c)$, so that K is perfectly sufficient for c . 31
- 13 [Left] An inconsistent set of claws with preconditions u and u' . [Right] Adding a finger to u such that u is sufficient for green renders the set consistent. 33
- 14 [Left] An inconsistent set of claws with postconditions v and v' sufficient for a . By Proposition 2, the set of claws should contain at least one more claw with a precondition in the disjunction. [Right] Adding a single new claw with a postcondition sufficient for a renders the set consistent. 33
- 15 u^1, u^2, u^3 form a consistent set of claws for a TAS that assembles a 2x2 square. 40
- 16 Shown here is a set of events in the probabilistic synoptic pattern of $\mathbb{T}_{a^n b^n}$. Pattern labels are taken from the tile set. Grey tiles map to the symbol ‘a’ in the alphabet of language $L = \{a^n b^n : n > 0\}$. 51
- 17 The synoptic pattern for the TAS in Example 12 is reminiscent of a herringbone pattern, with a nonterminal spine and terminal ribs. Patterns having most recently bound a C tile form the spine, and patterns having most recently bound a H tile form ribs of length 1. 54
- 18 Here is an fsa that recognizes the language of binary strings consisting of all strings containing an even number of ‘1’s. 61
- 19 In this Markov chain, the transient communication classes are circled. The remaining states belong to singleton recurrent communication classes. 75
- 20 The ideal RGB assembler is designed to grow two signals, one red from site R and one green from site G, toward a point of cooperation C. The seed is an $L \times L$ square perimeter where inputs R and G and output B are to be placed. Cooperation is achieved at the point C, where the signals would “meet” to effect the AND and produce a deterministic signal that is recorded at output B. 88

- 21 A tile set designed for a temperature 2 assembly system that uses cooperation to simultaneously detect nondeterministic input signals, red (R) and green (G), and from the designed point of cooperation output a deterministic signal to the blue (B) site. For cooperative assembly, double edges indicate strength 2 glues, and single edges strength 1. When the system is run in noncooperative mode, all glues are strength 1. R and G propagate from the inputs r and g by a series of R and G tiles. The D and E tiles are placed nondeterministically at some point along the signal to enable attachment of the cooperation tile C. At the point of cooperation, the O tile binds cooperatively and then grows the deterministic blue signal by a series of B tiles (only the first one is shown here.) The number of tiles in the blue signal is constrained by the size of the assembly space L .

88

INDEX OF TERMS

Term	Notation	Definition	Page
Successor Relation	$x \prec y$	pattern x makes y after some actions by an assembler	16
Synoptic Pattern	\mathbb{S}	the successor relation on a set of patterns	16
Cone	x^{++}	the future of a pattern in a synoptic pattern	19
Local Context	w	a neighborhood of labels around a center	20
Precondition	u	a local context with a blank center	21
Postcondition	v	a local context with a nonblank center	21
Claw	κ	a precondition and at least one postcondition	21
Domination	$u \rightarrow u'$	local context u contains u' as a subpattern	23
Disjunction	$\delta(u^1, u^2)$	a set of local contexts with labels from u^1 or u^2	26
Extension	$\mathcal{E}(x, \kappa)$	a set of patterns produced from x by claw extension	33
Probabilistic SP	\mathbb{P}	a probability model on a synoptic pattern	47
Similar modulo p	$x \sim_p y$	x and y of size n have pn matching labels	84
Strength	p	the mass of assembled target patterns of a given size	85
Impurity	q	an assembler's ratio of error mass to target mass	86
Efficiency	ε	an assembler's ratio of strength to impurity	86

CHAPTER 1

TILE ASSEMBLY SYSTEMS

Introduction

Self-assembly is a pervasive process in nature by which simple atomic components arrange themselves into complex structures. It plays a role in the formation of nanoscale structures like the DNA double helix or colloidal crystals and stellar-scale structures like star systems [1, 2, 3].

Modern DNA computing and nanoscale self-assembly were inspired by the combination of physical-chemical research and algorithmic design. Ned Seeman (a father of modern DNA crystallography and DNA computing) and Len Adleman (also known for his contributions to cryptography) are responsible for foundational work in molecular computing called DNA self-assembly. Self-assembly leverages DNA hybridization as a smart glue to form complex 2D and 3D nanoscale structures that are not otherwise manufacturable by traditional methods [3, 4, 5, 6]. Algorithmic models of self-assembly abstract the crystallography to simple binding relationships and prove that these assembly processes can be as complex as arbitrary computations in the Turing model [7].

A generic approach to spatial pattern formation loosely describes models of assembly fitting the following criteria:

1. Local interactions among atomic building blocks proceed without external guidance;
2. Complex, ordered patterns arise as a product of local interactions over space and time.

The abstract Tile Assembly Model (aTAM), introduced by Erik Winfree [7], is the touchstone model for algorithmic self-assembly and has invigorated the last 25 years of research, combining self-assembly with computer science, modern notions of power domains, and modern algorithmic design tools. Presented side-by-side with a more physically realistic kinetic Tile Assembly Model (kTAM), the aTAM provides an accessible, feasible, and powerful model on the order of

the Turing machine for massively parallel, massively nondeterministic self-assembly systems. Like the changing configuration of a Turing machine over time, an assembly sequence produced by a tile assembly system gives a single possible path of transformation from a fixed input. In analogy with crystallography, this input is called a seed assembly, or just a *seed*. New assemblies are formed from the seed by the attachment of tiles according to pairwise glue interactions and respecting an abstract temperature inspired by the physical melting temperature of nucleic acid chemical bonds. Collecting these assembly sequences together organizes the nondeterministic growth of the constituent patterns over time, but a simple family of assembly sequences does not capture the full picture of the relationship between these assembly sequences. In this dissertation, we build on the formal notions of assembly established by prior work with a new object called a synoptic pattern that organizes the disparate assembly sequences into a single comprehensive object akin to the computation tree of a nondeterministic Turing machine. We then adapt this object for probabilistic analysis by endowing it with a probability model to form a so-called probabilistic synoptic pattern.

Probabilistic analysis of self-assembly is a natural idea given the stochastic nature of real self-assembly systems. In the literature, there are essentially two approaches to establishing useful probabilistic models of assembly: tile concentration and probability of formation based on free-energy. Tile concentrations for each tile type model over time how often individual tiles and assemblies come into contact with each other in a well-mixed solution. We see the roots of both in the kTAM, where growth depends on a function of tile concentration called the forward rate and a function of the energy cost of breaking a bond, called binding strength. To prevent errorful growth, probabilistic techniques like proofreading [8] are used that rely on a series of dependent attachments to lock in an error. The probability of an error decreases with the scale of this technique, modeled as a random walk. Concentration is an accessible parameter to the experimenter and the introduction of the kTAM and subsequent inquiries building on its definition have had a great impact on developing actual nanoscale self-assembly systems, reducing error rates and improving yield. Models of self-assembly, such as a Markov process where states are assemblies

and the transition probability is a function of tile concentration or a probability of formation, provide stable ground to make strong statements about average- and worst-case complexity in abstract tile models of self-assembly. Abstractions of tile concentration have been shown to be effective theoretical tools for programming shape and size [9], as well as robust probabilistic assemblers [10, 11]. In Chandran et al's PTAM [11], nondeterminism and tile concentration are used to guarantee the realistic production of a target with high probability rather than deterministically in theory. Their construction employs a variation of a technique called tile concentration programming [12, 13, 14]. By treating the nondeterministic system probabilistically, works that employ tile concentration programming seek to understand the global behavior of the assembler without explicitly and directly constructing a single probability space that encompasses the potentially infinite number of all possible assemblies. The approach used in the PTAM and in later work [15] is particularly suited to their specific results, but does not scale well to other infinite sets of assemblies. Such a construction should assign some probability to each assemblable pattern relative to all other assemblies in a single comprehensive probability space. A naive approach could simply give an arbitrary mass to each assembly of a given size n , which can be normalized and extended to all patterns of size less than n . If only finitely many assemblies have positive mass, this technique can work. On the other hand, if the assembly process produces an infinite number of assemblies, it would assign a limit probability of 0 to infinitely many assemblies. The process of Bayesian inference from Bayesian statistics looks like an attractive choice to describe a statistically sound probability model of self-assembly. Inference proceeds from an initial probability model called the prior. We then sample and infer a posterior model based on the sample and prior. Unfortunately, attempting to produce a probability measure for an infinite sample space based on the traditional Bayesian inference is fraught. As shown in [16], there are choices of priors such that the process of Bayesian inference almost surely does not converge asymptotically. Even simply identifying a probability model that incorporates realistic assumptions from observations of nanoscale manufacturing systems is nontrivial. In chapter 2, our approach to address these issues is discussed at length. In this chapter, we first define the necessary

foundational concepts underlying our methodology to approach our research goal including basic notions of patterns, assemblies and the aTAM, as well as basic concepts such as graphs, probability spaces and probabilistic analysis methods.

General Definitions

In principle, assembly models assemble configurations (or assemblies) nondeterministically. The most well-known tile assembly model, the abstract Tile Assembly Model (aTAM), took inspiration from Wang tiles [17]. While Wang tile sets were originally designed with the goal of perfectly tiling the Euclidean plane, tile assembly systems in the aTAM are designed to algorithmically construct shapes or patterns or to compute some observable value within a collection of tiles. These tiles are usually labeled square units with labeled edges called *glues* and each is one of finitely many such tile types. Tiles are available in unlimited supply, but cannot be rotated, reflected or deformed. They arrange themselves in a container space according to a finite set of local rules applied uniformly everywhere. These local rules are expressed through a glue function and an abstract temperature value. Glue interactions below the temperature value are nonbinding, i.e., do not create a stable bond. When a tile is adjacent to more than one other tile, glue interactions are summed, creating a cooperative effect where multiple glue interactions among the adjacent tiles may be necessary to create a stable bond.

Typically, assemblies are 2-dimensional (2D) and composed of square tiles within Euclidean container spaces. The model is easily extended to higher dimensions, polyominoes, polygonal tiles, and other container spaces [18, 19, 20, 21, 22, 23, 24, 25, 26]. In this work, we will restrict ourselves mostly to 1D and 2D assembly systems in Euclidean spaces.

An assembly in the aTAM can be described as an arrangement of labels in a container space, i.e., as a *spatial pattern*. Traditionally, computer science is concerned with 1D spatial patterns called strings or words. Examples of spatial patterns in two dimensions include digital images, paintings, or graphs. Generally, spatial patterns consist of connected labeled regions. To define these

and other core concepts more precisely, we present here some well-known concepts and notation from mathematics and computer science.

Definition 1. A label alphabet is a finite nonempty set Σ including a special (“blank”) symbol λ . A spatial pattern x is a d -dimensional (dD) finite word over some alphabet Σ , with $d > 0$, i.e., a function that assigns a label to arbitrary positions \mathbf{i} in \mathbb{Z}^d (the unit cells or sites) from a label alphabet. A notion of size $|\ast|$ is a function of x that assigns a value $s \in \mathbb{Z}$ to every spatial pattern describing a digital volume in nonempty cells. A subpattern x' of x is a spatial pattern of the same dimension over the same alphabet such that x contains x' , i.e., there is some translation of x' such that for every translated cell in x' , either x has a matching label in that cell or x' has a λ label in that cell.

In this this work, we will only be interested in *finite patterns*, i.e., patterns where only finitely many cells have a nonblank label. Additionally, we will only be concerned with analyses of assembly systems producing *connected* spatial patterns, i.e., patterns where every nonempty cell is connected to other nonempty cells in its North, East, West or South directions, i.e., in the NEWS (also called the von Neumann) neighborhood. The notion of size for a pattern x we will commonly use is the number of nonempty unit cells in it, unless another notion such as *width* or *height* is explicitly mentioned, although results will apply to arbitrary notions of size with some mild conditions. We are concerned chiefly with assembly models that do not allow the detachment of tiles (the size of a pattern in an assembly sequence in these models is nondecreasing.)

Definition 2. Let G be a label alphabet of glues. A tile type is a k -tuple $(a, g_1, g_2, \dots, g_{k-1})$ where a is a tile label and $g_j \in G$. A 2D tile assembly system (TAS) $\mathbb{T} = (T, \sigma, \tau)$ in the aTAM consists of a set of 5-tuple tile types T with glues from a common G , a seed tile σ , and an integer τ called the (abstract) temperature of the system. A glue function $g: G \times G \rightarrow \mathbb{Z}_0^+$ must also be specified where the range values are glue strengths given by nonnegative integers. The special glue (tile) label λ is called the empty glue (empty tile) and it will be assumed that $g(g_i, g_j) = 0$ if either g_i or g_j are λ .

In the aTAM, we use the convention that glues in a tile type are listed in clockwise order from the north glue like so (a, g^N, g^E, g^S, g^W) .

In the aTAM, new assemblies are composed from the seed or smaller assemblies by tile accretion over time according to the glue function and temperature. When the sum of the glue strengths in a binding context around a binding site for a potential attachment tile is equal to or exceeds τ , the binding context is *sufficient* for a stable bond and the attachment does occur. When y is producible from x by the attachment of a tile from the tile set $T \in \mathbb{T}$, we write $x \mapsto_{\mathbb{T}} y$.

Example 1. *Constructing squares of fixed side length is a touchstone problem [27, 12] in the tile assembly literature. Consider the TAS $\mathbb{T} = (T, \sigma, \tau)$ with*

$$T = \{(s, \lambda, h_1, v_1, \lambda), (v_1, v_1, in, v_2, \lambda), (v_2, v_2, in, \lambda, \lambda), \\ (h_1, \lambda, h_2, in, h_1), (h_2, \lambda, \lambda, inh_2), (in, in, in, in, in)\},$$

$\sigma = (s, v_1, h_1, \lambda, \lambda)$, and $\tau = 2$. *The set of tile labels and the set of glue labels are identical; the glue function is diagonal and symmetric, i.e., $g(a, b) = 0$ when $a \neq b$ and $g(a, b) = g(b, a)$; and the strength of all positive glue interactions is 2 except $g(in, in) = 1$.*

This TAS assembles a 3×3 square using 6 tile types: $s, v_1, v_2, h_1, h_2, in$. The assembly process proceeds nondeterministically from a seed that allows v_1 or h_1 to stably bind. As soon as both v_1 and h_1 are bound, in may cooperatively bind. Each assembly sequence ends with a single unique terminal assembly produced after 8 tile attachments, as shown in Figure 2.

In the aTAM, stably bound tiles cannot be removed from assemblies. Thus, assemblies grow monotonically over time. Models like the kTAM and the restricted glue Tile Assembly Model [28] that include a mechanism for detachment of previously stably bound tiles, a reverse rate and negative strength glues, respectively, will not be considered in this work.

Definition 3. *A tile assembly system \mathbb{T} is monotonic if the attachment of any tile does not cause any detachments.*

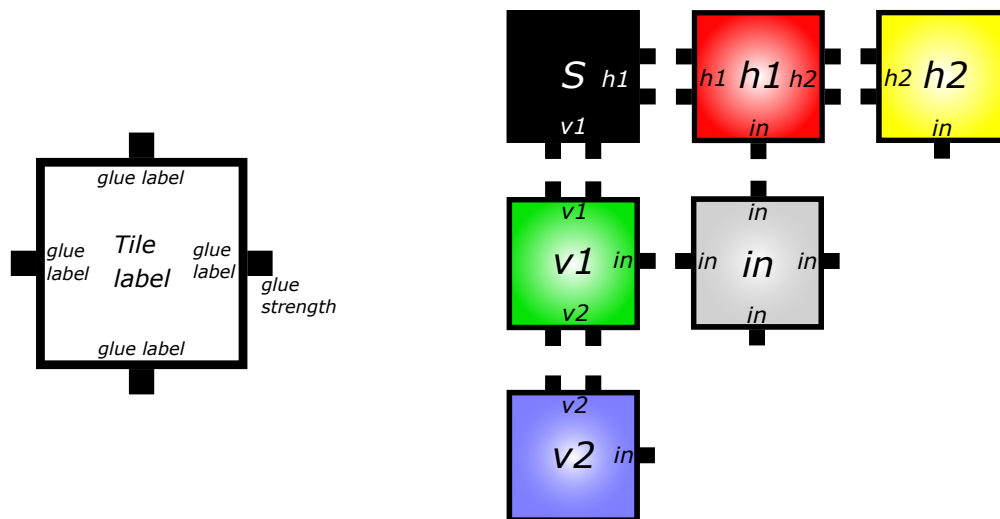


Figure 1: [Left] A tile diagram using a common convention for centered tile labels along with glue labels and prongs to show binding strength. Glue and tile labels may also be denoted by colors and/or shapes. [Right] A representation of a tile assembly system that uniquely assembles a 3x3 square, defined in Example 1.

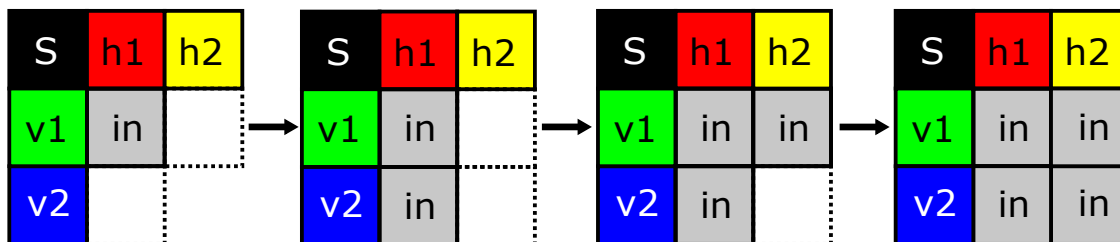


Figure 2: An assembly sequence produced by the TAS shown in Figure 1. Open binding sites where a tile from the tile set can attach by forming a temperature stable bond are shown in a dotted outline. The rightmost assembly is the single terminal produced by the TAS from seed tile s through successive stable attachments at open binding sites.

This work is concerned with the temporal relationships in the full set of assemblies produced by a TAS, including the local behavior, global properties, and long-term *dynamical* behavior of the TAS. However, we do not ignore the local composition of the patterns, including their shapes. A 1D spatial pattern like a length of yarn can be knit into a 2D pattern like a scarf or a 3D pattern like a glove. The use of additional dimensions to define additional aspects of the spatial pattern is intrinsic to the assembly system. A length of yarn is not a scarf unless it is composed in the form of a scarf.

Definition 4. A silhouette is a spatial pattern over a unary alphabet of one nonblank label and λ .

If a spatial pattern is made of stained glass, then a silhouette is that same stained glass with opaque paint slathered over it. When necessary, this paint may be scraped away in small, carefully chosen areas so some light shines through. Some labels in the spatial pattern may be maintained in a principled way to aid in comparisons between patterns. For example, in assemblies produced by an aTAM TAS, it is useful to preserve the seed label since all patterns grow from the seed and thus contain it as a point of common comparison.

Graphs are often used for representations of assembly sequences or relationships between assemblies. A *graph* is a collection of nodes and edges, where a node is typically drawn as a point and an edge is a line between two adjacent nodes. In a simple graph, adjacency is a symmetric property. If x is adjacent to y , then y is adjacent to x .

Definition 5. A directed graph (digraph) is a pair $G = (V, E)$, where V is a set of vertices and E is a subset of $V \times V = \{(x, y) : x, y \in V\}$, whose elements are called arcs, or directed edges. A graph is a digraph where for every arc $(x, y) \in E$, the arc $(y, x) \in E$.

A bipartite graph is a simple graph where V can be partitioned into two (disjoint) subsets A, B such that $E \subseteq A \times B$. The bipartite graph is called complete when $E = A \times B$. These graphs are denoted by $K_{i,j}$ where $i = |A|$ and $j = |B|$.

Definition 6. A partial order \leq is a relation on a set S satisfying the following properties for all $x, y, z \in S$

1. **Reflexive:** $x \leq x$
2. **Antisymmetric:** If $x \leq y$ and $y \leq x$, then $x = y$
3. **Transitive:** If $x \leq y$ and $y \leq z$, then $x \leq z$

A strict partial order $<$ is *irreflexive, antisymmetric, and transitive*.

We note the containment relationship between a pattern and subpattern is a partial order on the set of finite dD spatial patterns with labels from Σ . It is clear that the relation is reflexive and transitive. To show the relation is antisymmetric we note that the patterns are finite (contain finitely many non- λ labels) and connected with a tightly bounding perimeter of cells with λ labels such that every cell with a non- λ label is inside the perimeter. Since x contains y and y contains x this perimeter fixes the translation of x or y . Any shift would introduce a mismatched cell. Therefore, every cell is identically labeled in x and y and they are identical.

Definition 7. Given an aTAM TAS \mathbb{T} that produces a set of assemblies S , let the immediate successor relation be the relation \prec on the set S such that $x \prec y$ if and only if y is producible from x by the attachment of a single tile. We note that, by the monotonicity condition, the transitive closure of this relation denoted \prec^* is a strict partial order on S .

In the aTAM, $x \prec^* z$ if and only if z is producible from x by the attachment of a finite number of tiles in T in an assembly sequence. Assembly sequences represent successive changes in an assembly from the seed over time. Collecting these assembly sequences together and reasoning about the relationship between these sequences has led to fundamental concepts in the literature, including a notion of *directed assembly* and *local determinism* [29]. Locally deterministic assembly systems are computational systems that produce a single terminal assembly regardless of the effects of nondeterministic attachment choices that give rise to many distinct assembly sequences to a unique terminal. The strength of the locally deterministic constraint comes from the understanding that the time order of attachment can be critical to what assemblies are producible by some TAS as in Adleman’s partial order systems [12]. The aim of this work is to take a step back

and treat the collection of *all* assembly sequences as an interesting object in and of itself. The partially ordered set of assemblies captures the totality of the assembly system as an infinite object. The graph of the strict partial order of the 3x3 square TAS presented in Example 1 is shown in Figure 3.

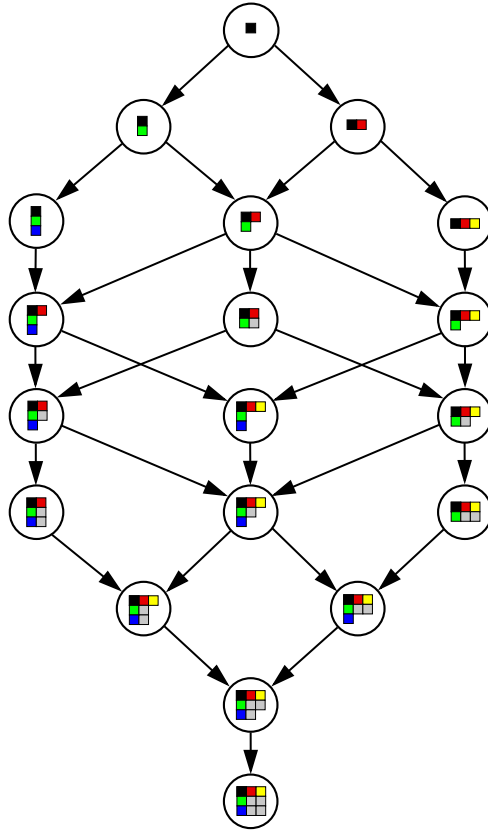


Figure 3: A graph of the immediate successor relation on spatial patterns assemblable by the 3x3 square TAS defined in Example 1.

In this work we will frequently return to the aTAM TAS whose terminal assemblies correspond to the string language $a^n b^n$ for illuminating examples. We introduce the tile set and minor details about the TAS below and will discuss other important features in more detail as needed.

Example 2. Consider an aTAM TAS $\mathbb{T}_{a^n b^n} = (T, s_t, 1)$ with a diagonal glue function and tile set as shown in Figure 4 and whose terminals are in correspondence with the string language $L = \{a^n b^n : n > 0\}$.

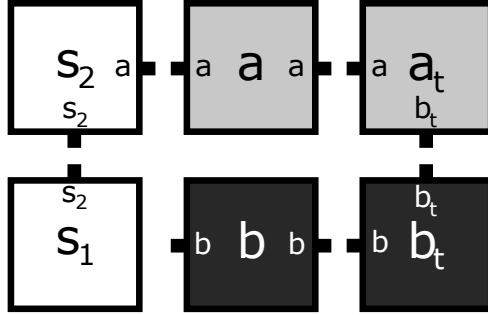


Figure 4: The tile set T belonging to TAS $\mathbb{T}_{a^n b^n}$. Grey tile types map to character ‘a’ and black tile types map to character ‘b’.

Probability Theory

This work is primarily concerned with understanding the global properties of a TAS to produce *families* of patterns as whole collections, perhaps consisting of single terminal assemblies but not necessarily specific single assemblies. Intrinsic to this idea is the notion of production, which requires an understanding of the dynamics of assembly, or how the assembly process unfolds and develops over time. Inherent in a nondeterministic computational model is uncertainty. The traditional way to address uncertainty is through probability theory [30, 31]. At the heart of probability theory is the notion of a *random experiment*, a procedure that produces one of a fixed number of outcomes where no single outcome can be predicted with certainty before performing the experiment. Flipping a coin or rolling a die are canonical examples of random experiments. The experiment must be repeatable any number of times.

Definition 8. A discrete sample space Ω is the finite or countably infinite set of all elementary events, where an elementary event is a possible outcome of a random experiment.

For rolling a 6-sided die, $\Omega = \{1 \dots 6\}$. Probabilistic analysis proceeds by assigning a measure of uncertainty, a *probability*, to these outcomes. Intuitively, a fair die should roll a 1 about as often as it rolls a 6 (or 5, etc.), and we might naturally extend this intuition and say that it should roll an even about as often as it rolls an odd, i.e., the probability of $\{2, 4, 6\}$ is equal to that of $\{1, 3, 5\}$. Rather than rolling a die, we could consider a TAS where the random experiment is to

choose a tile and attach it to a growing assembly to produce a new assembly. This random experiment is used in a continuing line of research to establish a time resource and complexity hierarchy [12] for self-assembly systems. Uncertainty in tile assembly occurs when there are several tile types that may bind at one or more binding sites. In extreme cases (e.g., a trick die that has a 6 on each face) the outcome may not be uncertain at all. TAS produced as a product of the zigzag transformation [18] from a deterministic Turing machine are similarly certain in their outcome. Ultimately, if we want to actually evaluate the probabilities for these *compound events*, we must satisfy certain formal conditions.

Definition 9. A σ -algebra is a collection of subsets (the measurable events) of a sample space Ω of a random experiment that

- contains the sample space,
- is closed under the operation of complementation, and
- is closed under the operation of taking countable unions.

For clarity, a set of more than one elementary event will be referred to as a compound event. Extreme examples of a σ -algebra include $\mathcal{F} = \{\emptyset, \Omega\}$ and the full σ -algebra of all subsets of Ω . A common way to define a σ -algebra is to define some subsets of Ω as desirable measurable sets (e.g., the intervals in the unit interval, with length as their probability measure), and then obtain the σ -algebra as the closure under complementation and countable unions (e.g., to obtain the so-called Borel measure from intervals.)

Definition 10. A probability measure (or mass function) for a σ -algebra \mathcal{F} is a function $\rho: \mathcal{F} \rightarrow [0, 1]$ such that $\rho(\emptyset) = 0$, $\rho(\Omega) = 1$, and for any set of pairwise disjoint sets of elementary events $\{E_1, E_2, \dots\}$ with $E_i \in \mathcal{F}$, $\rho(\bigcup_i E_i) = \sum_i \rho(E_i)$.

In measure theory, a σ -algebra describes the measurable subsets of a sample space. The sample space, σ -algebra and the probability measure taken together define a probability space.

Definition 11. A probability space is a triple $(\Omega, \mathcal{F}, \rho)$, where Ω is a sample space of elementary events, \mathcal{F} is a σ -algebra over Ω that contains the measurable events, and ρ is a probability

measure on \mathcal{F} .

Example 3. *The probability space associated with a fair six-sided die consists of*

- $\Omega = \{1, 2, 3, 4, 5, 6\}$,
- $\mathcal{F} = \mathcal{P}(\Omega)$, the power set of Ω , and
- $\rho(e) = \frac{1}{6}$, where $e \in \Omega$.

A uniform probability space can likewise be defined for any finite sample space Ω .

Probabilistic analysis is concerned with calculating or estimating the probability of events in a given probability space. For example, for any event $A \in \mathcal{F}$, $\rho(A \cup A^c) = \rho(\Omega) = 1$, so that $\rho(A^c) = 1 - \rho(A)$. Often, these events are defined by observations one can make on an outcome, like a gambler's winnings or losses on a die roll.

Definition 12. *Let Ω be a sample space. A discrete random variable is a map $X: \Omega \rightarrow \mathbb{R}$ assigning a value to every outcome of a finite or countably infinite random experiment.*

We introduce a discrete random variable to stipulate a constraint on the random experiments of self-assembly. In the aTAM, the seed is finite, the set of tile types is finite, and one tile is added at a time, thus the set of spatial patterns produced by an aTAM TAS is finite or countably infinite.

Example 4. *The random variable X associated with the random experiment "Roll a fair six-sided die 3 times and take the highest result" takes values 1, 2, 3, 4, 5, or 6. The probability of a particular outcome is computed from the probability space and can be described by the probability mass function ρ . For example, $\rho(X = 1) = \frac{1}{6^3}$ since the probability of a single 1 is $\frac{1}{6}$ and only one trial of the random experiment has value 1, the outcome where all three rolls have value 1. The event $X = 6$ is much larger, containing 91 trials with $\rho(X = 6) = 0.421\overline{296}$*

One way to summarize a collection of values with a single number is the *mean*. The mean of a set of numbers is the average of the values in the set, the sum divided by the size. Random variables have a similar summary, a sort of weighted average called the *expected value*.

Definition 13. The mean or expected value of a random variable X is defined as $E(X) = \sum_k x_k \rho(X = x_k) = \sum_k x_k \rho(x_k)$, where the x_k s range over all the values of the random variable.

The expected value is sometimes called the first moment of a random variable. Given a system of weights placed in the plane, the sum of the distances of the weights from the origin multiplied by the mass of the weight is the first moment about the origin. The expectation of a random variable is similar to the centroid of this system of weights, the first moment about the mean. The second moment about the mean provides another number to add to our helpful summary of a random variable and it is called the *variance*.

Definition 14. The variance of X is given by $\sum_k (x_k - E(X))^2 \rho(x_k) = E((X - E(X))^2)$.

Our approach to address the major goal of this work, as sketched above, is to define appropriate probability spaces to understand interesting properties of assemblies and assembly dynamics in a given TAS. A more interesting example for this work is choosing a tile type in a TAS and finding a possible attachment in an existing assembly, or whether the resulting assembly is *terminal*, i.e., cannot be further extended. By treating the abstract stochastic process of assembly with a probability space, computationally difficult questions can be approached in a structured way through methods from probability and statistics. In general, the problem of the assemblability of a specific pattern or a pattern with a specific property is algorithmically unsolvable, but with well-reasoned probabilistic arguments some aspects of the long-term behavior of some TAS can be partially described by a probabilistic statement. Furthermore, the presence or absence of a property-in-question may be determined by the formation of an ‘almost certainly’ argument about the property of spatial patterns corresponding to compound events in the probabilistic synoptic pattern. Toward this goal chapters 2, 3, and 4 present the central object of study, the *synoptic pattern*, the marriage of this object with a probability space to form a *probabilistic synoptic pattern*, and the operationalization of the probabilistic synoptic pattern to explore interesting questions about TAS. We focus on the capacity for a probabilistic synoptic pattern produced by certain families of assembly systems to be understood holistically through a Markov chain, a more manage-

able probabilistic object, in chapter 3, or even another probabilistic synoptic pattern in chapter 4.

CHAPTER 2

SYNOPTIC PATTERNS

Synoptic Patterns

Self-assembly systems generate spatial patterns that are the product of accretion. Tile assembly models of self-assembly accrete tiles, the atoms of the tile self-assembly process. Spatial patterns produced through self-assembly are assemblies that generally contain no explicit *temporal* information about the assembly process. Each pattern is a snapshot of the assembly process at a point in time. Collecting and exploring the full set of spatial patterns along with this temporal information produces sequences of spatial patterns that may reveal additional complexity beyond how the labels are arranged in space and what local rules control growth. The major aim of this chapter is to explore the structural properties of such spatio-temporal patterns.

Definition 15. A synoptic pattern $\mathbb{S} = (S, \prec)$ consists of a collection of dD spatial patterns S and an immediate successor binary relation \prec on S (or $x \prec y$ for the distinct elements x, y). The (irreflexive-) transitive closure of \prec is the successor relation on S and is also denoted by \prec (or \prec^* if the distinction is important.) A skeleton synoptic pattern is a synoptic pattern with silhouette labels. An abstract synoptic pattern is a synoptic pattern without labels.

A synoptic pattern of a monotonic self-assembly system is represented by a directed graph of the relation \prec , a strict partial order, where nodes are labeled with spatial patterns $x, y \in S$ and edge $(x, y) \in E(G)$ if and only if $x \prec y$. This is a directed and acyclic graph (DAG) and will be referred to as the abstract synoptic pattern associated with a given synoptic pattern. When it is useful to identify the portion of y not previously in x we use the notation (x, xa) where $y = (xa)$ and a is the additional label or labels used to construct y from x . In the aTAM, this is always a single label.

Example 5. An aTAM TAS \mathbb{T} generates a full set of assemblies S . A pair (x, y) with $x, y \in S$ is related by the immediate successor relation if y is obtained from x by the attachment of one tile.

The size n of a tile assembly x is the number of tiles in x . For a monotonic TAS (as in the aTAM), it is easy to verify this relationship induces a strict partial order on S . Thus, with S and \prec we have a synoptic pattern collecting the full history of all possible assemblies producible by \mathbb{T} . Figure 3 shows the synoptic pattern of a 2D TAS that assembles a 3×3 square. The abstract synoptic pattern is a rooted DAG where paths from the root encode assembly sequences in the TAS from the seed.

To explore conditions on the structure of the abstract synoptic pattern, we define the *overlap* of two spatial patterns. We say two spatial patterns have an overlap when the intersection of their label sets includes a common element called the seed and, after a suitable translation such that their seed labels are matched, every non- λ label in one pattern is matched by either the same label or λ in the other pattern. We take the position \mathbf{i} of each cell in these spatial patterns relative to the seed.

Definition 16. Given two dD spatial patterns x, y with labels from Σ that share at least a common seed label, the overlap z of x and y is a spatial pattern obtained by overlapping x and y at the common seed label, without rotation, and selecting all non- λ labels in x (y) matching labels in y (x) at the same positions relative to the seed and all non- λ labels in x (y) where y (x) has λ at the same position relative to the seed.

Thus, without loss of generality, if x contains a non- λ label at position \mathbf{i} from the common seed and y contains a different non- λ label at position \mathbf{i} , then x and y do not admit a valid overlap. By exploring the structure of the abstract synoptic pattern, we gain a better understanding of the behavior of aTAM TAS. We build the necessary formalism to address the question whether every rooted tree is induced as a (an abstract) synoptic pattern of some aTAM TAS.

Proposition 1. If a graph G is an abstract synoptic pattern associated with an aTAM TAS, then

1. G is a rooted directed and acyclic graph (a DAG);
2. Vertices in G can be layered by graph distance n (their size) from a node with in-degree 0 such that each node in a layer is adjacent to finitely many nodes in the next layer;

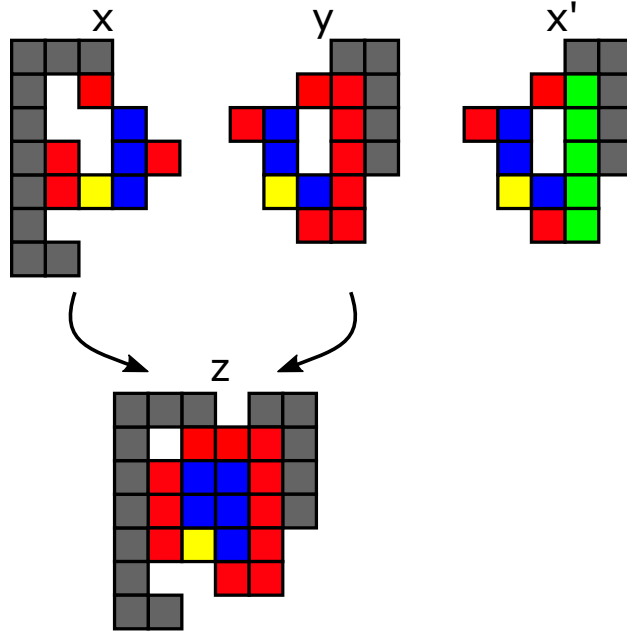


Figure 5: Spatial pattern z is the overlap of x and y relative to the yellow seed label. Although x' has labels in identical cells as y and the same seed label, x and x' do not form a valid overlap.

3. Vertices in G have finite out-degree, bounded by a function $O(|T|^{n^d})$ of the size n , where T is the tile set and assembly is performed in a dD container space;
4. G does not contain $K_{2,2}$ as a subgraph.

Proof. The successor relation \prec^* is a strict partial order, therefore the graph of the relation is a rooted DAG, with the seed at the root. Since each assembly is either the seed or grows from the seed by the attachment of tiles, a natural layering over time based on node labels (patterns) is constructible by following assembly sequences in the TAS (chains in the partial order.) Since each assembly is of finite size and the tile set has finitely many tile types, the number of assemblies producible from any pattern in one step is also finite and is bounded by the product of the count of open binding sites with the size of the subset of the tile set that consists of tiles that may form a stable bond at that binding site, at most $|T|$. G cannot contain $K_{2,2}$ as a *directed* subgraph because every pair of *different* patterns of size $n + 1$ that share a predecessor share a common subpattern of size n , which is their common predecessor and thus unique. \square

In the abstract synoptic pattern, the shape and pattern are absent and it is difficult to characterize

and derive the underlying tile assembler. Separating the influence of geometry from the algorithm implemented by the TAS is impossible in general. Skeleton synoptic patterns provide the step-by-step geometry of the growing assembly with none of the pattern information present in a full synoptic pattern (excepting seed information so that skeleton synoptic patterns can be reliably compared.) Comparisons between silhouette pattern labels in these synoptic patterns yield information about the underlying assembler, including when sites require cooperation to be occupied and time order relationships concerning when empty sites may be occupied.

Definition 17. *A cone in a synoptic pattern is the set of patterns $z^{++} = \{x : z \prec^* x\} \cup \{z\}$ consisting of all descendants from a pattern z by the successor relation and z itself. Conversely, an anticone is the set of patterns $z^{--} = \{x : x \prec^* z\} \cup \{z\}$ consisting of all predecessors of a pattern z and z itself.*

Intuitively, a cone is the potential future of an assembly in a synoptic pattern; an anticone is the necessary past of an assembly in a synoptic pattern. Any pattern in a cone may arise from the pattern at the tip of the cone, and any pattern in an anticone can produce the pattern at the tip of the anticone. In an aTAM TAS, this representation of long-term behavior comes from the sequential transformation of local regions in each assembly.

Local interactions in an aTAM TAS occur in a *local context*, a finite subregion of an assembly that contains sufficient information to decide whether a particular tile may stably bind. For simplicity, we will discuss aTAM TAS in 2D (i.e., $d = 2$), although most results below generalize to 3D TASs ($d = 3$.) In the 2D aTAM, this local context is exactly a cell and the NEWS adjacent cells. Although a local context may not consist of contiguous tiles (e.g., those occupying the NEWS neighborhood), an assembly resulting from some TAS can be decomposed into a set of subregions (that may overlap) that represent local contexts where local interactions can grow new assemblies by allowing for a tile to stably bind. We abstract this notion to spatial patterns. The *center* of a local context is mutable because the self-assembly process may change the label in the center cell as a result of local interactions. The local context with λ in the center cell is referred to as the *precondition* for this change. The precondition is sufficient for a *postcondi-*

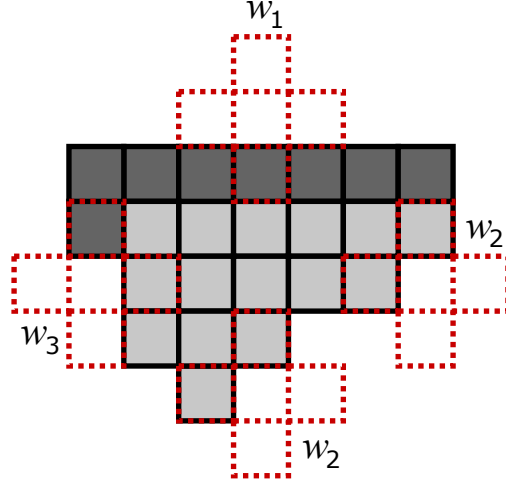


Figure 6: A spatial pattern with local contexts w_1, w_2, w_3 identified in red. A local context may appear in a spatial pattern many times.

tion, i.e., the label in the center cell after mutation along with the surrounding neighborhood. An example is shown in Figure 7.

Definition 18. A local context is a dD word w (called its support) with fixed orientation over a label alphabet Σ with a designated cell called its center. Each cell in the support is uniquely identified by its position $\mathbf{i} = (i_1, i_2, \dots, i_d)$, the vector of integer coordinates describing the cell's relative distance in each dimension from the center, which is assumed to be at position $\mathbf{0}$. The label in w at position \mathbf{i} is denoted $w_{\mathbf{i}}$. The size of a local context is the number of non- λ labels in it, excluding the center. The radius of the local context is the maximum Manhattan distance of any non- λ cell in w from the center.

When comparing local contexts it is often useful to overlay one on the other at their center so that a single position vector can coordinate the label in that cell in any of the local contexts.

We describe the relationship between local contexts in spatial patterns generated by some aTAM TAS with a special graph called a *claw* shown in Figure 7.

Definition 19. A claw is a tree of depth 1 rooted at a local context, called the precondition, and with one or more leaf node(s) called the postconditions. A precondition is a local context w with $w_{\mathbf{0}} = \lambda$. A postcondition consists of a copy of w except with a nonempty label $a \neq \lambda$ in the center.

The radius of the claw is the radius of the precondition. The pair (w, wa) with precondition w and postcondition wa is called a finger of the claw.

Every claw requires a postcondition, so that if no attachment can occur, no claw is added. For simplicity and without loss of generality, we exclude sets of dD claws where any pair of different claws have identical preconditions since they can be consolidated into a single claw.

Definition 19 is quite general and may be used to study assembly systems where local contexts do not correspond to sites in Euclidean spaces. A context makes explicit the geometry of assembly in a way that tiling models only mark implicitly. While in the following we will be mostly concerned with 1D or 2D spatial patterns, as illustrated in Figure 7, contexts and, by extension synoptic patterns can be useful tools in precisely determining the geometry of other assembly spaces as well because of this explicit description of assembly behavior.

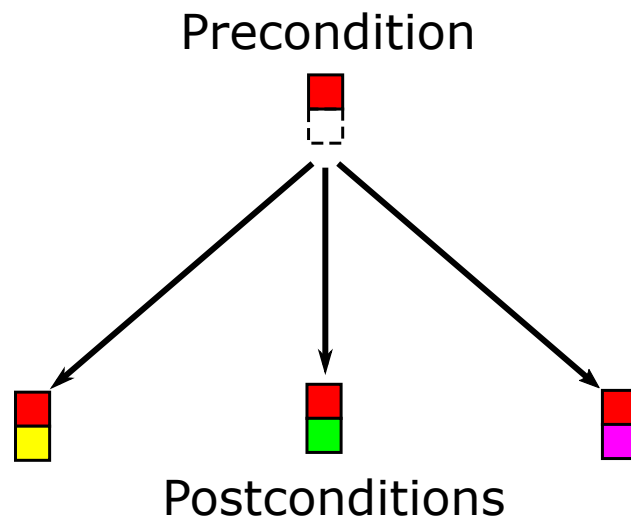


Figure 7: Each node in a claw is a local context. Sites with λ labels are left blank. At the root, the region surrounding the site in the dotted outline (the *center*) is the precondition. That same region in a leaf is a postcondition with a tile label in the center. A single precondition may lead to many postconditions in a nondeterministic assembly system.

We now establish that a set of claws can be produced from any given aTAM TAS.

Lemma 1. *Given a 2D aTAM TAS \mathbb{T} , there exists a finite set of claws with NEWS supports specifying all possible attachments in the synoptic pattern of \mathbb{T} .*

Proof. Preconditions are NEWS (von Neumann) neighborhoods of radius 1 anchored at an empty cell offering glues that enable an attachment at the center. Postconditions are associated with all possible attachments on these preconditions. A claw consists of a precondition and *all the possible arising postconditions* where the center would be stably occupied by a tile from the tileset, according to the glue function g and temperature τ . If only one attachment can occur deterministically, then the claw is a labeled path of length 2. \square

Note that although this set of claws makes no reference to glues, temperature, or attachments, it enables an assembly process that adds tiles to the preconditions as described by their postconditions. It is thus not clear *a priori* that any such finite set of claws can be conversely turned into a TAS in the aTAM, particularly if two different claws present contradictory postconditions from the preconditions, or are oblivious to some local contexts in a partial assembly. The full set of claws derived from a TAS might over-define the assembly system and indicate the assemblability of preconditions that are not *bona fide*, i.e., that do not appear in any actual assembly. During the assembly process, the container space or the assembly itself may limit new growth through excluded volume events, where a glue offers a positive strength bond, but the binding site is already occupied by some previously laid tile. In what follows, we consider these issues and demonstrate how we may account for them.

More generally, a synoptic pattern $\mathbb{S} = (S, \prec)$ of an aTAM TAS defines a set of local contexts as follows. Every λ cell adjacent to a non- λ cell in a pattern is the center cell of a local context with radius 1 (in Manhattan distance.) This local context (a precondition) taken together with the corresponding support with an occupied center in immediate successors across all spatial patterns in S (the postconditions) defines a claw. To find this corresponding support, we translate the patterns such that the center cells are overlaid. If the same local context occurs again anywhere in nodes in \mathbb{S} , the entire claw must also appear because of the locality of the interactions. This is true even though the labels in the spatial pattern outside the local context in the precondition may be different.

We continue to relate the aTAM and a set of claws by noting an implication of the rules for stable

binding in the aTAM. A small local context sufficient for some tile cannot be made insufficient by adding more tiles in the NEWS positions around the center. In a set of claws, there may be many preconditions that have some identical labels in the same positions relative to their common center cell. When the support of one precondition contains *as a subpattern* the support of another precondition translated so their center cells overlap, there is a relationship between the preconditions we call *domination*. The local contexts are similar, with the dominating one containing new nonempty labels not in the other.

Definition 20. *A local context u dominates u' , denoted $u \rightarrow u'$, if and only if u' is a subpattern of u relative to a common center cell, i.e., u contains all of the non- λ labels in u' in the same positions. When u does not dominate u' , we write $u \not\rightarrow u'$. If neither dominates the other, we write $u \leftrightarrow u'$.*

By definition, each postcondition dominates the precondition in the same claw. For example, in Figure 13, precondition $u \rightarrow u'$ and each postcondition of u dominates u .

It is clear that the relation \rightarrow induces a partial order in the set of all local contexts over a given label alphabet Σ , for any given linear ordering of the labels with λ as the minimum element. This partially ordered set (poset) may not be connected but may contain large connected components.

In this poset, a postcondition is an immediate successor to its precondition. The local context with only empty labels is a minimum element, that with only maximum labels is a maximum element, and any nonempty subset has maximal elements.

In the aTAM, tiles bind to assemblies one at a time, so every assembly sequence determines many connected components of local contexts in this poset. A small local context that is dominated by a much larger local context suggests that there should be some intermediate contexts between them. When the local contexts are preconditions and are sufficient for some postconditions, the intermediate preconditions should also be present upon examining the assembly sequence. We establish a notion of *completeness* in a set of claws as a necessary condition for the set of claws defining a TAS in the aTAM.

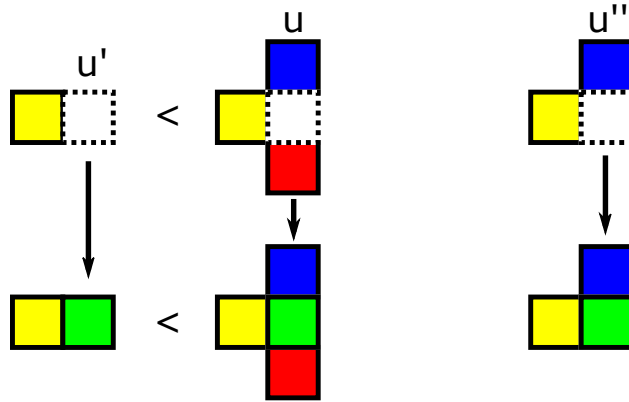


Figure 8: [Left] A set of two claws where precondition u dominates u' . It is not complete. [Right] We can add a claw with precondition u'' to make the set complete under domination.

Definition 21. A set of claws K is complete under domination (or simply, complete) if for every pair of preconditions u, u' in K where $u \rightarrow u'$, either u contains exactly one more non- λ label than u' or there exists an intermediate precondition $u'' \in K$ such that $u \rightarrow u'' \rightarrow u'$.

Every set of claws produced from a 2D aTAM TAS is complete under domination. An example of a complete set of claws associated with an aTAM TAS that assembles a 3×3 square is shown in Figure 9. A set of claws that is incomplete under domination cannot arise from an aTAM TAS if every precondition appears as a local context in an assembly produced by the TAS. Tiles are bound one at a time, so a large local context cannot suddenly appear in an assembly. It must be built up, tile by tile.

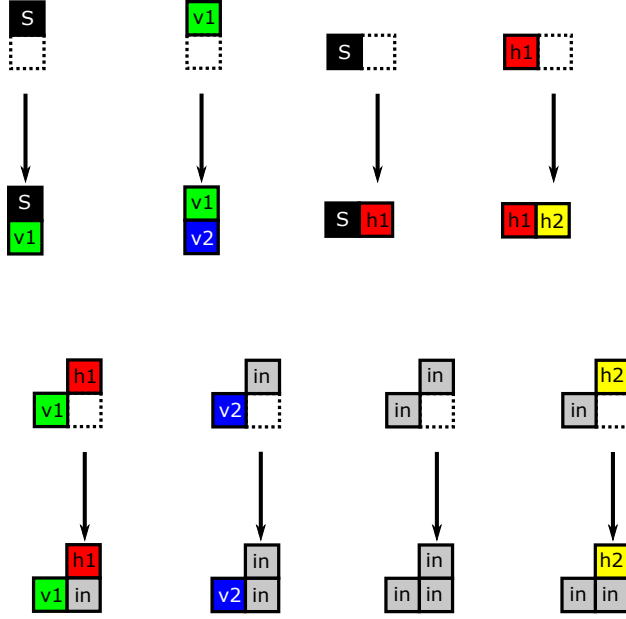


Figure 9: A complete set of claws representing the TAS that assembles a 3×3 square defined in Example 1.

We note here that a set of claws K with unique preconditions implies that, in the corresponding synoptic pattern, for every pair of spatial patterns x, y in the immediate successor relationship $x \prec y$, if the support of precondition u is a subpattern of x and u is sufficient for postcondition v , then the support of v is a subpattern of y .

As mentioned previously, in the aTAM, a sufficient set of glues for a stable bond cannot be made insufficient by adding more glues, but removing some glues might prevent the bond. We call this tipping point from sufficient to insufficient *barely sufficient*. We formalize this notion in a set of claws.

Definition 22. Let K be a set of claws and (u, ua) be a finger of a claw $\kappa \in K$. Precondition u is called *barely sufficient for a* if for every other claw in K with precondition u' sufficient for a , $u \not\rightarrow u'$. In this case, postcondition ua is also called *barely sufficient for a*.

As shown in Figure 10, if u is barely sufficient for a , u may dominate other preconditions in K for other labels. The definition only requires that ua dominate no other postcondition.

In the aTAM, a glue relation describes the strength of a bond between two adjacent tiles. This strength does not depend on any other information about the assembly or even the local context

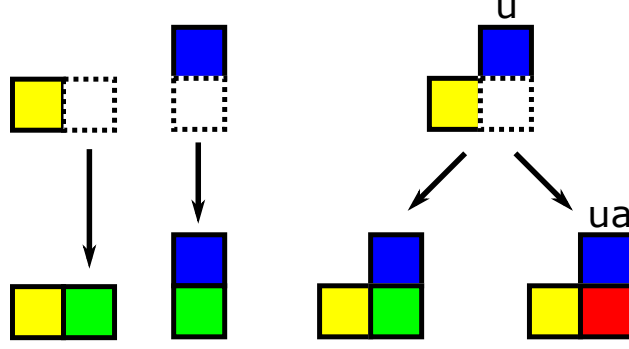


Figure 10: Precondition u dominates each other precondition in the set of claws. u is barely sufficient for ua as ua does not dominate any other postcondition.

outside of the pair of adjacent tiles. Provided a pair of local contexts in a tile assembly, each sufficient for the same tile, it seems reasonable that some type of disjunction of the two local contexts might also be sufficient for the tile to attach.

Definition 23. Let S be a set of dD spatial patterns over a label alphabet Σ and let u^1, u^2 be finite local contexts over Σ with common center label $a \in \Sigma$ and supports occurring in S . A dD finite local context v is a disjunction of u^1 and u^2 if and only if for every position \mathbf{i} in v , each label $v_{\mathbf{i}}$ matches either $u^1_{\mathbf{i}}$ or $u^2_{\mathbf{i}}$. For 2D local context u consisting of a center cell and the NEWS adjacent cells, we denote the labels about the center by a 4-tuple (u_n, u_e, u_s, u_w) . The set of all disjunctions of u^1 and u^2 is denoted $\delta(u^1, u^2)$. The union of the sets of all disjunctions of local contexts in W with common center label a is $\delta^a(W) = \bigcup_{u, v \in W} \delta(u, v)$. A disjunction of u^1, u^2 is sufficient for a with respect to S if its support appears as a subpattern of some spatial pattern in S . The set of all such disjunctions is denoted $\delta_{suf}^S(u^1, u^2)$. Likewise, the union of the sets of all such disjunctions of local contexts in set W with common center a is $\delta_{suf}^{S,a}(W) = \bigcup_{u, v \in W} \delta_{suf}^S(u, v)$.

Note that, in particular, $\delta(u^1, u^2)$ contains u^1 and u^2 . An example of the set of all disjunctions for a pair of local contexts is shown in Figure 11.

In what follows, we seek to establish a complete characterization of a special subset of the set of all disjunctions of any local contexts u^1, u^2 under some reasonable constraints. Our strategy is to show that each pair of suitably different local contexts with an identical center related to an

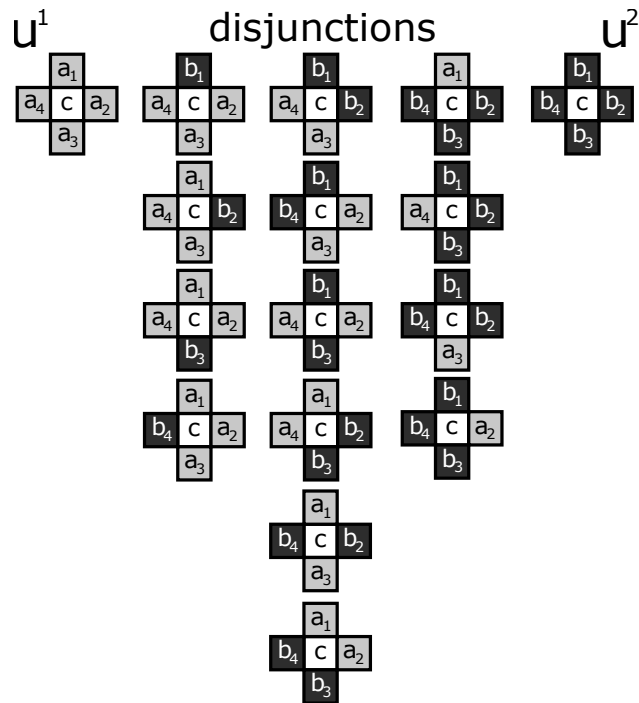


Figure 11: The set of all disjunctions of local contexts u^1, u^2 with common center c and labels from alphabet Σ including distinct labels $a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c$, one of which may be λ . The size of the set of disjunctions is 2^r where r is the number of positions at which the local contexts differ. If $a_1 = b_1$, then the set of disjunctions would be a subset of the above consisting of only 8 local contexts.

aTAM TAS has a nontrivial subset of disjunctions that is also sufficient for the same center. We then construct a finite set of such nontrivial subsets that serves as a *library* for sets of disjunctions (sets of local contexts) sufficient for that center. In order to identify if a set of claws contains an ideally *sufficient* set of disjunctions with center a as postconditions, we simply identify a set of postconditions with the necessary properties and check their local contexts against the library with some *matching-preserving isomorphism*. If the set has local contexts that are isomorphic to the local contexts in some set in the library, then it is a set of disjunctions sufficient for a . If no such isomorphism exists, then it is not such a set. We now formally define this notion of matching-preserving isomorphism and the library.

Definition 24. Let (u^1, u^2) be local contexts over alphabet Σ_u and (v^1, v^2) be local contexts over alphabet Σ_v . (u^1, u^2) and (v^1, v^2) are matching-preserving isomorphic (simply, isomorphic) if there exists a 1-1 correspondence between labels in Σ_u and Σ_v such that every pair of labels in (u^1, u^2) match if and only if their corresponding labels in (v^1, v^2) match. This matching is the isomorphism type of (u^1, u^2) .

A pair of local contexts is trivially isomorphic to itself. For any pair of 2D local contexts consisting of a center and the NEWS adjacent cells belonging to a set of claws with labels from Σ , at most 9 unique symbols from Σ label them. Such a pair is isomorphic to some pair of local contexts with similarly shaped supports and labels from $M_9 = \{m_1, \dots, m_9\}$ according to the isomorphism type of the pair since there are exactly 9 distinct possible comparisons between their labels across all five positions. We can construct such isomorphic local contexts by greedily labeling a pair of unlabeled supports with labels from M_9 respecting the isomorphism type.

Definition 25. Let S be a set of 2D spatial patterns over label alphabet Σ and W the set of 2D local contexts consisting of non- λ centers and the NEWS-adjacent sites whose supports appear as subpatterns of any spatial pattern in S . A library for S is a set of sets of disjunctions over the alphabet $M_9 = \{m_1, \dots, m_9\}$ constructed as follows

1. for each label $a \in \Sigma$, for each pair $u^i, u^j \in W$ with center a , construct $\Delta_{i,j}^a = \delta_{suf}^S(u^i, u^j) =$

$$\delta(u^i, u^j) \cap W$$

2. for all i, j construct the set of local contexts over M_9 of all contexts in $\Delta_{i,j}^a$ by the isomorphism type.

When S is generated by some 2D aTAM TAS, the library for S contains only local contexts representing stable bindings of tiles to growing assemblies.

Definition 26. A full library for the 2D aTAM is the union of all libraries for sets of spatial patterns generated by all possible 2D aTAM TASs.

This definition is not trivial. Every set of disjunctions in the complete library is a subset of $\delta(v^1, v^2)$ with v^1, v^2 over M_9 , but not every subset of $\delta(v^1, v^2)$ for every v^1, v^2 over M_9 is in the full library as indicated in Proposition 2. For example, consider barely sufficient local contexts drawn from some set of spatial patterns generated by a 2D aTAM TAS u^1, u^2 over Σ that share no labels except the center label. The full library contains sets with elements v^1, v^2 over M_9 isomorphic to u^1, u^2 . The full library does not contain the set $\{v^1, v^2, v\}$ for any v over M_9 .

We can build the full library without constructing a library for each of the infinitely many TAS by interpreting the binding constraints in the aTAM and exhausting the relationship between the sums of glue strengths and τ for any pair of local contexts isomorphic to v^1, v^2 over M_9 .

Having established the existence of a full library for the 2D aTAM and other necessary formalisms, we now show that the full library is not trivially empty. Note that for any pair of local contexts u^1, u^2 matching-preserving isomorphic to v^1, v^2 where $\delta(v^1, v^2)$ has some subset in a library, u^1, u^2 has the same subset in the same library under the same isomorphism. As shown above, we can expand this library to be agnostic of S (i.e., we can construct a full library) by exhausting all possible disjunctions of any pair of local contexts and constructing the necessary subsets of disjunctions sufficient for the center a . In support of this strategy, we now show these subsets exist for any two unique local contexts sufficient for the same tile by the constraints of the aTAM that guarantee that some, but not necessarily all, of their disjunctions will also be sufficient for that tile.

Proposition 2. *Let S be the set of spatial patterns produced by a 2D TAS $\mathbb{T} = (T, \sigma, g, \tau)$ in the aTAM. If two 2D preconditions $u^1 \neq u^2$ sufficient for label $a \in T$ satisfy the following conditions:*

1. *for any disjunction $u \in \delta(u^1, u^2)$, there exists a spatial pattern $x \in S$ that contains u 's support as a subpattern, and*
2. *$\{u^1, u^2\} \neq \delta(u^1, u^2)$,*

then there exists a third context $v \in \delta(u^1, u^2)$, $u^1 \neq v \neq u^2$, that is also sufficient for a .

Proof. Given the conditions, a third disjunction v^0 must appear in at least one spatial pattern produced as an assembly of \mathbb{T} . Therefore, there must exist two positions $\mathbf{i}_1 \neq \mathbf{i}_2$ where $u_{\mathbf{i}_k}^1 \neq u_{\mathbf{i}_k}^2$ for both $k = 1, 2$. For simplicity, we assume that $\mathbf{i}_1 = \mathbf{n}$ and $\mathbf{i}_2 = \mathbf{e}$ (other cases are similar.) Furthermore, since u^1, u^2 are sufficient for a , tiles with labels from u^1 (u^2) in the same positions stably bind the tile with label a and have total glue strength with a at least τ . The additive glue strengths in the aTAM allow the substitution of one tile for another to form a new local context sufficient for the center if the glue strength between the new tile and center is at least the strength of the glue between the old tile and the center. Consider the two disjunctions obtained from u^1 and u^2 respectively: $v^1 = (u_{\mathbf{n}}^1, u_{\mathbf{e}}^2, u_{\mathbf{s}}^1, u_{\mathbf{w}}^1)$ and $v^2 = (u_{\mathbf{n}}^1, u_{\mathbf{e}}^2, u_{\mathbf{s}}^2, u_{\mathbf{w}}^2)$. Notice that v^1 and v^2 consist of labels from u^1 and u^2 but are not identical to either. If $g(a, u_{\mathbf{e}}^1) \leq g(a, u_{\mathbf{e}}^2)$, v^1 is sufficient for a since u^1 is sufficient for a and the glue strength has increased in value between the center and the label at position \mathbf{e} . Likewise for v^2 when $g(a, u_{\mathbf{n}}^1) \leq g(a, u_{\mathbf{n}}^2)$. If neither condition is true, then v^2 is sufficient for a since the glue strength $g(a, u_{\mathbf{n}}^2)$ is less than $g(a, u_{\mathbf{n}}^1)$. Thus, one of the two disjunctions v^1, v^2 must be sufficient for a . □

Using the technique in the proof of the proposition, we define the set of disjunctions sufficient for a for any pair of local contexts sufficient for center label a found in any spatial pattern produced by a given tile assembly system.

We now characterize the sufficiency of a set of claws on label a .

Definition 27. *Let S be a set of spatial patterns, K a set of claws, $W_{su_f}^a$ the set of all postconditions in K sufficient for label a , and $W_{bsu_f}^a$ the set of all postconditions in K barely sufficient for a .*

K is perfectly sufficient for a if and only if $W_{suf}^a = \delta_{suf}^{S,a}(W_{bsuf}^a)$.

In a set of claws K , we take the set of postconditions barely sufficient for a and examine all disjunctions of the local contexts of those postconditions and compare that set to the set of all postconditions in K sufficient for a . If K is perfectly sufficient for a and S is generated by a 2D aTAM TAS, then K has a set of postconditions sufficient for a in the full library for the 2D aTAM. Furthermore, as demonstrated in the proof of Proposition 2 and contained in the full library for the 2D aTAM, for any set of disjunctions of sufficient local contexts with identical center labels with respect to the spatial patterns of some aTAM TAS, there exists such a nonempty sufficient subset.

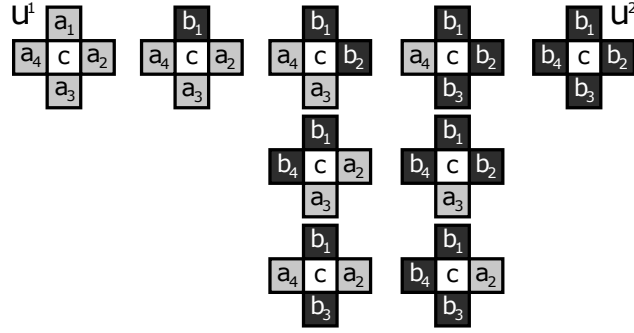


Figure 12: The local contexts shown here are a subset of $\delta(u^1, u^2)$ where u^1 shares only a center label with u^2 . If the depicted local contexts appear as postconditions in some set of claws K and K has no other postconditions sufficient for c , then $W_{bsuf}^c = \{u^1, u^2\}$ and $W = W_{suf}^c = \delta_{suf}^{S,c}(W_{bsuf}^c)$, so that K is perfectly sufficient for c .

Example 6. Consider the disjunctions of local contexts $u^1 = (a_1^n, a_2^e, a_3^s, a_4^w)$ and $u^2 = (b_1^n, b_2^e, b_3^s, b_4^w)$, both sufficient for c shown in Figure 12. These are the local contexts sufficient for c whose supports appear as subpatterns of some spatial pattern $s \in S$, where S is generated by a 2D aTAM TAS with the following relationship between glue strengths.

$$\begin{aligned} g(c, a_1) &< g(c, b_1) \\ g(c, a_2) &> g(c, b_2) \\ g(c, a_3) &> g(c, b_3) \\ g(c, a_4) &> g(c, b_4) \end{aligned}$$

When a set of claws K contains a pair of local contexts isomorphic to u^1 and u^2 and the set of postconditions sufficient for the label isomorphic to c $W_{suf}^c = \delta_{suf}^{S,c}(\{u^1, u^2\})$, then K is perfectly

sufficient for c . Furthermore, there exists an isomorphism between disjunctions in $\delta_{suf}^{S,c}(\{u^1, u^2\})$ and a set of disjunctions in the full library for the 2D aTAM.

Definition 28. A set of claws K over alphabet Σ is consistent if and only if it has the following properties:

1. K is complete under domination;
2. For each label $a \in \Sigma$, for each pair of claws $\kappa_1, \kappa_2 \in K$ with preconditions u^1, u^2 , respectively, for each finger $(u^2, u^2a) \in \kappa_2$, if $u^1 \rightarrow u^2$, then there exists a finger $(u^1, u^1a) \in \kappa_1$.
3. For each $a \in \Sigma$, K is perfectly sufficient for a .

If we consider the partial order of the domination relation on poset local contexts, we can restate the first two properties as follows. Every precondition belongs to a finite connected component such that its postconditions also belong to related finite connected components.

Example 7. Consider the set of two claws shown in Figure 8. The large precondition u dominates the small precondition u' and each postcondition of u' is dominated by some postcondition of u . Since u has 2 non- λ labels more than u' , the set is inconsistent. We can add the claw with precondition u'' to make the set consistent.

Example 8. Consider the set of two claws shown in Figure 13. u' is barely sufficient for green, u is sufficient for red and is not sufficient for green, and $u \rightarrow u'$. If the centers of these postconditions were the same, the set would be consistent. Since the centers are different, the set does not exhibit property 2 and is inconsistent. The set can be made consistent by adding a finger to u such that u is sufficient for green.

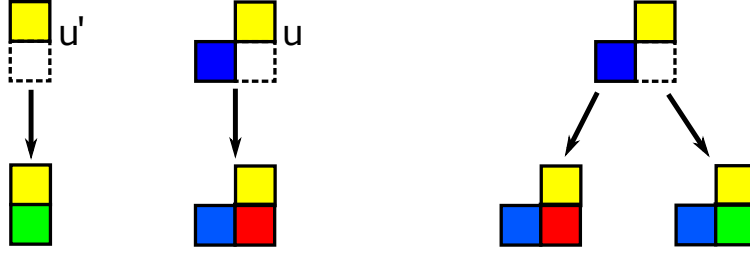


Figure 13: [Left] An inconsistent set of claws with preconditions u and u' . [Right] Adding a finger to u such that u is sufficient for green renders the set consistent.

Example 9. Consider the set of two claws shown in Figure 14. Postconditions v, v' are sufficient for a and $v \leftrightarrow v'$, thus v, v' are barely sufficient for a , i.e., $W_{suf}^a = W_{bsuf}^a = \{v, v'\}$. By Proposition 2, there does not exist a set of spatial patterns S generated by some 2D aTAM TAS such that $W_{suf}^a \neq \delta_{suf}^S(W_{bsuf}^a)$. Therefore, this set of claws is inconsistent as it is not perfectly sufficient for a and W_{suf}^a is not contained in the full library for the 2D aTAM. The set can be made consistent by adding the new claw on the right as shown.

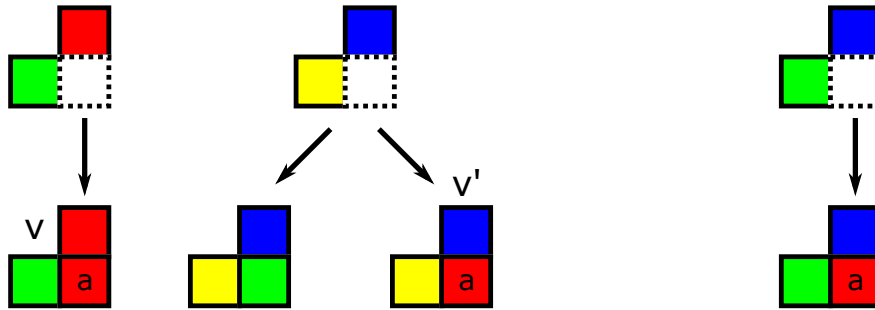


Figure 14: [Left] An inconsistent set of claws with postconditions v and v' sufficient for a . By Proposition 2, the set of claws should contain at least one more claw with a precondition in the disjunction. [Right] Adding a single new claw with a postcondition sufficient for a renders the set consistent.

Structure of Synoptic Patterns

With these preliminaries in place, we now describe how arbitrary synoptic patterns of aTAM TASs may be formed from a set of claws by the recursive application of the operation of claw extension from a seed spatial pattern and *vice versa*.

Definition 29. Let K be a set of claws over a label alphabet Σ and S be the set of all (finite) spatial patterns with labels from K . The extension function $\mathcal{E}: S \times K \rightarrow 2^S$ associates to a spatial pattern $x \in S$ and claw $\kappa \in K$, a set of spatial patterns obtained by replacing a subpattern of x matching the support of κ 's precondition u with the support of ua from finger (u, ua) of κ .

Starting from a seed pattern in $S_0 = \{\sigma\}$ (as in an aTAM TAS), the union of the extensions by all preconditions in a consistent set of claws produces a set of spatial patterns of size one larger containing the seed as a subpattern. Iteratively performing the extension of each pattern in this new set produces a sequence of layers of spatial patterns of larger and larger sizes, given by

$$\begin{aligned} S_0^\mathcal{E} &= S_0; \\ S_{i+1}^\mathcal{E} &= \bigcup_{\kappa \in K} \mathcal{E}(S_i^\mathcal{E}, \kappa) \text{ for } i \geq 0; \\ S_\omega^\mathcal{E} &= \bigcup_{i \geq 0} S_i^\mathcal{E}. \end{aligned}$$

Taken together, then, $(S_\omega^\mathcal{E}, \prec)$ is a synoptic pattern. It is clear that, performed with the set of claws equivalent to a given TAS, this extension closure produces the synoptic pattern generated by the TAS. We now expand Lemma 1 to the synoptic patterns of a 2D aTAM TAS and of a set of claws.

Lemma 2. *Given a 2D aTAM TAS with assemblies S , there exists a finite and consistent set of claws with NEWS supports specifying all possible attachments whose extension closure is a synoptic pattern identical to S .*

Proof. Take the set of claws K known to exist by Lemma 1 and produce a set of spatial patterns from the seed σ by extension closure $S_\omega^\mathcal{E}$. It is easy to verify that K is consistent. Produce a synoptic pattern $(S_\omega^\mathcal{E}, \prec)$. For each pair of spatial patterns (x, xa) in the synoptic pattern of the TAS, there exists a tile type a that extends x to be xa by stable binding to x and, by Lemma 1, produces a corresponding claw in K . Performing an extension on x with this claw produces a set of patterns that contains xa , i.e., $S \subseteq S_\omega^\mathcal{E}$. Conversely, for each pair of spatial patterns $y, yb \in S_\omega^\mathcal{E}$, there exists a claw in K with a precondition whose support is in y and postcondition with center b . b is a tile

type in the TAS and b can stably bind to y in the context consisting of tile types with labels in the support of the precondition. Therefore, the synoptic pattern of the TAS contains a pattern yb for all y and b . Thus, the synoptic patterns of the given 2D aTAM TAS and generated by the set of claws are identical. \square

The question arises as to whether, conversely, an arbitrary set of claws K determines an aTAM TAS, with a finite set of tiles, glues, and a temperature τ . This is not obvious but indeed so, as established in the remainder of this section.

Theorem 1. *For every synoptic pattern \mathbb{S} of a 2D aTAM TAS, the following two conditions are equivalent:*

1. \mathbb{S} is generated by a TAS \mathbb{T} ;
2. \mathbb{S} is generated by claw extension from a seed by a consistent set of claws K .

Proof. Lemma 2 proves that (1) implies (2). For the converse, given a synoptic pattern generated by claw extension on a consistent set of claws K , construct the associated tile assembly system $\mathbb{T} = (T, \sigma, g, \tau)$ in the following way. The tile set T consists of a finite number of k -tuples, where the tile label and glue information along the edges of the tile uniquely defines each tile type. The labels come directly from the labels on the pre- and postconditions. The seed σ is the smallest spatial pattern in the synoptic pattern. Algorithm 2 below produces a system of inequalities (an AMCS, precisely defined below) from K characterizing the attachment conditions in a 2D aTAM TAS. By Lemma 4 below, the AMCS always has a feasible solution if K is consistent. Each adjacent pair of tile labels in any postcondition has a glue interaction that is labeled uniquely by Algorithm 2. Glue strength and the temperature can be determined algorithmically via the b -rule on the AMCS to identify a feasible solution that represents an assignment of integers to glue interactions defined by g and the abstract temperature τ . (When the feasible solution has rational non-integer solutions, we multiply the solution to clear the denominators and produce a new solution with only integer components.) In this way, a TAS \mathbb{T} can be constructed from K . We show that the synoptic pattern of \mathbb{T} is identical to the synoptic pattern generated by claw extension on

K . By the construction, T contains a tile type t for the seed and for the center of each postcondition v in K and the glue function and temperature are such that t stably binds in a local context of tile types with labels from the support of v . Therefore, the synoptic pattern of \mathbb{T} contains patterns x, xt for any x with the support of v 's precondition as a subpattern and xt contains the support of v as a subpattern. By the definition of claw extension, the synoptic pattern generated by claw extension on K from seed σ contains σ and the set of patterns produced by iteratively replacing subpatterns identical to the support of some precondition in a claw κ in K with a subpattern identical to the support of a postcondition in κ . Thus, every pattern in the synoptic pattern generated by claw extension from σ by K is in the synoptic pattern of \mathbb{T} and vice versa. Therefore, (2) implies (1). □

We now provide the definitions and constructions in support of the proof of Theorem 1. First, we describe a system of inequalities called an *assembly model constraint system* (AMCS). Note that the set of tiles for any 2D aTAM TAS produced from a set of claws K is derived from K . Tile labels are cell labels in the pre- and postconditions. Glue labels can be described by coordinating pairs of tile labels in the corresponding direction of the gluing interaction. Attachment in the aTAM is determined by the total strength of glues at the binding site and the temperature. If we treat these as unknown variables (glue strengths and the temperature) we can construct a system of inequalities that characterize the conditions on binding as determined by pairs of preconditions and postconditions. If the system of inequalities afforded by the set of claws is feasible, then any integer solution constitutes valid glue strengths and temperature. The glue function with a domain of glue pairs and a range of glue strengths defined by a feasible solution, along with the temperature, taken together with the seed and tile set defines an aTAM TAS. The system of integer linear inequalities is guaranteed to be either feasible or infeasible by the well known Farkas's Lemma.

Lemma 3. [Farkas's Lemma [32]] *If $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, then exactly one of the following two statements is true:*

1. *There exists an $\mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \geq 0$.*

2. There exists a $\mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{A}^\top \mathbf{y} \geq 0$ and $\mathbf{b}^\top \mathbf{y} < 0$.

We operationalize the lemma using Algorithm 1 called the *b*-rule inspired by Bland’s pivot rule [33] in Linear Programming. The *b*-rule algorithm constructs a system of equations from a system of inequalities by introducing additional slack variables representing the slack in each inequality between the left-hand side and the right-hand side and then uses a series of substitutions to determine the (in)feasibility of the system.

Algorithm 1: [The *b*-rule [34]]

Input: A system of linear inequalities Γ

Output: A feasible solution or ‘Infeasible’

1. Index the inequalities $\alpha \leq \beta \in \Gamma$ with positive integers.
 2. Replace each inequality i by equation $x_i = \beta_i - \alpha_i$, where x_i is a slack variable.
 3. For each slack variable x_i , add inequality $x_i \geq 0$ to Γ .
 4. Temporarily set each variable in the right-hand side to 0 and choose an equation where the left-hand side now evaluates to be negative. (If many left-hand side variables evaluate to be negative, choose the one with the smallest index.) If no equation fits the criteria, the system is consistent. Return a feasible solution to the system of inequalities restricted to the original variables (i.e., the non-slack variables.)
 5. If no variable g_j with a positive coefficient exists in the selected equation, the system is inconsistent. Return ‘Infeasible’. Otherwise, solve for g_j in the equation with smallest index.
 6. Eliminate g_j in each other equation by substitution.
 7. Repeat steps 4-6.
-

To apply the *b*-rule, we must generate a representative system of inequalities. The system should be formed of three subsets: the *model* inequalities, the *glue* inequalities, and the *imputed glue* inequalities. The model inequalities represent the explicit constraints on interactions captured in the model of assembly, e.g., the nonnegative glue strength requirement for the aTAM. Different models may strengthen or relax constraints on gluing that influence the form and number of these inequalities. Additional one-off constraints (e.g., the abstract temperature) should also be represented by a model inequality. The glue inequalities are informed by the model’s defini-

tion of binding and are taken from the postconditions. Imputed glue inequalities are formed from the sufficient conditions for binding. When a dominated local context is insufficient for binding a tile the dominating local context binds, this can be represented as an imputed glue inequality. When a dominating local context is insufficient for binding a tile a dominated precondition binds, this is also an imputed glue inequality. We call such a system of inequalities an *assembly model constraint system* (AMCS). An AMCS for the aTAM is constructible via Algorithm 2. The unknowns $x_{a,v}^d$ are denoted by a tile label a indexed by a local context (either the associated postcondition v or a bespoke local context if no associated postcondition exists) and a direction d of interaction from the set $\{N, E, W, S\}$ forming the variable $x_{a,v}^d$ as a proxy for the glue g between the center labeled a of a local context v to the label in direction d in v .

Algorithm 2: Constructs an AMCS for a set of claws

Input: A finite set of claws K

Output: An assembly model constraint system (AMCS)

1. Let Σ consist of the labels in the supports of preconditions and postconditions in K and W consist of all local contexts w with labels from Σ .
 2. Add ' $x_\tau \geq 1$ '.
 3. For each local context w with center labeled $a \in \Sigma$ and each direction d , add ' $x_{a,w}^d \geq 0$ '.
 4. For each local context w with center labeled a and each direction d where w has label λ , add ' $x_{a,w}^d = 0$ '.
 5. For each postcondition v with center labeled a , add ' $\sum_{d \in D} x_{a,w}^d \geq x_\tau$ '.
 6. For each postcondition v' with center labeled a whose precondition u' is dominated by another precondition u and for each local context w dominated by u where, after mutating the center to label a , $w \rightarrow v'$, add ' $\sum_{d \in D} x_{a,w}^d \geq x_\tau$ '.
 7. For each local context w with center labeled a that is not a postcondition, add ' $\sum_{d \in D} x_{a,w}^d \leq x_\tau - 1$ '.
 8. For each maximal set of postconditions $W^a = \{v : v_{(0,0)} = a, v \text{ does not dominate any other postcondition in } K\}$, verify W_{suf}^a has a set of isomorphic local contexts in the full library. If not, add ' $x_\tau = 0$ '.
 9. Return the current system of inequalities as an AMCS.
-

Lemma 4. *If the AMCS generated by Algorithm 2 for a set of claws K is feasible, then K is consistent.*

Proof. We proceed by contradiction. Suppose the AMCS is feasible and generated from a set of claws, but the set of claws is inconsistent. To be inconsistent, the set of claws must fail to satisfy at least one of the three parts of the definition. To be incomplete, if precondition u^1 dominates no precondition with 1 fewer label and u^1 dominates precondition u^2 , then by step 7, the AMCS contains a glue inequality of the form ' $\sum_{d \in D} x_{a,w}^d \leq x_\tau - 1$ ', where $u^1 \rightarrow w \rightarrow u^2$, $v \neq u^1 \neq u^2$, and $v_{(0,0)} = a$. By step 6, the AMCS contains a glue inequality of the form ' $\sum_{d \in D} x_{a,w}^d \geq x_\tau$ ' and is infeasible, a contradiction. To violate the second property, the set of claws must contain at least one pair of different claws c_1 and c_2 where precondition $u^1 \in c_1$ dominates $u^2 \in c_2$ but no postcondition in c_1 is centered on some label a in a postcondition $v \in c_2$. In step 5, the inequality $\sum_{d \in D} x_{a,u^2}^d \geq x_\tau$ is added that represents the ability to stably bind a tile a to the local context u^2 . In step 7, the inequality $\sum_{d \in D} x_{a,u^1}^d < x_\tau$ is added that represents the inability to stably bind a tile a to the local context u^1 . Note that $u^1 \rightarrow u^2$ and therefore a subset of the terms have sum greater than or equal to x_τ while the sum of all terms is less than x_τ . Therefore, the AMCS is infeasible, a contradiction. To violate the third property, either K does not contain as a postcondition each local context in some sufficient disjunction of a set of postconditions W^a that are barely sufficient for a center labeled a , or K contains as a postcondition each local context in some sufficient disjunction of W^a and some other postcondition v with $v_{(0,0)} = a$ such that $\{v\} \cup \delta^a(W)$ is not a sufficient disjunction. In either case, $x_\tau = 0$ and $x_\tau > 0$, the AMCS is infeasible, a contradiction. In any case, we derive a contradiction. Therefore, the proposition holds. \square

Example 10. *For the set of claws for a 2×2 square shown in Figure 15, Algorithm 2 produces the AMCS described next. Glue labels are abstracted as pairs of adjacent unit labels, focused on the newly occupied center, along with a direction from the set $\{N, E, W, S\}$.*

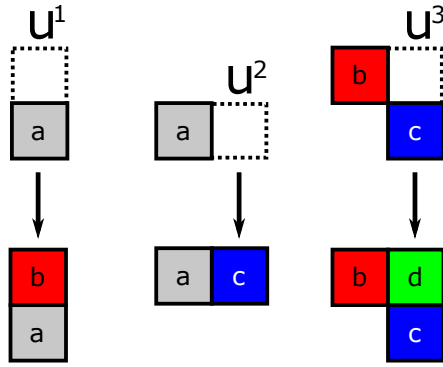


Figure 15: u^1, u^2, u^3 form a consistent set of claws for a TAS that assembles a 2×2 square.

Model	Glue
$x_{b,u^1}^S \geq 0$	$x_{b,u^1}^S \geq x_\tau$
$x_{c,u^2}^W \geq 0$	$x_{c,u^2}^W \geq x_\tau$
$x_{d,u^3}^W \geq 0$	$x_{d,u^3}^W + x_{d,u^3}^S \geq x_\tau$
$x_{d,u^3}^S \geq 0$	
$x_\tau > 0$	
Imputed	
$x_{d,u^3}^S < x_\tau$	
$x_{d,u^3}^W < x_\tau$	

By Algorithm 1, we introduce a nonnegative slack variable for each inequality and rewrite the

inequalities as equations.

$$x_1 = x_{b,u^1}^S$$

$$x_2 = x_{c,u^2}^W$$

$$x_3 = x_{d,u^3}^W$$

$$x_4 = x_{d,u^3}^S$$

$$x_5 = x_{b,u^1}^S - x_\tau$$

$$x_6 = x_{c,u^2}^W - x_\tau$$

$$x_7 = x_{d,u^3}^W + x_{d,u^3}^S - x_\tau$$

$$x_8 = -1 - x_{d,u^3}^W + x_\tau$$

$$x_9 = -1 - x_{d,u^3}^S + x_\tau$$

$$x_{10} = -1 + x_\tau$$

We perform the selection step, identifying the equation with x_8 and solving for variable x_τ , yield-

ing $x_\tau = x_8 + x_{d,u^3}^W + 1$. We then rewrite and simplify the system.

$$\begin{aligned}
x_1 &= x_{b,u^1}^S \\
x_2 &= x_{c,u^2}^W \\
x_3 &= x_{d,u^3}^W \\
x_4 &= x_{d,u^3}^S \\
x_5 &= x_{b,u^1}^S - x_8 - x_{d,u^3}^W - 1 \\
x_6 &= x_{c,u^2}^W - x_8 - x_{d,u^3}^W - 1 \\
x_7 &= x_{d,u^3}^S - x_8 - 1 \\
x_9 &= x_{d,u^3}^W - x_{d,u^3}^S + x_8 \\
x_{10} &= x_8 + x_{d,u^3}^W \\
x_\tau &= x_8 + x_{d,u^3}^W + 1
\end{aligned}$$

We select x_5 and solve for x_{b,u^1}^S , rewriting and simplifying.

$$\begin{aligned}
x_1 &= x_5 + x_8 + x_{d,u^3}^W + 1 \\
x_2 &= x_{c,u^2}^W \\
x_3 &= x_{d,u^3}^W \\
x_4 &= x_{d,u^3}^S \\
x_6 &= x_{c,u^2}^W - x_8 - x_{d,u^3}^W - 1 \\
x_7 &= x_{d,u^3}^S - x_8 - 1 \\
x_9 &= x_{d,u^3}^W - x_{d,u^3}^S + x_8 \\
x_{10} &= x_8 + x_{d,u^3}^W \\
x_\tau &= x_8 + x_{d,u^3}^W + 1 \\
x_{b,u^1}^S &= x_5 + x_8 + x_{d,u^3}^W + 1
\end{aligned}$$

Next, we select x_6 and solve for x_{c,u^2}^W , rewriting and simplifying.

$$x_1 = x_5 + x_8 + x_{d,u^3}^W + 1$$

$$x_2 = x_6 + x_8 + x_{d,u^3}^W + 1$$

$$x_3 = x_{d,u^3}^W$$

$$x_4 = x_{d,u^3}^S$$

$$x_7 = x_{d,u^3}^S - x_8 - 1$$

$$x_9 = x_{d,u^3}^W - x_{d,u^3}^S + x_8$$

$$x_{10} = x_8 + x_{d,u^3}^W$$

$$x_\tau = x_8 + x_{d,u^3}^W + 1$$

$$x_{b,u^1}^S = x_5 + x_8 + x_{d,u^3}^W + 1$$

$$x_{c,u^2}^W = x_6 + x_8 + x_{d,u^3}^W + 1$$

We now select x_7 , continuing to follow the selection algorithm, and solve for x_{d,u^3}^S before rewrit-

ing and simplifying.

$$x_1 = x_5 + x_8 + x_{d,u^3}^W + 1$$

$$x_2 = x_6 + x_8 + x_{d,u^3}^W + 1$$

$$x_3 = x_{d,u^3}^W$$

$$x_4 = x_7 + x_8 + 1$$

$$x_9 = x_{d,u^3}^W - x_7 - 1$$

$$x_{10} = x_8 + x_{d,u^3}^W$$

$$x_\tau = x_8 + x_{d,u^3}^W + 1$$

$$x_{b,u^1}^S = x_5 + x_8 + x_{d,u^3}^W + 1$$

$$x_{c,u^2}^W = x_6 + x_8 + x_{d,u^3}^W + 1$$

$$x_{d,u^3}^S = x_7 + x_8 + 1$$

Finally, we select x_9 and solve for x_{d,u^3}^W , producing $x_{d,u^3}^W = x_7 + x_9 + 1$. We perform substitution and produce a new system of equations with only slack variables and constants in any right-hand

side.

$$x_1 = x_5 + x_7 + x_8 + x_9 + 2$$

$$x_2 = x_6 + x_7 + x_8 + x_9 + 2$$

$$x_3 = x_7 + x_9 + 1$$

$$x_4 = x_7 + x_8 + 1$$

$$x_{10} = x_7 + x_8 + x_9 + 1$$

$$x_\tau = x_7 + x_8 + x_9 + 2$$

$$x_{b,u^1}^S = x_5 + x_7 + x_8 + x_9 + 2$$

$$x_{c,u^2}^W = x_6 + x_7 + x_8 + x_9 + 2$$

$$x_{d,u^3}^S = x_7 + x_8 + 1$$

$$x_{d,u^3}^W = x_7 + x_9 + 1$$

We set the slack variables in every right-hand side to 0, rewrite and simplify to produce a feasible

solution.

$$x_{d,u^3}^W = 1$$

$$x_{d,u^3}^S = 1$$

$$x_{c,u^2}^W = 2$$

$$x_{b,u^1}^S = 2$$

$$x_\tau = 2$$

$$x_1 = 2$$

$$x_2 = 2$$

$$x_3 = 1$$

$$x_4 = 1$$

$$x_{10} = 1$$

For our purposes, we restrict the feasible solution to the nonslack variables.

With these algorithms and results in place the proof of Theorem 1 is complete. We move our focus now to the analysis of the long-term behavior of assembly systems.

Probabilistic Synoptic Patterns

In addition to the difficulties with the naive probability model and Bayesian inference discussed in chapter 1, we also recognize that the process of assembly generally precludes the usual i.i.d. assumptions among a sequence of random variables on discrete spaces. It is incumbent on any probabilistic analysis of the long-term behavior of synoptic patterns, including those that may contain an infinite number of terminal patterns, that the underlying probability measure will assign probabilities to every spatial pattern in such that the process of self-assembly can be treated in a holistic way. We now address these issues by constructing several families of probability

measures on arbitrary synoptic patterns.

Definition 30. A probabilistic synoptic pattern $\mathbb{P}(\mathbb{S}, \rho)$ is a synoptic pattern \mathbb{S} along with a probability measure on the set of elementary events given by

$$\Omega = \{(x, y) \in \mathbb{P} : x \prec y\}. \quad (1)$$

The size of an elementary event (x, y) is $n = |(x, y)| = |x|$, the size of the spatial pattern x .

By analogy with claws, when discussing the spatial patterns in an elementary event, we refer to x as the *precondition* and y as the *postcondition* although they are full spatial patterns, not just contexts. We also expand the definition of cone and anticone with respect to probabilistic synoptic patterns.

Definition 31. In a probabilistic synoptic pattern \mathbb{P} , the cone at the event (x, xa) is $(x, xa)^{++} = \{(x, xa)\} \cup \{(y, z) : (y, z) \in \Omega \text{ and } xa \prec^* y\}$, and the anticone at (x, xa) is the event $(x, xa)^{--} = \{(y, z) : (y, z) \in \Omega \text{ and } y \prec^* x\} \cup \{(x, xa)\}$. When the postcondition is not specified, the cone or anticone also contains all events with the specified precondition and we use the notation x^{++} or x^{--} .

By extending the notion of from spatial patterns to elementary events, we can partition Ω into layers Ω_n containing pairs of patterns in the immediate successor relation with a given size n , for all size $n > 0$, such that $\Omega = \bigcup_{n>0} \Omega_n$ (disjointly.) Such a probabilistic synoptic pattern may be constructed from the set of producible assemblies of a TAS by the following construction on an arbitrary TAS.

Given a probability measure ρ_n defined on each Ω_n , let \mathcal{F} be a σ -algebra on Ω (e.g., the σ -algebra generated by unions of the form $\bigcup_{n>0} E_n$, where E_n is a measurable event in the corresponding σ -algebra for ρ_n .)

Lemma 5. Let \mathbb{T} be a TAS with a notion of size $|\ast|$ and let ρ_n be a sequence of probability measures defined on each Ω_n with corresponding σ -algebras \mathcal{F}_n in the corresponding layers Ω_n . The

family of subsets \mathcal{F} of Ω given by unions of the form $\bigcup_{n>0} E_n$, where E_n s are arbitrary measurable events in the corresponding σ -algebras, is also a σ -algebra.

Proof.

1. \mathcal{F} contains Ω since it is the union $\Omega = \bigcup \Omega_n$ of all full sets of elementary events Ω_n of size n .
2. \mathcal{F} is closed under complementation. Consider a compound event $E = \bigcup_{n>0} E_n$ of events $E_n \in \mathcal{F}_n$. Note the complement of E_n with respect to Ω_n is E_n^c . The complement of E is then $E^c = \Omega - \bigcup_{n>0} E_n = \bigcup_{n>0} \Omega_n - \bigcup_{n>0} E_n = \bigcup_{n>0} (\Omega_n - E_n) = \bigcup_{n>0} E_n^c$. Since \mathcal{F}_n is a σ -algebra for Ω_n , the complement E_n^c with respect to Ω_n is in \mathcal{F}_n . Therefore, E^c is also such a union and belongs to Ω .
3. \mathcal{F} is closed under countable unions. Consider a sequence of compound events E^k ($k > 0$) in each union $E^k = \bigcup_{n>0} E_n^k$, where $E_n^k \in \Omega_n$. The union of the sequence

$$E = \bigcup_{k>0} E^k = \bigcup_{k>0} \bigcup_{n>0} E_n^k = \bigcup_{n>0} \bigcup_{k>0} E_n^k.$$

Since \mathcal{F}_n is a σ -algebra for Ω_n , the union $\bigcup_{k>0} E_n^k$ also belongs to \mathcal{F}_n , and therefore the union E is also in \mathcal{F} .

□

The sequence of probability measures ρ_n can now be extended to a probability measure ρ for events $E \in \mathcal{F}$ in a variety of ways.

Proposition 3. *If w_n is an arbitrary sequence of nonnegative real numbers partitioning unity, i.e. $\sum_n w_n = 1$, then $(\Omega, \mathcal{F}, \rho)$ is a probability space for Ω , where ρ is given by*

$$\rho(E) = \sum_{n>0} w_n \rho_n(E \cap \Omega_n). \quad (2)$$

Proof. As defined, ρ is a function from Ω to $[0, 1]$ since $\rho(E) \leq \sum_n W_n = 1$. If $E = \Omega$, then $\rho_n(E \cap \Omega_n) = \rho_n(\Omega_n) = 1$, for all $n \geq 0$, yielding $\rho(\Omega) = \sum_{n=0}^{\infty} w_n = 1$. A set of pairwise disjoint sets of elementary events $E = \cup_i E_i$ can be written as $E = \cup_i (\cup_n E_{i,n}) = \cup_n (\cup_i E_{i,n}) = \cup E_n$, where $E_{i,n} = E_i \cap \Omega_n$ by intersection with the Ω_n s, a disjoint union. Hence, $\rho_n(E_n) = \sum_i \rho_n(E_{i,n})$. $\rho(E) = \rho(\cup_n E_n) = \sum_{n>0} w_n \rho_n(E_n) = \sum_{n>0} w_n \sum_i \rho_n(E_{i,n})$ Therefore $\rho(E) = \sum_i \sum_{n>0} \rho_n(E_{i,n}) + \sum_i \rho(E_i)$, as required. \square

Examples of such weight sequences include $w_n = 1/2^n$ (*geometric weights*), $w_n = K/n^s$ (*polynomial weights*, e.g., quadratic for $s = 2$ and $K = 6/\pi^2$), and $w_n = K/2^{n-n_0}$ (*radial geometric weights* defined more precisely below), for appropriate normalizing constants K to make them a partition of 1. Geometric weight concentrates most of the probability mass in events representing small assemblies and decreases exponentially fast as the size grows away from the transitions from the seed. Polynomial weights spread this probability across a larger initial segment of the synoptic patterns. Radial weights do likewise but place larger masses around a region of a particular size n_0 .

Example 11. For the synoptic pattern associated with the TAS $\mathbb{T}_{a^n b^n}$ in Figure 4, examples of events in Ω are shown in Figure 16. The size of an assembly is the number of tiles used in its construction. When partitioning Ω , an elementary event $e = (x_1, x_2)$ has size $|e| = |x_1|$. The σ -algebra associated with Ω is 2^Ω . The probability measure is as defined above with geometric weights and the standard counting measure for each Ω_n . The probability of the elementary event $e_1 = (x_1, x_2)$ shown in Figure 16 is

$$\rho((x_1, x_2)) = w_1 \rho_1((x_1, x_2)) = \frac{1}{2}. \quad (3)$$

The probability of $e_2 = (x_2, x_3)$ in Figure 16 is

$$\rho((x_2, x_3)) = w_2 \rho_2((x_2, x_3)) = \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}. \quad (4)$$

Note that there are two events in Ω_2, e_2 and the event where tile a_t attaches instead of tile a . Now

consider the probability of the compound event $E = \{e_1 = (x_1, x_2), e_2 = (x_2, x_3), e_3 = (x_3, x_4)\}$.

$$\rho(E) = \rho((x_1, x_2)) + \rho((x_2, x_3)) + \rho((x_3, x_4)) \quad (5)$$

$$= w_1 \rho_1((x_1, x_2)) + w_2 \rho_2((x_2, x_3)) + w_3 \rho_3((x_3, x_4)) \quad (6)$$

$$= \frac{1}{2} + \frac{1}{4} \cdot \frac{1}{2} + \frac{1}{8} \cdot \frac{1}{3} \quad (7)$$

$$= \frac{2}{3} \quad (8)$$

Intuitively, $\rho(E)$ is the probability of assembling pattern x_4 from pattern x_1 .

By substituting alternative weights, these calculations can vary. With polynomial weights inspired by the ζ function where $n = 2$

$$w = \sum_{i=1}^{\infty} w_i, \quad (9)$$

$$w_i = \frac{1}{\zeta(2) - 1} \cdot \frac{1}{(i+1)^2}. \quad (10)$$

and radial geometric weights defined by

$$w = \sum_{i=1}^{\infty} w_i, \quad (11)$$

$$w_i = \begin{cases} \frac{1}{2^{(2+(n_0-i))}} & 0 < i < n_0 \\ \frac{1}{2^i} & i = n_0 \\ \frac{1}{2^{2+(i-n_0)}} & n_0 < i < 2n_0 \\ \frac{1}{2^{n_0+(i-2n_0)+1}} & 2n_0 \leq i \end{cases} \quad (12)$$

centered at $n_0 = 2$, we can build the following table.

	Weight at size n	1st event	2nd event	Compound event
<i>Geometric</i>	$\frac{1}{2^n}$	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{2}{3}$
<i>Polynomial</i>	$\frac{1}{\zeta(n)-1}$	$\simeq 0.388$	$\simeq 0.086$	$\simeq 0.506$
<i>Radial₂</i>	<i>varies piecewise</i>	$\frac{1}{16}$	$\frac{1}{4}$	$\frac{3}{8}$

$$\begin{aligned}
e_1 &= \left(\begin{array}{|c|} \hline s_1 \\ \hline \end{array}, \begin{array}{|c|} \hline s_2 \\ \hline s_1 \\ \hline \end{array} \right) \\
e_2 &= \left(\begin{array}{|c|} \hline s_2 \\ \hline s_1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline s_2 & a \\ \hline s_1 & \\ \hline \end{array} \right) \\
e_3 &= \left(\begin{array}{|c|c|} \hline s_2 & a \\ \hline s_1 & \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline s_2 & a & a \\ \hline s_1 & & \\ \hline \end{array} \right)
\end{aligned}$$

Figure 16: Shown here is a set of events in the probabilistic synoptic pattern of $\mathbb{T}_{a^n b^n}$. Pattern labels are taken from the tile set. Grey tiles map to the symbol ‘a’ in the alphabet of language $L = \{a^n b^n : n > 0\}$.

This construction will be referred to as the *geometric*, *polynomial*, and *radial_{n₀} geometric* construction, respectively.

Theorem 2. *For any sequence of weights $\{w_n\}_{n>0}$ partitioning 1, any choice of local σ -algebras \mathcal{F}_n and local probability measures ρ_n , the induced probability space with probability measure ρ along with the original synoptic pattern is a probabilistic synoptic pattern.*

Proof. Define Ω to be the set of pairs of spatial patterns in the immediate successor relation for the synoptic pattern, layered as described above. Let ρ_n be a probability measure on \mathcal{F}_n , a σ -algebra of Ω_n . By Lemma 5, \mathcal{F} is a σ -algebra for Ω . Let ρ be an extension defined previously of the collection of ρ_n for all n . As shown above, ρ is a probability measure. This induced probability space along with the synoptic pattern is then a probabilistic synoptic pattern. \square

These families of probability spaces offer an incredible flexibility to the analyst. The weight function and local probability measures can be completely arbitrary. Any function that partitions the unit interval into countable infinitely many weights is a valid weight function. Different weight functions may aid the probabilistic methods used to discern the details of an assembly system underlying a probabilistic synoptic pattern. Polynomially decreasing weights may be useful when properties localized over more extended areas may be of interest (e.g., quantifying the uncertainty in a synoptic pattern or the constituent spatial patterns in the probability space.) Radial geometric weights centered about some critical size can aid the analysis around target patterns of a certain size or the time of their assembly. The geometric weights presented above shift focus to the critical, delicate beginnings of an assembly process, where any small deviation can have potentially radical effects arbitrarily far into the future in a monotonic assembly system. The choice of weights is a critical one to any subsequent analysis and the local probability measure can be tailored to specific laboratory conditions for a practical assembly and choice of Ω_n . The obvious choices are the counting measure or a bespoke assignment of measure in each finite Ω_n in our examples, but more statistically motivated or physically realistic local measures could be used as well.

Probabilistic Analysis of 1D Rulers

In the probabilistic Tile Assembly Model (PTAM) introduced by Chandran et al [11], assembly systems produce a family of 1D spatial patterns (so-called *rulers*) with expected length n_0 as terminal assemblies using $\Omega(\log n_0)$ tile types. The PTAM defines an assembly system as a tuple (T, E, g) , where the tile set T is a multiset of tile types and may contain many copies of each unique tile type. Tile types have glues on their east and west edges only and each has *multiplicity* equal to the number of its copies in T . The seed tile type $E \subset T$ has a noninteractive glue on its west edge so that growth happens only in one direction. g is the glue function as in the aTAM definition: glues are either interactive and form a stable bond with at least one glue or are non-

teractive and form no stable bonds with any glue. This forms a *de facto* temperature 1 assembly system (although no explicit abstract temperature τ is given.)

A PTAM assembly system is said to *probabilistically assemble a ruler of length n_0* [11] if the expected length of a terminal produced by the assembly system is n_0 . To compute this probability, they construct the probability of a tile binding in some context as the ratio of that tile type's multiplicity to the sum of the multiplicities of all tile types that could bind in that context. The PTAM thus models concentration and a rate of accretion in a well-mixed solution of tiles. The probability of a particular assembly x_k of size k is defined recursively as $P(x_k) = p_t P(x_{k-1})$, with the probability of the seed $P(x_1) = 1$. p_t is the probability of binding the tile that produces assembly x_k from assembly x_{k-1} . The probability of a set of assemblies (e.g., the set of terminals) is the sum of the individual probabilities.

Chandran et al. define a PTAM assembly system construction with parameterized multiplicity to assemble rulers of arbitrary expected length n_0 as (T, E, g) with $T = \{S, H, C\}$, $E = S$, and a glue function g that admits the following positive strength bonds: H^w binds to S^e and C^e , C^w binds to S^e and C^e . H has multiplicity 1 and C has multiplicity $n_0 - 2$. In the next example, we define a comparable aTAM TAS and carry out a similar analysis for the expected length of a terminal in our framework through the construction of a probabilistic synoptic pattern and relevant random variable.

Example 12. *We consider the aTAM assembly system $\mathbb{T} = (T, E, g, \tau)$ with $T = \{S, C, H\}$, $E = S$, and a glue function g that admits the following positive strength bonds: H^w binds to S^e and C^e , C^w binds to S^e and C^e . The temperature $\tau = 1$. Since H does not use its east glue to form a positive strength bond, an assembly becomes terminal after an H tile binds.*

We take the set of all spatial patterns S produced by the assembler \mathbb{T} along with the immediate successor relation \prec on S to form a synoptic pattern, as shown in Figure 17.

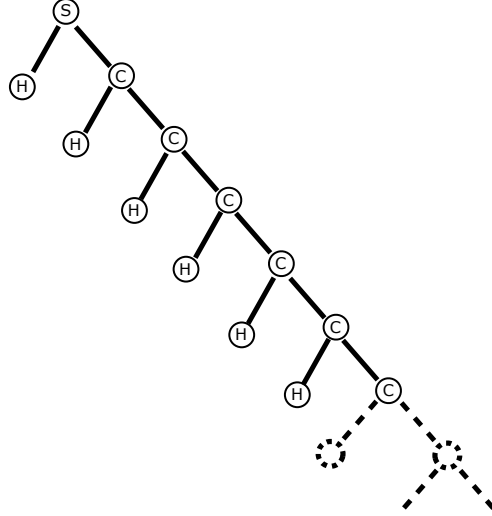


Figure 17: The synoptic pattern for the TAS in Example 12 is reminiscent of a herringbone pattern, with a nonterminal spine and terminal ribs. Patterns having most recently bound a C tile form the spine, and patterns having most recently bound a H tile form ribs of length 1.

To obtain a probabilistic synoptic pattern, we define the local probability spaces indexed by size, from individual sample spaces, σ -algebras, and local probability measures respectively given by:

- $\Omega_n = \{(x, y) : x \prec y \text{ and } |x| = n\}$,
- \mathcal{F}_n is the full σ -algebra on Ω_n , and
- $\rho_n((x, y)) = \begin{cases} .25 & \text{when } y \text{ is terminal} \\ .75 & \text{otherwise.} \end{cases}$

By Lemma 5, \mathcal{F} is a σ -algebra for Ω and, by Proposition 3, ρ is a probability measure for \mathcal{F} .

By Theorem 2, this probability space with an appropriate weight function and the above synoptic pattern forms a probabilistic synoptic pattern for \mathbb{T} . We may now calculate the expected size of a terminal assembly by considering the random variable on Ω given by

$$X(e = \{(x, y)\}) = \begin{cases} n = |y| & \text{when } y \text{ is terminal} \\ 0 & \text{otherwise} \end{cases}$$

Table 1 shows this probabilistic analysis of the length of a ruler for a trio of weight functions and three ρ_n of interest. Columns are labeled with the local probability of assembling a terminal for

Table 1: Probabilistic Analysis of Linear Assemblies, N=5

$E(X)$	$\rho_n((x,y)) = \frac{1}{4}$	$\rho_n((x,y)) = \frac{16}{17}$	$\rho_n((x,y)) \rightarrow 1$
Geometric	.75	2.824	3
Radial ₄	1.328	5	5.3125
Polynomial	∞	∞	∞

three different ρ_n and rows are labeled with three different weight functions. Generally, the expected length of a ruler is much shorter than in the PTAM probabilistic analysis. X observes 0 for infinitely many nonterminal assemblies. When the probability of formation of a terminal is increased, the expected length increases to a limiting value. Under the Radial₄ geometric weights, we can choose ρ_n such that $E(X) = 5$ and recover Chandran et al.'s analysis. Under the Polynomial weights, the expected length of a terminal is ∞ regardless of the local probability of assembling a terminal.

In our framework, the aTAM assembly system \mathbb{T} given in Example 12 can probabilistically assemble rulers of arbitrary integer length n_0 through a reasonable choice of weights and local probability measures ρ_n . The region of interest in the synoptic pattern should be around the terminals of size n_0 , so radial geometric weights centered on events representing terminals of size n_0 comprise our weight function. The local probability measures should weight terminal events such that $E(X) = n_0$. We can choose such a measure uniformly for every local probability space (independent of size) due to the similarity of Ω_n for all n . The sample space always consists of two events, one representing the assembly of a terminal and one representing the assembly of a non-terminal, regardless of n . We weigh terminals highly as a function of the desired expected length. For $n_0 > 2$, we define each ρ_n in this way

$$\rho_n((x,y)) = \begin{cases} 2^{n_0-1}/(2^{n_0-1} + 1) & \text{when } y \text{ is terminal} \\ 1/(2^{n_0-1} + 1) & \text{otherwise.} \end{cases} \quad (13)$$

Radial _{n_0-1} geometric weights together with these local probability measures on the appropriate local probability spaces complete the probabilistic synoptic pattern for any choice of $n_0 > 2$.

Another assembler presented by Chandran et al. is designed to produce rulers with expected length $n_0 = 2^k$ for $k \in \mathbb{N}$ with fewer than n_0 tile types. For $n_0 = 8$, Chandran et al.'s power-of-2 PTAM assembler $\mathbb{T}_8 = (T_8, E, g_8)$ has tile set $T_8 = \{S, S_1, T_1, T_2, R_1, R_2\}$ and includes seed $E = S$, each with multiplicity 1. The glue function g_8 defines the following tile interactions:

- S^e binds only S_1^w ,
- S_1^e binds T_2^w or R_2^w ,
- T_2^e binds T_1^w or R_1^w ,
- R_2^e binds T_2^w or R_2^w ,
- T_1^e is noninteractive, and
- R_1^e binds T_2^w or R_2^w .

Although \mathbb{T}_8 has fewer tile types than the expected length of the rulers Chandran et al. calculate the TAS produces, the resulting synoptic pattern is far more complex than in Example 12. New terminals are produced at a rate on the order of $\Theta(F(n))$ and nonterminals on the order of $\Omega(F(n))$, where $F(n)$ is the n th Fibonacci number. As a part of their probabilistic analysis, their work makes the implicit assumption that a terminal is inevitable. As we demonstrate here by a methodology that explicitly constructs a probability space and appropriate random variable, this assumption is not well founded. The behavior of an assembler in the limit and the qualities of the limiting pattern or patterns are dependent on the branching nondeterministic assembly process and specific choices in building the probability space in which the assembler is analyzed. In Example 13, we perform a probabilistic analysis on \mathbb{T} for $n_0 = 8$ with various weights and ρ_n .

Example 13. *Beginning with \mathbb{T} as described in Example 12, we form a synoptic pattern. We build the local probability spaces indexed by size, forming the individual sample spaces, σ -algebras, and local probability measures in the following way:*

- $\Omega_n = \{(x, y) : x \prec y \text{ and } |x| = n\}$,
- \mathcal{F}_n is the full σ -algebra, and

Table 2: Probabilistic Analysis of Linear Assemblies, N=8

$E(X)$	$\rho_n((x,y)) = .5$	$\rho_n((x,y)) = \frac{128}{129}$	$\rho_n((x,y)) \rightarrow 1$
Geometric	1.5	2.977	3
Radial ₇	4.031	8	8.063
Polynomial	∞	∞	∞

$$\bullet \rho_n((x,y)) = \begin{cases} .5 & \text{when } y \text{ is terminal} \\ .5 & \text{otherwise.} \end{cases}$$

As in Example 12, we compose these local probability spaces and a weight function to form a probability space. The synoptic pattern and probability space form a probabilistic synoptic pattern. We can calculate the expected size of a terminal assembly of this assembly system in a probabilistic synoptic pattern by considering the random variable on Ω given by

$$X(\{(x,y)\}) = \begin{cases} n = |y| & \text{when } y \text{ is terminal} \\ 0 & \text{otherwise} \end{cases}$$

We again take different weight schemes and local probability spaces under analysis to produce Table 2. Columns are labeled with the local probability of assembling a terminal for three different ρ_n . Rows are labeled with three different weight functions. The expected length of a ruler in various probabilistic synoptic patterns for various weights is again much shorter than in the PTAM probabilistic analysis. When the probability of formation of a terminal is increased, the expected length increases to a limiting value. With Radial₇ geometric weights and a ρ_n that concentrates mass in terminal events, we recover the PTAM probabilistic analysis. Under the Polynomial weights, the expected length of a terminal is again infinite (∞) regardless of the local probability of formation of a terminal.

In summary, Examples 12 and 13 show that setting a local probability by construction and defining a probability of formation without explicitly defining the probability space is not enough to establish a robust foundation for probabilistic analysis. Example 13 shows that the rate at which

terminals are produced in ratio to the nonterminals is important to any resulting analysis of an assembler. Ignoring large parts of the set of assembled patterns to focus on only the set of terminal patterns produces an incomplete analysis of the properties of the assembler, even when the view is restricted to only measurable qualities of the terminals. Fundamentally, a probabilistic synoptic pattern performs two roles in a well founded probabilistic analysis not present in the literature: the probability space is well defined for all spatial patterns and the relationships between them, and the synoptic pattern is rooted in the complete behavior of the assembler. Furthermore, this robust framework does not need a bespoke model to produce a reasonable probabilistic analysis, as shown by the flexibility of the simple aTAM assembly system \mathbb{T} .

CHAPTER 3

PROBABILISTIC ANALYSIS OF LONG-TERM BEHAVIOR

Traditionally, tile self-assembly has been focused on the production of single, unique patterns due to its usefulness as a representation of nanoscale manufacturing processes. It is desirable to be able to analyze tile self-assembly systems to understand the long-term behavior of the assembly process, especially if the systems generate arbitrarily large patterns. This would facilitate the process of producing more general self-assembly systems based on tile self-assembly models. Unfortunately, an algorithmic analysis is impossible in general because determining the products of aTAM TASs can be as complex as deciding the long-term behavior of arbitrary Turing machines, e.g., solving the Halting Problem [35]. The standard aTAM assumes assembly proceeds in the Euclidean plane without any errors and produces fully 2D spatial patterns as products. Relating an aTAM TAS and a Turing machine (TM) that terminates in some configuration is not so simple as examining the spatial pattern and the configuration of the Turing machine (TM) to find out whether they are identical. Strings are 1D words and spatial patterns are 2D words. Inherent in the representation of a computation by an aTAM TAS simulating a Turing machine is a notion of an interpreter to map the halting configuration to the spatial pattern and *vice versa*. Interpreting an arbitrary 2D word has a deep literature already [36, 37] with many interesting branches in natural language processing and graph theory. In tile assembly, this problem of interpretation is usually addressed by testing for an essentially linear subpattern. For example, the zigzag transformation described earlier produces the total history of configurations of the simulated TM as a spatial pattern. The final configuration is ‘read’ from a designated region of a terminal. This interpretation reduces the complexity of the spatial pattern to a linear subpattern that depends on the smaller assemblies that were produced before it like the halting configuration of the TM depends on the prior configurations. Hand-in-hand, restricting the aTAM to the 1D lattice also restricts its computational power. The natural dynamics of assembly are fundamentally different from the TM and reasonable choices must be made when comparing the long-term behavior of

the models. In this work, we are interested fundamentally in the dynamics of assembly and the process by which new assemblies are formed. We take the approach that the assembler should be responsible for producing a 1D word as directly as the associated grammar. We read the word starting from the seed, following positive strength glues until the spatial pattern is exhausted. This serialization respects the dynamics of assembly, but it does not greatly limit the generative power of self-assembly. Such systems are not Turing computation universal, but they penetrate into the Chomsky hierarchy and have been considered in the literature as interesting assemblers for both traditional and probabilistic analysis [38, 11].

In this work, we broaden the scope to include assembly systems that produce potentially infinitely many unique patterns of interest, but will continue the focus on sets of *terminal* patterns as representative of this long-term behavior. Furthermore, interesting sets of terminal patterns can be produced by relatively simple TAS. We construe self-assembly as a stochastic process [39] on probabilistic synoptic patterns in various ways and seek to find tools for predicting long-term behavior *probabilistically*. Markov chains [39] are well known probabilistic models that make these predictions possible, so we explore such models on probabilistic synoptic patterns. In this chapter, we discuss Markov chains of essentially 1D TASs, i.e. assemblers whose products can be described by single strings. The class of *regular assemblers* [40] are the computationally simplest TAS in the aTAM. These assembly systems are capable of simulating the class of right or left linear grammars and are capable of producing infinite sets of terminal patterns as described by regular sets in the Chomsky hierarchy.

1D Language Assemblers

Regular languages are the simplest class of string languages defined in the Chomsky hierarchy. There are many equivalent definitions of regular languages, including that a string language is regular if and only if there exists a deterministic finite automaton that recognizes it, i.e., that accepts every string in the language and does not accept any string not in it.

Definition 32. A deterministic finite-state automaton (fsa) is a 5-tuple $D = (Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of states,
- Σ is a (finite) alphabet of symbols,
- $\delta : Q \times \Sigma \rightarrow Q$ is a transition function,
- $q_0 \in Q$ is the initial state, and
- $F \subseteq Q$ is a set of accepting states.

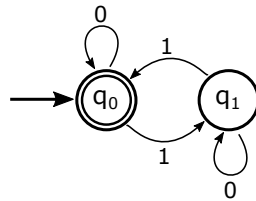


Figure 18: Here is an fsa that recognizes the language of binary strings consisting of all strings containing an even number of ‘1’s.

Given a string x (a 1D word) over the alphabet Σ and an fsa M , x takes M from the initial state q_0 through a sequence of states forming a path of length $n = |x|$ and labeled by x leading to some state q . M accepts x if and only if $q \in F$. The language recognized by M is the full set of strings accepted by M . A language L is regular if it is recognized by some fsa.

An equivalent definition of regular languages can be given with another type of production system, a Chomsky grammar. A *left (or right) linear grammar* is a generator for regular languages and is a particular case of a context-free grammar.

Definition 33. A context-free grammar $G = (V, \Sigma, R, S)$ consists of

- V , a finite set of variables;
- Σ , a finite set of terminals;
- \mathcal{R} , a set of production rules of the form $A \rightarrow \beta$ that describe how variables are rewritten into strings over the alphabet $\Sigma \cup V$ (sentential forms);
- S is the “sentencehood” variable in V .

The language generated by G is the set of strings in Σ^* generated from S by transforming sentential forms according to G 's rules. G is called right-linear if every rule in \mathcal{R} is either of the form $A \rightarrow aB$, i.e. $\beta = aB$, or $A \rightarrow a$ where $a \in \Sigma$, $B \in V$. G is called linear if every rule in \mathcal{R} has at most one variable in its right-hand side β .

Thus, iterated applications of the rules in a right-linear grammar only produce sentential forms with at most one nonterminal as the last character.

In [40], Winfree demonstrates a DNA tile self-assembly system that produces as a set of terminal assemblies precisely those patterns encoding all the strings of a given regular language. The proof is a robust algorithm for transforming a right (or left) linear grammar into a TAS in the aTAM. Using Winfree's algorithm, every regular language has a corresponding regular assembler that produces terminals described exactly by the strings in the language.

Algorithm 3: [Winfree's Algorithm [40]]

Input: A right linear grammar

Output: A regular assembler

1. Form a tile set T in the following way:
 - (a) For each rule in R of the form $X \rightarrow xY$, create a tile $(x, \lambda, Y, \lambda, X)$, where x is a tile label and the glue set contains X, Y
 - (b) For each rule in R with the form $X \rightarrow x$, create a tile $(x, \lambda, \lambda, \lambda, X)$
 2. Set the abstract temperature $\tau = 1$
 3. Create g , a diagonal glue function with all positive glue strengths equal to τ
 4. Create a seed tile $\sigma = (\lambda, \lambda, S, \lambda, \lambda)$
 5. Return the regular assembler (T, σ, g, τ)
-

Stochastic Processes for 1D Assemblers

We now formalize the notion of stochastic process. Intuitively, a stochastic process is a sequence of observations on successive outcomes of a random experiment in a probability space.

Definition 34. A stochastic process is a sequence of random variables $\{X_t\}_{t \in \mathbb{N}}$ defined on a common probability space $(\Omega, \mathcal{F}, \rho)$.

In the literature [12, 41, 42], continuous stochastic processes, often in analogy with physical chemical systems, have been used to analyze time complexity in tile assembly models. To establish a foundation for probabilistic analysis of tile assembly, we start with a formalization of a TAS as such a process. The underlying random experiment is the attachment of a tile, randomly selected from the tile set, at a particular position in an assembly. Generally, self-assembly is a stateful process with dependencies. The existence of a large pattern depends on having assembled smaller patterns under the constraints imposed by the glues and the container space. Iterating the random experiment highlights these dependencies between outcomes that *violate the usual assumptions in probability theory*, namely that random variables are *independent and have identical distribution* (the so-called i.i.d. assumption.) By using probabilistic synoptic patterns and the supporting constructions, we can design processes that factor in these dependencies and holistically analyze the long-term behavior of an assembler.

Example 14. Given a probabilistic synoptic pattern on a 1D TAS $\mathbb{T} = (T, \sigma, g, \tau)$ that assembles a set of spatial patterns P , consider a stochastic process that observes the t^{th} or the last tile in the postcondition of an elementary event (x, y) , i.e. for all $t \in \mathbb{N}^+$, $Y_t : \Omega \rightarrow T$ is given by

$$Y_t((x, y)) = \begin{cases} y_t & \text{if } t \leq |y| \\ y_{|y|} & \text{if } t > |y|. \end{cases} \quad (14)$$

Given a tile label $a \in T$, the event $[Y_t = a]$ consists of all elementary events where the spatial pattern y has a tile of type a in the t^{th} position (if $t \leq |y|$), along with the elementary events of length less than t with a spatial pattern ending with a tile of type a (if $t > |y|$.) In particular, the event $[Y_t = a]$ is empty if a tile of type a is never attached to an assembly. Therefore, the event $A_t = [Y_1 = a_1] \cap \dots \cap [Y_t = a_t]$ is the cone $(x_t, y_t)^{++}$ with $y_t = a_1 a_2 \dots a_t$. When $(y_t, y_t a_{t+1})$ exists, the conditional probability is then $\rho([Y_{t+1} = a_{t+1}] \mid A_t) = \frac{\rho(A_{t+1})}{\rho(A_t)}$. Otherwise, the conditional

probability is 0.

The union of the values of the random variables in stochastic processes in these examples could be an increasingly large or infinite set (which are usually called the *states* of the process), but we are primarily concerned with observation of a finite number of values. In these examples, we assume that the state set (either the tile set or a set of patterns) has been enumerated as positive integers by some indexing function in lexicographic order so that the RVs in the stochastic processes make numerical observations, as required. We restrict this union in various ways to maintain the accessibility of the representation to analysis. Ultimately, the goal is to identify stochastic processes that both capture the assembly process but are also amenable to analysis of their long-term behavior.

Example 15. *Given a probabilistic synoptic pattern on a 1D TAS $\mathbb{T} = (T, \sigma, g, \tau)$ that assembles a set of spatial patterns P , a stochastic process $\{X_t\}$ observes the tile label at the t^{th} position in the postcondition of an elementary event (x, y) , if such a tile exists, or declares no observation was made, i.e. for all $t \in \mathbb{N}^+$, $X_t : \Omega \rightarrow P$ is given by*

$$X_t((x, y)) = \begin{cases} y_t & \text{if } t \leq |y| \\ 0 & \text{else, if } t > |y|. \end{cases} \quad (15)$$

Given a tile label $a \in T$, the event $[X_t = a]$ consists of the union of the cones of elementary events where the postcondition y has a tile labeled a in the t^{th} position. The event $[X_t = 0]$ consists of all the elementary events of length less than t . In particular, the event $[X_t = 0] = \Omega$ if P is finite and t exceeds the size of the largest pattern in P . Therefore, the event $A_t = [X_1 = a_1] \cap \dots \cap [X_t = a_t]$ consists of cone $(x_t, y_t)^{++}$ with $y_t = a_1 a_2 \dots a_t$. As in Example 14, when $(y_t, y_t a_{t+1})$ exists, the conditional probability is then $\rho([Y_{t+1} = a_{t+1}] | A_t) = \frac{\rho(A_{t+1})}{\rho(A_t)}$. Otherwise, the conditional probability is 0.

In contrast to the previous examples, we next explore stochastic processes with an infinite number of observed states.

Example 16. Given a probabilistic synoptic pattern on a 1D TAS $\mathbb{T} = (T, \sigma, g, \tau)$ that assembles a set of spatial patterns P , a third stochastic process is given by

$$W_t((x, y)) = x \quad (16)$$

Given an observation spatial pattern $s \in P$, the event $[W_t = s]$ consists of the elementary events in the claw with precondition $x = s$. Therefore, the event $A_t = [W_1 = s_1] \cap \dots \cap [W_t = s_t]$ is either empty, or consists of events where $x_1 = s_1, \dots, x_t = \dots = s_t$ where the patterns s_k are nested as prefixes (denoted $s_t \leq s_{t+1}$) of the next, for all $1 \leq k \leq t - 1$. When $s_t \leq s_{t+1}$, the conditional probability is then $\rho([W_{t+1} = a_{t+1}] | A_t) = 1$ or 0 otherwise.

In the next example, we explore a stochastic process where the event corresponding to a particular observation at time t may be infinite. To facilitate an analysis of long-term behavior, it may be more suitable to have a finite state set of observations (discussed in the next section.) Therefore, in the following example we truncate the synoptic pattern at some size n and take the set of spatial patterns of size n or smaller to be the (finite) state space. The resulting stochastic process is called a *truncation*, or more precisely the *n-truncation*, on the associated probabilistic synoptic pattern, where *the random experiment consists of selecting at random an elementary event representing a preexisting assembly binding a tile at a particular position*. To ease discussion, we refer to a spatial pattern directly as a sequence of tile types $x_1x_2\dots x_n$ rather than by its index.

Example 17. Given a probabilistic synoptic pattern on a 1D TAS $\mathbb{T} = (T, \sigma, g, \tau)$ that assembles the set of spatial patterns P , a stochastic process $\{Z_t\}$ observes an initial segment of the assembly up to a certain size, for example,

$$Z_t((x, y)) = \begin{cases} x & \text{if } |(x, y)| < t \text{ and} \\ x_1x_2\dots x_t & \text{otherwise.} \end{cases} \quad (17)$$

Alternatively, we can put a bound on the size of the observation and define the *n-truncation* $\{Z_t^n\}$ just as $Z_t = Y_t$ for $t < n$ and $Z_t = Y_n$ otherwise. Given a spatial pattern $s \in P$, $[Y_t = s]$ is, depending

on $|s|$, either a claw of elementary events with precondition s or a cone s^{++} . In particular, when s is terminal (exists only as a postcondition in Ω) $[Y_t = s] = \emptyset$. Therefore, the event $A_t = [Y_1 = s_1] \cap \dots \cap [Y_t = s_t]$ is either empty or $[Y_t = s_t]$. For A_t to be empty, there must have been some event $[Y_i = s']$, $i < t$ where $|s'| < i$ and $[Y_{i+1} = s_{i+1}]$ with $s_{i+1} \neq s'$ or two events $[Y_i = s_i]$ and $[Y_j = s_j]$, $i < j$ where s_i is not a prefix of s_j . The conditional probability is thus either 0 or $\rho[X_{t+1} = s_{t+1} | A_t] = \rho(A_{t+1})/\rho(A_t)$.

Up to n , this is a faithful representation of the assembly process. Past n , the stochastic process lacks the resolution to represent the full status of the assemblies and it can only myopically observe initial chunks of size n in larger assemblies.

Markov Chains

Discrete Markov processes, a class of stochastic process, are particularly interesting because they admit a very elegant and relatively simple analysis of their long-term behavior. The assumptions on the model are what allow this analysis. Markov processes have bounded memory, i.e., their observations only depend on the observations immediately before, not on farther previous observations. We are especially interested in Markov processes with a finite number of states, commonly referred to as *Markov chains*.

Definition 35 (*Markov property (MP)*). A discrete stochastic process $\{X_t\}$ in a probability space $(\Omega, \mathcal{F}, \rho)$ on state space S is a Markov process if and only if

$$\rho(X_{t+1} = s_{t+1} | X_1 = s_1, X_2 = s_2, \dots, X_t = s_t) = \rho(X_{t+1} = s_{t+1} | X_t = s_t) \quad (18)$$

for every $t \geq 0$ and $s_1, s_2, \dots, s_t, s_{t+1} \in S$.

The Markov process is called time homogeneous if the probability in the right-hand side is independent of t , depending only on $(s_i, s_j) \in S^2$. In that case, \mathbb{M}_{s_i, s_j} (also denoted \mathbb{M}_{ij} or p_{ij}) is the transition probability given by $\rho(X_{t+1} = s_j | X_t = s_i)$. The process is a Markov chain if the set of

states S is finite.

We remark that it is customary to assume the MP is satisfied when events in the conditional probabilities (such as $[X_t = s_t]$) are empty, and the conditional probability is not defined (division by 0.) Obviously, the MP is not guaranteed for an arbitrary stochastic process that appears to represent the assembly process. For example, the stochastic process of a 1D aTAM TAS in Example 14 is not a Markov chain.

Example 18. *To demonstrate $\{Y_t\}$ does not have the MP, we note that spatial patterns belonging to unique runs may still have the same tile type in the same position. The event $[Y_t = a_t]$ may be a forest of cones with some scattered smaller events while $\cap_t [Y_t = a_t]$ is either a single cone or empty. The sequence of observations either corresponds to a valid run up to time t or it does not. Thus, when the intersection is not empty, $\rho(Y_{t+1} = s_{t+1} \mid Y_1 = s_1, Y_2 = s_2, \dots, Y_t = s_t) \neq \rho(Y_{t+1} = s_{t+1} \mid Y_t = s_t)$ and $\{Y(t), t \in \mathbb{N}^+\}$ is not a Markov chain.*

On the other hand, the stochastic process $\{Z_t\}$ of Example 17 is a Markov process.

Proposition 4. *The stochastic process $\{Z_t\}$ in Example 17 is a Markov process.*

Proof. To show that the MP holds for all t , we note that the intersection of two different claws is empty, the intersection of a claw and a cone is either the claw or empty, and the intersection of two cones in the synoptic pattern of a 1D TAS is either the cone pointed at the larger pattern or empty. To guarantee the MP, we analyze cases where $[Z_t = s_t]$ is a claw or a cone.

When $[Z_t = s_t]$ is a claw and the intersection is not empty, $[Z_1 = s_1] \cap \dots \cap [Z_t = s_t] = [Z_t = s_t]$ and the probability on the left-hand side and right-hand sides are equal. When $[Z_t = s_t]$ is a cone and the intersection is not empty, the intersection is again exactly $[Z_t = s_t]$ since no prior observation by any Z_i is of an event of size $|(x, y)| < i$ and every set of outcomes associated with the prior observations has $[Z_t = s_t]$ as a subset (i.e., is a cone that contains the cone s_t^{++} .) The conditional probabilities on the left-hand and right-hand sides are thus equal.

Furthermore, the intuitive idea of assembly is preserved in this Markov process through a representation of assembly as a nesting of cones pointed at successive patterns in an assembly se-

quence. It cannot be the case that the intersection is a claw when $[Z_t = s_t]$ is a cone. By the definition of the stochastic process, claws are the preimages of observations smaller than the index of the random variable. If $[Z_t = s_t]$ is a cone, it is pointed at s_t , a spatial pattern that is larger than every observation s_i for $1 \leq i < t$. Therefore, the subsequent observations of random variables in the stochastic process are indeed associated with nested cones in the synoptic pattern. \square

When discussing time homogeneous Markov chains, the *stochastic transition matrix* or stochastic matrix holds the key to an analysis of the long-term behavior. Operations on the associated stochastic matrix can help to understand the long-term behavior of a Markov chain.

Proposition 5 ([39]). *A transition matrix M of a Markov chain with state set $S = \{s_1, s_2, \dots, s_n\}$ satisfies the following conditions:*

- a. $0 \leq p_{ij} \leq 1$, for all s_i, s_j ,
- b. $\sum_j p_{ij} = 1$, for all s_i , and
- c. M_{ij}^k is the probability of being in state s_j after k steps given starting state s_i

Properties (a) and (b) define a stochastic matrix \mathbb{M} .

Although the process $\{Z_t\}$ in Example 17 is a Markov process and any n -truncation is a Markov chain, they are not time homogeneous. We can construct a new time homogeneous Markov chain informed by these stochastic processes. We derive a stochastic matrix from the pairwise matrices computed from the conditional probabilities $\rho(Z_{i+1} | Z_i)$. By the Kolmogorov Extension Theorem [43], this stochastic matrix has a backing stochastic process founded on a probability space.

Example 19. *We construct a homogenized stochastic matrix M_{sq} from the n -truncation $\{Z_t^n\}$ in Example 17 by an iterative construction from the empty matrix using Algorithm 4.*

For a probabilistic synoptic pattern with geometric weights and a local counting measure of the

Algorithm 4: [Compute Square Matrix M_{sq}]

Input: A matrix M , a stochastic process $X(t)$, and a size $n > 0$

Output: An $n \times n$ stochastic matrix M_{sq}

1. For t from 1 to n
 - (a) Compute $\rho(X_{t+1} = s_{t+1} | X_t = s_t)$ for each pair of observations s_t, s_{t+1}
 - (b) Build M_{t+1} from these values
 - (c) Compare M to M_{t+1} and compute M' by taking rows in M_{t+1} not in M and concatenating them with M
 - (d) Assign M' to M
 2. Add rows from the identity matrix to form a square matrix M_{sq} from M , filling any empty entries in M_{sq} with 0s if needed.
 3. Return M_{sq} .
-

simple herringbone synoptic pattern, a homogenized 5-truncation is shown here.

$$\begin{bmatrix} .5 & .5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .5 & .41667 & .08333 & 0 & 0 & 0 & 0 \\ 0 & 0 & .4 & 0 & .5 & .1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .4 & .5 & .1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For general n -truncations without the special row-wise structure of $\{Z_t^n\}$, averaging the individual matrices and padding appropriately to form a square matrix yields a stochastic matrix. This transformation of a series of matrices of conditional probabilities into a single stochastic matrix will prove useful in probing into the long-term behavior of a stochastic synoptic pattern.

It is indeed possible to use the thorough analysis afforded by Markov chains to learn something

useful about the long-term behavior of self-assembly systems through synoptic patterns. Since the probabilistic synoptic pattern is potentially infinite, the Markov chain will need to perform a kind of averaging over the whole space by mapping the potentially infinite set of states in the assembly process to the finite set of states in the Markov chain.

Definition 36. Let \mathbb{P} be a probabilistic synoptic pattern with weights w_n and probability measure ρ , let $\{X_t\}$ be a stochastic process on \mathbb{P} with state set R and let ε be a real number in the unit interval. A Markov chain M with states S and transition matrix $[p_{ij}]$ is an ε -approximation of $\{X_t\}$ if and only if, there exists a function $f: R \rightarrow S$ such that for all pairs (i, j) ,

$$|E(Z_{ij}) - p_{ij}| \leq \varepsilon, \quad (19)$$

where $Z_{ij}(t) = \rho(B_t^j | A_{t-1}^i)$ with $B_t^j = \{b \in \Omega : X_t(b) \in f^{-1}(j)\}$, $A_{t-1}^i = \{a \in \Omega : X_{t-1}(a) \in f^{-1}(i)\}$ and $E(Z_{ij}) = \sum_n w_n Z_{ij}(n)$.

We first examine the definition for soundness, i.e. is it true that a stochastic process on a probabilistic synoptic pattern that is a Markov chain is a 0-approximation of itself?

Proposition 6. *If the stochastic process $\{X_t\}$ on a probabilistic synoptic pattern is a time homogeneous Markov chain, then the process admits a 0-approximation.*

Proof. If f is the identity function on the state space of $\{X_t\}$, then $B_t^j = [X_t = j]$ and $A_{t-1}^i = [X_{t-1} = i]$ and the conditional probability $\rho(B_t^j | A_{t-1}^i) = \rho(X_t = j | X_{t-1} = i)$. $\rho(X_t = j | X_{t-1} = i)$ is independent of t , so the expectation of a constant random variable is its constant value and $|E[Z_{ij}] - p_{ij}| = 0 = \varepsilon$, as required. \square

We next examine a simple state Markov chain with three states that preserves the assembly relationship in the probabilistic synoptic pattern. The stochastic process is time nonhomogeneous, but only on the parity of the time index t . By alternating definitions of random variables, the intersection becomes the image of the observation of one random variable through the lens of another random variable's observation. If this is the stochastic process of a probabilistic synoptic

pattern, the related time homogeneous Markov chain is a $\frac{1}{3}$ -approximation, shown next. Through constraint optimization on the values in the resulting stochastic matrix, a derived time homogeneous Markov chain is shown to be a $\frac{1}{6}$ -approximation.

Example 20. *We first define the Parity stochastic process, a time nonhomogeneous stochastic process on a probabilistic synoptic pattern.*

$$X_{\text{odd}}((x,y)) = \begin{cases} x_{|x|} & \text{if } |(x,y)| \equiv 0 \pmod{2} \\ y_{|y|} & \text{else;} \end{cases}$$

$$X_{\text{even}}((x,y)) = \begin{cases} x_{|x|} & \text{if } |(x,y)| \equiv 1 \pmod{2} \\ y_{|y|} & \text{else.} \end{cases}$$

Thus, the event $[X_{\text{even}} = s_2] \cap [X_{\text{odd}} = s_1]$ consists of the events where s_1 binds s_2 . This behavior is brittle, though. When X_{odd} makes an observation, it is for the entire probabilistic synoptic pattern. The notions that s_1 binds s_2 and that s_2 binds some new tile type s_3 are incompatible (the intersection is empty) since X_{odd} would have to observe s_1 and s_3 simultaneously. The behavior is interesting enough for an assembly process where limited memory precludes such concerns.

We can homogenize the Parity stochastic process any number of ways. If we take the nonzero conditional probabilities between the odds and evens directly from the definition of the stochastic process, we produce the following stochastic matrix.

	0	1	S
0	.25	.25	.5
1	.25	.25	.5
S	.5	.5	0

It is natural to test this new time homogeneous Markov chain as an approximation of the original stochastic process. Take f to be the identity and use the probabilistic synoptic pattern's geometric weights for each $E(Z_{ij})$. The result is a $\frac{1}{3}$ -approximation. With the same f , we can optimize

the stochastic matrix to form a new time homogeneous Markov chain that is a witness for a $\frac{1}{6}$ -approximation.

While the Parity stochastic process is simple, it does illuminate the long-term behavior of the approximated probabilistic synoptic pattern. The ε is sensitive to the weight function in the probabilistic synoptic pattern. In Example 20, the geometric weights essentially set the transition probabilities from S to other states. The details of the probabilistic synoptic pattern are set by the experimenter. Weighting small patterns highly implies a slow assembly process. After a long time, the experimenter expects to see more small patterns in their set of assemblies.

The n -truncation as an approximation is similar to the Parity stochastic process. It is dependent on the choice of weight function, but can give additional insight into the assembly process without compromising the analysis of the long-term behavior as aggressively as partitioning the state space into 3 parts.

Example 21. *We first demonstrate a reasonable application of the definition for a particular stochastic process, namely the MP of self-assembly on a probabilistic synoptic pattern. For this fixed stochastic process on a probabilistic synoptic pattern with geometric weights we show the n -truncation is a .25-approximation for any n with this function*

$$f(r) = \begin{cases} r & |r| < n \\ r_1 r_2 \dots r_n & |r| \geq n \end{cases} \quad (20)$$

B_t^r is a compound event that is exactly the cone r^{++} when $|r| > t$ or a claw with precondition r when $|r| \leq t$. By definition, X_t has no observations of size larger than t . A_{t-1}^r only contains observations of size smaller than t . Considering the conditional probability, when the intersection exists, written $\frac{B_t^j}{A_{t-1}^i}$ it has mass 1 when $j = i < t, r$ and 0 when $j \neq i < t, r$. The n -truncation sets

$p_{ij} = \frac{\rho(B_{|j|}^j)}{\rho(A_{|j-1|}^i)}$. The weighted sum is, then,

$$E(Z_{ij}) \in \left\{ \sum_{k=2}^{|j|} w_k \frac{\rho(B_k^j)}{\rho(A_{k-1}^i)} + w_{|j|}, \sum_{k=2}^{|j|} w_k \frac{\rho(B_k^j)}{\rho(A_{k-1}^i)}, 0 \right\} \quad (21)$$

Either the intersection exists at size $|j|$ and larger (meaning $i = j$), or the intersection exists exactly at size $|j|$ (meaning $B_{|j|}^j$ is in $A_{|j-1|}^i$), or the intersection is empty. In any case, by shuffling the geometric weights such that the most weight is given to observations at time $|j|$, the absolute value of the difference is no more than half the ratio of two adjacent weights, i.e., $\varepsilon = .25$. We can continue to improve this epsilon by using deeper knowledge of the assembler. We match weights for each Z_{ij} to the relative mass of cones represented by $B_{|j|}^j$ and $A_{|j-1|}^i$. The higher the ratio, the higher the weight, since the remaining values will either be 1 (receiving more weight) or 0 (no weight consideration is given).

Previously, we have loosely used the idea of ‘long-term behavior’ to mean how the modeled assembler spends most of its time. Through the definition of ε -approximation, we can tie this loose idea of long-term behavior to the precise concept of *communication classes* used for the same purpose in the analysis of in Markov chains by partitioning the state set into specially identified classes.

Definition 37. Let M be a Markov chain with transition matrix $[p_{ij}]$ and its powers $[p_{ij}^{(n)}]$. A state j is said to be accessible from state i , written $i \rightarrow j$, if there exists $n \in \mathbb{Z}^+$ such that

$$p_{ij}^{(n)} > 0, \quad (22)$$

i.e., there is a positive probability of reaching j from i in finitely many steps. Two states i and j communicate, denoted $i \leftrightarrow j$, if and only if j is accessible from i and i is accessible from j . The period of a state i is the greatest common divisor of $\{n \geq 1 : p_{ii}^{(n)} > 0\}$. The chain is ergodic if and only if there is one communication class (the chain is irreducible) and the period of any state is 1 (the chain is aperiodic.)

Communication defines a binary equivalence relation among states in a Markov chain and thus the state set S can be decomposed into equivalence (called *communication*) classes. Two states belong to the same class if and only if they communicate with each other.

Proposition 7 (Decomposition Theorem for Markov Chains [39]). *The communication classes C_1, C_2, \dots, C_k form a partition of the state set of a Markov chain, i.e., for every i, j*

$$C_i \cap C_j = \emptyset \text{ (for } i \neq j\text{), and}$$

$$\bigcup_i C_i = S.$$

Thus, any two states in the same class transition from one to the other, but states from different classes will not.

Definition 38. *A state i is recurrent if and only if $\sum_{n=1}^{\infty} p_{ii}^{(n)} = \infty$. Equivalently, the probability of returning to state i is 1 for infinitely many times n . If state i is not recurrent, it is called transient. A communication class that contains recurrent (transient) states is a recurrent (transient, respectively) communication class.*

Once a Markov chain enters a recurrent state, one can guarantee that the system will return to that state again and again, infinitely often. Our interest is in those communication classes containing states representing spatial patterns produced by the assembler whose presence is guaranteed in the limit, the recurrent communication classes. Were the experimenter to leave and return, in what condition would they find their experiment? An ε -approximation is a time homogeneous Markov chain and can provide these answers through the calculation of its communication classes.

Example 22. *Consider the n -truncation $\{Z_t^n\}$ on the herringbone probabilistic synoptic pattern. The synoptic pattern has a regular structure, existing patterns are either extended by a self-binding tile type, or enter a terminal branch. The state set is S , labeled by spatial patterns up to size n . We recall the assembly partial order, the notion that large patterns cannot grow to*

taining both 0_c and 1_c . $\{0_c, 1_c\}$ is a recurrent communication class. After a long time, we expect the assembler to be producing spatial patterns corresponding to longer and longer binary strings.

Example 24. Consider the Markov chain $\{X_t\}$ on the probabilistic synoptic pattern of a TAS that assembles infinitely many unique patterns, some of which are terminal. $\{X_t\}$ has two states, one labeled “terminal” s and one “nonterminal” r , along with the conditions that $p_{ss} = 1$, $p_{rs} > 0$, and $p_{rr} > 0$. We ignore the natural choice of f to produce an ε -approximation with ε close to 1 by mapping states that represent nonterminals to s and states that represent terminals to r . $\{s\}$ is a recurrent communication class that fails to capture the long-term behavior of the TAS through the probabilistic synoptic pattern.

The truncation technique to produce a Markov chain from a Markov process can be refined for the analysis of long-term behavior through communication classes by the use of a mapping between the spatial patterns and a finite set of states. In the case of the size truncation for arbitrary assemblers, the function is the identity up to size n . Each regular assembler has an associated fsa that recognizes the regular language assembled. Take the fsa mapping function to be the function $f : A \rightarrow S$ that maps spatial patterns in A to states in the fsa by simulating the fsa on the string representation of the spatial pattern and mapping the spatial pattern to the state the fsa is in when the string representation is exhausted. When f has a finite range, the size truncation is unnecessary and f maps the infinite set of spatial patterns onto the state space of the fsa.

Example 25. We utilize the truncation technique on a finite state stochastic process associated with the regular assembler for the language of strings divisible by three. This produces a Markov chain with three states $\bar{0}, \bar{1}, \bar{2}$. The truncation for this Markov process is a homogenization of the time nonhomogeneous Markov chain, resulting in the following stochastic matrix of a Markov chain.

As in the Parity example, this homogenization is not unique. Any procedure that dutifully transforms the potentially infinite set of time-dependent matrices of conditional probabilities is a valid

	$\bar{0}$	$\bar{1}$	$\bar{2}$
$\bar{0}$.85	.15	0
$\bar{1}$.16	.68	.16
$\bar{2}$.13	.19	.68

homogenization. Above, the geometric weights from the probabilistic synoptic pattern are used.

Next, we homogenize with the radial weights.

	$\bar{0}$	$\bar{1}$	$\bar{2}$
$\bar{0}$.94	.06	0
$\bar{1}$.08	.84	.08
$\bar{2}$.06	.12	.82

A homogeneous Markov chain is amenable to the analysis of its long-term behavior through a steady state analysis.

	Geometric	Radial
$\bar{0}$.275	.557
$\bar{1}$.358	.31
$\bar{2}$.367	.133

Both homogenizations are ergodic and the single communication class is faithful to the long-term behavior of the associated assembler.

The homogenization procedure remains an opportunity for effective summary, regardless of the source time nonhomogeneous Markov chain. Any procedure that faithfully transforms the potentially infinite set of time-dependent matrices of conditional probabilities is a valid homogenization. When the language recognized by a fsa is infinite, the finite prefixes and suffixes outside of the pumpable portion of strings in the language have an outsized effect at small sizes, so we also explore the use of radial weights in addition to geometric weights from the probabilistic synoptic pattern.

Taken together, this approach provides the analyst a robust technique to examine the long-term behavior of any regular assembler. It extends from an fsa, a deterministic automaton, to a tile assembly system, a stochastic model of growth and is a well founded and sound statistical method-

ology. In the specific case of regular assemblers, the use of the weights from the probabilistic synoptic pattern will produce a 0-approximation, a stochastic model from a deterministic one.

Theorem 3. *The homogenized truncation of a regular assembler with matching weights is a 0-approximation of the associated stochastic process.*

Proof. Take f to be the function that maps spatial patterns to the state the fsa is in after processing the binary string representation of that spatial pattern. In Definition 36, B_t^j is the set of elementary events such that observations by X_t get mapped to j by f . Since the set of observations of random variables in the stochastic process is identical to the states in the Markov chain, f is the identity. Thus, $B_t^j = [X_t = j]$. $Z_{ij}(t)$ is then the conditional probability $\rho(X_t = j | X_{t-1} = i)$ and $E(Z_{ij}) = \sum_n w_n \rho(X_t = j | X_{t-1} = i) = p_{ij}$ where w_n is the weight at size n from the probabilistic synoptic pattern. Therefore, $|E(Z_{ij}) - p_{ij}| = 0$ for all pairs (i, j) . \square

Thus, for the regular assemblers, we have a tool to gain some understanding of the long-term behavior in the recurrent communication classes with the steady state vector as well as a good approximation of the stochastic process of assembly in a probabilistic synoptic pattern.

ε -Approximability of Linear Languages

Linear string languages are those generated by the linear grammars, i.e., context-free grammars with at most one nonterminal on the right-hand side of every rule. This class of languages extends the notion of the regular languages described by the right-linear grammars used in the construction of the class of regular assemblers. These languages are known to be recognized by one-turn pushdown automata (pda), finite state machines with access to an infinite stack and the caveat that after some transition decreases the size of the stack, no subsequent transition can increase the size of the stack.

Definition 39. *A pushdown automaton (pda) is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$, where*

1. Q is the finite set of states,

2. Σ is the finite input alphabet,
3. Γ is the finite stack alphabet,
4. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow \mathcal{P}(Q \times \Gamma_\varepsilon^*)$ is the transition function
5. $q_0 \in Q$ is the initial state, and
6. $F \subseteq Q$ is the set of accepting states.

A one-turn pda has the additional conditions that

1. $Q = Q_1 \cup Q_2$, where $Q_1 \cap Q_2 = \emptyset$,
2. $q_0 \in Q_1$, and
3. each transition $\delta(q, a, z) = (q', z')$ is defined such that either
 - (a) $q \in Q_1, q' \in Q, a \in \Sigma_\varepsilon, z, z' \in \Gamma^+$, OR
 - (b) $q, q' \in Q_2, a \in \Sigma_\varepsilon, z \in \Gamma, z' \in \Gamma_\varepsilon$.

Theorem 4 ([44]). *The class of one-turn pda recognizes the linear languages.*

Following the blueprint from the prior section, we present an algorithm to construct an aTAM TAS to produce a linear tile representation of strings in the language generated by a linear grammar.

We note that, as with the regular assemblers, the abstract synoptic pattern of the resulting aTAM TAS is a tree (i.e., a rooted acyclic graph.)

Example 26. *We present a linear grammar that produces the language $L = \{a^n b^n : n \geq 1\}$,*

$$S \rightarrow aSb. \tag{23}$$

Algorithm 5: [Linear Grammar to Linear Assembler]

Input: A linear grammar G that generates language L

Output: An aTAM TAS T whose terminals correspond to strings in L

1. Convert each rule that is not right- or left-linear into two rules, one right-linear and one left-linear by introducing a new variable;
 2. For each right-linear rule, create a tile type as in Winfree's algorithm3;
 3. For each left-linear rule, create a tile type as in Winfree's algorithm, except the east and west glues are mirrored;
 4. For any introduced variables or pre-existing variables with rules that are a mix of left-linear and right-linear or with at least one linear rule, produce a set of tile types that describe the deterministic substrings of terminals derived. If a variable can derive itself (the substring can be arbitrarily long), add k new tile types labeled with the k terminals derivable from the variable in any sentential form;
 5. When the ratio of the lengths of dependent substrings is not 1, produce a set of tile types representing a deterministic substring by turning north (south) and reproducing the deterministic substring in part before turning back south (north) to return to the main axis of assembly. This ensures the rate at which substrings are generated in the first half of the string matches the rate at which they are generated in the second half;
 6. The center of any of these substrings is represented with a special pair of 'turning' tile types, the first with a glue on its west and south, the second with a glue on its north and west, with the west glue marked as mirrored with a superscript m ;
 7. Create a seed tile and a tile type whose east glue is the start variable as in Winfree's algorithm;
 8. Glues labeled with matching variables have positive strength. Glues on mirrored tile types with the same label are differentiated by a superscript m and are given positive strength;
 9. Glues on tile types designed to interact deterministically are given positive strength;
 10. Return the resulting TAS.
-

First, we split the rule $S \rightarrow aSb$ by adding a new variable V_1 to produce the equivalent grammar

$$\begin{aligned} S &\rightarrow aV_1 \\ V_1 &\rightarrow Sb. \end{aligned} \tag{24}$$

Then we create tile type $\{a, \lambda, V_1, \lambda, V_1\}$ for the rule $S \rightarrow aV_1$. Next, we create tile type $\{b, \lambda, V_1^m, \lambda, V_1^m\}$ for the rule $V_1 \rightarrow Sb$. The ‘turning’ tile types are next, $\{a, \lambda, \lambda, T, V_1\}$ and $\{a, T, \lambda, \lambda, V_1^m\}$. Finally, we create the seed construction tile types $\{\varepsilon, S, \lambda, \lambda, \lambda\}$ and $\{\varepsilon, \lambda, V_1, S, \lambda\}$. The glue function is diagonal and the strength of each non-zero interaction is τ , the temperature of the system. This TAS is the one presented in Figure 4.

The one-turn pda that recognizes this language is a finite state machine and we use these states in a definition of a stochastic process on the probabilistic synoptic pattern of the TAS produced by the above algorithm.

Example 27. Given a probabilistic synoptic pattern on a 1D TAS $\mathbb{T} = (T, \sigma, g, \tau)$ that assembles a set of spatial patterns whose terminals correspond to strings in a language recognized by a one-turn pda P , consider a stochastic process that observes in an elementary event (x, y) , the state of P having processed the string corresponding to x .

By applying the truncation technique, we can derive from this time nonhomogeneous Markov chain a homogenized Markov chain, as in the previous section.

Theorem 5. The homogenized truncation of an aTAM TAS corresponding to a linear grammar with matching weights is a 0-approximation of the associated stochastic process.

Proof. We follow in the steps of the proof strategy used in the previous section, instead using this new construction from the one-turn pda. Again, we take f to be the function that maps spatial patterns to the state the one-turn pda is in after processing the binary string representation of that spatial pattern. In Definition 36, B_t^j is the set of elementary events whose observations by X_t get mapped to j by f . Since the set of observations of random variables in the stochastic process is

identical to the states in the Markov chain, f is the identity. Thus, $B_i^j = [X_t = j]$. $Z_{ij}(t)$ is then the conditional probability $\rho(X_t = j \mid X_{t-1} = i)$ and $E(Z_{ij}) = \sum_n w_n \rho(X_t = j \mid X_{t-1} = i) = p_{ij}$ where w_n is the weight at size n from the probabilistic synoptic pattern. Therefore, $|E(Z_{ij}) - p_{ij}| = 0$ for all pairs (i, j) . □

The insight that supports these 0-approximations of essentially linear TAS is three-fold. We first note the stochastic process is a production experiment, each iteration produces a spatial pattern. This is true for the stochastic process of any self-assembler. Uniquely to the essentially linear TAS, the produced assemblies have spatial patterns that are serializable without affecting the dynamics of the assembler in producing them. We can use the n -truncation for the stochastic process as a 0-approximation because for any time t , the first t positions of any spatial pattern represent the first t tiles laid in the corresponding assembly. Since there is no confluence, these tiles in this sequence define a unique spatial pattern in the synoptic pattern. Finally, finite state automata have decidable halting problems and thus there is a procedure to decide, for any finite string (spatial pattern), the state of the finite state machine after processing that finite string. If any of these three supporting conditions fails for a recognizer or the assembler, we cannot guarantee a 0-approximation for the assembler with these methods. Whether these approximations can be extended to other families remains an interesting question for further research.

CHAPTER 4

APPROXIMATION BY SELF-ASSEMBLY

Examining the long-term behavior of a computationally complex assembler through approximation by Markov chain does not fully address the connection between theoretical algorithmic self-assembly and practical algorithmic self-assembly. Theory informs the design of experimental systems, but the physical chemical systems themselves often require herculean experimental efforts and dedicated design tools to fine-tune and reduce errors in the final product. Historically, these tools rely on making errors less energetically favorable through techniques like proofreading and algorithmically limiting facets for spurious nucleation [8]. Robust error-free assembly in these contexts requires scaling, redundancy, and careful design. A growth error where a tile experiences a weak, transient bond but is geometrically or energetically locked-in by subsequent growth can be made to have an arbitrarily low probability at the cost of scaling, i.e. a sequence of n incorrect attachments must be formed before a temperature stable growth error occurs. The costs of these solutions are not especially high and they have been shown to be effective *in vitro*. In terms of the products of these assembly processes, the transformed TASs exhibiting this robust error-free assembly exhibit dynamics that may not reflect the desired dynamics exhibited by an ideal assembler (one in an error-free model.) Proofreading scales the assembled patterns or requires particular dynamics that force certain sequences of attachments. Reducing spurious nucleation in hierarchical models requires the use of careful design principles such that certain small patterns cannot possibly be assembled by the assembler. These heuristics may be effective, but are not comprehensive.

In Moore et al. [45], we approach the problem of robust error-free assembly through the approximation of an ideal assembler by a realistic assembly process, under mild assumptions on the nature of the assembly model (not necessarily aTAM.) That process is taken to produce the entire set of all assembled patterns with no assumption about the method of assembly proceeding as intended, but rather allowing for potential mishaps. Here we extend that analysis to more general

situations by enhancing the notion of *probabilistic assemblability* into an approximation methodology, leveraging the framework developed in chapters 1-3. When an assembler \mathcal{G} is intended to assemble a set of spatial patterns in a synoptic pattern \mathbb{H} and errors occur, then \mathcal{G} only approximately assembles \mathbb{H} . In this chapter we introduce the notion of probabilistically approximable assembly to quantify and assess the quality of such errorful assemblers.

Probabilistically Approximable Assemblers

We begin with a notion of pattern similarity. We say one pattern is similar to another pattern of the same size when they can be overlapped so that some positions share matching labels. It contains exactly the maximal subpattern with only matching labels (nonblank matches identical nonblank, blank matches blank) for a fixed translation. The degree of similarity is a ratio of matching nonblank labels to total nonblank labels.

Definition 40. *Let p be a real number $0 \leq p \leq 1$. Two spatial patterns x, y are similar modulo p , denoted $x \sim_p y$, if and only if they are of the same size n and there exists a spatial pattern z that is a subpattern of x and y with size $|z| > pn$.*

Two patterns are thus similar modulo $p = 1$ if and only if they are identical. They are similar modulo $\frac{1}{2}$ if they can be overlapped so that the labels of at least half of their unit cells match. Since every pair of patterns of the same size are similar modulo 0, we assume that $p > 0$.

Definition 41. *Let H be a set of target patterns, \mathcal{G} an assembler, and G the set of patterns assembled by \mathcal{G} . Let ρ_n be a sequence of probability measures for the probability space indexed by a notion of size n on the set of elementary events $\Omega = G \cup H$ as described in chapter 2. The assembler \mathcal{G} probabilistically approximates H if and only if there exists a constant $0 < p \leq 1$ such that*

$$\forall n, \rho_n(G_n | H_n) \geq p, \tag{25}$$

and

$$\forall n \forall y \in G_n \exists x \in H_n \text{ such that } x \sim_p y, \quad (26)$$

where G_n (H_n , respectively) denotes the set of patterns of size n assembled by \mathcal{G} (just contained in H , respectively.) The constant p is called the strength of \mathcal{G} .

Thus, every pattern x in the target set probabilistically approximated by an assembler with strength p has a witness assembly y that is similar to x modulo p , although the complementary subpatterns in x and y may be entirely different. Thus, even when the assembler makes patterns outside of the target set, a significant proportion of these nontarget assemblies are close to target patterns of the same size.

This concept also allows us to focus on the ability for one assembler to approximate another through probabilistic assemblability.

Definition 42. *An assembler \mathcal{H} that assembles a set of patterns H is p -approximable if there exists an assembler \mathcal{G} that probabilistically approximates H with some strength $p > 0$*

In chapter 3, we were concerned with exhibiting an approximation that preserves the local behavior of the stochastic process of assembly to some degree ϵ in order to obtain an informative description of the long-term behavior of the assembler. Armed with a notion of probabilistic assemblability, we now can directly understand the long-term behavior of a target assembler via a potentially simpler assembler whose long-term behavior may be more amenable to analysis. In the aTAM, there is an open question about the relative computational power of cooperative, sometimes called temperature 2, and noncooperative (temperature 1) assembly. It has been shown under a variety of constraints that some terminal patterns are assemblable at temperature 2 but not at temperature 1, but the more general question about the ability of temperature 1 systems to simulate a significant class of temperature 2 systems is under intense scrutiny from a variety of interesting directions [27, 46, 47, 48, 28, 49, 47]. We approach the question through an idea of approximation that differs from extant analyses of approximate assembly [50, 51]. When we relax the notion of simulation to one of approximation, some of the power of cooperativity to pro-

duce interesting patterns can be harnessed in noncooperative systems at the price of a notion of *efficiency* that involves the strength and a notion of *purity* (inversely, *impurity*) of the approximation. The notion of purity separates those assemblies that are responsible for useful assembly work (have some target pattern in their cone) from those that cannot possibly contribute to assembling target patterns of a larger size. *Promising* patterns are equivalent to some target pattern or may grow to become a target pattern, and are positive examples for approximation.

Definition 43. *In the context of Definition 41, a promising pattern is a spatial pattern x in G assembled by \mathcal{G} whose cone x^{++} contains at least one spatial pattern in the target set H .*

Proposition 8. *The assemblies of an assembler \mathcal{G} that probabilistically approximates a target set H can be partitioned into two sets G' (the promising patterns), and E (the errors or junk patterns) given by $G' = \{a : \exists b \in H \cap a^{++}\}$ and $E = G - G'$.*

An assembler that generates only promising patterns may not become a 1-approximation, as a promising pattern may not be similar modulo 1 to any pattern in the target set, but the intuition should be that such an assembler is responsibly producing patterns in the target set. With this notion of promising patterns we can now define the purity of an assembler, or how carefully it manages assembly of nontarget patterns.

Definition 44. *A set of patterns H is called approximable with impurity q if there exists some assembler \mathcal{G} assembling a set of patterns G , a constant $q > 0$, a probability space $(\Omega = G \cup H, \sigma, \rho)$, and a notion of size n such that*

$$\forall n, \rho_n(G_n - G'_n) \leq q \rho_n(H_n) \quad (27)$$

If H is probabilistically approximable by an assembler \mathcal{G} with strength p and impurity q , the ratio $\varepsilon = \frac{p}{q}$ is called the efficiency of \mathcal{G} . When $\varepsilon = \frac{p}{q} > 1$ for some assembler, the target set H is efficiently probabilistically approximable.

We now present an example of how one assembler may approximate a target set to show why

this concept is useful to the study of tile self-assembly. This example is adapted from a series of examples originally presented in [45].

Example 28. *We construct a simple tile assembly system called the RGB gadget that exhibits a signalling behavior in a modified aTAM . The seed is a containing square of length L on a side, with seed tiles from the tile set incorporated into it at positions $\frac{L}{2}$ along the west and south sides. The red and green signals grow from the seed tiles and can cooperate to produce a blue signal, but nondeterminism in the model may prevent this cooperation through a blocking behavior. Here we consider how we can use a cooperative (temperature 2) RGB gadget G_c to approximate a target set H consisting of a blue signal shown in Figure 20 and all assemblable subpatterns. \mathcal{G}_c consists of a tile set as shown in Figure 21, with the expected glue function and $\tau = 2$. To determine the strength p and impurity q , we first construct a probability space $(\Omega = G_c \cup H, \sigma, \rho)$ with the Manhattan distance as the notion of size n . Since every target pattern is produced, the patterns that are least similar to any target pattern in the set dictate an appropriate value of p . In this case, this pattern is the one that binds two errorful tiles to the seed and $p = \frac{1}{2}$. q is sensitive to the local measure ρ_n . Using the counting measures for ρ_n , we recover the q as in [45], otherwise q is the ratio of the mass of the errors to the mass of the targets at each size. Assuming ρ_n is not highly sensitive to n (the mass of an error at size n is given by some $O(n)$ function), the efficiency ϵ_c under the counting measure is then $\Omega(\frac{1}{L})$, where L is the length of a side of the container seed square. Weighting targets highly at some choice sizes, or partitioning the weights between targets and assembled patterns according to an $\Omega(n)$ function affects q directly and can overwhelm the effect of the size of the seed such that the efficiency is more sensitive to the changing ρ_n than changing the size of the seed.*

Theorem 6. *If a set of patterns P is probabilistically approximable with probability p , impurity q and efficiency $\epsilon = \frac{p}{q}$ by a monotonic pattern assembler \mathcal{G} , then*

1. *target patterns are assembled with positive probability at least $\frac{Kp}{p+q} = \frac{K\epsilon}{1+\epsilon} > 0$ where K is the minimum positive probability of attachment of any single tile in the assembler; promis-*

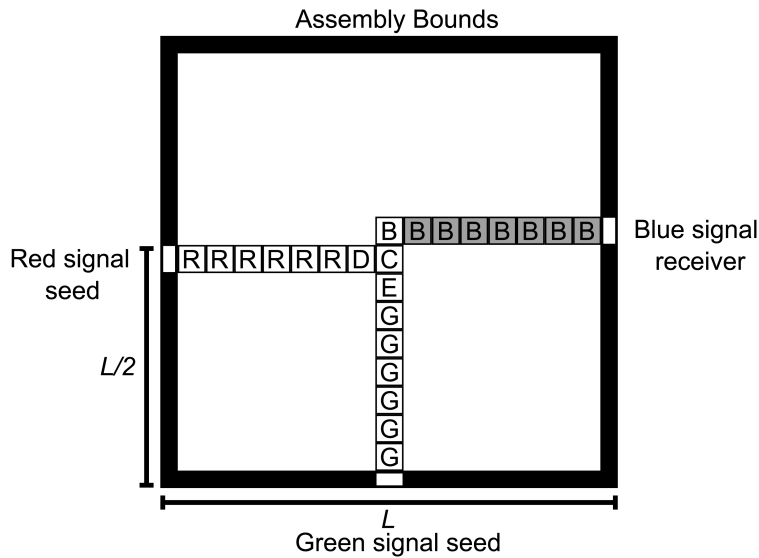


Figure 20: The ideal RGB assembler is designed to grow two signals, one red from site R and one green from site G, toward a point of cooperation C. The seed is an $L \times L$ square perimeter where inputs R and G and output B are to be placed. Cooperation is achieved at the point C, where the signals would “meet” to effect the AND and produce a deterministic signal that is recorded at output B.

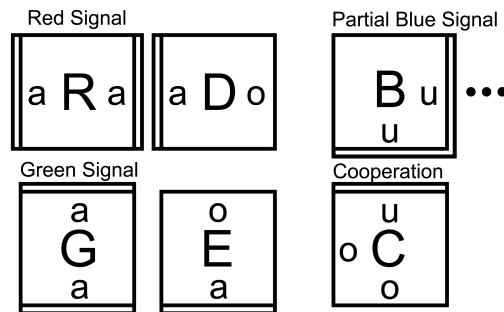


Figure 21: A tile set designed for a temperature 2 assembly system that uses cooperation to simultaneously detect nondeterministic input signals, red (R) and green (G), and from the designed point of cooperation output a deterministic signal to the blue (B) site. For cooperative assembly, double edges indicate strength 2 glues, and single edges strength 1. When the system is run in noncooperative mode, all glues are strength 1. R and G propagate from the inputs r and g by a series of R and G tiles. The D and E tiles are placed nondeterministically at some point along the signal to enable attachment of the cooperation tile C. At the point of cooperation, the O tile binds cooperatively and then grows the deterministic blue signal by a series of B tiles (only the first one is shown here.) The number of tiles in the blue signal is constrained by the size of the assembly space L .

ing patterns are assembled with positive probability at least $\frac{p}{p+q} = \frac{\varepsilon}{1+\varepsilon} > 0$;

2. nontarget (erroneous + nontarget promising) patterns are assemblable with probability at most $1 - \frac{Kp}{p+q} = \frac{(1-K)p+q}{p+q} = \frac{1+(1-K)\varepsilon}{1+\varepsilon}$.
3. With respect to all assemblies, erroneous patterns are assembled with a complementary probability to that of promising patterns and within the interval $[0, \frac{1}{1+\varepsilon}]$

Proof. For a size n , let the probability of assembling a target pattern, the probability of assembling a promising pattern, and the probability of assembling an error be p_n , a'_n , and e_n , respectively. Let $p_n = \rho_n(H_n | G_n)$ be the local conditional probability of assembling a target pattern and \bar{A} denote the complementary $\Omega - A$. By the hypotheses, $\frac{\rho_n(G_n \cap H_n)}{\rho_n(H_n)} \geq p$, so $p_n = \frac{\rho_n(G_n \cap H_n)}{\rho_n(H_n)} \frac{\rho_n(H_n)}{\rho_n(G_n)} \geq p \frac{\rho_n(H_n)}{\rho_n(G_n)}$ and $e_n = \rho_n(\overline{H_n \cup G'_n} | G_n) = \frac{\rho_n(G_n - G'_n)}{\rho_n(G_n)} \leq q \frac{\rho_n(H_n)}{\rho_n(G_n)} \leq q \frac{p_n}{p} \leq \frac{q}{p} G'_n$, and thus $G'_n \geq \frac{p}{p+q} > 0$. Since a target pattern is the extension of a promising pattern of a smaller size by the attachment of a tile, the probability of a target pattern x of size n is the probability of attaching a new tile to a promising pattern a of size $n - 1$ is given by $K \frac{p}{p+q}$, where K is the minimum probability of attachment of a single tile. The probability of assembly of a nontarget pattern is then $e_n + a'_n - p_n = \rho_n(G_n) - \rho_n(H_n | G_n) \leq 1 - \frac{\rho_n(H_n \cap G_n)}{\rho_n(G_n)} \leq 1 - K \frac{p}{p+q} = \frac{1+(1-K)\varepsilon}{1+\varepsilon}$. \square

Theorem 7. *If a set of patterns H is efficiently probabilistically approximable by a pattern assembler \mathcal{G} , then one of the three following alternatives must hold:*

1. *the rate at which erroneous and nontarget promising patterns are assembled stabilizes at a constant value;*
2. *the rate at which erroneous and nontarget promising patterns are assembled switches in value in an alternating fashion;*
3. *or the rate at which erroneous and nontarget promising patterns are assembled is at least exponential.*

This proof requires the following concepts in real analysis in ordinary Euclidean space. A limit point of a given sequence of real numbers $\{a_n\}_n$ is the limit of some converging subsequence $\{a_{n_j}\}$. Thus, a converging sequence has only one limit point (its limit), but nonconvergent sequences such as $\{(-1)^n\}$ can have more than one limit point (± 1 .) The Bolzano-Weierstrass Theorem [52] says that a bounded sequence of points in a real closed interval always has at least one limit point. This nonempty set of limit points for a sequence $\{a_n\}_n$ will be denoted $Lim(\{a_n\})$.

Proof of Theorem 7. Define $p_n = \rho_n(G'_n | G_n)$ and $v_n = \rho_n((G_n - G'_n) | G_n)$ to be the infinite sequences for the corresponding sets of assembly for promising and erroneous patterns, respectively. It follows from Theorem 6 that $0 \leq p_n + v_n < 1 - \frac{Kp}{p+q}$ for all sizes $n \geq 0$. Therefore, these sequences are bounded and $Lim(p_n)$ and $Lim(v_n)$ are nonempty sets, by the Bolzano-Weierstrass Theorem.

If $Lim(p_n)$ and $Lim(v_n)$ consist of single elements, (a) holds. Otherwise, let p and v be the supremum of their respective nonempty limit sets. If $p = v$, then for infinitely many n $p_n \geq v_n$ and for infinitely many n , $p_n \leq v_n$, thus (b) holds. Otherwise $p \neq v$, meaning the smallest upper bound of one sequence is greater than the smallest upper bound of the other, and either the promising patterns or the errors eventually dominates.

Given a promising or error pattern x of size n , let the *context* of an assembled pattern refer to a site where one tile may stably attach by assembly in the given assembler and let the *boundary* be the set of all contexts for an assembly of size n . If x is an error, every extension of x is also an error. Let $\alpha_n(x) = |x^{++}|$ be the number of all such possible attachments in x^{++} . If x is promising, x^{++} contains at least one promising pattern (e.g., x .) Call $\beta_n(x)$ the number of promising attachments. The number of errorful or promising attachments are complementary in the set of all possible attachments. Let α_n and β_n be the minimum of $\alpha_n(x)$ and $\beta_n(x)$ over all patterns x of size n . Let α and β be the infimum, the greatest lower bound, of all $\alpha_n > 0$ and $\beta_n > 0$ over all $n > 0$. Since $p \neq v$, at least one must be positive and, in fact, greater than 1 for infinitely many n . Therefore $p_n \geq c\alpha^n$ or $v_n \geq c\beta^n$ for infinitely many n . □

Conclusions

In this work, we introduced a novel object for the study of the long-term behavior of assembly processes, synoptic patterns, a holistic representation of the process designed to focus on the relationship between the products of assembly overall, rather than the relationships between atoms within single assemblies. In chapter 2, we described the necessary conditions an arbitrary DAG must meet to be recognizable as a synoptic pattern of some aTAM TAS. Next, we showed how a synoptic pattern can be reconstructed from a set of special claw graphs by an operation called extension from a seed. We connected these model-agnostic (e.g. glue-independent) claws to the aTAM by construction and showed that any synoptic pattern associated with an aTAM TAS is also associated with a consistent set of claws in the aTAM and *vice versa*. We then showed how to add on a probability space to form probabilistic synoptic patterns where a probabilistic analysis of the long-term behavior of assemblers can be undertaken. Events in the probabilistic synoptic pattern of an aTAM TAS represent the stable binding of a tile to a preexisting assembly. We show how changing the probability measure in the probabilistic synoptic pattern provides a useful tool for the analysis of self-assembly systems by comparing our object to a TAS in the PTAM and demonstrating the recovery of their analysis for a particular choice of probability measure parameterized by the same length that parameterizes the corresponding TAS. We contrast our construction with the *ad hoc* representation of probabilistic self-assembly as the local probability of stable binding, showing the flexibility of the probabilistic synoptic pattern does not forgo the expressivity of prior work in the area. In chapter 3, we expanded on the applications of a probabilistic synoptic pattern by demonstrating the usefulness of a well founded probabilistic representation of the total products of a self-assembly system by introducing a notion of approximation of string languages by aTAM TAS through the probabilistic synoptic patterns. We show that building from the finite-state automata and Markov processes, we can exhibit a 0-approximation of the linear languages. Finally, in chapter 4 we show how the author's prior work on efficient probabilistic approximation can be expanded with the use of the probabilistic synoptic pattern. Instead

of just counting hits and misses, the probabilistic synoptic pattern has to put a similar mass on the matches while still conforming closely to the contents of those assemblies. We recover the conclusions in our prior work and expand them to more general notions of assemblers and probability measures with support from the underlying definition of probabilistic synoptic pattern. In summary, this work presents and explores a new approach to probabilistic analysis of self-assembly by introducing the synoptic pattern and a careful construction of a probability space to produce the novel concept of probabilistic synoptic pattern. We demonstrate its usefulness in a robust probabilistic analysis of self-assembly, providing new perspectives on known results in the literature and in characterizing a family of errorful but robust assemblers of high purity and efficiency.

Bibliography

- [1] L.M. Adleman. Toward a mathematical theory of self-assembly (extended abstract). Technical report, University of Southern California, 1999.
- [2] Michael A Boles, Michael Engel, and Dmitri V Talapin. Self-assembly of colloidal nanocrystals: From intricate structures to functional materials. *Chemical reviews*, 116(18):11220–11289, 2016.
- [3] Nadrian C Seeman. Nucleic acid junctions and lattices. *Journal of theoretical biology*, 99(2):237–247, 1982.
- [4] Tsu Ju Fu and Nadrian C Seeman. Dna double-crossover molecules. *Biochemistry*, 32(13):3211–3220, 1993.
- [5] L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266 5187:1021–4, 1994.
- [6] Paul WK Rothmund. Folding dna to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.
- [7] Erik Winfree. *Algorithmic self-assembly of DNA*. PhD thesis, California Institute of Technology, 1998.
- [8] Erik Winfree and Renat Bekbolatov. Proofreading tile sets: Error correction for algorithmic self-assembly. In *International Workshop on DNA-Based Computers*, pages 126–144. Springer, 2003.
- [9] David Doty. Randomized self-assembly for exact shapes. *SIAM Journal on Computing*, 39(8):3521–3552, 2010.
- [10] Cameron T Chalk, Bin Fu, Alejandro Huerta, Mario A Maldonado, Eric Martinez, Robert T Schweller, and Tim Wylie. Flipping tiles: concentration independent coin flips in tile self-assembly. In *International Workshop on DNA-Based Computers*, pages 87–103. Springer, 2015.
- [11] Harish Chandran, Nikhil Gopalkrishnan, and John Reif. Tile complexity of linear assemblies. *SIAM Journal on Computing*, 41(4):1051–1073, 2012.
- [12] Leonard Adleman, Qi Cheng, Ashish Goel, and Ming-Deh Huang. Running time and program size for self-assembled squares. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 740–748, 2001.
- [13] David Doty, Jack H Lutz, Matthew J Patitz, Scott M Summers, and Damien Woods. Random number selection in self-assembly. In *International Conference on Unconventional Computation*, pages 143–157. Springer, 2009.
- [14] Ho-Lin Chen and Ming-Yang Kao. Optimizing tile concentrations to minimize errors and time for dna tile self-assembly systems. In *International Workshop on DNA-Based Computers*, pages 13–24. Springer, 2010.

- [15] Neil Dalchau, Harish Chandran, Nikhil Gopalkrishnan, Andrew Phillips, and John Reif. Probabilistic analysis of localized dna hybridization circuits. *ACS synthetic biology*, 4(8):898–913, 2015.
- [16] David A Freedman. On the asymptotic behavior of bayes estimates in the discrete case ii. *The Annals of Mathematical Statistics*, 36(2):454–456, 1965.
- [17] Hao Wang. Proving theorems by pattern recognition—ii. *Bell system technical journal*, 40(1):1–41, 1961.
- [18] Matthew Cook, Yunhui Fu, and Robert Schweller. Temperature 1 self-assembly: deterministic assembly in 3d and probabilistic assembly in 2d. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 570–589. SIAM, 2011.
- [19] Gagan Aggarwal, Qi Cheng, Michael H Goldwasser, Ming-Yang Kao, Pablo Moisset De Espanes, and Robert T Schweller. Complexities for generalized models of self-assembly. *SIAM Journal on Computing*, 34(6):1493–1515, 2005.
- [20] Erik D Demaine, Martin L Demaine, Sándor P Fekete, Mashhood Ishaque, Eynat Rafalin, Robert T Schweller, and Diane L Souvaine. Staged self-assembly: nanomanufacture of arbitrary shapes with $o(1)$ glues. *Natural Computing*, 7(3):347–370, 2008.
- [21] Chris Luhrs. Polyomino-safe dna self-assembly via block replacement. *Natural Computing*, 9(1):97–109, 2010.
- [22] Cameron Chalk, Eric Martinez, Robert Schweller, Luis Vega, Andrew Winslow, and Tim Wylie. Optimal staged self-assembly of general shapes. *Algorithmica*, 80(4):1383–1409, 2018.
- [23] Jacob Hendricks, Matthew J Patitz, Trent A Rogers, and Scott M Summers. The power of duples (in self-assembly): It’s not so hip to be square. *Theoretical Computer Science*, 743:148–166, 2018.
- [24] Bin Fu, Matthew J Patitz, Robert T Schweller, and Robert Sheline. Self-assembly with geometric tiles. In *International Colloquium on Automata, Languages, and Programming*, pages 714–725. Springer, 2012.
- [25] Daniel Hader and Matthew J Patitz. Geometric tiles and powers and limitations of geometric hindrance in self-assembly. *Natural Computing*, pages 1–16, 2021.
- [26] Oscar Gilbert, Jacob Hendricks, Matthew J Patitz, and Trent A Rogers. Computing in continuous space with self-assembling polygonal tiles. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 937–956. SIAM, 2016.
- [27] Paul WK Rothmund and Erik Winfree. The program-size complexity of self-assembled squares. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 459–468, 2000.

- [28] Matthew J Patitz, Robert T Schweller, and Scott M Summers. Exact shapes and turing universality at temperature 1 with a single negative glue. In *International Workshop on DNA-Based Computers*, pages 175–189. Springer, 2011.
- [29] David Soloveichik and Erik Winfree. Complexity of self-assembled shapes. *SIAM Journal on Computing*, 36(6):1544–1569, 2007.
- [30] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media, 2005.
- [31] Robert V Hogg, Elliot A Tanis, and Dale L Zimmerman. *Probability and statistical inference*. Pearson/Prentice Hall Upper Saddle River, NJ, USA:, 2010.
- [32] David Gale, Harold W Kuhn, and Albert W Tucker. Linear programming and the theory of games. *Activity analysis of production and allocation*, 13:317–335, 1951.
- [33] Robert G Bland. New finite pivoting rules for the simplex method. *Mathematics of operations Research*, 2(2):103–107, 1977.
- [34] David Avis and Bohdan Kaluzny. Solving inequalities and proving farkas’s lemma made easy. *The American Mathematical Monthly*, 111(2):152–157, 2004.
- [35] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.
- [36] Dora Giammarresi and Antonio Restivo. Two-dimensional languages. In *Handbook of formal languages*, pages 215–267. Springer, 1997.
- [37] Meenakshi Paramasivan. *Operations on Graphs, Arrays and Automata*. PhD thesis, University of Trier, 2018.
- [38] Leonard Adleman, Q Cheng, Ashish Goel, Ming-Deh Huang, and Hal Wasserman. Linear self-assemblies: Equilibria, entropy and convergence rates. In *Sixth international conference on difference equations and applications*. Taylor and Francis, 2001.
- [39] Ming Liao. *Applied stochastic processes*. CRC Press, 2013.
- [40] Erik Winfree, Furong Liu, Lisa A Wenzler, and Nadrian C Seeman. Design and self-assembly of two-dimensional dna crystals. *Nature*, 394(6693):539–544, 1998.
- [41] Constantine G Evans and Erik Winfree. Physical principles for dna tile self-assembly. *Chemical Society Reviews*, 46(12):3808–3829, 2017.
- [42] Damien Woods, Ho-Lin Chen, Scott Goodfriend, Nadine Dabby, Erik Winfree, and Peng Yin. Active self-assembly of algorithmic shapes and patterns in polylogarithmic time. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 353–354, 2013.

- [43] Terence Tao. *An introduction to measure theory*, volume 126. American Mathematical Society Providence, RI, 2011.
- [44] Seymour Ginsburg and Edwin H. Spanier. Finite-turn pushdown automata. *SIAM Journal on Control*, 4(3):429–453, 1966.
- [45] Tyler G Moore, Max H Garzon, and Russell J Deaton. Probabilistic analysis of pattern formation in monotonic self-assembly. *Plos one*, 10(9):e0137982, 2015.
- [46] Pierre-Étienne Meunier and Damien Woods. The non-cooperative tile assembly model is not intrinsically universal or capable of bounded turing machine simulation. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 328–341, 2017.
- [47] David Doty, Matthew J Patitz, and Scott M Summers. Limitations of self-assembly at temperature 1. *Theoretical Computer Science*, 412(1-2):145–158, 2011.
- [48] David Furcy and Scott M Summers. Optimal self-assembly of finite shapes at temperature 1 in 3d. *Algorithmica*, 80(6):1909–1963, 2018.
- [49] Pierre-Etienne Meunier, Matthew J Patitz, Scott M Summers, Guillaume Theyssier, Andrew Winslow, and Damien Woods. Intrinsic universality in tile self-assembly requires cooperation. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*, pages 752–771. SIAM, 2014.
- [50] Ming-Yang Kao and Robert Schweller. Randomized self-assembly for approximate shapes. In *International Colloquium on Automata, Languages, and Programming*, pages 370–384. Springer, 2008.
- [51] Harish Chandran, Nikhil Gopalkrishnan, and John Reif. Tile complexity of approximate squares. *Algorithmica*, 66(1):1–17, 2013.
- [52] Robert G Bartle and Donald R Sherbert. *Introduction to real analysis*, volume 2. Wiley New York, 2000.