

UC Davis

UC Davis Electronic Theses and Dissertations

Title

Design and Verification of Modular Components in Thermodynamic Binding Networks

Permalink

<https://escholarship.org/uc/item/6xc1w2kc>

Author

Haley, David Russell

Publication Date

2021

Peer reviewed|Thesis/dissertation

Design and Verification of Modular Components in Thermodynamic
Binding Networks

By

DAVID RUSSELL HALEY
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Applied Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

David Doty, Chair

Raissa D'Souza

James Crutchfield

Committee in Charge

2021

Copyright © 2021 by
David Russell Haley
All rights reserved.

This work is dedicated to everyone who has in some way supported me or encouraged me over the course of my career. This group is in total too numerous to name here, but the individuals know who they are, even though most will be modest and try to underplay their particular role.

I especially thank my wife, Kelly Meyer, for her exceptional patience over the better part of a decade during which she had front-row tickets with which to observe the often trying experiences of graduate education.

CONTENTS

Abstract	vi
1 Introduction	2
2 Thermodynamic Binding Networks Model	8
2.1 Model	8
2.1.1 TBN	8
2.1.2 Configuration	8
2.1.3 Path	9
2.1.4 Energy	10
2.1.5 Barrier	12
2.2 Saturated paths	13
2.2.1 Bounds on energy change	13
2.2.2 Saturated paths suffice	14
2.3 Modeling bonds	16
2.3.1 Bond-Aware Model	16
2.3.2 Equivalence	17
3 Engineering Systems with Programmable Energy Barriers	19
3.1 Translator cycle	20
3.1.1 Construction	20
3.1.2 Proof of Barrier	30
3.2 Grid gate	32
3.2.1 Construction	32
3.2.2 Proof of Barrier	33
3.2.3 Catalysis	34
3.2.4 Autocatalysis	39
3.3 Reducing binding site complexity for multistable TBNs	41
3.3.1 Golfergate: Pairwise intersection 1 of state monomers	42

3.3.2	Catalysts	44
3.3.3	Three states	45
3.3.4	Many states	46
3.3.5	Catalyst with Barrier $n/2 - \sqrt{n \ln n}$	48
4	PSPACE-completeness of Reachability	51
4.1	Introduction	51
4.1.1	Related work	52
4.2	Model	54
4.2.1	Chemical reaction networks	54
4.2.2	Constraints on CRNs	55
4.3	PSPACE-completeness of binary, reversible, singular, catalytic CRNs	56
4.3.1	Definition of CRN simulation	56
4.3.2	Construction	58
4.3.3	Proof	61
4.4	Singular ≤ 3 -catalytic CRNs can simulate singular catalytic CRNs	69
4.4.1	Simulation construction	69
4.5	TBNs can simulate (restricted) CRNs	74
4.5.1	Thermodynamic Binding Networks Model	74
4.5.2	Simulation of CRNs	76
4.5.3	Grid gate	78
4.5.4	Modifications to grid gate for simulating CRNs	79
5	Verification Software	89
5.1	Preliminaries	89
5.1.1	Definitions	89
5.1.2	Solvers	91
5.2	Computing stable configurations of TBNs	92
5.2.1	Finding stable configurations of TBNs	92
5.2.2	Casting StableConfigs as an IP	93

5.2.3	Empirical running time measurements	98
5.3	Computing bases of locally stable configurations of TBNs	99
5.3.1	Equivalence of polymer bases and Hilbert bases	99
5.3.2	Using the polymer basis to reason about TBN behavior	102
5.3.3	A case example: Circular Translator Cascade	103
6	Grid Gate Experiment	106
6.1	Designs	106
6.2	Designing Orthogonal Domains	110
6.2.1	Constraints	110
6.3	Results	116
6.3.1	Assay technique	116
6.3.2	Stable formation of Grid Gate	118
6.4	Next Steps	123

ABSTRACT

Design and Verification of Modular Components in Thermodynamic Binding Networks

Designing engineered molecular systems typically requires specialized knowledge of the particular substrate; however, one can also reason about such systems in a *substrate-independent* fashion, by examining the underlying energetics that govern any chemical substrate: the formation of molecular bonds and the number of complexes formed. The thermodynamic binding networks (TBN) model was developed to study such systems, and in particular, to determine fault-tolerance in molecular systems such as DNA strand displacement cascades. This dissertation details an extended form of the model in which complexes can merge together or split apart at an energetic benefit/cost. This extension allows one to also reason about reachability of configurations with respect to energy barriers. Several theoretical constructions are presented here which demonstrate that such energy barriers can be programmably large, implement catalytic and autocatalytic behavior, and be part of larger, modular systems in which complex behavior can be realized. Indeed, reasoning about the energy barrier between configurations in such systems is proved here to be PSPACE-hard, even to a c -factor approximation. This dissertation also contains details of integer and constraint programming formulations that can solve certain questions related to a system's energetics. Also made formal here is the connection between TBNs and the well-studied combinatorial concept of Hilbert bases, and examples are given which illustrate how one can use a Hilbert basis to verify particular aspects of TBN designs. Finally, the details of an experiment attempting to implement one of the programmable TBN constructions are given, along with empirical results and interpretations.

CONTENTS

Chapter 1

Introduction

Recent experimental breakthroughs in DNA nanotechnology [19] have enabled the construction of intricate molecular machinery whose complexity rivals that of other biological macromolecules, even executing general-purpose algorithms [59]. A major challenge in creating synthetic DNA molecules that undergo desired chemical reactions is the occurrence of erroneous “leak” reactions [45], driven by the fact that the products of the leak reactions are more energetically favorable. A promising design principle to mitigate such errors is to build “thermodynamic robustness” into the system, ensuring that leak reactions incur an energetic cost [55, 57] by logically forcing one of two unfavorable events: either many molecular bonds must break—an “enthalpic” cost—or many separate molecular complexes (called *polymers* in this document) must simultaneously come together—an “entropic” cost.

The model of *thermodynamic binding networks* (TBNs) [25] was defined as a combinatorial abstraction of such molecules, deliberately simplifying substrate-dependent details of DNA in order to isolate the foundational energetic contributions of forming bonds and separating polymers. A TBN consists of *monomers* containing specific *binding sites*, where binding site a can bind only to its complement a^* . A key aspect of the TBN model is the lack of geometry: a monomer is an *unordered* collection of binding sites such as $\{a, a, b^*, c\}$. A *configuration* of a TBN describes which monomers are grouped into *polymers*; bonds can only form within a polymer. One can formalize the “correctness” of a TBN by requiring that its desired configuration(s) be *stable*: the configuration maximizes

the number of bonds formed, a.k.a., it is *saturated*, and, among all saturated configurations, it maximizes the number of separate polymers.¹ See Fig. 1.1 for an example. Stable configurations are meant to capture the *minimum free energy structures* of the TBN. Unfortunately, answering basic questions such as “*Is a particular TBN configuration stable?*” turn out to be NP-hard [11].

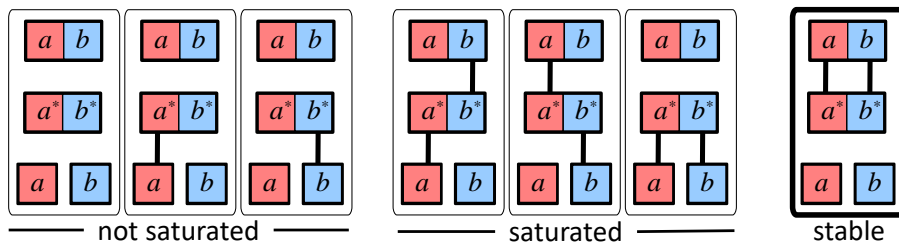


Figure 1.1: Example of a simple thermodynamic binding network (TBN). There are four monomers: $\mathbf{m}_1 = \{a^*, b^*\}$, $\mathbf{m}_2 = \{a, b\}$, $\mathbf{m}_3 = \{a\}$, $\mathbf{m}_4 = \{b\}$, with seven configurations shown: four of these configurations are saturated because they have the maximum of 2 bonds. Of these, three have 2 polymers and one has 3 polymers, making the latter the only stable configuration. Despite the suggestive lines between binding sites, the model of this paper ignores individual bonds, defining a configuration solely by how it partitions the set of monomers into polymers, assuming that a maximum number of bonds will form within each polymer. (Thus other configurations exist besides those shown, which would merge polymers shown without allowing new bonds to form.)

Abstract mathematical models of molecular systems, such as chemical reaction networks, have long been useful in *natural* science to study the properties of natural molecules. For a chemical system designed to perform computation, we can prescribe a chemical program with abstract chemical reactions such as



In particular, a program may require Eq. 1.1 and forbid Eq. 1.2. But what remains hidden at this level of abstraction is a well-known chemical constraint: if Eq. 1.1 is possible, then Eq. 1.2 must also be, no matter the exact substances. Knowing this, we might try to slow Eq. 1.2 by ensuring B has high free energy. But then $B + C$ must also have high free energy, so Eq. 1.1 slows in tandem. The only option to slow Eq. 1.2 but not Eq. 1.1 is to

¹This definition captures the limiting case (often approximated in practice in DNA nanotechnology) corresponding to increasing the strength of bonds, while diluting (increasing volume), such that the ratio of binding to unbinding rate goes to infinity.

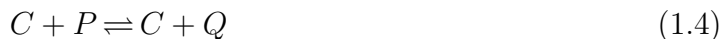
use a *kinetic barrier*: designing A so that, although it is possible for A to reconfigure into B , the system must traverse a higher energy (less favorable) intermediate in the absence of C .

It seems difficult to engineer kinetic energy barriers and catalysis in a way that is independent of the particular chemical substrate. For example, the development of novel protein enzymes requires the precise positioning of hydrophobic or electrostatic interactions, or otherwise chemically active sites, which are often hard to engineer from first principles. Catalysts based on DNA strand displacement reactions arguably promise the highest degree of programmability [62, 60].

Yet, kinetic barriers here usually crucially depend on a specific toehold-sequestering mechanism. Importantly, such DNA-based catalysis suffers from a relatively large rate of the uncatalyzed reaction, and prior work lacks a method to arbitrarily increase the uncatalyzed energy barrier.

We augment the TBN model with a notion of kinetic paths (changes in configuration) due to merging of different complexes and splitting them up (and in Section 2.3 making, breaking, or exchanging bonds). This gives rise to a notion of *paths* of configurations, with different energies. Define the *height* of a path starting at γ as the maximum energy difference $E(\delta) - E(\gamma)$ over all configurations δ on the path. Then the kinetic energy *barrier* separating configuration δ from configuration γ is the height of the minimum-height path from γ to δ .

Moving beyond a single catalytic reaction, we consider larger catalytic networks. Intuitively, this requires generalizing the construction in a number of ways. First, a single catalyst may catalyze two different reactions:



This can be achieved in the original grid gate construction simply by using two grid gates with orthogonal binding sites and merging their respective catalysts. Second, a single

substrate may be converted to two different products via different catalysts:



This requires extending the original grid gate construction to have multiple endpoint states besides “ H ” and “ V ”, with catalysts for each. Our construction (called *golfergate* on account of its relation to a combinatorial puzzle known as the “social golfer” problem) achieves exponentially more endpoint states— $\Theta(n)$ states with n^2 binding site types and barrier $\Theta(n)$ to unwanted reactions. A drawback is that it is not possible to use multiple catalysts (for the same grid) simultaneously.

The constructions discussed up to now realize “bipartite” catalytic networks in the sense that every species is either a catalyst in every reaction it participates in or a catalyst in no reaction.² In general catalytic networks, a species may appear as both a catalyst and a substrate, such as C below:



Such networks are prevalent in protein-protein networks where a single protein can be a target of phosphorylation, as well as the kinase phosphorylating other proteins (with the two actions often allosterically linked).

It was previously demonstrated that the thermodynamic driving force of maximizing the number of separate complexes as captured in the TBN model is sufficient to realize complex computation at equilibrium, including arbitrary logic circuits [25, 17]. We believe the same driving force is also sufficient for inducing complex *kinetic* behavior by programming the entire energy landscape. Catalysis is a complex kinetic behavior (the equilibrium is unchanged by catalysts) and thus illustrates a difficult test case for this hypothesis.

²For example, metabolic networks are bipartite since the enzymes are proteins while the substrates are small molecules.

Given the generality of the TBN model, our results could suggest design strategies for preventing undesired kinetic behavior in a variety of molecular systems, including autocatalysts (which are a special case of non-bipartite catalytic networks), and nucleation barriers.

In Chapter 2 we introduce the TBN model and its associated formalism. We define the notion of paths and reachability, and give a sufficient condition for large kinetic barriers. We show that there is a finite threshold past which the trade-off between enthalpic bond energy and entropic complex formation does not change the behavior of any TBN, eliminating the need to consider such systems in an idealized state of dilution (i.e. in the limit of infinite bond energy).

In Chapter 3 we begin by showing two constructions for catalytic systems: the translator cascade and the grid gate. Both constructions yield families of TBNs parametrized by a complexity parameter n such that the uncatalyzed energy barrier scales linearly with n . The catalyzed energy barrier is always 1. We show that the grid gate system can be modified to create an autocatalytic TBN, with an arbitrarily large energy barrier to undesired triggering, that exponentially amplifies its input signal (Section 3.2.3). We also show how to create a barrier-programmable system with more than two stable states that allows selective catalysis (at some reduction of the barrier and under the constraint that only one catalyst can be present simultaneously).

In Chapter 4, we show that in TBNs, deciding reachability subject to a given barrier is PSPACE -complete, even to a c -factor approximation. We do this through a chain of simulations in which TBNs can simulate a restricted subclass of chemical reaction networks (CRNs), and then that subclass can simulate a much larger subclass of CRNs. We then present a CRN from the larger subclass in which a reachability oracle could be used to solve QSat in polynomial time. By virtue of the simulations, this extends to show that a reachability oracle for TBNs could be used to solve QSat in polynomial time, and therefore that deciding reachability in TBNs is PSPACE -complete.

In Chapter 5, we now embrace the NP -hardness of the natural questions that arise out of the model, and present an efficient formulation that uses integer programming and

constraint programming solvers in tandem to compute a complete set of stable states for any input TBN. We provide open-source software in Python which uses the formulation to solve for the stable states of TBNs. We also show that the problem of characterizing all saturated states of a TBN can be cast as a problem of finding a polymer basis, and that solving for this polymer basis is equivalent to finding a Hilbert basis of a particular convex cone which we describe. While solving for the polymer basis of complex systems can be time-consuming (and our software can do this also), we expect that the connection will provide a suitable bridge for further theoretical advances as Hilbert bases are well-studied in mathematical research; here we provide a way for theoretical advances in combinatorics to inform our design principles in the molecular computing community.

Lastly, Chapter 6 describes the partial results of a chemical experiment that was designed to implement the grid gate in solution. As TBNs do not consider geometry as part of the model, it was necessary to design a geometry that was sound according to our chosen substrate (DNA), but also adhered to the kinetic behavior intended. Furthermore, the TBN model assumes orthogonality of its domains, and this cannot be perfectly realized in DNA, so the choice of DNA sequence required significant computational effort and we describe the algorithm and process used. Lastly, we show data collected and provide some interpretations as well as advice for future directions with similar experiments. Most notably, we advise not to choose to begin an experiment in March of 2020, for the author was ejected from the lab at U.T. Austin where he had been performing the experiment due to public safety measures implemented in an attempt to reduce the potential spread of COVID-19.

Chapter 2

Thermodynamic Binding Networks Model

This chapter was published as Sections 2.1-3.8 and 5.1-5.2 in [10].

2.1 Model

Our work here build upon the model of thermodynamic binding networks (TBN) [25, 12]. Intuitively, we model a chemical system as a collection of molecules, each of which has a collection of binding sites, which can bind if they are complementary. Although the TBN model is more general, DNA domains can be thought of as the prototypical example of binding sites. No geometry is enforced, which allows the model to handle topologically complex structures, such as pseudoknots.

2.1.1 TBN

(See Fig. 2.1.) Formally, a *TBN* is a multiset of monomer types. A *monomer type* is a multiset of site types. A *site type* is a formal symbol, such as a , and has a *complementary* type, denoted a^* . We call an instance of a monomer type a *monomer* and an instance of a site type a *site*.

2.1.2 Configuration

(See Fig. 2.1.) We may describe the configuration of a TBN at any moment in terms of which monomers are grouped into polymers. This way a *polymer* is a multiset of

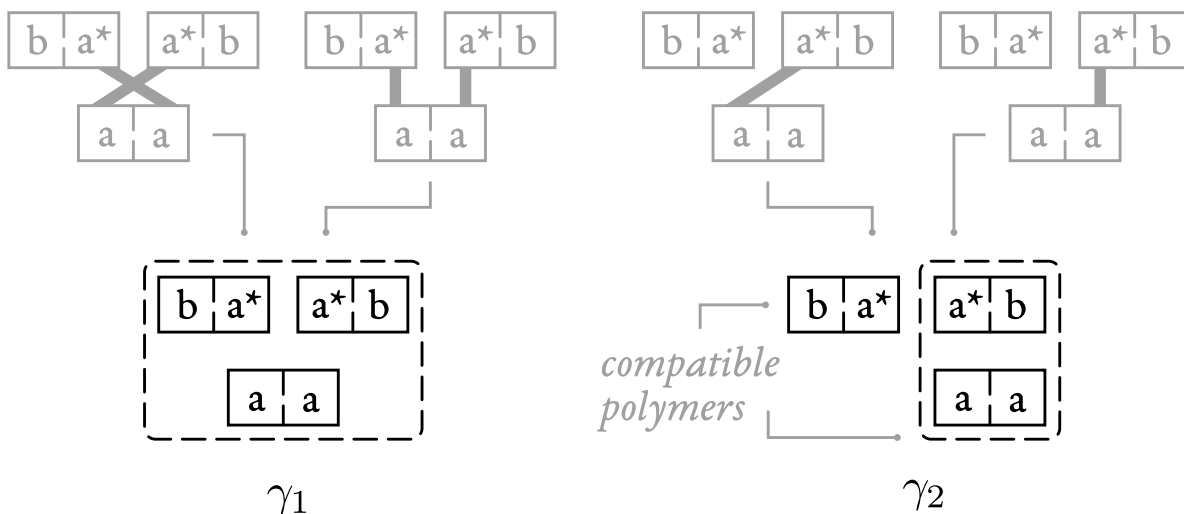


Figure 2.1: Two configurations γ_1 and γ_2 of the TBN $\mathcal{T} = \{\{a, a\}, \{a^*, b\}, \{a^*, b\}\}$. Note that \mathcal{T} has 3 monomers but 2 monomer types and 6 sites but 3 site types. A dashed box indicates monomers that are part of the same polymer. A single configuration (bottom) can correspond to multiple ways of binding complementary sites (top), which are not distinguished in our model. In γ_2 the polymer on the left has exposed sites $\{b, a^*\}$ and the polymer on the right $\{a, b\}$; they are thus compatible since the exposed site a^* of the left is complementary to exposed site a of the right. Since γ_2 has compatible polymers it is not saturated, but γ_1 is.

monomer types, and a *configuration* is a partition of the TBN into polymers.¹

The *exposed sites* of a polymer is the multiset of site types that would remain if one were to remove as many complementary pairs of sites as possible. Each such pair is counted as a *bond*. Note that bonds are not specified as part of a configuration, and intuitively we think of polymers as being maximally bonded. Two polymers are *compatible* if they have some complementary exposed sites. A configuration is *saturated* if no two polymers are compatible. This is equivalent to having the maximum possible number of bonds.

2.1.3 Path

(See Fig. 2.2.) One configuration can change into another by a sequence of elementary steps. If γ can become δ by replacing two polymers in γ with their union, then γ *merges* to δ and δ *splits* to γ , and we write $\gamma \succ^1 \delta$.² We denote by \succeq^1 , \succ , \succeq the reflexive,

¹Consider swapping two monomers of the same type between two polymers in a configuration. We do not consider the result a different configuration. Note that monomers of the same type correspond to an entropy contribution that we ignore (see also footnote 4).

²Note that this would form a lattice of partitions if the configurations were sets instead of multisets.

transitive, and reflexive transitive closures of \succ^1 . A *path* is a nonempty sequence of configurations where each merges or splits to the next. Note that there is a path between any two configurations.³

We could imagine smaller steps that manipulate individual bonds. But surprisingly, such a bond-aware model leads to essentially equivalent kinetic barriers, which we prove in Section 2.3.

2.1.4 Energy

(See Fig. 2.2.) For a configuration γ , denote by $H(\gamma)$ the number of bonds summed over all polymers. Denote by $S(\gamma)$ the number of polymers.⁴ Note that a saturated configuration has maximum $H(\gamma)$. The *energy* of γ is

$$E(\gamma) = -wH(\gamma) - S(\gamma), \quad (2.1)$$

where the bond strength w represents the benefit from gaining a bond relative to gaining a polymer.⁵ Note that $H(\gamma) \geq 0$ and $S(\gamma) > 0$, so $E(\gamma) < 0$, and that lower energy,

³Our model allows incompatible polymers to be merged (i.e., two polymers merge without forming any new bonds). This represents spontaneous co-localization and comes with an energy penalty, as discussed later. For instance, to get from any configuration to any other, we can merge all initial polymers into one and then split into the desired end polymers; however, such a path could be very energetically unfavorable.

⁴The quantities $H(\gamma)$ and $S(\gamma)$ are meant to evoke the thermodynamic quantities of enthalpy and entropy, although the mapping is not exact. Indeed, the free energy contribution of forming additional bonds typically contains substantial enthalpic and entropic parts. Further, while $S(\gamma)$ captures entropy due to independent positions of separate polymers, chemical free energy may consider a variety of other entropic contributions. These may include geometric configurations of a single polymer, as well as the entropy due to swapping indistinguishable monomers. We can justify focusing on $S(\gamma)$ because its contribution arbitrarily predominates in taking the limit of large solution volume—that is, the low concentration regime [25].

⁵Our notion of energy idealizes the physical Gibbs free energy. In typical DNA nanotechnology applications, the Gibbs free energy is likewise a linear combination of $H(\gamma)$ and $S(\gamma)$. We can estimate the Gibbs free energy $\Delta G(\gamma)$ of a configuration γ as follows. Bonds correspond to domains of length l bases, and forming each base pair is favorable by $\Delta G_{\text{bp}}^\circ$. Thus, the contribution of $H(\gamma)$ to $\Delta G(\gamma)$ is $(\Delta G_{\text{bp}}^\circ \cdot l)H(\gamma)$. At 1 M concentration, the free energy penalty due to decreasing the number of separate complexes by one is $\Delta G_{\text{assoc}}^\circ$. At lower concentration c M $<$ 1 M, this penalty increases to $\Delta G_{\text{assoc}}^\circ + RT \ln((1 \text{ M})/c)$. The point of zero free energy is taken to be the configuration with no bonds, and all monomers separate. Thus, the contribution of $S(\gamma)$ to $\Delta G(\gamma)$ is $(\Delta G_{\text{assoc}}^\circ + RT \ln((1 \text{ M})/c))(m - S(\gamma))$, where m is the total number of monomers. To summarize,

$$\Delta G(\gamma) = (\Delta G_{\text{bp}}^\circ \cdot l)H(\gamma) + (\Delta G_{\text{assoc}}^\circ + RT \ln((1 \text{ M})/c))(m - S(\gamma)). \quad (2.2)$$

Note that, as expected, this is a linear combination of $H(\gamma)$ and $S(\gamma)$, and that increasing the length of domains l weighs $H(\gamma)$ more heavily, while decreasing the concentration c weighs $S(\gamma)$ more heav-

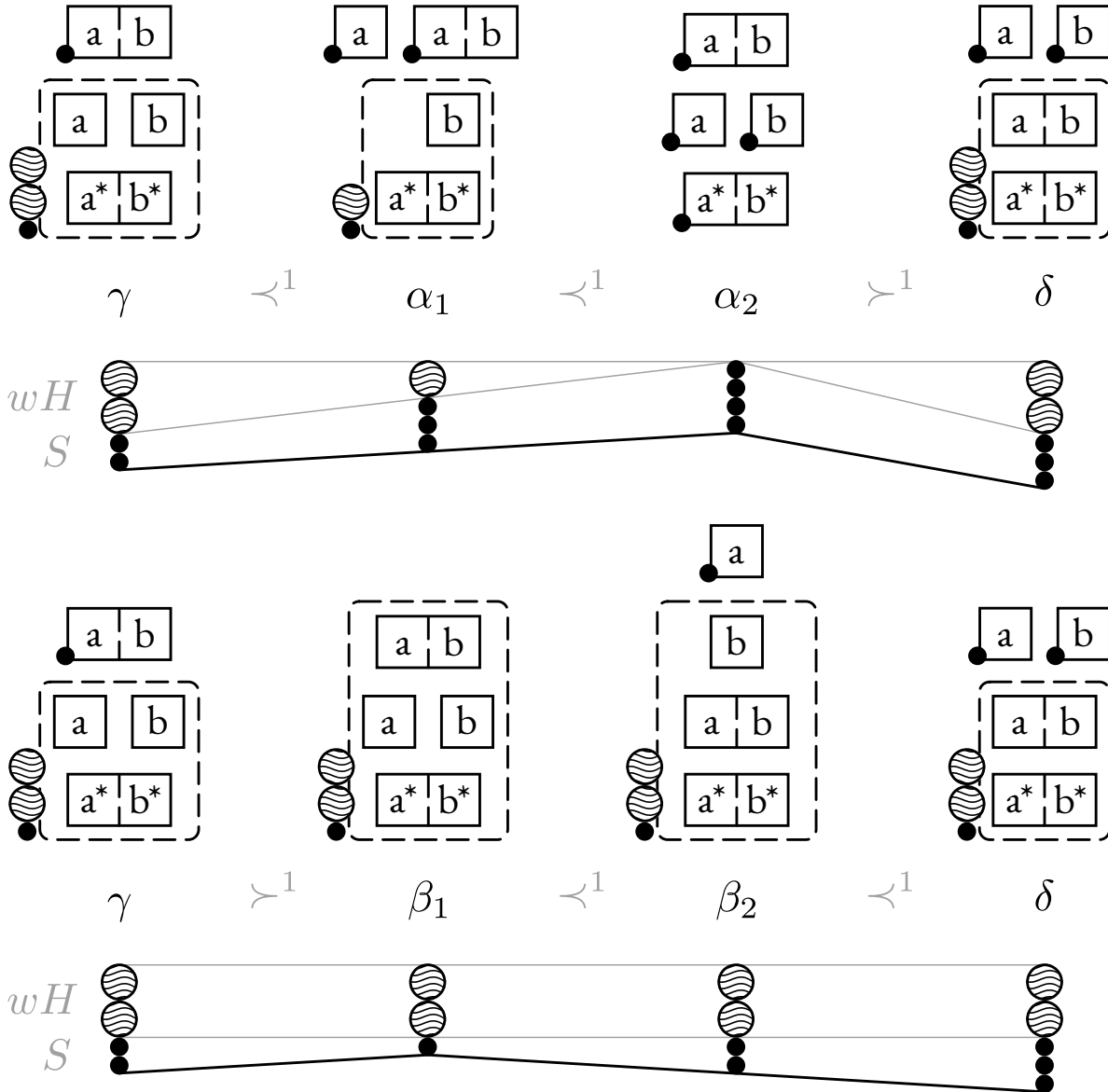


Figure 2.2: A path 1 consisting of γ , α_1 , α_2 , δ and a path 1' consisting of γ , β_1 , β_2 , δ in the TBN $\mathcal{T} = \{\{a\}, \{b\}, \{a, b\}, \{a^*, b^*\}\}$. The energy of each configuration is shown graphically below it. A large wavy disc indicates energy due to a bond. A small solid disc indicates energy due to a polymer. Lower is more favorable. Here bond strength $w = 2$, so a wavy disc is twice as tall as a solid disc. The height of 1 is $h(1) = E(\alpha_2) - E(\gamma) = (-4) - (-2w - 2) = 2$. The height of 1' is $h(1') = E(\beta_1) - E(\gamma) = (-2w - 1) - (-2w - 2) = 1$.

which results from more bonds or more polymers, is more favorable. (The choice to make favorability correspond to lower energy, that is more negative, is motivated by consistency with the standard physical chemistry notion of free energy.) We call a minimum energy

ily. Domains are routinely 15 – 25 bases long, and at 100 nM concentration at room temperature this corresponds to a relative bond strength w of 1.9–3.2.

configuration *stable*. However, since we allow the number of polymers to be potentially infinite, we will sometimes use the equivalent notion that stable configurations are those that can be “constructed” by starting with the configuration whose polymers are all singletons with one monomer, performing the minimum number of merges necessary to reach a saturated configuration. For example, consider the TBN consisting of monomer types $\mathbf{t} = \{a\}$, $\mathbf{b} = \{a^*\}$, with counts $\infty \cdot \mathbf{t}$ and $2 \cdot \mathbf{b}$. The unique stable configuration has polymers $\{2 \cdot \{\mathbf{b}, \mathbf{t}\}, \infty \cdot \mathbf{t}\}$, since two merges of a \mathbf{b} and a \mathbf{t} are necessary and sufficient to create this configuration from the individual monomers.

Merging incompatible polymers forms no additional bonds and so is unfavorable, since $S(\gamma)$ drops without $H(\gamma)$ rising. In contrast, when bond strength $w > 1$, merges between compatible polymers are energetically favorable. So every stable (that is, minimum energy) configuration is saturated. This regime is typical of many real systems, and in particular, we can engineer DNA strand displacement systems [55] to have large bond strength w by increasing the length of domains.

2.1.5 Barrier

(See Fig. 2.2.) There are many paths from a start configuration γ to an end configuration δ . The *height* $h(1)$ of a particular such path 1 is the greatest energy difference $E(\alpha) - E(\gamma)$ between any α along 1 and γ . This measures the difficulty of traversing 1. Notice that $h(1) \geq E(\gamma) - E(\gamma) = 0$.

Another reasonable definition of height is the greatest energy difference $E(\beta) - E(\alpha)$ between any α and later β on the path. We will be considering paths between stable (lowest energy) configurations, where the two definitions are equivalent.

Going from one configuration to another is difficult if each path has large height. The *barrier* $b(\gamma, \delta)$ from γ to δ is the least height of any path from γ to δ . Notice that $b(\gamma, \delta) \geq 0$ as well. Since paths are reversible, it is easy to show that if $E(\gamma) = E(\delta)$ then $b(\gamma, \delta) = b(\delta, \gamma)$.

2.2 Saturated paths

Analyzing TBNs is simpler if we reason only about saturated configurations. Clearly, in the limit of large bond strength w , breaking a bond is so unfavorable that a least height path has only saturated configurations. The main result of this section, Corollary 8, shows that the barrier remains the same even if we consider paths that traverse only saturated configurations as long as $w \geq 2$. Such a threshold may be surprising: it might seem that breaking some bonds, even if locally unfavorable, might allow a path to bypass an otherwise large barrier elsewhere.

We prove Corollary 8 in Section 2.2.2 by showing how to transform an arbitrary path into a saturated path with height no greater. Section 2.2.1 defines a relation among configurations that allows us to keep track of changes in energy as we transform the path.

2.2.1 Bounds on energy change

When two polymers merge, knowing whether they are compatible makes the change in energy predictable. Recall that merging incompatible polymers results in no more bonds, so overall energy increases by 1. Merging compatible polymers results in at least one more bond, so overall energy decreases by at least $w - 1$. To make this precise, let $\gamma \succ^1 \delta$ (and let $\gamma \succ_{\bullet}^1 \delta$) mean that γ merges to δ by combining two incompatible (compatible) polymers. Let \succeq (\succeq_{\bullet}) be the reflexive, transitive closure of \succ^1 (\succ_{\bullet}^1).

Observation 1. *If $\gamma \succ^1 \delta$, then $E(\delta) = E(\gamma) + 1$. If $\gamma \succ_{\bullet}^1 \delta$, then $E(\delta) \leq E(\gamma) + 1 - w$.*

Observation 2. *Let $\Delta = S(\gamma) - S(\delta)$. If $\gamma \succeq \delta$, then $E(\delta) = E(\gamma) + \Delta$. If $\gamma \succeq_{\bullet} \delta$, then $E(\delta) \leq E(\gamma) + \Delta(1 - w)$.*

To apply these bounds to the general case $\gamma \succeq \delta$, we decompose \succeq into \succeq_{\bullet} and \succeq . This allows us to identify an intermediate configuration that has least energy (is most favorable).

Theorem 3. *If $\gamma \succeq \delta$, then some α has $\gamma \succeq_{\bullet} \alpha \succeq \delta$.*

Proof. Let $\gamma \succeq \delta$, and let α be a “most merged” configuration with $\gamma \succeq_{\bullet} \alpha \succeq \delta$ (no $\alpha' \preceq \alpha$ has $\gamma \succeq_{\bullet} \alpha' \succeq \delta$). Then $\alpha = \beta_0 \succ^1 \dots \succ^1 \beta_n = \delta$ for some configurations β_i . Consider the polymers 1 and 1 in β_k merged by $\beta_k \succ^1 \beta_{k+1}$. We claim 1 and 1 are incompatible.

To see so, note that $1 = 1_1 \cup \dots \cup 1_x$ and $1 = 1_1 \cup \dots \cup 1_y$ for some polymers 1_i and 1_j in α . If any 1_i and 1_j are compatible, then merging them in α would produce a configuration that contradicts α being most merged. So they are pairwise incompatible. Letting $[X]$ denote the exposed sites of X , we have $[1] \subseteq [1_1] \cup \dots \cup [1_x]$ and $[1] \subseteq [1_1] \cup \dots \cup [1_y]$. So 1 and 1 are incompatible.

So $\beta_k \succ^1 \beta_{k+1}$ for each k , and $\alpha = \beta_0 \succeq \beta_n = \delta$. □

2.2.2 Saturated paths suffice

A *saturated path* is a path along which every configuration is saturated. For example, the bottom path $1'$ in Fig. 2.2 is saturated. If γ and δ are saturated, then let $b_{\text{sat}}(\gamma, \delta)$ denote the barrier from γ to δ when allowing only saturated paths. Since a saturated path is a path, $b_{\text{sat}}(\gamma, \delta) \geq b(\gamma, \delta)$. It turns out that if bond strength $w \geq 2$, then the reverse inequality also holds, so $b_{\text{sat}}(\gamma, \delta) = b(\gamma, \delta)$. And if $w \geq 1$, then the reverse inequality “almost” holds.

To show the reverse inequality, we turn an arbitrary path into a saturated path without increasing its height (much). We do this step by step by always merging just enough polymers to achieve saturation. To make this precise, let $[\gamma]$ denote the set of saturated γ' with $\gamma \succeq \bullet \gamma'$, and let $E_{\text{max}}(1)$ be the maximum energy of any configuration along the path 1 .

First we show how to saturate a split step.

Lemma 4. *Let $w \geq 1$. If $\gamma \preceq^1 \delta$ and $\gamma' \in [\gamma]$, then some $\delta' \in [\delta]$ and some saturated path $1'$ from γ' to δ' has $E_{\text{max}}(1') \leq E(\gamma)$.*

Proof. Let $\gamma \preceq^1 \delta$ and $\gamma' \in [\gamma]$. Then $\gamma \succeq \gamma'$, so $\delta \succeq \gamma'$. So by Theorem 3, some δ' has $\delta \succeq \bullet \delta' \succeq \gamma'$. By assumption, γ' is saturated, so δ' is, so $\delta' \in [\delta]$. Now let $1'$ be a path guaranteed by $\gamma' \preceq \delta'$. Then $E_{\text{max}}(1') = E(\gamma') \leq E(\gamma)$. □

To show how to saturate a merge step, we rely on being able to transfer a merge from one context to another.

Lemma 5. *If $\gamma \succ^1 \delta$ and $\gamma \succeq \gamma'$, then some α has $\gamma' \succeq^1 \alpha$ and $\delta \succeq \alpha$.*

Proof. Let $\gamma \succ^1 \delta$ and $\gamma \succeq \gamma'$. Let 1 be the polymer merged by $\gamma \succ^1 \delta$, and let γ^* be γ but with all polymers intersecting 1 merged. This way $\gamma^* = \delta$ and $\gamma' \succeq^1 \gamma'^*$.

Now $\gamma = \beta_0 \succ^1 \dots \succ^1 \beta_n = \gamma'$ for some configurations β_i . So $\delta = \gamma^* = \beta_0^* \succeq^1 \dots \succeq^1 \beta_n^* = \gamma'^*$. So choose $\alpha = \gamma'^*$. \square

Now we show how to saturate a merge step.

Lemma 6. *Let $w \geq 1$. If $\gamma \succ^1 \delta$ and $\gamma' \in [\gamma]$, then some $\delta' \in [\delta]$ and some saturated path $1'$ from γ' to δ' has $E_{\max}(1') \leq \max\{E(\gamma), E(\delta)\} + \max\{0, 2 - w\}$.*

Proof. Let $\gamma \succ^1 \delta$ and $\gamma' \in [\gamma]$. If $\gamma' = \gamma$, then let $\delta' = \delta$, and let $1' = \gamma, \delta$. Then $E_{\max}(1') = E(\delta)$.

Otherwise $\gamma' \neq \gamma$. Now by Lemma 5, some α has $\gamma' \succeq^1 \alpha$ and $\delta \succeq \alpha$. So by Theorem 3, some δ' has $\delta \succeq \bullet \delta' \succeq \alpha$. Since γ' is saturated, α is, so δ' is, so $\delta' \in [\delta]$. Now let $1'$ be a path guaranteed by $\gamma' \succeq^1 \alpha \preceq \delta'$. Then $1'$ is saturated. Also, $E_{\max}(1') = E(\alpha)$. And $\gamma' \succeq^1 \alpha$, so $E(\alpha) \leq E(\gamma') + 1$. Since $\gamma' \in [\gamma]$ and $\gamma \neq \gamma'$, we have $\gamma \succ \bullet \gamma'$, so $E(\gamma') \leq E(\gamma) + 1 - w$. So $E_{\max}(1') \leq E(\gamma) + 2 - w$, and the result follows from the identity $x \leq \max\{x, y\}$. \square

To saturate a full path, we saturate each step.

Theorem 7. *Let bond strength $w \geq 1$ and γ and δ be saturated. Then*

$$b_{\text{sat}}(\gamma, \delta) \leq b(\gamma, \delta) + \max\{0, 2 - w\}. \quad (2.3)$$

Proof. Let α_1 and α_n be saturated. Consider a path $1 = \alpha_1, \dots, \alpha_n$. Let $\alpha'_1 = \alpha_1$. Then $\alpha'_1 \in [\alpha_1]$. So by Lemmas 4 and 6, for each i we get $\alpha'_{i+1} \in [\alpha_{i+1}]$ and saturated $1'_i$ from α'_i to α'_{i+1} with $E_{\max}(1'_i) \leq \max\{E(\alpha_i), E(\alpha_{i+1})\} + M_w$ where $M_w = \max\{0, 2 - w\}$. Now α_n is saturated, so $\alpha'_n = \alpha_n$. So let $1'$ be the concatenation of the $1'_i$. Then $1'$ is a saturated path from α_1 to α_n . And we have

$$E_{\max}(1') = \max_i E_{\max}(1'_i) \quad (2.4)$$

$$\leq \max_i \max\{E(\alpha_i), E(\alpha_{i+1})\} + M_w \quad (2.5)$$

$$= \max_i E(\alpha_i) + M_w \quad (2.6)$$

$$= E_{\max}(1) + M_w. \quad (2.7)$$

So $h(1') \leq h(1) + M_w$. So $b_{\text{sat}}(\alpha_1, \alpha_n) \leq b(\alpha_1, \alpha_n) + M_w$. \square

Notice that we need bond strength $w \geq 1$ in Theorem 7. If $w < 1$, then $b_{\text{sat}}(\gamma, \delta)$ can be larger than $b(\gamma, \delta)$ by an arbitrary amount.

Also notice that $\max\{0, 2 - w\}$ is tight. To see this, the reader may check that $b_{\text{sat}}(\gamma, \delta) = b(\gamma, \delta) + \max\{0, 2 - w\}$ for the following example: $\gamma = \{\{1_1, 1_2\}, \{1_3\}\}$ and $\delta = \{\{1_1\}, \{1_2, 1_3\}\}$ where $1_1 = \{a, b\}$, $1_2 = \{a^*\}$, $1_3 = \{a, c\}$.

Now since $b_{\text{sat}}(\gamma, \delta) \geq b(\gamma, \delta)$, we have the following corollary of Theorem 7, which is the main result of this section.

Corollary 8. *Let bond strength $w \geq 2$ and γ and δ be saturated. Then $b_{\text{sat}}(\gamma, \delta) = b(\gamma, \delta)$.*

2.3 Modeling bonds

The model, in Section 2.1, represents bonds implicitly. For example, as Fig. 2.1 shows, a single configuration can correspond to multiple ways of pairing up binding sites. This makes it easier to manipulate and reason about configurations.

But does this simplification of configurations affect the height of kinetic barriers? One might imagine that by having to manipulate individual bonds, one would need to overcome larger energy barriers than in our original model where all possible bonds are “automatically made”. Since bonds do change on an individual basis in a chemical system, this would mean that a barrier that exists in the bond-oblivious model is misleading. However, we show that the bond-aware model is essentially equivalent.

We now define the bond-aware model analogously to the definitions of Section 2.1.

2.3.1 Bond-Aware Model

A *configuration* γ of a TBN is a matching among its complementary sites along with a partition of the components of that matching. A *polymer* of γ is a part of this partition. A *bond* is an edge in the matching. A configuration is *saturated* if the matching is maximal.

For a configuration γ , denote by $H(\gamma)$ the total number of bonds. Let $S(\gamma)$ and $E(\gamma)$ be as before.

A *make* adds a missing bond in a polymer. A *break* removes an existing bond in a polymer. A three-way *swap* is changing one endpoint of a bond to another. A four-way swap is swapping the endpoint of one bond with the endpoint of another. A *path* is a sequence of configurations where each gets to the next by a merge, split, make, break, or swap.

$h(\cdot)$ is as before. Let $b_{bond}(\gamma, \delta)$ be the barrier in the bond-aware model. Let $b_{sat-bond}(\gamma, \delta)$ be over only paths with no break (and so no make).

2.3.2 Equivalence

We can view the original coarse kinetic model in Section 2.1 as a simplification of the more detailed model. To move between the two perspectives, we introduce some notation. For a polymer 1 of the bond-aware model, let $\langle 1 \rangle$ be the collection of monomers in 1 (which is the corresponding polymer in the bond-oblivious model). For a configuration γ , let $\langle \gamma \rangle$ be the collection of $\langle 1 \rangle$ for each polymer 1 in γ (which is the corresponding configuration in the bond-oblivious model).

Lemma 9. $E(\langle \alpha \rangle) \leq E(\alpha)$.

Proof. $S(\langle \alpha \rangle) = S(\alpha)$ and $H(\langle \alpha \rangle) \geq H(\alpha)$. □

The bond-aware model allows as much as the polymer model and more, so intuitively, a barrier in the bond-aware model should be no higher.

Theorem 10. *If $E(\gamma) = E(\langle \gamma \rangle)$, then $b_{bond}(\gamma, \delta) \leq b(\langle \gamma \rangle, \langle \delta \rangle)$.*

Proof. Consider a path $1 = \gamma_1, \dots, \gamma_n$. Let $\langle 1 \rangle = \langle \gamma_1 \rangle, \dots, \langle \gamma_n \rangle$, and let $\langle \gamma_i \rangle$ have highest energy along $\langle 1 \rangle$. By Lemma 9, $E(\langle \gamma_i \rangle) \leq E(\gamma_i)$. So if $E(\langle \gamma_1 \rangle) = E(\gamma_1)$, then $h(\langle 1 \rangle) \leq h(1)$. □

If γ is saturated, then $\langle \gamma \rangle$ is saturated and $E(\gamma) = E(\langle \gamma \rangle)$. So this proof also proves the inequality for saturated paths in the two models.

Lemma 11. *If γ and δ are saturated, then $b_{sat-bond}(\gamma, \delta) \leq b_{sat}(\langle \gamma \rangle, \langle \delta \rangle)$.*

What may be more surprising is that we can also establish a reversed inequality.

Theorem 12. $b_{bond}(\gamma, \delta) + 1 \geq b(\langle \gamma \rangle, \langle \delta \rangle)$.

Proof. Consider a path 1 from $\langle \gamma \rangle$ to $\langle \delta \rangle$. Form a path 1' from γ to δ by adding makes, breaks, and swaps as follows. Before each split, swap enough to break as few bonds as possible. After each merge, make as many bonds as possible. \square

A saturated path simply needs no makes or breaks, so this proof also proves the inequality for saturated paths in the two models.

Corollary 13. $b_{sat-bond}(\gamma, \delta) + 1 \geq b_{sat}(\langle \gamma \rangle, \langle \delta \rangle)$.

Chapter 3

Engineering Systems with Programmable Energy Barriers

Sections 3.1-3.2 of this chapter were published as Sections 4.1-4.2 in [10].

In this chapter we first present two constructions. Each is a family of TBNs indexed by an integer n . We call certain configurations of those TBNs *initial* and *triggered* and show an energy barrier between them. As n increases, the size of the energy barrier increases linearly. Each also has a catalyst, which reduces the energy barrier to 1 when added.

The first construction (translator cycle), discussed in Section 3.1, is based on a DNA strand displacement catalyst. Progress from the initial to triggered configurations with the catalyst can be physically implemented as a strand displacement cascade. Although this system has been previously proposed[55, 58], we rigorously prove an energy barrier for the first time.

The second construction (grid gate), discussed in Section 3.2, does not have an evident physical implementation (e.g., as a strand displacement system), but surpasses the translator cycle system in two ways. First, the grid gate can exhibit *autocatalysis*—that is, it can be modified so that the catalyst transforms the initial polymer into a polymer that has the same exposed binding sites as the catalyst, which can itself catalyze the transformation of additional initial polymers (leading to exponential amplification). Second, the grid gate is *self-stabilizing*, which we define to mean that from any configuration, there is a barrier of zero to reach either an initial or triggered configuration. This intuitively

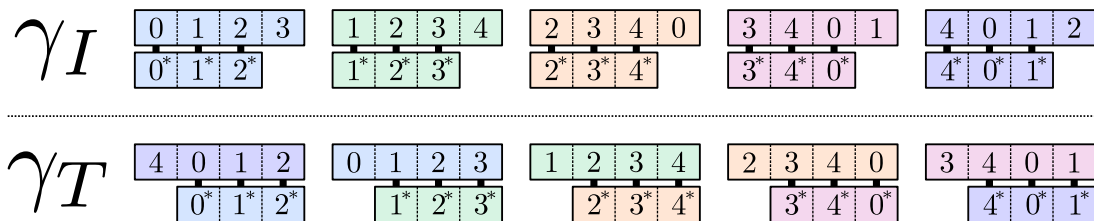


Figure 3.1: The two stable configurations of a translator cycle with complex length $1 = 3$ and number of complex types $1 = 5$. In place of binding site x_i , we write i for clarity.

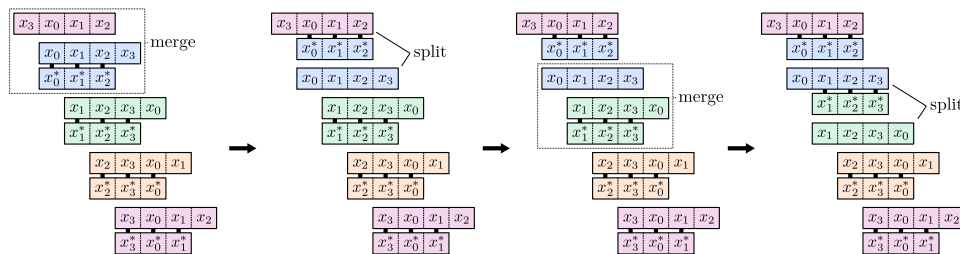


Figure 3.2: A segment of the height 1 path which is possible because an extra copy of a top monomer, $\{x_4, x_0, x_1, x_2\}$, is present to act as a catalyst. The dotted arrow signifies a sequence of merge/split steps. In place of binding site x_i , we write i for clarity.

ensures that the system cannot get stuck in an undesired local energy minimum.

Throughout both sections, we assume $w \geq 2$, so that by Corollary 8, we can determine energy barriers by considering only saturated paths. If we weaken this assumption to $w \geq 1$, then by Theorem 7 the barrier proved is within 1 of the barrier in the unrestricted pathway model (allowing unsaturated configurations). We believe that for these systems an $\Omega(n)$ energy barrier exists even if $w < 1$ but sufficiently large. However, studying the $w < 1$ regime is left for future work.

The constructions demonstrate that catalysts and autocatalysts with arbitrarily high energy barriers can be engineered solely by reference to the general thermodynamic driving forces of binding and formation of separate complexes, which are captured in the TBN model.

3.1 Translator cycle

3.1.1 Construction

Consider the TBN illustrated in Fig. 3.1. There are two particular configurations that interest us, an initial configuration 1 and a triggered configuration 1. The two config-

urations are stable. In the presence of a catalyst monomer $\{x_4, x_0, x_1, x_2\}$ (or an extra copy of any *top monomer*—any of the monomers with unstarred binding sites), a height one pathway exists to reach 1, illustrated by Fig. 3.2. If the catalyst is not present, we prove there is a barrier which can be made arbitrarily large by including more and longer monomer types. Further, this catalytic system is realizable as a DNA strand displacement cascade; more information about this connection can be found in [58]. In the case of many copies of each complex, since the catalyst is in fact any of the top monomers, the system may be used as an amplifier: at the end of the pathway shown in Fig. 3.2, another monomer with binding sites $\{x_4, x_0, x_1, x_2\}$ becomes free which can catalyze another set of complexes which are in the initial configuration.

To program a large energy barrier, we give a formal definition for generalizing the translator cycle, parameterized by *complex length* 1 and *number of complex types* 1. Given $1 \leq z$, a $(1, 1)$ -*translator cycle* is a TBN with monomer types t_i (*top monomers*) and b_i (*bottom monomers*) for $i \in \mathbb{Z}_1$, where

$$t_i = \{x_i, x_{i+1 \pmod{1}}, \dots, x_{i+1 \pmod{1}}\},$$

$$b_i = \{x_i^*, x_{i+1 \pmod{1}}^*, \dots, x_{i+1 \pmod{1}}^*\}.$$

A $(1, 1)$ -translator cycle may have any number of each monomer type as long as (1) for all i , the number of t_i is equal to the number of b_i and (2) there is at least one of each t_i and b_i . To justify constraint (1) note that including an extra top monomer can catalyze the cycle so the barrier disappears, while extra bottom monomers merge the two-monomer complexes to saturate, disrupting the desired initial and triggered configurations. Constraint (2) is required for the catalytic pathway (Fig. 3.2).

The *initial configuration* has each b_i in a polymer $\{b_i, t_i\}$, and a *triggered configuration* 1 is any saturated configuration which contains a subset $\{\{b_i, t_{i-1}\} \mid i \in \mathbb{Z}_c\}$ (at least one set of complexes in the triggered state). The rest of this section is dedicated to proving that the barrier between 1 and 1 depends on the complex length 1 and the number of complex types 1, and can be made arbitrarily large: Formally, we prove that if $1^2 = 1$, then $b(1, 1) > \frac{z}{2+z-1}$.

To motivate our choice of $l^2 = 1$, it is worth describing three available pathways that influence how the upper bound for the uncatalyzed barrier scales with l and 1 . For the first path, we can reach a configuration with a free top monomer, which can subsequently be used as a catalyst as in Fig. 3.2. For example, merging the three polymers with exposed sites x_4, x_0 , and x_1 with the polymer $\{\{x_4, x_0, x_1, x_2\}, \{x_4^*, x_0^*, x_1^*\}\}$ allows the top monomer $\{x_4, x_0, x_1, x_2\}$ to be split. In the general case, this requires merging l polymers and then following a height l path, and the path in total has height l . The second path brings all complexes together while reducing the number of polymers by $l - 1$, and then splits them into the triggered complexes, resulting in a height $l - 1$ path. These paths show that the barrier is not larger than z or $c - 1$. Surprisingly, it is still not sufficient to set $l = l - 1 = n$ to attain a barrier of n ; there is a complicated third path which has height $\frac{2l}{l} - 1$. The third path shows that l must be asymptotically larger than l to achieve a non-constant barrier. Thus we fix $z = n$ and $c = n^2$ for the remainder of the section.

Before we get into details, we give an overview of the proof that $b(1, 1) > \frac{n}{2+n-1}$. First, we show that we can restrict our attention to configurations where polymers must have the same amount of top monomers as bottom monomers (denoted as *normal form*), since other configurations have low polymer count. We think of pairing top and bottom monomers in normal form polymers. Initially, top and bottom monomers with the same indices are paired. In the triggered configuration, the top and bottom monomers are paired with different indices, notably t_i is bound to b_{i+1} ; we say the top index is offset “to the left” of the bottom index by one, or has offset “minus one”. We will formalize this notion of offset, and show that the sum of all offsets between pairs in the configuration, initially zero, does not change with merges and splits in paths of normal form configurations. In the single-copy case, this contradicts any path which reaches a triggered configuration, which must have a negative total offset.

In the multi-copy case, the negative offset of the triggered complexes can be canceled by positive offset elsewhere, resulting in zero total offset, so the argument is not as simple. We will show that polymers providing net positive offset have a size which grows along

with the offset. The large size of these polymers then implies a barrier.

First, we provide an easy way to count how many complexes are in a configuration of a cycle. Note that since there is at least one unstarred binding site for each starred site, the starred sites *must* be bound in a saturated configuration. So we call the starred sites *limiting*. We can use this fact to argue about the number of separate polymers in saturated configurations. Since the starred binding sites are limiting and γ is saturated, each bottom monomer must be bound to at least one top monomer, so we can count $S(\gamma)$ by counting the number of top monomers in separate polymers. This leads to:

Observation 14. *Given a (z, c) -translator cycle with k instances of top monomers, in a saturated configuration γ , if there is a polymer with m top monomers, then $S(\gamma) \leq k - m + 1$.*

Now we restrict the configurations and paths we must consider by describing a *normal form* for polymers.

Definition 15. *Given a configuration of an (n, n^2) -translator cycle, a polymer is normal form if its number of top monomers is equal to its number of bottom monomers. A configuration is normal form if every polymer is normal form. A path is normal form if every configuration is normal form.*

Normal form paths are more restricted than arbitrary paths, and will be easier to reason about. To motivate them, we show that saturated paths from γ_I to γ_T that are not normal form must have a large height via a large polymer in some configuration, and so low height paths (if they existed) would be normal form.

The following lemma is a technical fact used in proofs of Lemmas 17 and 18. It gives properties for polymers in saturated configurations with x bottom monomers with n binding sites each, and y top monomers with $(n + 1)$ sites each, with $x > y$. This will help us restrict to normal form polymers, which have $x = y$.

Lemma 16. *Assume $x, y, n \in \mathbb{N}$, $y(n + 1) \geq xn$, and $x > y$. Then $y \geq n$ and $x \geq n + 1$.*

Proof. If $x > y$, then $x \geq y + 1$, so $y(n + 1) \geq xn \geq (y + 1)n$. So $yn + y \geq yn + n$. So $y \geq n$. Since $x > y$, $x \geq n + 1$. □

The next lemma shows the saturated configurations which are not normal form must have a large (size $\Omega(n)$) polymer.

Lemma 17. *If a saturated configuration is not normal form, then some polymer has at least n top monomers.*

Proof. Since the configuration is not normal form, some polymer 1 has either fewer or more top than bottom monomers. If 1 has fewer, then let $1' = 1$. If 1 has more, then some other polymer $1'$ has fewer top monomers than bottom monomers. Let t (resp., b) be the number of top (resp., bottom) monomers in $1'$. The number of unstarred (resp., starred) binding sites in $1'$ is $t(n+1)$ (resp., bn). Recall that the starred sites are limiting. So for each starred site in $1'$, there is at least one corresponding unstarred site, so $bn \leq t(n+1)$. Since $1'$ has $b > t$, Lemma 16 gives $t \geq n$, so P' has at least n top monomers. \square

So saturated paths with low height must consist of normal form configurations, since otherwise they would have a polymer with many top monomers which implies a large height by Observation 14.

Now we formalize the offset of a pair of compatible monomers. (Recall that two monomers are compatible if they have complementary binding sites.) For $k \in \mathbb{N}$ and $a, b \in \mathbb{Z}_k$, define the sequence

$$[a, b]_k = \langle a, a+1, \dots, b \rangle \pmod{k}. \quad (3.1)$$

For example, $[1, 3]_5 = \langle 1, 2, 3 \rangle$ and $[3, 1]_5 = \langle 3, 4, 0, 1 \rangle$. Also let ℓ_S be the index of element ℓ in sequence S . Then for monomers b_i and t_j , we define the *offset* to be $f(b_i, t_j) = j_S - i_S$, where $S = [i-n, i+n-1]_{n^2}$.

We will define the offset of a normal form polymer in terms of compatible pairs of top and bottom monomers. To choose the pairs, we use the notion of *matchings* from graph theory. Given a normal form polymer P , let T be the set of top monomers and B be the set of bottom monomers. Define a bipartite graph $G_P = \{T, B, E\}$ where $\{b_i, t_j\} \in E$ if and only if b_i and t_j are compatible.

Lemma 18. *If a polymer P is in a normal form saturated configuration and has size $|P| < 2n + 1$, then there exists a perfect matching on G_P .*

A proof of Lemma 18 appears in [10].

For a perfect matching M on G_P , the matching offset is $f(M) = \sum_{m \in M} f(m)$. It will turn out that for small polymers, the offset of every perfect matching is the same. So we will use the matching offset to define the offset of a polymer. To do so, it will be useful to define a notion of “leftmost” and “rightmost” monomers in a polymer. For small polymers (of size less than about n), these are intuitively well-defined since there are not enough monomers to “wrap-around” the entire cycle. We formalize this notion via a *cutoff* value for a polymer:

Definition 19. *Given a normal form polymer P , a cutoff value $c_P \in \mathbb{Z}_{n^2}$ satisfies the following: let C be $[c_P, c_P - 1]_{n^2}$, then there is no edge $\{b_i, t_j\} \in G_P$ such that $i_C \leq n$ and $j_C > n^2 - n$ or $j_C \leq n$ and $i_C > n^2 - n$ (recall that i_C denotes the index of i in C).*

If a cutoff value exists, it means no possible bonds in the polymer—and equivalently no possible edges in any matching—cross the cutoff. Then the leftmost and rightmost pairs are easily defined with respect to the cutoff sequence $[c_P, c_P - 1]_{n^2}$. We prove a sufficient condition for there to exist a cutoff point:

Lemma 20. *For a normal form polymer P with size $|P| < \frac{2n^2}{2n+1}$, there exists a cutoff value c_P .*

Proof. First, we give the reasoning behind the choice of $|P| < \frac{2n^2}{2n+1}$. Intuitively, we want to choose a cutoff point at an index where the top monomer with that index is not compatible for any bottom monomer in the polymer. If there are k bottom monomers in a polymer, the union of the sets of compatible top monomers for those bottom polymers is at most $k(2n+1)$. If we have $k(2n+1) < n^2$, there will exist an index for a top monomer which is not compatible to any bottom in the polymer. This gives $k < \frac{n^2}{2n+1}$, and there are $2k$ monomers in the polymer, so we must have $|P| = 2k < \frac{2n^2}{2n+1}$. Then let c_P be the index of that incompatible top monomer.

We now show that c_P is in fact a cutoff point as in Definition 19. Towards contradiction, let C be $[c_P, c_P - 1]_{n^2}$ and assume there does exist an edge $\{b_i, t_j\} \in G_P$ such that $i_C \leq n$ and $j_C > n^2 - n$ or $j_C \leq n$ and $i_C > n^2 - n$. Then t_j is compatible for b_i , but c_P

is in between i and j , so then t_{c_P} is a compatible top monomer for b_i , which contradicts our choice of c_P as an index of an incompatible top monomer for any bottom monomer in the polymer. \square

To use the cutoff value in future lemmas, we consider configurations and paths with polymers' size restricted to less than $\frac{2n^2}{2n+1}$. We let $n' = \frac{2n^2}{2n+1}$ and define the following:

Definition 21. *A polymer is n' -bounded if its size is less than n' . A configuration is n' -bounded if every polymer is n' -bounded. A path is n' -bounded if every configuration is n' -bounded.*

We can use a cutoff to show that the offset of every matching is the same.

Lemma 22. *For an n' -bounded polymer P in a normal form saturated configuration, for any two perfect matchings M and M' on G_P , $f(M) = f(M')$.*

A proof of Lemma 22 appears in [10].

So we define the *polymer offset* of P , $f(P)$, simply as the offset of any perfect matching on G_P . Given a configuration γ , we can define the *configuration offset* as $\sum_{P \in \gamma} f(P)$. We now show that under certain conditions, merges and splits do not change the configuration offset.

Lemma 23. *In a normal form saturated n' -bounded path, if $\gamma \succ^1 \delta$, then $f(\gamma) = f(\delta)$.*

Proof. Consider the two polymers which merge, P_1 and P_2 , and call the polymer which is their union P . Let M_{P_1} and M_{P_2} be any perfect matchings of G_{P_1} and G_{P_2} . Then $M_P = M_{P_1} \cup M_{P_2}$ is a perfect matching on G_P , and $f(M_P) = f(M_{P_1}) + f(M_{P_2})$. Since the polymers are n' -bounded, we have by Lemma 22 that the polymer offsets equal the offsets of any matching, so $f(P) = f(P_1) + f(P_2)$. Then $f(\gamma) = f(\delta)$ since their only different summands are $f(P)$ and $f(P_1), f(P_2)$. \square

From the above lemma, one can prove that in the case of one copy of each monomer type, the (n, n^2) -translator cycle has a barrier of $\Omega(n)$ to trigger. An informal proof follows: any path which is not normal form or does not have small polymers must have a

large height due to a polymer with many top monomers (see Observation 14). Otherwise, if we restrict paths to saturated normal form n' -bounded paths, Lemma 23 gives that from the initial configuration offset of zero, there is no path which can change the configuration offset to $-n^2$, which is the offset of the triggered configuration. We leave out a formal statement of this proof for brevity, as we prove a more general theorem later—a barrier in the multi-copy case.

The above argument is not sufficient in the multi-copy case because the $-n^2$ offset given by triggered polymers can be canceled out by positive n^2 offset in other polymers. We will argue that having a large positive offset is (roughly) proportional to having a large polymer or several large polymers, so the n^2 positive offset would require merging many complexes. To do so, we define and prove existence of a *sorted* matching on polymers. Intuitively, a sorted matching is a matching which has no crossing edges when the indices are sorted.

Lemma 24. *Given an n' -bounded polymer P with cutoff c_P , there exists a sorted matching M on G_P which satisfies that there does not exist $\{b_{i_1}, t_{j_2}\}, \{b_{i_2}, t_{j_1}\} \in M$ with $i_1 \leq i_2$ and $j_1 \leq j_2$ with respect to the ordering given by the cutoff value sequence, $[c_P, c_P - 1]_{n^2}$.*

A proof of Lemma 24 appears in [10].

Using the sorted matching, we show that the maximum offset (if it is positive) of any one pair in a polymer is proportional to the size of the polymer. First we relate exposed sites to the size of a polymer, then relate the exposed sites to the maximum offset.

Lemma 25. *If a polymer P in a saturated normal form configuration has k exposed sites, it has size $2k$.*

Proof. Assume P is of size $2s$ for some s . Then P has sn starred domains which must be bound in a saturated configuration. P has $s(n+1)$ unstarred domains. So P has exactly s exposed sites for any s . Then to have k exposed sites, it must have size $2k$. \square

Lemma 26. *Given a normal form n' -bounded polymer P , consider the sorted matching M of G_P . Let m be the value of the maximum offset of any pair in M , then $|P| \geq 2(m+1)$.*

Proof. We will show that P has at least $m + 1$ exposed sites, and thus by Lemma 25 is of size at least $2(m + 1)$. Consider the pairs ordered by smallest bottom index to largest with respect to the cutoff c_P given by Lemma 20. Imagine constructing P by adding one pair at a time in order. We will show that when adding a pair, the number of exposed sites cannot decrease due to the order, and when we add the pair with offset m , the constructed polymer has $m + 1$ exposed sites.

First, consider adding a pair p with nonnegative offset $f(p)$ to a polymer with k exposed unstarred sites. Note that the polymer containing only the two monomers in the pair p has $f(p) + 1$ exposed unstarred sites, and $f(p)$ exposed starred sites. By the ordering, we know that the $f(p) + 1$ exposed unstarred sites cannot be bound by any bottom monomers in the polymer constructed thus far. Further, at most $f(p)$ of the k exposed unstarred sites on the polymer prior to adding p can be bound after adding p , since only $f(p)$ exposed starred sites are added. Thus the net change in exposed unstarred sites is plus one. Further, the number of exposed sites on the constructed polymer after adding p is at least $f(p) + 1$.

Next, consider adding a pair p with negative offset to a polymer with k exposed unstarred sites. Note that due to the ordering, the bottom monomer in the pair has no domains in common with the unstarred exposed sites of the polymer constructed thus far. So the number of exposed sites does not decrease.

In both cases, the number of exposed unstarred sites cannot decrease by adding polymers. Consider the point in this construction where we have just added the polymer with positive offset m . The constructed polymer thus far has the $m + 1$ exposed unstarred sites given by the offset m , and the number of exposed sites cannot be reduced by adding the remaining polymers, so the final polymer P must have at least $m + 1$ exposed sites. So by Lemma 25, $|P| \geq 2(m + 1)$. \square

We prove two more lemmas before the proof of the barrier of the translator cycle. The first key lemma is that triggered polymers' negative offset must be canceled out by polymers with positive offset, but since positive offset results in large polymers (or many slightly larger polymers), such a configuration implies a large height for the path which

contains it.

Lemma 27. *Given a normal form saturated n' -bounded configuration γ , if there exists a subset of polymers $\mathcal{P} = \{P_1, \dots, P_k\}$ such that $\sum_{P_i \in \mathcal{P}} f(P_i) \geq n^2$, then $S(\gamma_I) - S(\gamma) > 2n + 1$.*

Proof. Consider γ and for each polymer P_i in γ , fix any sorted matching M_i on G_{P_i} given by Lemma 24 and denote the set of M_i by \mathcal{M} . Then for each $P_i \in \mathcal{P}$,

$$f(P_i) \leq \frac{|P_i|}{2} \max_{p \in M_i} f(p),$$

since there are $\frac{|P_i|}{2}$ pairs each with offset at most the max over the offsets. Since γ is n' -bounded, $|P_i| < n'$, so

$$f(P_i) < \frac{n'}{2} \max_{p \in M_i} f(p).$$

So we have the following:

$$\sum_{M_i \in \mathcal{M}} \frac{n'}{2} \max_{p \in M_i} f(p) = \frac{n'}{2} \sum_{M_i \in \mathcal{M}} \max_{p \in M_i} f(p) \tag{3.2}$$

$$> \sum_{P_i \in \mathcal{P}} f(P_i) \tag{3.3}$$

$$\geq n^2, \tag{3.4}$$

and further

$$\sum_{M_i \in \mathcal{M}} \max_{p \in M_i} f(p) > \frac{2n^2}{n'}. \tag{3.5}$$

Now we show that

$$S(\gamma_I) - S(\gamma) \geq \sum_{M_i \in \mathcal{M}} \max_{p \in M_i} f(p).$$

Consider γ' , the (unsaturated) configuration which is given by taking the polymers in γ and splitting them into pairs of top and bottom monomers based on the matchings M_i . Each bottom monomer is in a polymer with exactly one top monomer, so $S(\gamma_I) = S(\gamma')$. For each P_i with sorted matching M_i in \mathcal{P} , consider the pair p satisfying $\max_{p \in M_i} f(p)$.

We know that in γ , each polymer P_i with $p \in M_i$ must have size at least $2(f(p) + 1)$ by Lemma 26. So P_i must be a polymer containing at least $f(p)$ other pairs. So

$$S(\gamma') - S(\gamma) \geq \sum_{M_i \in \mathcal{M}} \max_{p \in M_i} f(p).$$

Since $S(\gamma_I) = S(\gamma')$, Eq. 3.5 gives us $S(\gamma_I) - S(\gamma) > \frac{2n^2}{n'} = 2n + 1$. \square

Theorem 28. *Given an (n, n^2) -translator cycle, $b(\gamma_I, \gamma_T) \geq \frac{n}{2+n-1}$.*

Proof. We split the possible paths from γ_I to γ_T into three cases. Case 1: if a saturated path is normal form and n' -bounded, we have by Lemma 23 that for any γ_i in the path, $f(\gamma_i) = f(\gamma_I) = 0$. Consider the final configuration on the path, the triggered configuration γ_T . By definition, in γ_T there exists at least one pair of each b_i bound to $t_{i-1 \pmod{n^2}}$. Each of these pairs has offset minus one, contributing minus n^2 to the offset. However, we know that the configuration offset must be zero. So there must exist a subset of polymers $\mathcal{P} = \{P_1, \dots, P_k\}$ such that $\sum_{i=1}^k f(P_i) \geq n^2$, so by Lemma 27, $S(\gamma_I) - S(\gamma_T) > 2n + 1$.

Case 2: if a path is not normal form, then by Lemma 17, there exists a configuration γ with a polymer with n top monomers. Let k be the total number of top monomers in the cycle and note that $S(\gamma_I) = k$. Then by Observation 14, $S(\gamma) \leq k - n + 1$. Then $S(\gamma_I) - S(\gamma) \geq n - 1$.

Case 3: if a path is normal form but is not n' -bounded, then by definition there exists a polymer of size at least n' with an equal number of top and bottom monomers; i.e., a polymer with at least $\frac{n'}{2}$ top monomers. Let k be the total number of top monomers in the cycle and note that $S(\gamma_I) = k$. Then by Observation 14, $S(\gamma) \leq k - \frac{n'}{2} + 1$. Note that $S(\gamma_I) = k$. Then $S(\gamma_I) - S(\gamma) \geq \frac{n'}{2} - 1 = \frac{n^2}{2n+1} - 1$.

By Corollary 8, we restrict analysis to saturated paths. Then $H(\gamma_I) = H(\gamma)$, and so $E(\gamma) - E(\gamma_I) = -(S(\gamma_I) - S(\gamma))$. Among the three cases, the smallest lower bound on the height is $\frac{n^2}{2n+1}$, so the barrier is at least $\frac{n^2}{2n+1} = \frac{n}{2+n-1}$. \square

3.1.2 Proof of Barrier

Lemma 18. *If a polymer P is in a normal form saturated configuration and $|P| < 2n + 1$, then there exists a perfect matching on G_P .*

Proof. Given a set S of vertices, let $N(S)$ be the set of vertices adjacent to a vertex in S . Hall's condition states that a perfect matching exists on a bipartite graph $\{V_1, V_2, E\}$ if and only if for all subsets $S \subseteq V_1$, $|S| \leq |N(S)|$. We prove this holds for G_P .

Consider any subset $S \subseteq B$. There are $n|S|$ starred (limiting) binding sites. The number of sites on compatible top monomers for the set S is given by $(n+1)|N(S)|$. Since P is saturated, the $n|S|$ starred sites must be bound to the $(n+1)|N(S)|$ unstarred sites, so we have $n|S| \leq (n+1)|N(S)|$. If $|S| > |N(S)|$, Lemma 16 gives us that $|S| > n+1$ and $|N(S)| > n$, so to avoid contradicting the assumption that $|P| < 2n+1$, it must be that $|S| \leq |N(S)|$. \square

Lemma 22. *For an n' -sized polymer P in a normal form saturated configuration, for any two perfect matchings M and M' on G_P , $f(M) = f(M')$.*

Proof. Intuitively, first we will shift the indices of the top and bottom monomers so that the leftmost index has value zero. Let c_P be a cutoff value given by Lemma 20. We rewrite each b_i or t_i as $b_{i-c_P \bmod n^2}$ or $t_{i-c_P \bmod n^2}$. Since originally no edge crossed the cutoff, now no edge crosses zero. Note that this does not change the offset of any pair and thus does not change the offset of any matching. For each $f(b_i, t_j)$, since no edge $\{b_i, t_j\}$ crosses zero we can think of the indices on a line, so we can rewrite the offset as $f(b_i, t_j) = j - i$. Then $f(M) = \sum_k (j_k - i_k) = \sum_k j_k - \sum_k i_k$. This expression is independent of the matching and only depends on the indices of the monomers in the polymer, so for any two matchings M and M' , $f(M) = f(M')$. \square

Lemma 24. *Given an n' -sized polymer P with cutoff c_P , there exists a sorted matching M on G_P which satisfies that there does not exist $\{b_{i_1}, t_{j_2}\}, \{b_{i_2}, t_{j_1}\} \in M$ with $i_1 \leq i_2$ and $j_1 \leq j_2$ with respect to the ordering given by the cutoff value, $[c_P, c_P - 1]_{n^2}$.*

Proof. Given any matching M' which is not sorted, we show that we can swap the offending edges, resulting in a new matching which is in sorted order. Let S_{i_k} be $[i_k - n, i_k + n - 1]_{n^2}$, the sequence giving the indices of compatible top monomers for b_{i_k} . For any $\{b_{i_1}, t_{j_2}\}, \{b_{i_2}, t_{j_1}\} \in M$ with $i_1 \leq i_2$ and $j_1 \leq j_2$, note that $j_2 \in S_{i_1}$ and $j_1 \in S_{i_2}$. The orderings given by the sequences S_{i_1}, S_{i_2} are the same orderings as given by $[c_P, c_P - 1]_{n^2}$,

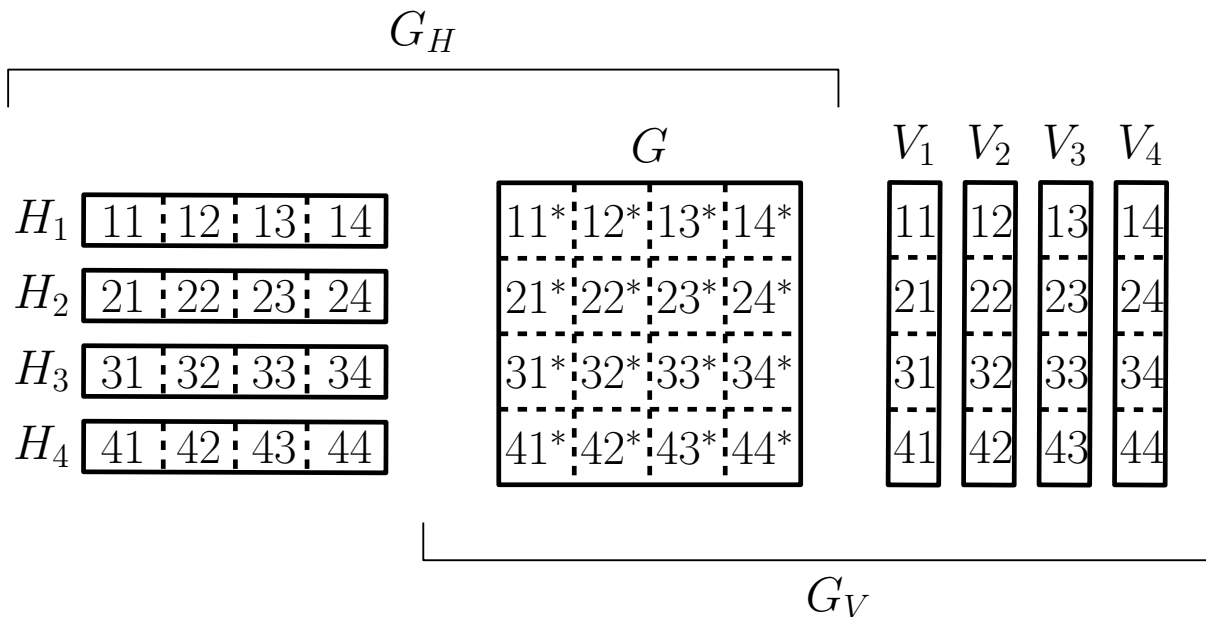


Figure 3.3: The monomer types in the grid gate TBN for the case $n = 4$. In the figure, any two digit number ij represents domain x_{ij} , e.g. x_{23}^* is represented as 23^* .

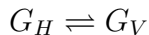
since S_{i_1}, S_{i_2} are both subsequences of $[c_P, c_P - 1]_{n^2}$. Since $i_1 \leq i_2$, S_{i_1} contains no elements greater than any of those in S_{i_2} , and S_{i_2} contains no elements less than any of those in S_{i_1} . So $j_2 \in S_{i_1}$ and $j_1 \in S_{i_2}$ with $j_1 \leq j_2$ gives that both $j_1, j_2 \in S_{i_1}$ and further $j_1, j_2 \in S_{i_2}$. So both t_{j_1} and t_{j_2} are compatible for both b_{i_1} and b_{i_2} . Since the edges of G_P are between bottom monomers and their compatible tops, we can swap $\{b_{i_1}, t_{j_2}\}, \{b_{i_2}, t_{j_1}\}$ with $\{b_{i_1}, t_{j_1}\}, \{b_{i_2}, t_{j_2}\}$ and the result is a matching on G_P . So given any perfect matching on G_P , we can construct a sorted matching by swapping the offending edges one-by-one. \square

3.2 Grid gate

3.2.1 Construction

Consider the TBN illustrated in Fig. 3.3. We focus on two polymer types G_H and G_V depicted in the figure, and show that there is a barrier $n \in \mathbb{N}$ to convert G_H to G_V and vice versa. We parameterize the construction by n as follows. Define the following monomer types: “horizontal” $H_i = \{x_{ij}\}_{j=1}^n$ for $i \in \{1, \dots, n\}$, “vertical” $V_j = \{x_{ij}\}_{i=1}^n$ for $j \in \{1, \dots, n\}$, and “gate” $G = \{x_{ij}^*\}_{i,j=1}^n$. In the notation of chemical reaction networks,

the net reaction



can occur in the presence of sufficiently many free H_i 's and V_j 's, but an energy barrier of n must be surmounted in order for this conversion to happen. In Section 3.2.3 we will show how this energy barrier can be reduced to 1 in the presence of a *catalyst* monomer, corresponding to the chemical notion of a catalyst reducing the activation energy required for a reaction to occur.

Note that throughout this section, the configurations considered are saturated, so that for two configurations γ and δ , we have $H(\gamma) = H(\delta)$, and so $E(\delta) - E(\gamma) = -(S(\delta) - S(\gamma))$. That is, the energy difference between two configurations is the opposite of the difference in their polymer counts.

We fix a network \mathcal{T}_{gg} that contains any number of any of these monomer types, so long as there are enough of other monomers to completely bind all the G monomers (i.e., in saturated configurations there are no exposed starred sites). We define *base* configurations of the network to be those configurations that contain polymers of type G_H or G_V , with all other monomers in separate polymers by themselves. In Theorem 35 we show that these base configurations are stable (take $m = 0$).

The following lemma establishes a necessary condition in any saturated configuration: that any G must be in a polymer with either all of the horizontal monomers or all of the vertical monomers.

Lemma 29. *In a saturated configuration of \mathcal{T}_{gg} , a polymer containing G also contains $\{H_i\}_{i=1}^n$ or $\{V_j\}_{j=1}^n$ as a subset.*

Proof. Suppose a polymer contains G but neither H_i nor V_j for some i and j . Then site x_{ij}^* on G is exposed, and so by definition of \mathcal{T}_{gg} , the configuration is not saturated. \square

3.2.2 Proof of Barrier

The following theorem then establishes that any saturated configuration in \mathcal{T}_{gg} is *self-stabilizing*, that is, it can reach a stable (base) configuration via a path with barrier 0 (i.e. using all splits).

Theorem 30. *For any saturated γ of \mathcal{T}_{gg} , some base π has $\gamma \preceq \pi$.*

Proof. Consider a saturated configuration γ . Suppose γ has a non-base polymer 1. If 1 contains no G , then we can split into polymers of type H, V . Otherwise, 1 contains G , and by Lemma 29 we can split off a G_V or G_H polymer. The theorem holds by repeating this process. \square

Note that since the base configurations are stable (this follows from Theorem 35 with $m = 0$), and by Theorem 30 any other saturated configuration can reach a base configuration using only splits, the base configurations are also the only stable configurations in this network.

We now prove the desired energy barrier between different base configurations.

Theorem 31. *The barrier between different base configurations of \mathcal{T}_{gg} is n .*

Proof. Consider a saturated path 1 from a base configuration γ to another, δ . Notice that $\delta \not\preceq \gamma$. So $\delta \not\preceq \gamma$. So some first β along 1 has $\beta \not\preceq \gamma$. But by Theorem 30, some other base $\pi \neq \gamma$ does have $\beta \preceq \pi$.

Now take α just before β along 1. Then $\alpha \preceq \gamma$ by definition of β . Since α and β are adjacent on 1, either $\alpha \prec^1 \beta$ or $\alpha \succ^1 \beta$. The latter contradicts $\beta \not\preceq \gamma$. So $\alpha \prec^1 \beta$, implying $\alpha \preceq \pi$.

Let $f(\gamma)$ count the H monomers with a G in γ . Since γ and π are different bases, *wlog*, $f(\pi) \geq n + f(\gamma)$. Consider a path of k merges corresponding to $\gamma \succeq \alpha$. It can increase $f(\cdot)$ by at most k . So $f(\alpha) \leq f(\gamma) + k$. A path of splits does not increase $f(\cdot)$, so $\alpha \preceq \pi$ implies $f(\alpha) \geq f(\pi)$. So we get

$$k \geq f(\alpha) - f(\gamma) \tag{3.6}$$

$$\geq f(\pi) - f(\gamma) \geq n. \tag{3.7}$$

So $S(\gamma) - S(\alpha) = k \geq n$. \square

3.2.3 Catalysis

The kinetic barrier shown for the grid gate can be disrupted by the presence of new monomer types. In fact, the model admits a *catalyst* monomer C that lowers the energy

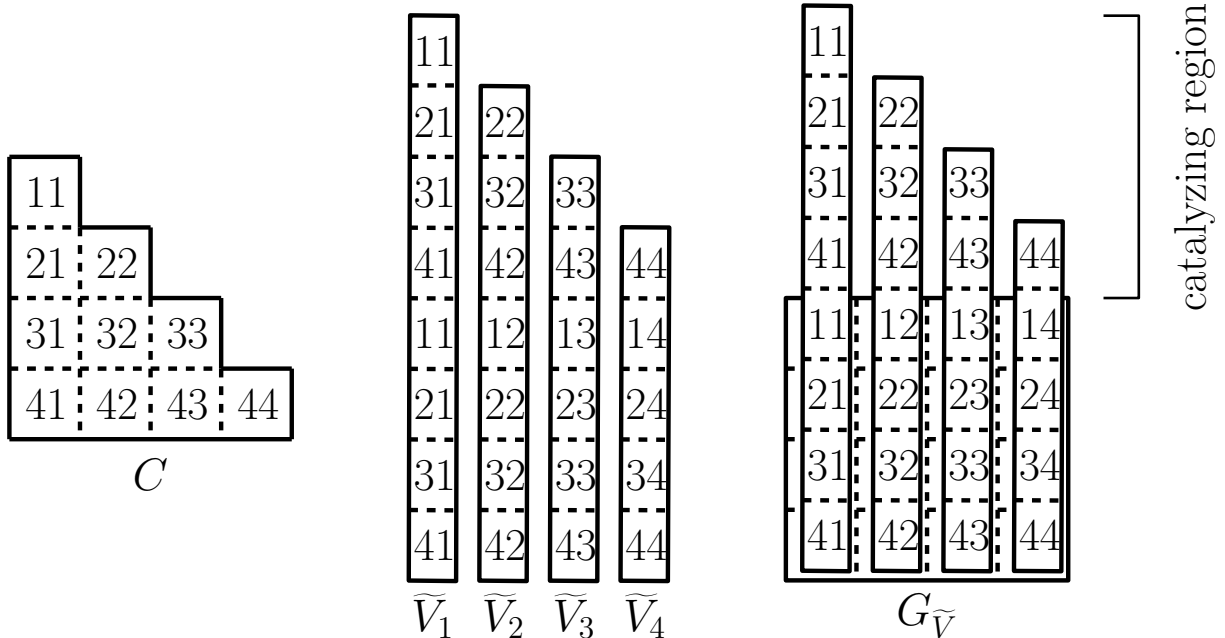
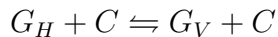


Figure 3.4: Catalysts and autocatalysts in the grid gate TBN for the case $n = 4$. left: C is a single monomer that acts as a catalyst to convert between G_H and G_V . middle: Modified vertical monomers $\{\tilde{V}_j\}_{j=1}^n$ with extra sites. right: After C converts G_H to $G_{\tilde{V}}$ with modified vertical monomers, $G_{\tilde{V}}$ has the same excess sites as C and acts as a catalyst itself (i.e. is “active” as a catalyst).

barrier from n to 1, i.e., in the presence of one or more C , a G_H can be converted into a G_V , and vice versa, with a sequence of merge-split pairs. In the notation of chemical reaction networks, this binding network implements the net reaction



with energy barrier 1, while maintaining a large energy barrier for the reaction $G_H \rightleftharpoons G_V$.

For the grid gate of size $n \times n$, we define a catalyst: $C = \{x_{ij} \mid 1 \leq j \leq i \leq n\}$ illustrated in Fig. 3.4 (left). C is a monomer consisting of the “lower triangle” of the unstarred sites. The mechanism by which C can transform G_H to G_V with merge-split pairs is by an alternating processes of merges and splits shown in Fig. 3.5. Intuitively, in each step of the catalyzed reaction $G_H + C \rightarrow G_V + C$, G switches its association with H_i (left) to its counterpart on V_j (right) by merging the evolving polymer (center) with V_j and then splitting off H_i .

Consider a network $\{G, C\} \cup \{H_i\}_{i=1}^n \cup \{V_j\}_{j=1}^n$ which includes one instance of every monomer type that has been introduced, as well as the catalyst. As before, we shall be

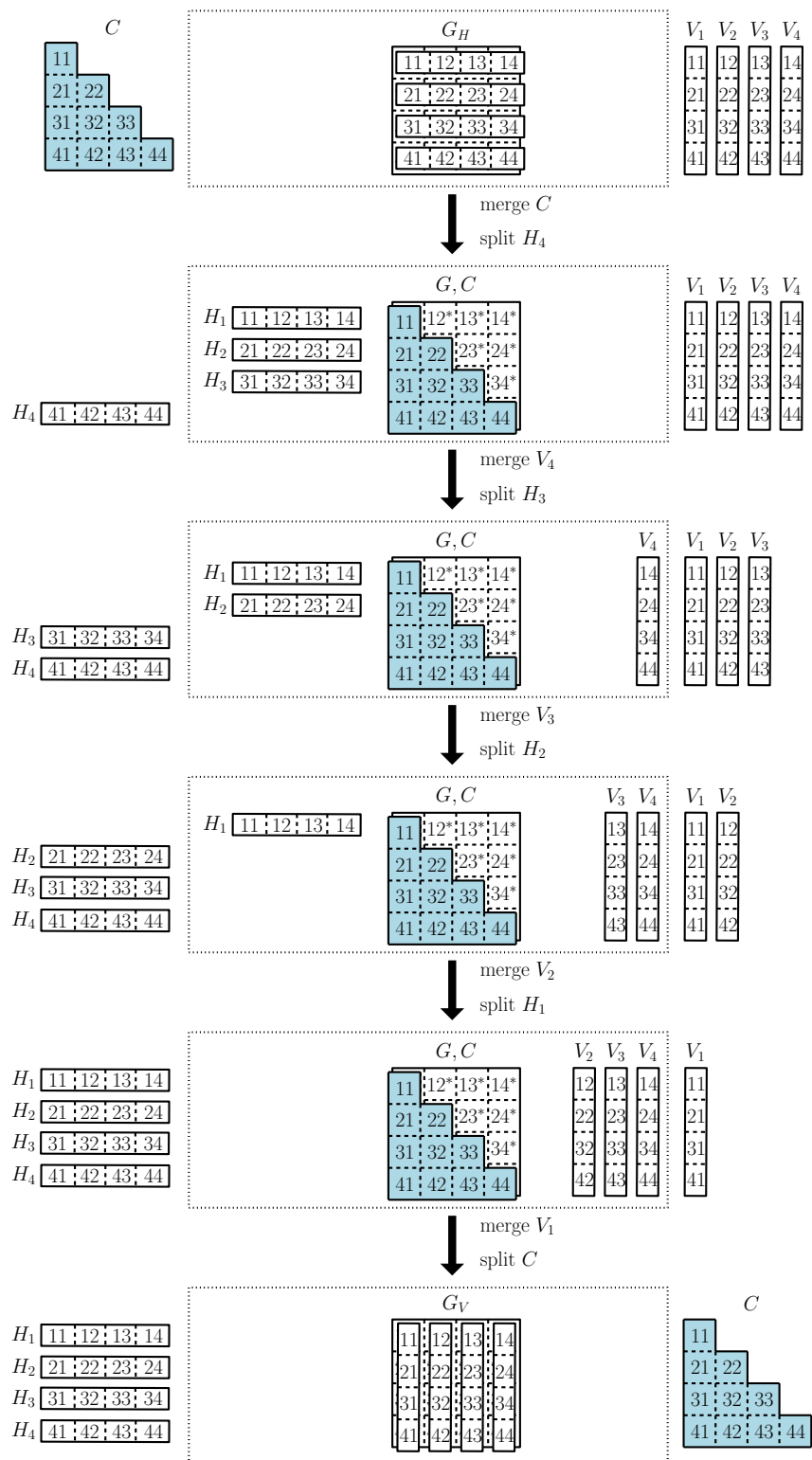


Figure 3.5: Full pathway for reaction $G_H + C \rightarrow G_V + C$. In each stage, exactly one merge and one split occurs, and the center polymer remains saturated.

interested in net transitions between G_H and G_V , and so for this network we define the following configurations: $\gamma_H^C = \{G_H, C\} \cup \{V_j\}_{j=1}^n$ and $\gamma_V^C = \{G_V, C\} \cup \{H_i\}_{i=1}^n$.

Theorem 32 states that transitions in the single-copy case, having arbitrarily large energy barriers according to Theorem 31, in the presence of C have their barrier reduced to one.

Theorem 32. $b(\gamma_H^C, \gamma_V^C) = b(\gamma_V^C, \gamma_H^C) = 1$.

Proof. Consider the following saturated merge-split pathway that begins with configuration γ_H^C and ends with γ_V^C (illustrated in Figure 3.5).

- Merge G_H with C .
- Split H_n from the resulting polymer P_1 .
- For $1 \leq i \leq n - 1$, iteratively merge V_{i+1} with P_i , then split H_i from P_i to form P_{i+1} .
- Merge V_1 with P_n and split off C to form G_V .

This path maintains saturation while never decreasing the polymer count by more than one, and so by Corollary 8 we have that $b(\gamma_H^C, \gamma_V^C) \leq 1$. As it is not possible to reach γ_V^C from γ_H^C in a saturated merge-split path that uses only splits, it will not be possible to have a zero barrier, giving $b(\gamma_H^C, \gamma_V^C) = 1$.

This merge-split path can be executed in the reverse fashion to show that $b(\gamma_V^C, \gamma_H^C) = 1$. □

To generalize the result to the multi-copy setting, we first observe that the height 1 pathway guaranteed in Theorem 32 still exists. What remains is to show that the base configurations, plus zero or more separate catalyst monomers, are stable.

In the arguments that follow, it will be useful to define the set $D_\uparrow = \{x_{i,i+1}\}_{i=1}^{n-1} \cup \{x_{n,1}\}$, which consists of the domains from the “shifted diagonal”. For $m \in \mathbb{N}$, and let $\mathcal{T}_{gg}^m = \mathcal{T}_{gg} \cup \{m \cdot C\}$ denote the TBN \mathcal{T}_{gg} with m additional catalyst monomers.

The next lemma states that each G in a polymer in a saturated configuration must be accompanied by n additional monomers, thus giving a lower bound on the size of any polymer as a function of the number of G 's contained within it.

Lemma 33. *If there are k G 's in a polymer P in a saturated configuration δ of \mathcal{T}_{gg}^m , then $|P| \geq k(n + 1)$.*

Proof. Consider the domains from D_{\uparrow} . As the starred versions of these domains are present on each G , to maintain saturation in δ , it must be the case that each G in δ is joined in P with a set of monomers that include these domains; however, no monomer in $\{C, H_1, \dots, H_n, V_1, \dots, V_n\}$ has more than one of these domains. Thus, to be saturated, if there exist k instances of G in P , there must be at least kn additional monomers in P to bind the above chosen domains. \square

The following lemma uses Lemma 33 to show that no saturated configuration has more polymers than are contained in the base configurations, even when some catalyst monomers are present.

Lemma 34. *If there are k G 's in \mathcal{T}_{gg}^m , then any saturated configuration δ has $S(\delta) \leq |\mathcal{T}_{gg}^m| - kn$.*

Proof. Consider the polymers P_1, \dots, P_j in δ containing all k copies of G , where P_i has k_i copies. Then by Lemma 33, $\sum |P_i| \geq \sum k_i(n + 1) = k(n + 1)$. So $S(\delta) \leq j + |\mathcal{T}_{gg}^m| - k(n + 1) \leq k + |\mathcal{T}_{gg}^m| - k(n + 1) = |\mathcal{T}_{gg}^m| - kn$. \square

The next Theorem establishes that the base configurations, which were stable in the original network \mathcal{T}_{gg} , are still stable even if any number of catalysts should also be present in the network.

Theorem 35. *Let $m \in \mathbb{N}$ and let γ be a base configuration of \mathcal{T}_{gg} . Then $\gamma_m = \gamma \cup (m \cdot \{C\})$ is stable.*

Proof. The base configurations satisfy Lemma 34 with equality, thus they are stable. \square

These results show that the catalyzed network is copy tolerant; that is, it behaves in the expected way even should the amounts of the constituent monomers (and catalyst) vary.

3.2.4 Autocatalysis

The grid gate can also be modified to act in an *autocatalytic* manner. By modifying the vertical monomers it is possible for G_V to have a set of exposed monomers acting as a “catalyzing region”, which has the same structure and function as C (see Figure Fig. 3.4, middle and right).

To obtain an autocatalytic system, we modify the vertical monomers of the network to include additional sites that, when combined with G , form a catalyzing region that can act in the same manner as the catalyst C . See Figure Fig. 3.4 for an illustration.

Formally, we define the modified vertical monomers (see Fig.) as:

$$\tilde{V}_j = \{x_{ij}\}_{i=1}^n \cup \{x_{ij}\}_{i=j}^n$$

We define $G_{\tilde{V}} = \{G\} \cup \{\tilde{V}_j\}_{j=1}^n$ to be the version of G_V that uses the modified monomers. This polymer is the so-called auto-catalyst.

We now consider the network $\tilde{\mathcal{T}}_{gg} = \{2 \cdot G\} \cup \{H_i\}_{i=1}^n \cup \{2 \cdot \tilde{V}_j\}_{j=1}^n$ which includes enough monomers to create the autocatalyst as well as retain enough monomers to analyze transitions between G_H and $G_{\tilde{V}}$. The two configurations that we will be most interested in are:

$$\tilde{\gamma}_H = \{G_H, G_{\tilde{V}}\} \cup \{\tilde{V}_j\}_{j=1}^n$$

$$\tilde{\gamma}_V = \{2 \cdot G_{\tilde{V}}\} \cup \{H_i\}_{i=1}^n$$

The following theorem establishes that the autocatalyzed configurations $\tilde{\gamma}_H$ and $\tilde{\gamma}_V$ are stable, with a low energy barrier between them in the presence of the autocatalyst.

Theorem 36. $\tilde{\gamma}_H$ and $\tilde{\gamma}_V$ are stable.

Proof. Note that there are $3n+2$ total monomers in $\tilde{\mathcal{T}}_{gg}$, and that $S(\tilde{\gamma}_H) = S(\tilde{\gamma}_V) = n+2$. To show that these are stable, it suffices to show any other saturated configuration δ obeys $S(\delta) \leq n+2$. Consider the set of domains $D_{\uparrow} = \{x_{i,i+1}\}_{i=1}^{n-1} \cup \{x_{n,1}\}$, as in Lemma 33.

Each monomer has at most one of each *type* of these domains, and except for $x_{n,1}$, has exactly one instance of each. The exception is \tilde{V}_1 , which has two instances of $x_{n,1}$.

Let δ be any saturated configuration of $\tilde{\mathcal{T}}_{gg}$. We consider two cases: 1) that the G 's are in separate polymers, and 2) that the G 's are in the same polymer. In the case that the G 's are in separate polymers, as each G contains the starred versions of the n domain types in D_{\uparrow} , each polymer P with a single G must have n additional monomers to bind each of the starred versions of these domains. Note that in this case a single \tilde{V}_1 cannot bind both instances of $x_{n,1}^*$, for this would result in both G 's being on the same polymer. This leaves at most $(3n + 2) - 2(n + 1) = n$ remaining monomers. Thus $S(\delta) \leq n + 2$.

Now consider the case that the G are present in the same polymer. Let P be a polymer in δ that contains two G 's. Then in P , the two instances of site $x_{n,1}^*$ can be bound to one instance of \tilde{V}_1 . (Note that if there were two \tilde{V}_1 in this polymer, then one copy could be split into its own polymer while still maintaining saturation, increasing S and putting us in the first case). The remaining sites in D_{\uparrow} of these G 's must be bound to monomers containing $2 \cdot \{x_{i,i+1}\}_{i=1}^{n-1}$. No single monomer contains more than one site from this set, so this requirement must be satisfied by the inclusion of $2(n - 1)$ additional non- G monomers in P . Then there are at least $2n + 1$ monomers in P , leaving at most $(3n + 2) - (2n + 1) = n + 1$ additional monomers. Thus $S(\delta) \leq 1 + (n + 1) = n + 2$.

Since δ was arbitrary, this shows $\tilde{\gamma}_H$ and $\tilde{\gamma}_V$ have maximal S , thus are stable. \square

In particular, the desired feature of this network is that $G_{\tilde{\gamma}}$ acts as an autocatalyst.

Theorem 37. $b(\tilde{\gamma}_H, \tilde{\gamma}_V) = 1$.

Proof. Figure Fig. 3.4 shows that the exposed sites of $G_{\tilde{\gamma}}$ are exactly the sites of C . The proof then follows as in Theorem 32. \square

Like the catalyzed network, the results for the autocatalyzed network can be extended to the multi-copy case. The proof follows in straightforward fashion by an inductive argument.

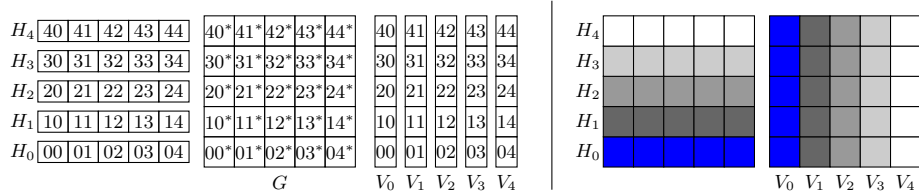


Figure 3.6: The grid gate shown as sets of binding sites (left) and in a more compact notation (right) where monomers with unstarred domains are shown as the same-color points in a single grid in 2D. The template monomer G and the unbound monomers are implicit. The geometric depiction does not imply any underlying geometric arrangement, but is merely for ease of illustration and to aid in analysis.

3.3 Reducing binding site complexity for multistable TBNs

In this section we consider the problem of creating as many states as possible, with programmable barriers to transitioning between states. The main result of this section, Theorem 38, shows that $n+1$ states can be constructed using monomers of size n (and one of size n^2), with catalysts that lower the energy barrier to 1 between specific pairs of states while maintaining an energy barrier of $n/4$ between all other pairs of states. However, multiple catalysts of different types cannot be present simultaneously while maintaining the energy barriers between unwanted pairs of states. It is an open problem to do so with reduced binding site complexity.

In this section we omit a *copy-tolerant* analysis (that is, an analysis of the case in which the system is a union of many duplicates of the base design) concentrating on the case where there is exactly one of each monomer type. However, we are confident that techniques from our previous work [9] can be used to prove copy-tolerance for the construction in this section as well.

Rather than generalizing the grid gate to higher dimensions, in this section we stick to 2 dimensions but generalize how we partition the points into state monomers. Fig. 3.6 shows a notational shorthand in representing the TBNs in this section. Since all binding sites are elements of the geometric space $[n]^2$ for some $n \in \mathbb{N}^+$, we represent them as sets of points in 2D Euclidean space, without always labeling each binding site. It is assumed there is always one undepicted *template* monomer G with a single starred version of every

point in $[n]^2$, so only monomers with unstarred domains are depicted.

Formally we define a state to be a partition of the binding sites $[n]^2$ into n monomers M_1, \dots, M_n , each with n binding sites. A polymer represents a particular state if it contains one of each monomer in that partition joined together with the template monomer G . In this notation, the original grid gate has two states: “horizontal” $H = \{H_0, \dots, H_{n-1}\}$ and “vertical” $V = \{V_0, \dots, V_{n-1}\}$, where $H_i = \{(x, i) \mid x \in [n]\}$ and $V_j = \{(j, y) \mid y \in [n]\}$.

In this section we consider a single copy of template monomer G , and thus we can identify states with configurations of the TBN. As before, we require these configurations to be stable. If a catalyst is present, there will be additional stable configurations (intuitively transitioning between states). In a configuration corresponding to a state, the catalyst must be unbound.

3.3.1 Golfergate: Pairwise intersection 1 of state monomers

Recall the grid gate from Fig. 3.3. Without the catalyst, there is a barrier of n for any monomers to displace an H_i from G_H because, for each $H_i \in H$ and each $M \in V \cup H \setminus \{H_i\}$, $|H_i \cap M| \leq 1$. That is, the pairwise intersection of H_i with any other monomer in the network is at most one. Thus n monomers must merge to G_H before any H_i can be split. Note that this is stronger than merely requiring a barrier of n to transition all the way from state H to V . It shows that the construction is “self-stabilizing”: any configuration reached via paths of height $< n$ can return to the start just by splitting.

Motivated by this observation, we require that any additional state P also has the property that each of its state monomers has pairwise intersection at most 1 with each other state monomer in the network; then a barrier of n will remain to remove any monomer of a state from the polymer. We consider the problem of finding as many states as possible, where each state is a set of n disjoint monomers, each of n binding sites from the set $[n]^2$, i.e., each state is a partition of $[n]^2$ into n sets of size n each, with pairwise intersection 1 between monomers of different states.

Let $n \geq 2$ and let $P \sqsubset [n]^2$ denote that $P = \{M_0, \dots, M_{n-1}\}$ is a partition of $[n]^2$ into n sets $M_0, \dots, M_{n-1} \subset [n]^2$ of cardinality n each. Let $p(n)$ be the largest number of such

partitions of $[n]^2$ such that all pairs of sets M, M' in the various partitions have pairwise intersection ≤ 1 ; formally,

$$p(n) = \max\{k \in \mathbb{N} \mid \exists P_1, \dots, P_k \sqsubset [n]^2 \text{ such that} \\ \forall i \neq j \quad (M \in P_i \wedge M' \in P_j) \implies |M \cap M'| \leq 1\}.$$

Note that this definition would be equivalent if we had written “ $= 1$ ” instead of “ ≤ 1 ”; for if an intersection were 0, by pigeonhole another would be ≥ 2 .

This characterization of partitions with “pairwise intersection 1” is equivalent to the *Mutually Orthogonal Latin Squares* problem in combinatorics, and is a special case of the *Social Golfers* problem [21, Ch. 22],[1], motivating the *golfergate* name of the construction of Theorem 38. The following are well-known and easy to show for all $n \geq 2$:

1. $p(n) \geq 3$. The “diagonal” D of Section 3.3.3 works as a third partition.
2. $p(n) \leq n + 1$. Any partition $P \neq V$ has some set $M \in P$ containing $(0, 0)$. The point $(1, i) \in V_1 \cap M$ is unique to P : for any partition $P' \neq P$, defining $M' \in P'$ to contain $(0, 0)$, we must have $(1, i) \notin M'$, or else $|M \cap M'| \geq 2$. Since there are n choices for i , there are $\leq n$ partitions other than V .

The following are also known, though not as straightforward as the above facts:

1. If n is a prime power, then $p(n) = n + 1$ [21]. Fig. 3.7 shows an example.
2. For some n not a prime power, $p(n) < n + 1$; e.g., $p(6) = 3$ [51].
3. If $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$ where p_i is prime, then $p(n) \geq \min_i \{p_i^{e_i} + 1\}$ [21].

Thus, an infinite (and relatively “frequent”) number of n realize the upper bound of $n + 1$, producing a number of states that scale linearly with n . A Golfergate network containing all of the monomers of Fig. 3.7 simultaneously has six stable configurations and retains an energy barrier of n to transition between them (assuming no other monomers in the system, such as catalysts). An intriguing property of such dense networks is that they admit catalysts, which we describe in the next subsection, albeit catalysts that cannot be present simultaneously.

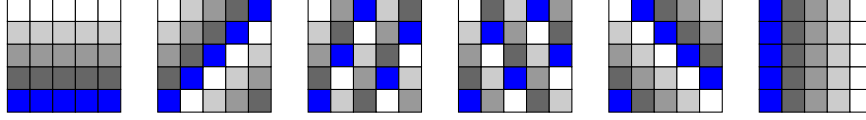


Figure 3.7: Example of $n = 5$ golfergate having $n + 1 = 6$ states with intersection ≤ 1 between any pair of monomers. The construction is explained in the proof of Theorem 38, where these groups of monomers are called P_0 through P_5 . Note that each monomer is (in the field of size 5) a “line of slope s ” for $s \in \{0, 1, 2, 3, 4, \infty\}$.

3.3.2 Catalysts

If we have states $P_1, \dots, P_k \sqsubset [n]^2$, then for each $1 \leq q \neq r \leq k$, a catalyst C_{qr} is a monomer that reduces the barrier between P_q and P_r to 1, while maintaining a barrier of $\Omega(n)$ between any other pair of states.

A barrier-1 path between states is a sequence of merge-split pairs. Suppose state q has monomers $\{Q_0, \dots, Q_{n-1}\}$, and state r has monomers $\{R_0, \dots, R_{n-1}\}$. The catalytic pathway merges C_{qr} with the polymer representing a state q , then for each $i \in [n]$, splits Q_{n-i-1} , followed by merging R_i , and finally splits C_{qr} .

We now show that a large energy barrier (n) is impossible in the golfer gate for any construction of catalysts. Instead the golfer gate barrier is at most $\lfloor \frac{n+1}{2} \rfloor$ if another catalyst is present.

We first argue that $|C_{qr}| = \frac{n(n+1)}{2}$, and then show that this implies the barrier upper bound. Observe that $Q_{n-1} \subseteq C_{qr}$ for Q_{n-1} to split immediately after C_{qr} merges. Further, $Q_{n-2} \not\subseteq C_{qr}$ since otherwise Q_{n-2} would immediately split without an intermediate merge, and the configurations corresponding to states q and r would not be stable. Since Q_{n-2} splits upon merging of R_0 , we have $Q_{n-2} \subseteq C_{qr} \uplus R_0$.

Continuing by induction shows that to achieve such a catalytic pathway, for each $i \in [n]$, we must have $|C_{qr} \cap Q_i| = |C_{qr} \cap R_i| = i + 1$. This condition implies that $|C_{qr}| = \sum_{i=0}^{n-1} (i + 1) = \frac{n(n+1)}{2}$.

Let state $t \notin \{q, r\}$ be another state with monomers T_0, \dots, T_{n-1} . By the symmetry of barriers between stable configurations of a network, it suffices to show that there remains a large energy barrier to displace any T_i , i.e., to convert t to another state. Intuitively, we must ensure $|C_{qr} \cap T_i|$ is small for all $i \in [n]$. The smallest such intersection achievable

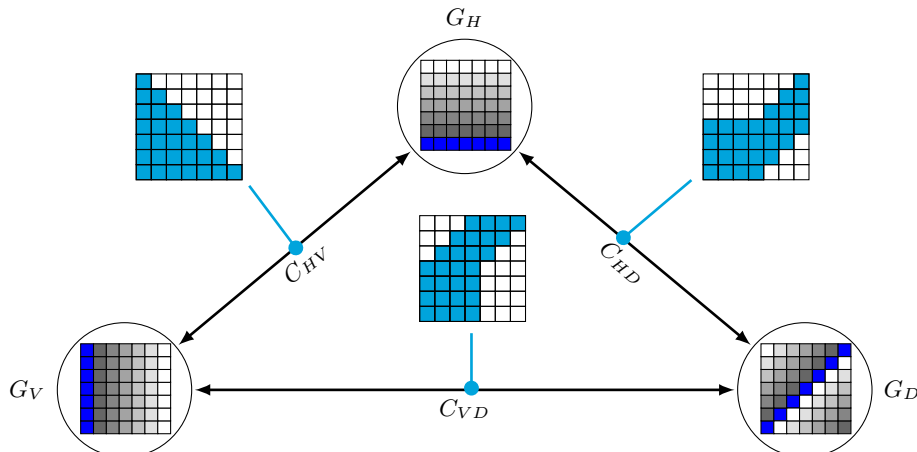


Figure 3.8: Example golfergate with three states. The catalyst binding sites are shown in light blue. A black arrow indicates a reaction pathway which is catalyzed by the catalyst monomer connected to the arrow with a blue line.

for *all* i , is to have $|C_{qr} \cap T_i| = \lceil \frac{n+1}{2} \rceil$ for at least one i since $|C_{qr}| = \frac{n(n+1)}{2}$. This results in an energy barrier of $\lceil \frac{n+1}{2} \rceil$. This minimum is achieved with the three-state TBN of Section 3.3.3.

3.3.3 Three states

As a first example of a golfergate with more than two states we can consider a new state $D = \{D_0, \dots, D_{n-1}\}$ consisting of the “diagonal” lines on the grid. Fig. 3.8 illustrates the states and the respective catalysts between them. Although Section 3.3.4 shows how to achieve more than three states, this is provably impossible for some values of n (e.g., $n = 6$ [51]). The three-state TBN described here works for any value of $n \geq 2$, and retains a barrier $n/2$ to uncatalyzed state transitions.

This example helps to illustrate many of the key concepts used in the more complex construction of Section 3.3.4: each pair of monomers has intersection ≤ 1 , each catalyst C intended to catalyze a set of monomers has the property that the monomers can be put in an order M_0, \dots, M_{n-1} so that $|C \cap M_i| = i + 1$, and for any monomer M *not* part of a state catalyzed by C , $|C \cap M|$ is “significantly” smaller than n ($n/2$ in this case).

We omit a detailed analysis of this system, as it is simple to verify these properties by inspection. The next section shows how to extend to more than three states.

3.3.4 Many states

This section describes our golfergate construction for an arbitrarily large number of states. Let n be any prime power; our construction achieves up to $n + 1$ states.

Intuitively, the construction works as follows. We take one of the states to be G_V (as in Fig. 3.3); we then construct the other n states by envisioning monomers as functions $f_{ib}(x)$ over the field \mathbb{F}_n with the form $i \cdot x + b$. Fig. 3.7 shows an example of a complete set of states for the case that $n = 5$. In each state, the blue monomer has $b = 0$ and passes through the origin in the lower left; note that the slopes of the different partitions are the integers from 0 to $n - 1$. All other monomers in the same partition are vertical shifts of that monomer ($b \neq 0$), wrapping points from top to bottom if they go above the top line. Since the integers mod a prime n are a field with addition and multiplication being defined modulo n , this degree-1 polynomial obeys the property that its intersection with any other degree-1 polynomial is at most 1.

Note that it is not the case in general that the construction gives such visual intuition (e.g. for non-prime n), and in the theorem below we prove that the construction holds in the more general case that n is a prime power.

Theorem 38. *Let $n \in \mathbb{N}$ be a prime power. Then there are $n + 1$ states $P_0, \dots, P_n \sqsubset [n]^2$ such that the following holds. For any $q \neq r$, there is a barrier of n to convert between P_q and P_r .*

Proof. Because n is a prime power, there is a finite field \mathbb{F}_n of size n . For the remainder of this proof, when the operations of addition and multiplication are indicated, they refer to the canonical field operations in \mathbb{F}_n .

Let the partition $P_n = V$ as in Fig. 3.6 (right). We will define the other partitions P_0, \dots, P_{n-1} as the families of graphs of first-order polynomials in \mathbb{F}_n that have the same coefficient on their first-order term; formally $P_i = \{p_{ib}(x)\}_{b \in [n-1]} = \{i \cdot x + b\}_{b \in [n-1]}$ for each $i \in [n - 1]$.

These define partitions: For any i and any two $b \neq c$, $p_{ib}(x) - p_{ic}(x) = b - c \neq 0$ (any two functions in the same partition are a nonzero additive constant apart), so p_{ib} does not

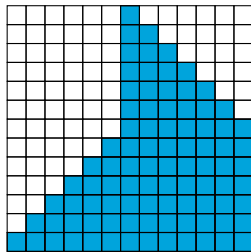


Figure 3.9: Catalyst C_{hv} of Theorem 38, converting between states H and V with energy barrier 1, while retaining energy barrier $n/4$ between all other pairs of states, for $n = 13$.

intersect p_{ic} . Thus the monomers in P_i are pairwise disjoint. Since there are n monomers with n binding site types each, their union is $[n]^2$.

They obey the pairwise intersection ≤ 1 property: Let $i, j, b, c \in [n]$ with $i \neq j$. Then $p_{ib}(x) - p_{jc}(x) = (i - j) \cdot x + (b - c)$ is a linear polynomial with coefficient $(i - j) \neq 0$. By the Fundamental Theorem of Algebra (which holds in \mathbb{F}_n), this polynomial has at most one root. Thus p_{ib} intersects p_{jc} on at most (in fact *exactly*) one point. As observed in Section 3.3.1, this pairwise-one intersection property gives the desired barrier of n . \square

We now describe the construction for a catalyst. The particular construction we give is a catalyst C_{hv} converting between H and V , and is depicted in Fig. 3.9. It is designed to maintain an energy barrier of $n/4$ to displace any other monomer (those with positive finite slope). We will provide justification for why this catalyst works for prime n by appealing to its continuous equivalent.¹

Intuitively, a catalyzed pathway exists because the catalyst intersects H monomers and V monomers in a decreasing order, starting with intersection n and progressing down to intersection 1, enabling a pathway functionally similar to that in Fig. 3.4.² It maintains energy barrier of $n/4$ to displace any other monomer (those with positive finite slope) by the following arguments. Monomers of slopes 1 and $n - 1$ (also is slope -1 in the field of size n) are special cases that intersect C_{hv} on $3n/4$ points in the worst case. Monomers of even slope can be shown to have at least $1/4$ of their points lying in the upper-left

¹It is easier to give intuition in the case that n is prime, for if n is prime, the field operations in \mathbb{F}_n are the normal integer addition and multiplication modulo n , and this paradigm admits certain geometric intuitions as they relate to slopes of lines. The construction presented in this section would need to be modified for grids of non-prime dimension.

²For H , the order is H_0, H_1, \dots, H_{n-1} . For V , the order is $V_{\lfloor n/2 \rfloor}, V_{\lfloor n/2 \rfloor + 1}, \dots, V_{n-1}, V_{\lfloor n/2 \rfloor - 1}, V_{\lfloor n/2 \rfloor - 2}, \dots, V_0$.

“quadrant”, which is entirely outside of C_{hv} . Monomers of odd slope have $\Theta(n)$ points in this quadrant, but must also appeal to the other vacant regions of the graph in order to obtain $n/4$ points that do not intersect the catalyst. Then to displace any such monomer requires an additional $\approx n/4$ monomers, since each monomer can only displace one binding site.

While the construction of C_{hv} seems to be somewhat specific to the states H and V , this construction can be permuted in such a way as to catalyze between any two pairs of states. Specifically, for each q, r such that $\{q, r\} \neq \{h, v\}$, a linear transformation T_{qr} can be applied that permutes binding sites such that q becomes h and r becomes v . Because the transformation and its inverse T_{qr}^{-1} are linear operations performed on linear functions, they preserve the high barriers between other pairs of states. The desired catalyst is thus obtained by defining $C_{qr} = T_{qr}^{-1}(C_{hv})$.

The energy barrier achieved with this catalyst, though scaling linearly with n , is not exactly n . In fact this is a necessary consequence of our setup that reuses the same binding sites in $[n]^2$ to have more than two states. Even for the three-state system described in Section 3.3.3, any catalyst C_{hv} must have at least $\frac{n(n+1)}{2}$ binding sites. This is true by the Generalized Pigeonhole Principle, for any other state with monomers M_0, \dots, M_{n-1} , there is an $i \in [n]$ such that $|C_{hv} \cap M_i| \neq (n+1)/2$, lowering the barrier to displacing M_i to at most $n/2$.

We conjecture that catalysts exist that retain a (maximal) barrier of $n/2$ to pathways between uncatalyzed states. Numerical experiments seem to confirm a probabilistic argument that with high probability such catalysts exist in cases for prime values of n .

3.3.5 Catalyst with Barrier $n/2 - \sqrt{n \ln n}$

We first describe a random process that creates a random catalyst C_σ that converts between H and V . Let $\sigma : [n] \rightarrow [n]$ be a uniform random permutation. Then we can define the catalyst

$$C_\sigma = \{(x, y) \in [n]^2 : y \leq \sigma(x)\}.$$

If σ is the identity permutation, this gives the original triangle-shaped C_{HV} catalyst. Thus what we have here is a random permutation of the columns. By symmetry, the

same proof holds that C_σ is a catalyst between the states H and V .

We now consider the barrier to pathways between uncatalyzed states. We will show that some random choice of C_σ has a barrier of $n/2 - \sqrt{n \ln n}$. This comes from an upper bound on the overlap $C_\sigma \cap M$ for any other monomer M , which will give by the union bound an upper bound on the overlap with any other monomer. Thus for large enough values of n , we can have barrier cn for c arbitrarily close to the optimal value of $1/2$.

Let M be an arbitrary monomer that is not an H or V monomer. Thus M contains exactly one element in each row and column, and we can represent $M : [n] \rightarrow [n]$ as a permutation. Then let the random variable $X = |C_\sigma \cap M|$. Summing over the columns of C_σ , we can write X as the sum of indicators

$$X = \sum_{i=1}^n \mathbf{1}_{\sigma(i) \geq M(i)} = \sum_{i=1}^n \mathbf{1}_{M^{-1}(\sigma(i)) \geq i}$$

Since $M^{-1} \circ \sigma$ is also a uniform random permutation, X has the same distribution as the number of *weak excedences* in a uniform random permutation [49] (the number of values i such that $\sigma(i) \geq i$).

By linearity of expectation, it is straightforward to calculate

$$E(X) = \sum_{i=1}^n \frac{i}{n} = \frac{n+1}{2},$$

but we require a large deviation bound for $P(X > \frac{n}{2} + t)$. These indicators are not independent, so a straightforward Chernoff bound will not work. Instead, we will identify a different random process with the exact same distribution as X that is easier to analyze.

By Proposition 1.4.3 in [49], there is a bijection that shows that the number of permutations $\sigma \in S_n$ with k weak excedences is the same as the number of permutations with $k-1$ *descents*: values i such that $\sigma(i+1) < \sigma(i)$ ³. Thus X has the same distribution as $1 + |\{i : \sigma(i+1) < \sigma(i)\}|$ for a uniform random permutation σ .

We now describe a process that generates a uniform random permutation, to give an even more convenient process with the same distribution as X . Let U_1, \dots, U_n be IID uniform[0, 1] random variables. Let $S_j = \sum_{i=1}^j U_i$ denote the partial sums, and

³This is referred to as the Eulerian number $A(n,k)$.

$F_j = S_j - \lfloor S_j \rfloor$ denote the fractional part of the partial sums. It is straightforward to verify that F_1, \dots, F_n are also IID uniform $[0, 1]$ random variables. Now since the F_i are distinct with probability 1, let σ be the permutation given by the order of the F_i : $\sigma(i) = |\{j : F_j \leq F_i\}|$. Since the F_i are IID, their order σ is a uniform random permutation. Now

$$\sigma(i+1) < \sigma(i) \iff F_{i+1} < F_i \iff \lfloor S_{i+1} \rfloor > \lfloor S_i \rfloor,$$

thus the number of descents in σ exactly counts the number of times the partial sums increment past an integer. Thus X has the same distribution as $1 + \lfloor S_n \rfloor$.

Now we can bound

$$P(X > \frac{n}{2} + 1 + t) = P(\lfloor S_n \rfloor > \frac{n}{2} + t) \leq P(S_n - \frac{n}{2} > t)$$

using Hoeffding's Inequality. Since S_n is the sum of independent variables bounded in $[0, 1]$ and $E(S_n) = n/2$, taking $t = \sqrt{n \ln n}$ gives

$$P(S_n - n/2 > t) < e^{-2t^2/n} = 1/n^2.$$

This is a bound on the probability that C_σ overlaps a fixed monomer at more than $n/2 + \sqrt{n \ln n} + 1$ binding sites. Thus displacing the monomer M will require at least $n - (n/2 + \sqrt{n \ln n} + 1) + 1 = n/2 - \sqrt{n \ln n}$ merges.

Now there can be at most $n(n-1)$ other monomer types (excluding the H and V monomers), so taking the union bound over all monomer types gives

$$P(C_\sigma \text{ has barrier} < n/2 - \sqrt{n \ln n}) < \frac{n(n-1)}{n^2} < 1.$$

Thus there exists some catalyst C_σ that leaves a barrier of at least $n/2 - \sqrt{n \ln n}$ to all other pathways.

Chapter 4

PSPACE-completeness of Reachability

4.1 Introduction

Our main result in this chapter is that the following problem is PSPACE-complete: given two configurations of a TBN, compute the energy barrier between them (more precisely, its decision variant: given threshold τ , is the barrier $\leq \tau$?). This follows via several intermediate results that may be of independent interest, concerning simulation and reachability in CRNs.

In Section 4.5, we show how a TBN can simulate a CRN, with programmably large energy barrier to any spurious reaction occurring in the absence of any of its reactants, if the CRN obeys four constraints. The simulated CRN must be (1) *reversible*: all reactions are reversible, (e.g., $A + B \rightleftharpoons C + D$), (2) *catalytic*: in each reaction, exactly one reactant changes (i.e., all other reactants are *catalysts*, e.g., $A + B + X \rightleftharpoons A + B + Y$, also written $X \xrightarrow{A+B} Y$), (3) *binary*: each species comes in two forms S^F and S^T , and can only be changed between these two (e.g., $S^F \xrightarrow{A+B+C} S^T$), and (4) *catalytically non-competitive*: if a non-catalytic reactant S appears in multiple reactions, each has only one catalyst, e.g., $S^F \xrightarrow{C_1} S^T$ and $S^F \xrightarrow{C_2} S^T$.

In our simulation of a chemical reaction (e.g., $S^F \xrightarrow{A+B+C} S^T$) with a TBN, changing S^F to S^T in the presence of k catalysts (e.g. A , B , and C) can be done with small barrier k , whereas a programmably large barrier exists to changing S^F into S^T if any

catalyst is absent. More precisely, the simulation is parameterized by a parameter w (where increasing w increases the size/complexity of the TBN), such that a reaction with k catalysts has energy barrier $k + w - 1$ to transform the reactant between S^F and S^T if any catalyst is absent.

Reactions with many catalysts are unrealistic since many molecules must simultaneously collide, and they furthermore have a lower energy barrier for a fixed parameter n in our TBN simulation. In Section 4.4, we show that any catalytic CRN that is *singular* (no catalyst has multiplicity greater than 1, e.g., disallowing $X \xrightarrow{2C} Y$ or $C \xrightarrow{C} D$) can be simulated by a singular, catalytically non-competitive CRN with at most *three* catalysts per reaction, preserving the constraints of binary and reversible if they are obeyed by the former CRN.

In Section 4.3, we show that the reachability problem for reversible, singular, catalytic, binary CRNs is PSPACE-complete (thus by the simulation described above, PSPACE-complete even for catalytically non-competitive CRNs with at most three catalysts per reaction). Combining these results, the problem of computing the energy barrier between two TBN configurations is PSPACE-complete as well. Since the energy barrier of spurious reactions can be made arbitrarily large compared to the barrier k when all k catalysts are present, this shows that even approximating the energy barrier within a constant factor is PSPACE-hard.

4.1.1 Related work

The Petri net model is exactly equivalent to the CRN model. The reachability problem in CRNs is known to be TOWER-hard, i.e., not even elementary [23], but this assumes that reactions can increase molecular counts. In a *1-conservative* CRN [38], every reaction conserves the total number of molecules, so its reachability problem is in PSPACE.

A special case of 1-conservative CRNs is the model of *population protocols* [3], in which each reaction has two reactants and two products, e.g., $A + B \rightarrow X + Y$. Reachability analysis has been important for verifying correctness of population protocols, both in the case of one fixed initial configuration [18, 20, 43, 50] and infinitely many initial configurations [7, 8], as well as general CRNs [42, 13, 14, 16, 33, 48]. Our model is similar to

the subclass of *immediate observation* population protocols [4, 27, 26, 46], in which all reactions have a catalyst that does not change, e.g., $C + X \rightarrow C + Y$. Our catalytic CRNs generalize this to allow more than 2 reactants and products, retaining the requirement that exactly one reactant can change.

A notable result of Esparza, Raskin, and Weil-Kennedy [27] is that the reachability problem for immediate observation population protocols is PSPACE-complete. Our PSPACE-completeness result concerns CRNs that are not directly comparable to immediate observation population protocols—we relax one constraint while tightening others—so the techniques of [27] are inapplicable to our setting. We allow more than one catalyst, but we require the system to be binary (S^F and S^T can change into each other \implies they cannot change into anything else), reversible, and catalytically non-competitive. The reversibility constraint can possibly be relaxed in the PSPACE-hardness proof of [27], by simulating a space-efficient reversible Turing machine [5]. However, the binary constraint presents a more significant challenge for the reduction of [27], which crucially relies on non-binary species for reactions such as $off[\sigma, n] \xrightarrow{C} on[\sigma, n]$ and $on[\sigma, n] \xrightarrow{C'} off[\sigma', n]$ for $\sigma \neq \sigma'$. The catalytically non-competitive constraint is a further challenge that we address in the simulation of Theorem 45.

As part of our presentation we give a construction that simulates a CRN with an arbitrary number of catalysts by a CRN with at most 3 catalysts; in some ways this resembles a result in population protocols by Blondin, Esparza, and Jaax [6] which showed that arbitrary k -way population protocols could be simulated by 2-way population protocols. However, while our construction outputs ≤ 4 -way protocols (a weaker constraint), our CRNs have the additional constraints of being catalytic and binary. The construction of [6] is noncatalytic and crucially uses non-binary species (e.g., in [6], a single agent can represent any of q_1 , d_1 , or r_1).

Our reduction more closely follows that of Thachuk and Condon [53], who showed, by reduction from the QSAT problem, PSPACE-completeness of reachability in a different class of CRNs appropriate for modeling certain DNA strand displacement systems. However, the construction of [53] is also crucially non-binary, as well as non-catalytic (more

than one reactant can change in each reaction), so novel techniques are required in our reduction.

Finally, there is significant work on computing energy barriers in DNA systems, where configurations are defined by the set of base pairs formed among strands [36, 54, 37, 22, 41, 52]. One can consider a single DNA base as a TBN binding site and a DNA strand as a TBN monomer. However, these results are not applicable to our setting, because they concern the subset of configurations (a.k.a. *secondary structures*) that are *unpseudoknotted*, a geometric constraint inexpressible in the TBN model. Furthermore, with the exception of Thachuk’s study of multi-stranded systems [52], the rest concern single-strand systems, considering the energetic contribution of forming bonds, but not that of separating complexes.

4.2 Model

The TBN model will be defined formally in Section 4.5.1; here we formally define CRNs and the various restrictions on them considered in this paper.

4.2.1 Chemical reaction networks

\mathbb{N} denotes the set of nonnegative integers. If Λ is a finite set (in this paper, of chemical species), we write \mathbb{N}^Λ to denote the set of functions $f: \Lambda \rightarrow \mathbb{N}$. Equivalently, we view an element $\mathbf{c} \in \mathbb{N}^\Lambda$ as a vector of $|\Lambda|$ nonnegative integers, with each coordinate “labeled” by an element of Λ . Given $S \in \Lambda$ and $\mathbf{c} \in \mathbb{N}^\Lambda$, we refer to $\mathbf{c}(S)$ as the *count of S in \mathbf{c}* . Given $\mathbf{c}, \mathbf{c}' \in \mathbb{N}^\Lambda$, we define the vector component-wise operations of addition $\mathbf{c} + \mathbf{c}'$, subtraction $\mathbf{c} - \mathbf{c}'$ (when $\mathbf{c} \geq \mathbf{c}'$), and scalar multiplication $n\mathbf{c}$ for $n \in \mathbb{N}$. We equivalently view a vector $\mathbf{c} \in \mathbb{N}^\Lambda$ as a *multiset* (set with multiplicities) of elements of Λ , e.g., $\mathbf{c}(A) = 2, \mathbf{c}(B) = 0, \mathbf{c}(C) = 1$ is the multiset $\{2A, C\}$. We may also equivalently write this as $2A + C$.

A *reaction* is a pair $\alpha = (\mathbf{r}, \mathbf{p}) \in \mathbb{N}^\Lambda \times \mathbb{N}^\Lambda$, such that $\mathbf{r} \neq \mathbf{p}$, specifying the stoichiometry of the reactants and products, e.g., $2A + B \rightarrow A + 3C$ is the pair $((2, 1, 0), (1, 0, 3))$. A (*finite*) *chemical reaction network (CRN)* is a pair $\mathcal{N} = (\Lambda, \mathcal{R})$, where Λ is a finite set of chemical *species*, and \mathcal{R} is a finite set of reactions. A *configuration* of a CRN $\mathcal{N} = (\Lambda, \mathcal{R})$

is a vector $\mathbf{c} \in \mathbb{N}^\Lambda$. A reaction $\alpha = (\mathbf{r}, \mathbf{p})$ is *applicable* to a configuration \mathbf{c} if $\mathbf{r} \leq \mathbf{c}$, i.e., \mathbf{c} contains enough of each of the reactants for the reaction to occur.

If α is applicable to \mathbf{c} , then write $\alpha(\mathbf{c})$ to denote the configuration $\mathbf{c} + \mathbf{r} - \mathbf{p}$ (i.e., the configuration that results from applying reaction α to \mathbf{c}). If $\mathbf{d} = \alpha(\mathbf{c})$ for some reaction $\alpha \in \mathcal{R}$, we write $\mathbf{c} \rightarrow_{\mathcal{N}} \mathbf{d}$, or merely $\mathbf{c} \rightarrow \mathbf{d}$ when \mathcal{N} is clear from context, and we write $\mathbf{c} \xrightarrow{\alpha} \mathbf{c}'$ when emphasizing which reaction took place. We say that \mathbf{d} is *reachable* from \mathbf{c} , writing $\mathbf{c} \Rightarrow_{\mathcal{N}} \mathbf{d}$, or merely $\mathbf{c} \Rightarrow \mathbf{d}$ when the CRN \mathcal{N} is clear from context, if zero or more reactions can take \mathbf{c} to \mathbf{d} .

4.2.2 Constraints on CRNs

A CRN $\mathcal{N} = (\Lambda, \mathcal{R})$ is *reversible* if, for every $(\mathbf{r}, \mathbf{p}) \in \mathcal{R}$, also $(\mathbf{p}, \mathbf{r}) \in \mathcal{R}$; we write such reactions as $\mathbf{r} \leftrightarrow \mathbf{p}$. A reaction (\mathbf{r}, \mathbf{p}) is *1-conservative* if $|\mathbf{r}| = |\mathbf{p}|$, i.e., it has the same number of products and reactants, e.g., $A + B + C \rightarrow X + Y + Z$. The reaction is *singular* if $\mathbf{r}(S) \leq 1$ for all $S \in \Lambda$, e.g., forbidding $2A \rightarrow X + Y$. A CRN is *1-conservative* if each reaction is 1-conservative (possibly with different numbers of reactants *between* different reactions, e.g., \mathcal{R} can have both reactions $A + B \rightarrow C + D$ and $X \rightarrow Y$). A CRN is *singular* if each reaction is singular.

For $k \in \mathbb{N}$, a reaction (\mathbf{r}, \mathbf{p}) is *k-catalytic* if $|\mathbf{r}| = |\mathbf{p}| = k + 1$ and for two species $R, P \in \Lambda$, $\mathbf{r}(R) = \mathbf{p}(P) = 1$, $\mathbf{p}(R) = \mathbf{r}(P) = 0$, and $\mathbf{r}(S) = \mathbf{p}(S)$ for all other $S \in \Lambda \setminus \{R, P\}$. In other words, the reaction is 1-conservative, the count of one reactant is decremented and the count of one product is incremented, and all other species are *catalysts* that don't change in the reaction. We write such a reaction with catalysts C_1, \dots, C_k and reactant R that changes to product P as $R \xrightarrow{C_1 + \dots + C_k} P$. Note that a unimolecular conservative reaction $A \rightarrow B$ is 0-catalytic. Note that a singular catalytic reaction forbids $C \xrightarrow{C} D$, since this corresponds to $2C \rightarrow C + D$. A CRN is *≤k-catalytic* if each reaction is *k'*-catalytic for some $0 \leq k' \leq k$, and it is *catalytic* if it is *≤k*-catalytic for some $k \in \mathbb{N}$. A CRN with only 1-catalytic reactions is equivalent to an immediate observation population protocol [4, 27, 26, 46].

A catalytic CRN is *binary* if, for any pair of reactions $R \xrightarrow{C_1 + \dots + C_k} P$ and $R' \xrightarrow{C_1 + \dots + C_k} P'$ we have $R = R' \implies P = P'$ and $R = P' \implies R' = P$, i.e., any species that can

change between two states in any reaction can change *only* between those two states in the entire CRN. This forbids, for example, reactions such as $X \xleftrightarrow{C} Y$ and $Y \xleftrightarrow{C'} Z$. By convention we write such species as X^F and X^T , and we refer to F and T as the two *states* of X . A catalytic CRN is *catalytically non-competitive* if for every species S , if S is a non-catalyst in multiple reactions, then each reaction has only one catalyst, e.g., $S^F \xleftrightarrow{C_1} S^T$ and $S^F \xleftrightarrow{C_2} S^T$.

A configuration \mathbf{c} of a binary CRN is *1-safe* if $\mathbf{c} \in \{0, 1\}^\Lambda$ and $\mathbf{c}(S^F) + \mathbf{c}(S^T) = 1$ for all binary pairs $S^F, S^T \in \Lambda$, i.e., each binary pair (S^F, S^T) has exactly one representative in the configuration. A binary CRN \mathcal{N} is *1-safe* if, for any 1-safe configuration \mathbf{c} , if $\mathbf{c} \Rightarrow_{\mathcal{N}} \mathbf{d} \in C$, then \mathbf{d} is 1-safe, i.e., starting with 1 representative from each binary pair, every reachable configuration also has exactly 1 representative from each binary pair.

Observation 39. *Every 1-conservative, binary, reversible, singular CRN is 1-safe.*

4.3 PSPACE-completeness of binary, reversible, singular, catalytic CRNs

The main result of this section is Theorem 40.

Theorem 40. *It is PSPACE-complete to decide, given two configurations \mathbf{c} and \mathbf{d} of a binary, reversible, catalytic, singular, 1-safe CRN \mathcal{N} , whether $\mathbf{c} \Rightarrow_{\mathcal{N}} \mathbf{d}$.*

The proof appears below in Section 4.3.3.

4.3.1 Definition of CRN simulation

Many notions of CRN equivalence exist, based, for example, on weak bisimulation [33], strong bisimulation [15], and pathway decomposition [48]. We use Milner’s definition of *weak bisimulation* [39], adapted for our particular needs in Theorem 45, but we simply use the term *simulate*.¹

¹Unfortunately, for various technical reasons, none of the cited papers on CRN simulation use definitions matching the sort of simulation we have. The notions of simulation in [33, 48] extend beyond Milner’s definition primarily for the purpose of more gracefully handling CRNs with unbounded counts. Since all the CRNs we study are 1-conservative (they conserve total molecular count), Milner’s theory of bisimulation suffices for our purposes.

Definition 41 (formal and implementing CRNs, interpretations). Let $\mathcal{S} = (\Lambda_{\mathcal{S}}, \mathcal{R}_{\mathcal{S}})$ and $\mathcal{N} = (\Lambda_{\mathcal{N}}, \mathcal{R}_{\mathcal{N}})$ be CRNs such that $\Lambda_{\mathcal{N}} \subseteq \Lambda_{\mathcal{S}}$; we consider \mathcal{S} to be simulating \mathcal{N} . We refer to \mathcal{N} as the formal CRN and \mathcal{S} as the implementation CRN, and similarly to their configurations as formal and implementation configurations, respectively. The species in $\Lambda_{\mathcal{N}}$ are formal species, and the species in $\Lambda_{\mathcal{S}} \setminus \Lambda_{\mathcal{N}}$ are intermediate species. For every configuration $\mathbf{c}_{\mathcal{S}} \in \mathbb{N}^{\Lambda_{\mathcal{S}}}$, define its interpretation $m(\mathbf{c}_{\mathcal{S}}) \in \mathbb{N}^{\Lambda_{\mathcal{N}}}$ to be the configuration restricted to formal species, e.g., if $\Lambda_{\mathcal{N}} = \{A, B, C\}$ and $\Lambda_{\mathcal{S}} = \{A, B, C, x, y, z\}$, then $m(\{2A, 3B, 4C, 5x, 6y, 7z\}) = \{2A, 3B, 4C\}$.

A reaction $\alpha_{\mathcal{S}} = (\mathbf{r}, \mathbf{p}) \in \mathcal{R}_{\mathcal{S}}$ is *silent* if $m(\mathbf{r}) = m(\mathbf{p})$, i.e., it only changes the counts of intermediate species. For a reaction α and configurations \mathbf{c}, \mathbf{d} of the same CRN, recall that $\mathbf{c} \xrightarrow{\alpha} \mathbf{d}$ denotes that α is applicable to \mathbf{c} , and applying it results in \mathbf{d} . If $\alpha_{\mathcal{N}} \in \mathcal{R}_{\mathcal{N}}$ is a formal reaction and $\mathbf{c}_{\mathcal{S}}, \mathbf{d}_{\mathcal{S}} \in \mathbb{N}^{\Lambda_{\mathcal{S}}}$ are implementation configurations, we write $\mathbf{c}_{\mathcal{S}} \overset{\alpha_{\mathcal{N}}}{\rightsquigarrow} \mathbf{d}_{\mathcal{S}}$ to denote that, for some (non-silent) $\alpha_{\mathcal{S}} \in \mathcal{R}_{\mathcal{S}}$, $\mathbf{c}_{\mathcal{S}} \xrightarrow{\alpha_{\mathcal{S}}} \mathbf{d}_{\mathcal{S}}$, and $m(\mathbf{c}_{\mathcal{S}}) \xrightarrow{\alpha_{\mathcal{N}}} m(\mathbf{d}_{\mathcal{S}})$. In other words, some \mathcal{S} reaction can take $\mathbf{c}_{\mathcal{S}}$ to $\mathbf{d}_{\mathcal{S}}$, and looking at only the formal species in $\mathbf{c}_{\mathcal{S}}$ and $\mathbf{d}_{\mathcal{S}}$, this “looks like” applying reaction $\alpha_{\mathcal{N}}$.

We write $\mathbf{c}_{\mathcal{S}} \overset{\perp}{\rightarrow} \mathbf{d}_{\mathcal{S}}$ to denote that $\mathbf{d}_{\mathcal{S}}$ is reachable from $\mathbf{c}_{\mathcal{S}}$ by a sequence of silent reactions. We write $\mathbf{c}_{\mathcal{S}} \xrightarrow{\alpha_{\mathcal{N}}} \mathbf{d}_{\mathcal{S}}$ to denote that $\mathbf{c}_{\mathcal{S}} \overset{\perp}{\rightarrow} \mathbf{c}'_{\mathcal{S}} \overset{\alpha_{\mathcal{N}}}{\rightsquigarrow} \mathbf{d}'_{\mathcal{S}} \overset{\perp}{\rightarrow} \mathbf{d}_{\mathcal{S}}$, i.e., a sequence of reactions takes $\mathbf{c}_{\mathcal{S}}$ to $\mathbf{d}_{\mathcal{S}}$, one of which looks like $\alpha_{\mathcal{N}}$, and the rest are silent.

Definition 42 (CRN simulation). Let $\mathcal{S} = (\Lambda_{\mathcal{S}}, \mathcal{R}_{\mathcal{S}})$ and $\mathcal{N} = (\Lambda_{\mathcal{N}}, \mathcal{R}_{\mathcal{N}})$ be as in Definition 41. Say \mathcal{S} simulates (a.k.a., weakly bisimulates) \mathcal{N} if the following conditions hold:

sound: For every reachable² implementation configuration $\mathbf{c}_{\mathcal{S}} \in \mathbb{N}^{\Lambda_{\mathcal{S}}}$ and non-silent reaction $\alpha_{\mathcal{S}} \in \mathcal{R}_{\mathcal{S}}$, if $\mathbf{c}_{\mathcal{S}} \xrightarrow{\alpha_{\mathcal{S}}} \mathbf{d}_{\mathcal{S}}$, then for some $\alpha_{\mathcal{N}} \in \mathcal{R}_{\mathcal{N}}$, $\mathbf{c}_{\mathcal{S}} \overset{\alpha_{\mathcal{N}}}{\rightsquigarrow} \mathbf{d}_{\mathcal{S}}$ and $\mathbf{c}_{\mathcal{N}} \xrightarrow{\alpha_{\mathcal{N}}} \mathbf{d}_{\mathcal{N}}$, where $\mathbf{c}_{\mathcal{N}} = m(\mathbf{c}_{\mathcal{S}})$ and $\mathbf{d}_{\mathcal{N}} = m(\mathbf{d}_{\mathcal{S}})$. (If a non-silent reaction occurs in $\mathbf{c}_{\mathcal{S}}$, then it “looks like”—restricted to formal species—a formal reaction $\alpha_{\mathcal{N}}$, and $\alpha_{\mathcal{N}}$ is applicable to formal configuration $\mathbf{c}_{\mathcal{N}} = m(\mathbf{c}_{\mathcal{S}})$, and goes to the expected formal configuration $\mathbf{d}_{\mathcal{N}} = m(\mathbf{d}_{\mathcal{S}})$.)

²Some definitions of weak bisimulation have no reachability constraint. For correctness of our simulation, we require count ≤ 1 of certain intermediate species, a bound preserved in reachable implementation configurations by the fact our CRN is catalytic, reversible, and singular.

complete: For every reachable implementation configuration $\mathbf{c}_S \in \mathbb{N}^{\Lambda_S}$ such that $m(\mathbf{c}_S) = \mathbf{c}_N$, if $\mathbf{c}_N \xrightarrow{\alpha_N} \mathbf{d}_N$ for some $\alpha_N \in \mathcal{R}_N$, then $\mathbf{c}_S \xrightarrow{\alpha_N} \mathbf{d}_S$, where $m(\mathbf{d}_S) = \mathbf{d}_N$. (If α_N can occur in \mathbf{c}_N , then it can be simulated in any \mathbf{c}_S representing \mathbf{c}_N , possibly by first executing some silent reactions from \mathbf{c}_S .)³

The primary objective of this definition of simulation, for the purpose of connecting Theorem 40 with Corollary 43, is that reachability of formal configurations is preserved by the implementation CRN: $\mathbf{c}_S \Rightarrow \mathbf{d}_S$ for $\mathbf{c}_S, \mathbf{d}_S \in \mathbb{N}^{\Lambda_S}$ if and only if $m(\mathbf{c}_S) \Rightarrow m(\mathbf{d}_S)$.

Applying Theorem 45 to the CRN in the proof of Theorem 40 gives the following:

Corollary 43. *It is PSPACE-complete to decide, given two configurations \mathbf{c} and \mathbf{d} of a binary, reversible, ≤ 3 -catalytic, catalytically non-competitive, singular, 1-safe CRN \mathcal{N} , whether $\mathbf{c} \Rightarrow_{\mathcal{N}} \mathbf{d}$.*

4.3.2 Construction

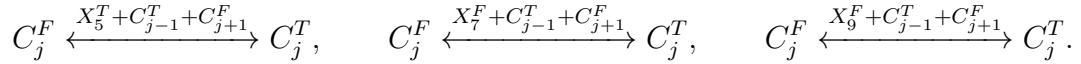
Since a catalytic CRN is 1-conservative, configurations reachable from \mathbf{c} can be represented in space $O(|\mathbf{c}|)$, so the problem is in PSPACE. We now show that QSAT, the PSPACE-complete problem of evaluating a quantified Boolean CNF formula [40], is polynomial-time many-one reducible to this problem. Throughout Section 4.3, let \mathcal{N} refer to the CRN constructed below.

Let $\psi = \forall x_1 \exists x_2 \forall x_3 \exists x_4 \dots \phi(x_1, \dots, x_n)$ be a quantified Boolean formula, where ϕ is in 3CNF with m clauses $c_1 \wedge \dots \wedge c_m$, with $c_i = w_{i,1} \vee w_{i,2} \vee w_{i,3}$ for literals $w_{i,j}$, each literal equal to some variable x_k or its negation $\neg x_k$. We have binary species (each with F and T states) $X_1, \dots, X_n, C_1, \dots, C_m, L_1, \dots, L_n$, and R_1, \dots, R_n , and Ψ . Uppercase species $X_1, \dots, X_n, C_1, \dots, C_m$, and Ψ represent lowercase Boolean formula variables x_i , and clauses c_j , and the whole quantified formula ψ , respectively. L_i and R_i represent the left and right child in the tree (see Fig. 4.1) of x_i under the current truth assignment to x_1, \dots, x_i represented by the F/T states of species X_1, \dots, X_n . Let \mathbf{c} be the configuration

³In the theory of weak bisimulation allowing more general interpretation functions, one would typically also require that every formal configuration is represented by some implementation configuration. This holds trivially for our definition of interpretation, so we omit the explicit condition.

with one copy of the F state of each species, and let \mathbf{d} have one copy of the T state of each species.

NP-hardness. To build intuition, it is useful first to consider the simpler task of showing NP-hardness by transforming the formula $\exists x_1 \exists x_2 \dots \exists x_n \phi(x_1, \dots, x_n)$ into a CRN. We allow the CRN to nondeterministically try every possible assignment to x_1, \dots, x_n , while ensuring that the current assignment cannot change while the clause values are being evaluated. To do this, use “assignment” reactions $X_i^F \xrightleftharpoons{C_1^F} X_i^T$ for each $1 \leq i \leq n$, and for each $1 \leq j \leq m$ add three **clause evaluation** reactions, supposing for example that clause $c_j = (x_5 \vee \neg x_7 \vee \neg x_9)$:



If $j = 1$, then omit the C_{j-1}^T catalyst, and if $j = n$, then omit the C_{j+1}^F catalyst. The X_i catalyst in each reaction ensures that the clause can evaluate to true only if at least one of its literals is true. The other catalysts ensure that clauses are evaluated sequentially in order, and only if none have been evaluated, i.e., C_1^F is present, can the assignment reactions alter the truth assignment to the x_i 's. In this construction, ϕ is satisfiable if and only if C_m^T is producible, but in a configuration that reveals the satisfying assignment. To allow the configuration \mathbf{d} to be computable easily from ϕ , add the “scrambling” reactions $X_i^F \xrightleftharpoons{C_m^T} X_i^T$ for each $1 \leq i \leq n$; then the “all- T ” configuration \mathbf{d} is reachable from the “all- F ” configuration \mathbf{c} if and only if ϕ is satisfiable.

Our reduction from QSAT is similar to this, but it uses the L_i and R_i species to conduct a more intricate search of the truth assignments to handle the alternating quantifiers of ψ . The clause evaluation reactions are identical. We explain below how to modify the assignment reactions and add other reactions.

Consider the complete binary tree of depth $n + 1$ whose leaves represent all 2^n assignments to $x_1 \dots x_n$, with the nodes at depth $1 \leq i \leq n$ (root is depth 1) labeled by x_i ; see Fig. 4.1. At any time, the current truth assignment to variables x_1, \dots, x_n encoded in $X_1^{F/T} \dots X_n^{F/T}$ represents a root-to-leaf path in the tree, so when we say “node x_i ”, we mean “the node labeled with x_i along the current path”. A leaf node evaluates to true if the assignment it represents satisfies ϕ . An \exists node evaluates to the Boolean OR of its

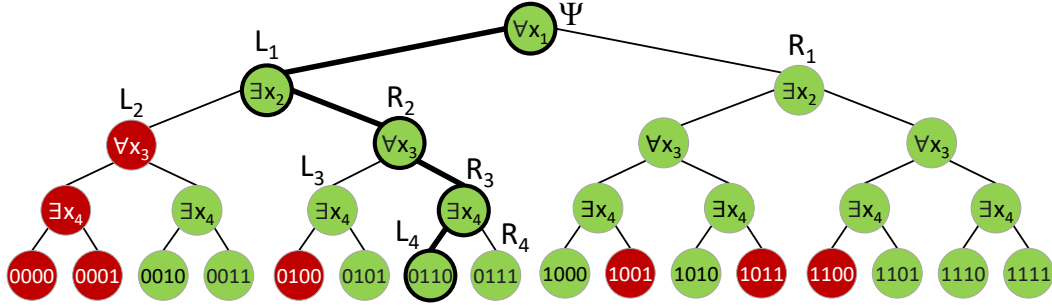


Figure 4.1: Tree representing all assignments to formula $\psi = \forall x_1 \exists x_2 \forall x_3 \exists x_4 [(x_1 \vee x_2 \vee x_3) \wedge (\neg x_2 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee \neg x_4)]$. Red (dark) nodes evaluate to false and green (light) nodes evaluate to true. Every configuration encodes some truth assignment to the x_i 's, corresponding to a root-to-leaf path; the example path shown in bold corresponds to the assignment $x_1 = F, x_2 = T, x_3 = T, x_4 = F$. Depending on the assignment, the L_i, R_i species reference the pair of children of x_i nodes along that path. All L_i, R_i start in state F , as does the root Ψ , and we enforce that L_i, R_i can only be set to T under a particular assignment of x_i 's if either its node or one of its ancestors evaluates to T .

children, and a \forall node evaluates to the Boolean AND of its children. The entire quantified formula ψ is true if the root node evaluates to true.

The species L_i^T (respectively, R_i^T) represents “the left (resp., right) child of node x_i evaluates to true under the current assignment, and $x_i = \text{false}$ (resp., true)”.

We first add reactions to evaluate the tree based on the results of the clause evaluation reactions. For $i = n$, set L_n (left) or R_n (right) to true if the formula is true under the current assignment (C_m^T is present) and $X_n^{F/T}$ represents the appropriate leaf child: left (X_n^F) or right (X_n^T) **leaf evaluation** reactions:



We also have **internal evaluation** reactions. Internal \forall nodes (odd $i < n$) require both left and right children to be true:



Internal \exists nodes (even $i < n$) require at least one child to be true:



The entire formula is evaluated as a \forall node based on the values of L_1 and R_1 in the **root evaluation** reaction: $\Psi^F \xleftarrow{L_1^T + R_1^T} \Psi^T$.

If a node labeled x_i has evaluated to true (i.e., if either L_i^T or R_i^T is present), then we can “reset” the values L_j, R_j for $j > i$ storing its descendants’ values (getting L_j, R_j back to state F , to allow testing new values of variable X_j via assignment reactions below), in order to reuse the species to compute in other branches of the tree; add the following **reset** reactions for all $1 \leq i < j \leq n$:



If the entire formula ψ is true, then we “scramble” the F/T states to enable \mathbf{d} to have every species in state T ; for all species $S \neq \Psi$, add **scrambling** reaction $S^F \xrightarrow{\Psi^T} S^T$.

We now require **assignment** reactions to alter the assignment of truth values to variables. We retain the reaction $X_n^F \xrightarrow{C_1^F} X_n^T$ from the NP-hardness reduction, but for $i < n$, modify the reactions adjusting truth assignments of x_i to have more catalysts: $X_i^F \xleftarrow{C_1^F + L_{i+1}^F + R_{i+1}^F + \dots + L_n^F + R_n^F} X_i^T$. In other words, we can change a variable x_i only if the L_j/R_j species representing its descendants (those with $j > i$) are in state F . If the node labeled with x_i evaluates to true, then either L_i^T or R_i^T (depending on whether x_i itself is false or true) will be produced, after which the descendant L_j/R_j ’s are no longer required to determine the truth value of x_i ’s subtree.

By inspection the CRN is 1-conservative, reversible, and singular, so it is 1-safe by Observation 39.

4.3.3 Proof

Proof of Theorem 40. We described the construction in Section 4.3.2. Now we argue that it reduces QSAT to our reachability problem, by showing that ψ is true if and only if the “all- T ” configuration \mathbf{d} is reachable from the the “all- F ” configuration \mathbf{c} .

We apply Lemma 44, stated and proven below. Because it is proven by induction, it depends on some complex definitions prior to the statement, but our use of it here is with the simplest form of those definitions, those listed in the examples under “ $k = 0$ ” just after the statement of Lemma 44.

(\implies): Assume ψ is true. Since the first variable is a \forall variable, this implies that the quantified formulas $\psi(F)$, obtained by setting x_1 to F , and $\psi(T)$, obtained by setting

x_1 to T , are both true. Let A_ε be the set of configurations where the following are present: C_1^F , Ψ^F , and L_i^F, R_i^F for all $i \in \{2, \dots, n\}$. Note that $\mathbf{c} \in A_\varepsilon$. We apply Lemma 44 with this A_ε and $\mathcal{S}(F) = \{R_1, \Psi^F\}$, which gives that $\mathbf{c} \Rightarrow_{R_1, \Psi^F} \mathbf{x}$ (i.e., \mathbf{c} can reach to \mathbf{x} with Ψ^F present the whole time, while preserving the state of R_1), where $\mathbf{x}(L_1^T) > 0$, such that Ψ^F is present the whole time, and R_1 , which starts as R_1^F , is preserved as R_1^F . Then $\mathbf{x} \in A_\varepsilon$.

Thus we can apply Lemma 44 again (with $\mathcal{S}(T) = \{L_1, \Psi^F\}$) to get that $\mathbf{x} \Rightarrow_{L_1, \Psi^F} \mathbf{y}$ such that $\mathbf{y}(R_1^T) > 0$. Since L_1 appears in the reachability subscript, and since we produced L_1^T in the previous step, L_1^T is preserved along the path, so $\mathbf{y}(L_1^T) > 0$. Since L_1^T and R_1^T are both present in \mathbf{y} , apply the root evaluation reaction $\Psi^F \xleftarrow{L_1^T + R_1^T} \Psi^T$ to create Ψ^T . Then apply the scrambling reactions $S^F \xleftrightarrow{\Psi^T} S^T$ forward on all species $S \neq \Psi$ to reach \mathbf{d} .

(\Leftarrow): Assume ψ is false. Then at least one of $\psi(F)$ or $\psi(T)$ is false; assume without loss of generality $\psi(F)$ is false. Suppose for the sake of contradiction that we could produce L_1^T and R_1^T while keeping Ψ^F . Then no matter what is the value of x_1 , the reset reactions $L_j^F \xleftarrow{L_1^T + x_1^F} L_j^T$, $L_j^F \xleftarrow{R_1^T + x_1^T} L_j^T$, $R_j^F \xleftarrow{L_1^T + x_1^F} R_j^T$, $R_j^F \xleftarrow{R_1^T + x_1^T} R_j^T$ for all $j \geq 2$ can create $L_2^F, R_2^F, \dots, L_n^F, R_n^F$. Then we would have $\mathbf{c} \Rightarrow_{R_1, \Psi^F} \{L_1^T, L_2^F, R_2^F, \dots, L_n^F, R_n^F, ?\}$, violating Lemma 44 since $\psi(F)$ is false. So we cannot create both L_1^T and R_1^T while Ψ^F is present, so the root evaluation reaction $\Psi^F \xleftarrow{L_1^T + R_1^T} \Psi^T$ can never execute. Thus the all- T configuration \mathbf{d} is not reachable from \mathbf{c} . \square

We now build up the technical definitions required to state and prove Lemma 44.

For $x_1, \dots, x_i \in \{F, T\}$, let $\psi(x_1, \dots, x_i)$ represent ψ with the first i variables assigned the values x_1, \dots, x_i . For example, if $\psi = \forall x_1 \exists x_2 \forall x_3 [(x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3)]$, then for $x_1 = T$, $\psi(x_1) = \exists x_2 \forall x_3 [(\neg x_2 \vee x_3) \wedge (x_2 \vee x_3)]$, whereas for $x_1 = F$, $\psi(x_1) = \exists x_2 \forall x_3 [(x_2) \wedge (x_2 \vee x_3)]$. $\psi(x_1, \dots, x_i)$ corresponds to the evaluated value (as described in the construction) of the subtree labeled by truth values x_1, \dots, x_i , e.g., $\psi(F)$ is the value of the left subtree of the root, $\psi(T)$ is the value of the right subtree of the root, and $\psi(x_1, \dots, x_n)$ is simply the value of the unquantified formula ϕ on the given assignment

(value of a leaf node). In the context of a CRN configuration, we write $\psi(x_1, \dots, x_i)$ to mean the above with truth values taken from the configuration's assignment to variables represented by species $X_1^{F/T}, \dots, X_i^{F/T}$.

We now define some (unfortunately dense) technical notation to establish the intuitive claim that the CRN properly evaluates the formula ψ .

For a sequence of $0 \leq k \leq n$ Boolean values $b_1 \dots b_k \in \{F, T\}^k$, let $A_{b_1 \dots b_k}$ be the set of all reachable configurations of \mathcal{N} (intuitively, $A_{b_1 \dots b_k}$ represents a particular truth assignment in the X_i species, with some constraints on other species) with the following species present:

- for all $i \in \{1, \dots, k\}$: $X_i^{b_i}$
- for all $i \in \{1, \dots, k\}$: L_i^F if $b_i = F$, and R_i^F if $b_i = T$
- for all $i \in \{k+2, \dots, n\}$: L_i^F, R_i^F
- C_1^F (Thus all clause species C_j for $j > 1$ are also state F .)
- Ψ^F

For example, if $n = 7$, A_{TTFT} is the set of configurations with the following species present: $X_1^T, R_1^F, X_2^T, R_2^F, X_3^F, L_3^F, X_4^T, R_4^F, C_1^F, \Psi^F, L_6^F, L_7^F, R_6^F, R_7^F$. The following species can be any state:

- variable species X_5, X_6, X_7 (indices $> k$)
- left/right species L_1, L_2, R_3, L_4 (indices $\leq k$, where interpreting L as F and R as T , each is the negation of the corresponding b_j , i.e., siblings of ancestors of X_5)

As another example, A_ε (with no subscript Booleans) is the set of configurations with the following species present: L_i^F, R_i^F for all $1 < i \leq n$, C_1^F , and Ψ^F .

Note that $A(b_1, \dots, b_{k-1}, b_k) \subset A(b_1, \dots, b_{k-1})$, since the former has all the constraints of the latter.

Below, if we have two sets of species, for example $\{X, Y, Z\}$ and $\{D, E\}$, and a set of configurations A , we write $A \Rightarrow_{X,Y,Z} \{D, E, \dots\}$ to denote that the CRN can reach from

every configuration in A to a configuration that has D and E present, while maintaining that X, Y, Z are present the entire time. Recall that species are binary. The binary species may be labeled with F or T , or not. If we write, for example, $A \Rightarrow_{X^F, Y^T, Z} \{D, E, ?\}$, it means X^F, Y^T are present the entire time, and whichever is the state of Z at the start (whether Z^F or Z^T), that state is also present the entire time.

For $b_k \in \{F, T\}$, let $\text{child}(b_k) = L_k$ if $b_k = F$ and $\text{child}(b_k) = R_k$ if $b_k = T$. Define $\mathcal{S}(b_1, \dots, b_k)$ as the following set of binary species (for use in the relation $\Rightarrow_{\mathcal{S}(b_1, \dots, b_k)}$, so some possibly are unlabeled with F or T as above) See Fig. 4.1.

$$\begin{aligned} \mathcal{S}(b_1, \dots, b_k) = \{ & X_1^{b_1}, \text{child}(b_1)^F, \text{child}(\neg b_1), \\ & X_2^{b_2}, \text{child}(b_2)^F, \text{child}(\neg b_2), \\ & \dots \\ & X_{k-1}^{b_{k-1}}, \text{child}(b_{k-1})^F, \text{child}(\neg b_{k-1}), \\ & \text{child}(\neg b_k), \\ & \Psi^F \}. \end{aligned}$$

For example, $\mathcal{S}(F) = \{R_1, \Psi^F\}$, $\mathcal{S}(T) = \{L_1, \Psi^F\}$,

$$\begin{aligned} \mathcal{S}(F, F) &= \{X_1^F, L_1^T, R_1, R_2, \Psi^F\}, \\ \mathcal{S}(F, T) &= \{X_1^F, L_1^T, R_1, L_2, \Psi^F\}, \\ \mathcal{S}(T, F) &= \{X_1^T, R_1^T, L_1, R_2, \Psi^F\}, \\ \mathcal{S}(T, T) &= \{X_1^T, R_1^T, L_1, L_2, \Psi^F\}, \end{aligned}$$

and

$$\begin{aligned} \mathcal{S}(T, T, F, T, F) = \{ & X_1^T, R_1^F, L_1, \\ & X_2^T, R_2^F, L_2, \\ & X_3^F, L_3^F, R_3, \\ & X_4^T, R_4^F, L_4, \\ & R_5, \\ & \Psi^F \}. \end{aligned}$$

Intuitively, $\mathcal{S}(b_1, \dots, b_k)$ represents a path from the root to a node (internal or leaf), in the following way: X_i 's for $1 \leq i \leq k-1$ set consistently with b_i 's, the “child” species

L_i/R_i set consistently with those, e.g., L_1^T if X_1^F , since this means first non-root node is a left child (and similarly R_1^T if X_1^T), and the siblings of each node on the path (including the final node at index k) are included but have no superscript. Since we use \mathcal{S} in the $\Rightarrow_{\mathcal{S}}$ relation, this means that those sibling species can be either F or T , but must stay that way along the entire path.

The following technical lemma expresses the intuitive idea that it is possible to produce L_i^T/R_i^T only if the node it currently represents in the tree (given values of x_1, \dots, x_i) is true, or one of its ancestors is true.⁴

Lemma 44. *Let $0 \leq k < n$ and let $b_1, \dots, b_k \in \{F, T\}^k$ be a sequence of k Boolean values.*

Then

$$\psi(b_1, \dots, b_k, F) \iff A_{b_1 \dots b_k} \Rightarrow_{\mathcal{S}(b_1, \dots, b_k, F)} \{L_{k+1}^T, L_{k+2}^F, R_{k+2}^F, \dots, L_n^F, R_n^F, ?\}$$

and

$$\psi(b_1, \dots, b_k, T) \iff A_{b_1 \dots b_k} \Rightarrow_{\mathcal{S}(b_1, \dots, b_k, T)} \{R_{k+1}^T, L_{k+2}^F, R_{k+2}^F, \dots, L_n^F, R_n^F, ?\}.$$

Furthermore, the reached configurations in each case are also in A_{b_1, \dots, b_k} .

For example, here are concrete statements of the lemma for a few values of b_1, \dots, b_k :

$k = 0$:

$$\psi(F) \iff A_{\varepsilon} \Rightarrow_{R_1, \Psi^F} \{L_1^T, L_2^F, R_2^F, \dots, L_n^F, R_n^F, ?\}$$

$$\psi(T) \iff A_{\varepsilon} \Rightarrow_{L_1, \Psi^F} \{R_1^T, L_2^F, R_2^F, \dots, L_n^F, R_n^F, ?\}$$

$k = 1$:

$$\psi(F, F) \iff A_F \Rightarrow_{X_1^F, L_1^F, R_1, R_2, \Psi^F} \{L_2^T, L_3^F, R_3^F, \dots, L_n^F, R_n^F, ?\}$$

$$\psi(F, T) \iff A_F \Rightarrow_{X_1^F, L_1^F, R_1, L_2, \Psi^F} \{R_2^T, L_3^F, R_3^F, \dots, L_n^F, R_n^F, ?\}$$

$$\psi(T, F) \iff A_T \Rightarrow_{X_1^T, R_1^F, L_1, R_2, \Psi^F} \{L_2^T, L_3^F, R_3^F, \dots, L_n^F, R_n^F, ?\}$$

$$\psi(T, T) \iff A_T \Rightarrow_{X_1^T, R_1^F, L_1, L_2, \Psi^F} \{R_2^T, L_3^F, R_3^F, \dots, L_n^F, R_n^F, ?\}$$

⁴The statement about ancestors deserves some clarification. In our construction, suppose $x_1 = F$, and suppose the left child $\exists x_2$ node is true because its right child is true, as in Fig. 4.1. Then we can produce R_2^T when $x_1 = F, x_2 = T$. Then we use R_2^T to create L_1^T via the \exists internal evaluation reaction $L_1^F \xleftarrow{x_1^F + R_2^T} L_1^T$. Finally, use L_1^T to produce L_2^T via the reset reaction $L_2^F \xleftarrow{L_1^T + x_1^F} L_2^T$, even though the left child of x_2 is false in the tree (as in Fig. 4.1).

$k = 2$:

$$\begin{aligned}
\psi(F, F, F) &\iff A_{FFF} \Rightarrow_{X_1^F, L_1^F, R_1, X_2^F, L_2^F, R_2, R_3, \Psi^F} \{L_3^T, L_4^F, R_4^F, \dots, L_n^F, R_n^F, ?\} \\
\psi(F, F, T) &\iff A_{FFT} \Rightarrow_{X_1^F, L_1^F, R_1, X_2^F, L_2^F, R_2, L_3, \Psi^F} \{R_3^T, L_4^F, R_4^F, \dots, L_n^F, R_n^F, ?\} \\
\psi(F, T, F) &\iff A_{FTF} \Rightarrow_{X_1^F, L_1^F, R_1, X_2^T, R_2^F, L_2, R_3, \Psi^F} \{L_3^T, L_4^F, R_4^F, \dots, L_n^F, R_n^F, ?\} \\
\psi(F, T, T) &\iff A_{FTT} \Rightarrow_{X_1^F, L_1^F, R_1, X_2^T, R_2^F, L_2, L_3, \Psi^F} \{R_3^T, L_4^F, R_4^F, \dots, L_n^F, R_n^F, ?\} \\
\psi(T, F, F) &\iff A_{TFF} \Rightarrow_{X_1^T, R_1^F, L_1, X_2^F, L_2^F, R_2, R_3, \Psi^F} \{L_3^T, L_4^F, R_4^F, \dots, L_n^F, R_n^F, ?\} \\
\psi(T, F, T) &\iff A_{TFT} \Rightarrow_{X_1^T, R_1^F, L_1, X_2^F, L_2^F, R_2, L_3, \Psi^F} \{R_3^T, L_4^F, R_4^F, \dots, L_n^F, R_n^F, ?\} \\
\psi(T, T, F) &\iff A_{TTF} \Rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, R_3, \Psi^F} \{L_3^T, L_4^F, R_4^F, \dots, L_n^F, R_n^F, ?\} \\
\psi(T, T, T) &\iff A_{TTT} \Rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, L_3, \Psi^F} \{R_3^T, L_4^F, R_4^F, \dots, L_n^F, R_n^F, ?\}
\end{aligned}$$

For example, if $n = 4$, the final statement says that $\psi(T, F, T)$ is true if and only if, starting from any configuration \mathbf{x} with $X_1^T, R_1^F, L_1, X_2^F, L_2^F, R_2, L_3, \Psi^F$, (i.e., any configuration in A_{TTF} without R_3^T) it is possible to produce R_3^T , with the following species present the whole time: $X_1^T, X_2^T, R_1^F, R_2^F, \Psi^F$, and whatever were the states of the following in \mathbf{x} also preserved the whole time: L_1, L_2, L_3 . If we interpret T, F, T to be a length-3 path in the tree, it's saying that since the final value representing x_3 is T , we can produce R_3^T , without disturbing L_3 (whatever its state), and without disturbing any other *sibling* species L_1, R_2 farther *up* in the tree. But direct *ancestor* species R_1^F, L_2^F are fixed to state F the entire time, and match the assignment X_1^T, X_2^F (and those ancestor assignment species are also fixed).

Proof. The proof is by induction on the number $n - k$ of non-fixed variables in ψ .

Base case $k = n - 1$: We have some assignment $x_i = b_i \in \{F, T\}$ for each $i \in \{1, \dots, n - 1\}$. Assume for the sake of concreteness that each $b_i = T$ (the other cases are similar). In this case the lemma statement is

$$\psi(T, \dots, T, F) \iff A_{T, \dots, T} \Rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_{n-1}^T, R_{n-1}^F, L_{n-1}, R_n, \Psi^F} \{L_n^T, ?\}$$

and

$$\psi(T, \dots, T, T) \iff A_{T, \dots, T} \Rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_{n-1}^T, R_{n-1}^F, L_{n-1}, L_n, \Psi^F} \{R_n^T, ?\}.$$

We show just the first statement; the other is symmetric. Assume $\psi(T, \dots, T, F) = \phi(T, \dots, T, F)$ is true. Let $\mathbf{x} \in A_{T, \dots, T}$, so that $\mathbf{x}(X_i^T) = 1$ for all $i \in \{1, \dots, n-1\}$, and $\mathbf{x}(C_1^F) = 1$.

Apply assignment reaction $X_n^F \xleftrightarrow{C_1^T} X_n^T$ to create X_n^F if it is not already present in \mathbf{x} .

Apply the clause evaluation reactions in order to create C_m^T ; since $\phi(T, \dots, T, F)$ is true and $X_1^T, \dots, X_{n-1}^T, X_n^F$ are present, this is possible.

Apply leaf evaluation reaction $L_n^F \xleftrightarrow{X_n^F + C_m^T} L_n^T$ to create L_1^T . (For the case of the second part of the lemma where the final variable is T , instead apply $R_n^F \xleftrightarrow{X_n^T + C_m^T} R_n^T$ to create R_1^T .)

Apply the clause evaluation reactions in reverse to create C_1^F .

This configuration is contained in $A_{T, \dots, T, F}$ and has a L_1^T , so the lemma holds for the base case.

Inductive case $k < n - 1$:

As in the base case, for the sake of concreteness but without (much) loss of generality, we choose a particular assignment of $x_i = b_i = T$ to the first k variables x_1, \dots, x_k .

(\implies): For $b_1 = \dots = b_k = T$, assume $\psi(b_1, \dots, b_k)$ is true. We need to show

$$A_{b_1, \dots, b_{k-1}} \implies_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_{k-1}^T, R_{k-1}^F, L_{k-1}, L_k, \Psi^F} \{R_k^T, ?\}.$$

(If $b_k = F$ instead of T , then we would have to show the same with R_k in place of L_k and L_k^T in place of R_k^T above, but the proof is symmetric.)

Assume x_{k+1} is a \forall variable; the \exists case is similar (and slightly simpler since we only need to produce one of L_{k+1}^T or R_{k+1}^T below). Then $\psi(b_1, \dots, b_k, F)$ and $\psi(b_1, \dots, b_k, T)$ are both true. Then inductively we have

$$\psi(b_1, \dots, b_k, F) \iff A_{b_1, \dots, b_k} \implies_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_k^T, R_k^F, L_k, R_{k+1}, \Psi^F} \{L_{k+1}^T, ?\}$$

and

$$\psi(b_1, \dots, b_k, T) \iff A_{b_1, \dots, b_k} \implies_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_k^T, R_k^F, L_k, L_{k+1}, \Psi^F} \{R_{k+1}^T, ?\}.$$

To begin, let $\mathbf{x} \in A_{b_1, \dots, b_{k-1}}$. If R_k^T is present already in \mathbf{x} then we are done. So assume R_k^F is present. Now, \mathbf{x} already has X_1^T, \dots, X_{k-1}^T , and if necessary can immediately

change the state of X_k to T via assignment reaction $X_k^F \xleftarrow{C_1^F + L_{k+1}^F + R_{k+1}^F + \dots + L_n^F + R_n^F} X_k^T$, since those catalysts are by definition present in $A_{b_1, \dots, b_{k-1}}$. Then after producing X_k^T , the resulting configuration is in A_{b_1, \dots, b_k} (since we assumed R_k^F is present, and the other requirements of A_{b_1, \dots, b_k} follow from the species present in \mathbf{x} due to its being in $A_{b_1, \dots, b_{k-1}}$). Neither R_{k+1}^T nor L_{k+1}^T is present since we started in $\mathbf{x} \in A_{b_1, \dots, b_{k-1}}$; the goal is to produce each of them in sequence.

Produce L_{k+1}^T as follows. Since $\psi(b_1, \dots, b_k, F)$ is true and we've reached a configuration in A_{b_1, \dots, b_k} , by induction L_{k+1}^T is producible via

$$A_{b_1, \dots, b_k} \Rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_k^T, L_k^F, L_k, R_{k+1}, \Psi^F} \{L_{k+1}^T, ?\}$$

Apply reset reactions to set all L_i, R_i to state F for $i \geq k+2$. Then we are again in a configuration in A_{b_1, \dots, b_k} . Since $\psi(b_1, \dots, b_k, T)$ is true, by induction R_{k+1}^T is producible via

$$A_{b_1, \dots, b_k} \Rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_k^T, L_k^F, L_k, L_{k+1}, \Psi^F} \{R_{k+1}^T, ?\}.$$

Since L_{k+1} is a subscript in the reachability constraint, its value L_{k+1}^T that we created in the first part is preserved. Thus both L_{k+1}^T and R_{k+1}^T are now present.

Finally apply the \forall internal evaluation reaction $R_k^F \xleftarrow{X_k^T + L_{k+1}^T + R_{k+1}^T} R_k^T$ to create R_k^T .

(\Leftarrow): For $b_1 = \dots = b_k = T$, assume $\psi(b_1, \dots, b_k)$ is false. Then at least one of $\psi(b_1, \dots, b_k, F)$ or $\psi(b_1, \dots, b_k, T)$ is false; assume without loss of generality that it is $\psi(b_1, \dots, b_k, F)$. We need to show

$$A_{b_1, \dots, b_{k-1}} \not\Rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_{k-1}^T, R_{k-1}^F, L_{k-1}, L_k, \Psi^F} \{R_k^T, ?\}.$$

Inductively we have that

$$A_{b_1, \dots, b_k} \not\Rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_k^T, R_k^F, L_k, R_{k+1}, \Psi^F} \{L_{k+1}^T, ?\}$$

We claim that the following cannot occur: R_k becomes R_k^T while X_{k-1}^F is present, and subsequently X_{k-1}^F changes to X_{k-1}^T : The assignment reactions disallow X_{k-1} to change while R_k^T is present.

So assume R_k^F becomes R_k^T while X_{k-1}^T is present. We examine the first time that this occurs, so that R_k^F was fixed before this point. That is, we are holding fixed $\Psi^F, R_1^F, \dots, R_k^F$. We examine which reactions might change R_k to state T for the first time. The leaf evaluation reactions do not apply since $k < n$.

We assumed x_k is a \forall variable, and so we look at the \forall internal evaluation reaction $R_k^F \xleftarrow{X_k^T + L_{k+1}^T + R_{k+1}^T} R_k^T$. It requires X_k^F and L_{k+1}^T , but since we've shown that X_1^T, \dots, X_{k-1}^T are present as well, this violates the inductive hypothesis. Similar reasoning applies to the case of an \exists variable, but using two induction hypothesis based on the fact that both $\psi(b_1, \dots, b_k, F)$ and $\psi(b_1, \dots, b_k, T)$ are false.

For $i < k$, the reset reactions $R_k^F \xleftarrow{L_i^T + X_i^F} R_k^T$ and $R_k^F \xleftarrow{R_i^F + X_i^T} R_k^T$ cannot occur because X_i^T and R_i^F are present. (These facts are more complex to state but still true if we assume the assignment to x_1, \dots, x_{k-1} is something other than $TT \dots T$.)

The scrambling reaction $R_k^F \xleftrightarrow{\Psi^T} R_k^T$ cannot occur because Ψ^F is present.

Thus $A_{b_1, \dots, b_{k-1}} \not\rightarrow_{X_1^T, R_1^F, L_1, X_2^T, R_2^F, L_2, \dots, X_{k-1}^T, R_{k-1}^F, L_{k-1}, L_k, \Psi^F} \{R_k^T, ?\}$. □

4.4 Singular ≤ 3 -catalytic CRNs can simulate singular catalytic CRNs

The main result of this section is Theorem 45. Section 4.3.1 contains the formal definition of CRN simulation employed in the theorem statement.

4.4.1 Simulation construction

Although this paper applies the following theorem only to the reversible, binary, 1-safe (all species counts ≤ 1) CRN \mathcal{N} of Theorem 40, the theorem also applies to CRNs \mathcal{N} with larger counts (although the simulating CRN \mathcal{S} has extra species not in \mathcal{N} , whose counts are required to start at 0 or 1), and that may be irreversible or non-binary.

Intuitively, the construction works as follows. Since \mathcal{N} is singular, it suffices to test that each catalyst of a reaction is present—no need to test exact counts—since no reaction requires two or more species of the same type to execute. For each of \mathcal{N} 's reactions $X \xrightarrow{C_1 + \dots + C_k} Y$, we need to test whether C_1, \dots, C_k are present using a sequence of 3-catalytic reactions. The difficulty is that, supposing C_1 is present but C_4 is absent, a

naïve implementation could get partway through the test and verify that C_1, C_2, C_3 are present, but perhaps another pathway in the network allows $C_1 \rightarrow C_4$. Once C_4 is present, the sequence can continue, although C_1 and C_4 were not present *simultaneously*, as required for $X \xrightarrow{C_1+\dots+C_k} Y$ to be applicable. The key is to use “mutexes” to lock other simulated reactions while testing for the presence of $C_1 + \dots + C_k$, so that they cannot appear and disappear during the test.

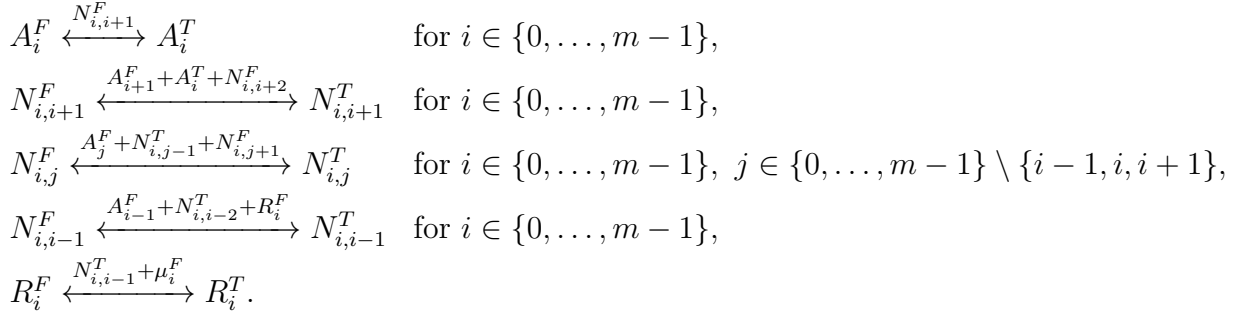
For intuition, first consider a simpler 1-catalytic, but non-binary, construction. Start with a single leader species $L_{0,0}$. If \mathcal{N} has reactions $\alpha_0, \dots, \alpha_{m-1}$, then add to \mathcal{S} reactions $L_{i,0} \leftrightarrow L_{i',0}$ for all $0 \leq i \neq i' \leq m-1$. From state $L_{i,0}$, the leader can sequentially test for the catalysts in reaction $\alpha_i : X \xrightarrow{C_1+\dots+C_k} Y$ via $L_{i,j-1} \xleftrightarrow{C_j} L_{i,j}$ for $1 \leq j \leq k$. Reaction $X \xrightarrow{L_{i,k}} Y$ completes the simulation of α_i . Since the test reactions are reversible, the leader can reverse through them back to $L_{i,0}$ to simulate another reaction. The challenge is to achieve something similar without a single leader that can take on more than two states, and furthermore to obey the constraint of catalytic non-competitiveness.

Each species $X \in \Lambda_{\mathcal{N}}$ is represented by the same species $X \in \Lambda_{\mathcal{S}}$. All species below not explicitly identified as being in $\Lambda_{\mathcal{N}}$ are new species in $\Lambda_{\mathcal{S}}$. For each reaction $\alpha_0, \dots, \alpha_{m-1} \in R_{\mathcal{N}}$, we introduce binary species R_i^F, R_i^T for $i \in \{0, \dots, m-1\}$. In the initial configuration of \mathcal{S} , we have one copy of each R_i^F , and we ensure that at most one R_i^T exists in any reachable configuration. When R_i^T is present, this will allow a sequence of reactions in \mathcal{S} that test for the catalysts of $\alpha_i : X \xrightarrow{C_1+\dots+C_k} Y$.

There will be two main stages in simulating any reaction α_i : we first *activate* α_i , which lets us produce the mutex μ_i^T , and then we *test* for the presence of the catalysts of α_i . We want to make a random choice of which reaction α_i to activate, but if one reaction is activated, we want to shut down the possibility of activating any other reaction, to stop from possibly changing the count of C_1, \dots, C_k (or any other species in $\Lambda_{\mathcal{N}}$) during the sequential test for their presence. This chain of events is illustrated in Fig. 4.2.

To simplify specification, we assume in the following reactions that all addition is mod m , i.e., if $i = m-1$, then $i+1 = 0$, and if $i = 0$, then $i-1 = m-1$. Start with species A_i^F for each $i \in \{0, \dots, m-1\}$ and $N_{i,j}^F$ for each $i \in \{0, \dots, m-1\}$ and $j \in \{0, \dots, m-1\} \setminus \{i\}$.

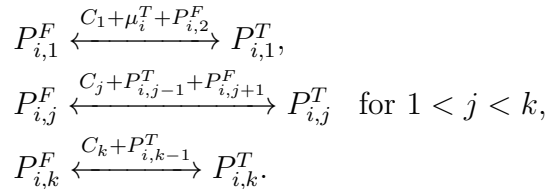
Intuitively, A_i^T means “the CRN is currently attempting to activate reaction α_i ”, and $N_{i,j}^T$ additionally means “and the CRN is not currently attempting to activate any reactions $\alpha_{i+1}, \dots, \alpha_j$.” For example, if $m = 10$ and $N_{7,1}^T$ and $N_{3,5}^T$ are present (which requires A_7^T and A_3^T to be present), the CRN is attempting to activate α_7 and α_3 and not attempting to activate $\alpha_8, \alpha_9, \alpha_0$, or α_1 (due to $N_{7,1}^T$), nor α_4 or α_5 (due to $N_{3,5}^T$). The species R_i^T will mean “reaction α_i has successfully been activated”. (Once α_i is activated, the CRN can produce the mutex μ_i^T .) Add the following **activation** reactions:



Once any A_i^T is present, for every $j \neq i$, some reaction in the second, third, or fourth line above will prevent $N_{j,i}^T$ from being produced. $N_{j,j-1}^T$ can only be produced if all previous (in the circular order mod m) $N_{j,i}^T$ are produced; this holds by similar reasoning to the test reactions described below. So at most one $N_{i,i-1}^T$, therefore at most one R_i^T , are present.

Once reaction α_i is activated, we can *acquire* its mutex. Add the **mutex** reaction $\mu_i^F \xleftarrow{R_i^T + P_{i,1}^F} \mu_i^T$. The presence of the “mutex” μ_i^T means that we are currently simulating reaction α_i (i.e., testing for the presence of its catalysts); we will see below what the other catalyst $P_{i,1}^F$ accomplishes.⁵

Now we explain how to use μ_i^T to test sequentially for the presence of catalysts C_1, \dots, C_k of reaction $\alpha_i \in R_{\mathcal{N}}$. Add the following **catalyst test** reactions:



⁵The mutex reaction is not strictly necessary, but if we removed it, we would have to add catalysts from the activation reactions to the catalyst test reactions, and vice versa. The mutex reaction serves as a “buffer” to help cleanly and separately understand the activation and catalyst test reactions.

These reactions ensure that, if C_l is the first catalyst *not* present, then we can set $P_{i,j}$ to state T for all $1 \leq j \leq l - 1$, but not for any $j \geq l$. In other words, $P_{i,j}^T$ represents “ C_1, \dots, C_j are all present.” The additional catalysts $P_{i,j-1}^T$ and $P_{i,j+1}^F$ ensure monotonicity: for all $1 \leq j < j' \leq k$, $P_{i,j'}^T$ implies $P_{i,j}^T$. Since all reactions are reversible, if some catalyst C_1, \dots, C_k of α_i is missing, the sequence of test reactions can “back out” by setting all $P_{i,j}$ to state F , which allows us to reverse the initial mutex reaction $\mu_i^F \xleftarrow{R_i + P_{i,1}^F} \mu_i^T$, restoring μ_i^F , and finally reversing the activation reactions, allowing the random choice of reaction activation to continue. The monotonicity of states T along the $P_{i,j}$ ’s ensures that this cannot happen unless *all* $P_{i,j}$ are state F . Thus all catalysts must be present when first setting μ_i to T in the mutex reaction in order to reach the end of α_i ’s test reactions and set $P_{i,k}$ to T .

Thus, if and only if all catalysts C_1, \dots, C_k are present, $P_{i,k}^T$ is producible. The **commit** reaction $X \xrightarrow{P_{i,k}^T} Y$ completes the simulation of \mathcal{N} ’s reaction $\alpha_i : X \xrightarrow{C_1 + \dots + C_k} Y$. If α_i is reversible, then use $X \xleftrightarrow{P_{i,k}^T} Y$ instead, to ensure \mathcal{S} is reversible if \mathcal{N} is.

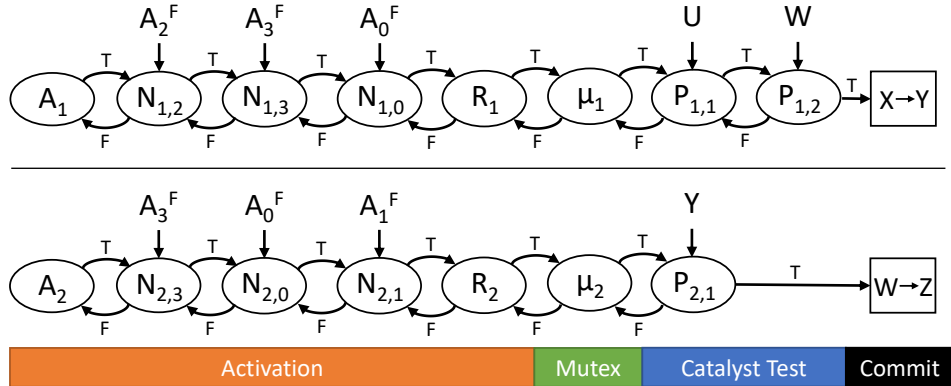


Figure 4.2: The figure shows an example of a simulation with four formal reactions, with simulation paths shown for reactions 1 and 2. Each bubble in the figure corresponds to a binary species in the simulation; the species can only change states if the species pointing into it are present (e.g., there are reactions $A_1^F \xleftrightarrow{N_{1,2}^F} A_1^T$ and $N_{1,2}^F \xleftrightarrow{A_1^T, A_2^F, N_{1,3}^F} N_{1,2}^T$). To simulate formal reaction 1, $X \xrightarrow{U, W} Y$ (top of figure), a chain of dependencies must be resolved, beginning from activation attempt A_1 and ending with the commit reaction $X \xrightarrow{P_{1,2}^T} Y$. To simulate formal reaction 2, $W \xrightarrow{Y} Z$ (bottom of figure), the chain begins with activation attempt A_2 and ends with commit reaction $W \xrightarrow{P_{2,1}^T} Z$. The activation reactions ensure that only one R_i can be true at any given time (and hence, at any given time only one chain can potentially reach the commit reaction). The mutex reaction ensures that R_i remains true while the rest of the chain proceeds. The catalyst test reactions then check for the existence of the correct catalysts before allowing the commit reaction to occur.

The CRN is catalytically non-competitive, even if the simulated CRN is not. This is because the single-catalyst commit reactions of \mathcal{S} simulate the original reactions of \mathcal{N} , which by inspection are the only places where a non-catalytic reactant can appear in multiple reactions. The initial configuration of \mathcal{S} is that of \mathcal{N} , plus one copy each of the F states of all new species described above.

Theorem 45. *For every singular, catalytic CRN \mathcal{N} , there is a singular, ≤ 3 -catalytic, catalytically non-competitive CRN \mathcal{S} that simulates \mathcal{N} . Furthermore, if \mathcal{N} is reversible, then \mathcal{S} is reversible, and if \mathcal{N} is binary, then \mathcal{S} is binary.*

Proof. The construction is described and explained intuitively above. Here we explain how it satisfies the conditions of Definition 42 to justify that \mathcal{S} simulates \mathcal{N} .

sound: The commit reactions are the only implementation reactions that correspond to formal reactions. Let $\mathbf{c}_{\mathcal{S}} \in \mathbb{N}^{\Lambda_{\mathcal{S}}}$ be a reachable configuration to which a commit reaction is applicable. Since $\mathbf{c}_{\mathcal{S}}$ is reachable, it has count one of each intermediate species (each in either the F or T state). Thus applying $\alpha_{\mathcal{S}}$ to $\mathbf{c}_{\mathcal{S}} \in \mathbb{N}^{\Lambda_{\mathcal{S}}}$ results in implementation configuration $\mathbf{d}_{\mathcal{S}} \in \mathbb{N}^{\Lambda_{\mathcal{S}}}$, i.e., $\mathbf{c}_{\mathcal{S}} \xrightarrow{\alpha_{\mathcal{S}}} \mathbf{d}_{\mathcal{S}}$, with one fewer X and one more Y . As argued in the construction, if a commit reaction $\alpha_{\mathcal{S}} : X \xrightarrow{P_{i,k}^T} Y$ is applicable, then all formal catalysts C_1, \dots, C_k are present. Thus formal reaction $\alpha_{\mathcal{N}} : X \xrightarrow{C_1, \dots, C_k} Y$ is applicable to formal configuration $\mathbf{c}_{\mathcal{N}} \in \mathbb{N}^{\Lambda_{\mathcal{N}}}$, where $\mathbf{c}_{\mathcal{N}} = m(\mathbf{c}_{\mathcal{S}})$. Applying it results in a formal configuration $\mathbf{d}_{\mathcal{N}} \in \mathbb{N}^{\Lambda_{\mathcal{N}}}$ with one fewer X and one fewer Y . Since $m(\mathbf{c}_{\mathcal{S}}) = \mathbf{c}_{\mathcal{N}}$ and $m(\mathbf{d}_{\mathcal{S}}) = \mathbf{d}_{\mathcal{N}}$, this means $\mathbf{c}_{\mathcal{N}} \xrightarrow{\alpha_{\mathcal{N}}} \mathbf{d}_{\mathcal{N}}$. This establishes the soundness condition.

complete: Let $\mathbf{c}_{\mathcal{S}} \in \mathbb{N}^{\Lambda_{\mathcal{S}}}$ be a reachable implementation configuration. Since $\mathbf{c}_{\mathcal{S}}$ is reachable, it has count one of each intermediate species (each in either the F or T state). Let $\alpha_i : (X \xrightarrow{C_1 + \dots + C_k} Y) \in \mathcal{R}_{\mathcal{N}}$ be a formal reaction.

Some steps explained below may be skipped if they are only used to generate an intermediate species that is already present.

- If $\mu_{i'}^T$ is present for $i' \neq i$, execute the test reactions in reverse until $P_{i',0}^F$ is present.

- Execute the mutex reaction $\mu_{i'}^F \xleftarrow{R_{i'}+P_{i',0}^F} \mu_{i'}^T$ in reverse to generate $\mu_{i'}^F$.
- Execute the activation reactions in reverse until all species changed by the activation reactions are in state F .
- Execute the activation reactions for α_i forward until R_i^T is present.
- Execute the mutex reaction $\mu_i^F \xleftarrow{R_i+P_{i,0}^F} \mu_i^T$ forward to generate μ_i^T .
- Execute the catalyst test reactions forward in order until $P_{i,k}^T$ is present.
- Execute the commit reaction $X \xrightarrow{P_{i,k}^T} Y$.

All of these are silent reactions but the last. They reach configuration $\mathbf{d}_S \in \mathbb{N}^{\Lambda_S}$ such that $m(\mathbf{d}_S) = \mathbf{d}_N$, where \mathbf{d}_N has one fewer X and one more Y than $m(\mathbf{c}_S)$. This sequence of reactions shows that $\mathbf{c}_S \xrightarrow{\alpha_N} \mathbf{d}_S$, satisfying the completeness condition.

□

4.5 TBNs can simulate (restricted) CRNs

The main result of this section is Theorem 48, showing that TBNs can simulate a restricted class of CRNs, which have a PSPACE-complete reachability problem by Corollary 43. The rest of this section is devoted to proving Theorem 48.

4.5.1 Thermodynamic Binding Networks Model

A *multiset* is an unordered collection of objects allowing duplicates, e.g., $\mathbf{v} = \{2 \cdot a, b, 5 \cdot d\}$. Equivalently, a multiset with elements from a finite set U is a vector $\mathbf{v} \in (\mathbb{N} \cup \{\infty\})^U$ describing the counts, indexed by U ; in the example above, if $U = \{a, b, c, d\}$, then $\mathbf{v}(a) = 2$, $\mathbf{v}(b) = 1$, $\mathbf{v}(c) = 0$, and $\mathbf{v}(d) = 5$. The *cardinality* of a multiset $\mathbf{v} \in \mathbb{N}^U$ is $|\mathbf{v}| = \sum_{u \in U} \mathbf{v}(u)$; a *finite multiset* \mathbf{v} obeys $|\mathbf{v}| < \infty$. A *site type* is a formal symbol, such as a , representing a specific binding site on a molecule; in Fig. 1.1 the site types are a, a^*, b, b^* . Each site type has a corresponding *complement type* which is denoted by a star: e.g. a^* . Complementarity is an involution: i.e. $(a^*)^* = a$. A site and its complement can form an attachment called a *bond*. We follow the convention that for any complementary pair of sites a, a^* , the total count of a^* across the whole TBN is at most that of a , i.e.,

the starred sites are *limiting*. A *monomer type* is a finite multiset of site types. When context implies a single instance of a monomer/site type, we may interchangeably use the term *monomer/site*.⁶

A *thermodynamic binding network (TBN)* is a multiset of monomer types; equivalently, a vector in \mathbb{N}^m , if m is the number of monomer types and we have fixed some standardized ordering of them. We allow some monomer counts to be infinite in order to capture the case where some monomers are added in “large excess” over others, a common experimental approach [47, 45]. A *polymer* (alt. *complex*) is a finite multiset of monomer types, equivalently, a vector in \mathbb{N}^m , where m is the number of monomer types in a standardized ordering.⁷ Note that despite the suggestive lines representing bonds in Fig. 1.1, this definition does not track which pairs of complementary sites are bound within a polymer.

The *exposed sites* of a polymer are a finite multiset of site types that results from removing as many (site, complement) pairs from a polymer as possible. For example, in the polymer $\{\{a^*, b^*\}, \{a, c\}, \{a, b, c\}\}$, the exposed sites are $\{a, 2 \cdot c\}$.

A *configuration* of a TBN is a partition of the TBN into polymers. A polymer is *self-saturated* if it has no exposed starred sites. A configuration is *saturated* if all of its polymers are self-saturated; since we assume starred sites are limiting, this is equivalent to stipulating that the maximum number of bonds are formed.

The *entropy of a configuration* is the number of polymers within a configuration, representing the energetic contribution of many complexes in a system. A configuration is *stable* if it is saturated and has maximum entropy⁸.

We allow saturated configurations to transition from one to another via *paths*. These

⁶Concretely, in a DNA nanotech design, a monomer corresponds to a strand of DNA, whose sequence is logically partitioned into binding sites corresponding to long (5-20 base) regions, e.g., 5'-AAAGG-3', intended to bind to the Watson-Crick complementary sequence 3'-TTTCC-5' that is part of another strand.

⁷The term “polymer” is chosen to convey the concept of combining many atomic objects into a complex, but it is not necessarily a linear chain of repeated units.

⁸Note that as we only consider saturated configurations, the enthalpic contribution of bonds between sites is constant for each TBN and can be omitted from relative energy calculations here; for a more detailed analysis of energetic contributions and a proof of why saturated configurations are sufficient for consideration, see [10].

paths consist of the sequence of configurations reachable consecutively by applying two types of atomic operations: a *merge* in which two polymers join and become the union polymer, and a *split* in which a single polymer is partitioned into two polymers so long as the overall configuration remains saturated.

The *barrier of a path* is the largest difference in entropy between two configurations β and γ in the path, so long as γ occurs later in the path than β . We will generally consider barriers of paths that originate from a stable state, and in such cases it is sufficient to take the largest entropy difference between the starting configuration and any intermediate configuration.

The *energy barrier between configurations* is the smallest possible barrier of any pathway between the two configurations.

4.5.2 Simulation of CRNs

To show that TBNs can (weakly bi-)simulate binary, reversible, catalytic, singular, and catalytically non-competitive CRNs, we will use similar soundness and completeness definitions as were used in Definition 41 and Definition 42, but now applicable to TBNs.

Definition 46 (formal CRNs and implementing TBNs, interpretations). *Let \mathcal{T} be a TBN and let $\mathcal{N} = (\Lambda_{\mathcal{N}}, \mathcal{R}_{\mathcal{N}})$. Let $B \in \mathbb{N}$ and let $\Gamma_{\mathcal{T}}^B$ denote the set of saturated configurations of \mathcal{T} that are reachable from some stable configuration of \mathcal{T} by a path with barrier at most B . We refer to B as the global barrier. We refer to configurations of \mathcal{N} as formal configurations (of the CRN) and to configurations in $\Gamma_{\mathcal{T}}^B$ as the implementation configurations (of the TBN). Let $m : \Gamma_{\mathcal{T}}^B \rightarrow \mathbb{N}^{\Lambda_{\mathcal{N}}}$ be a surjective function mapping every TBN configuration in $\Gamma_{\mathcal{T}}^B$ to a corresponding CRN configuration in $\mathbb{N}^{\Lambda_{\mathcal{N}}}$. We refer to elements in the range of m as the interpretation of elements in the pre-image. We consider \mathcal{T} subject to global barrier B to be simulating \mathcal{N} .*

If a merge or a split can take a configuration $\beta \in \Gamma_{\mathcal{T}}^B$ to a configuration $\gamma \in \Gamma_{\mathcal{T}}^B$, we write $\beta \rightarrow \gamma$. Furthermore, the merge or split is *silent* if $m(\beta) = m(\gamma)$, i.e., it does not change the interpretation. If $\alpha_{\mathcal{N}} \in \mathcal{R}_{\mathcal{N}}$ is a formal reaction and $\beta, \gamma \in \Gamma_{\mathcal{T}}^B$ are implementation configurations, we write $\beta \overset{\alpha_{\mathcal{N}}}{\rightsquigarrow} \gamma$ to denote that, for some (non-silent) merge or split, $\beta \rightarrow \gamma$

and $m(\beta) \xrightarrow{\alpha_{\mathcal{N}}} m(\gamma)$.

We write $\beta \xrightarrow{\perp} \gamma$ to denote that γ is reachable from β by a sequence of silent merges and splits. We write $\beta \xrightarrow{\alpha_{\mathcal{N}}} \gamma$ to denote that $\beta \xrightarrow{\perp} \beta' \xrightarrow{\alpha_{\mathcal{N}}} \gamma' \xrightarrow{\perp} \gamma$, i.e., a sequence of merges and splits takes $\mathbf{c}_{\mathcal{S}}$ to $\mathbf{d}_{\mathcal{S}}$, in one operation the interpretation changes as it would in $\alpha_{\mathcal{N}}$, and the rest of the operations are silent.

Definition 47 (CRN simulation by TBN). *Let \mathcal{T} be a TBN, $\mathcal{N} = (\Lambda_{\mathcal{N}}, \mathcal{R}_{\mathcal{N}})$ be a CRN, and let $B \in \mathbb{N}$, as in Definition 46. Say \mathcal{T} subject to global barrier B simulates (a.k.a., weakly bisimulates) \mathcal{N} if the following conditions hold:*

sound: *For every implementation configuration $\beta \in \Gamma_{\mathcal{T}}^B$, if an operation $\beta \rightarrow \gamma$ is not silent, then for some $\alpha_{\mathcal{N}} \in \mathcal{R}_{\mathcal{N}}$, $\beta \xrightarrow{\alpha_{\mathcal{N}}} \gamma$ and $m(\beta) \xrightarrow{\alpha_{\mathcal{N}}} m(\gamma)$. (If a non-silent transition occurs in β , then it “looks like” a formal reaction $\alpha_{\mathcal{N}}$, and $\alpha_{\mathcal{N}}$ is applicable to the formal configuration $m(\beta)$, and goes to the expected formal configuration $m(\gamma)$.)*

complete: *For every implementation configuration $\beta \in \Gamma_{\mathcal{T}}^B$, if $m(\beta) \xrightarrow{\alpha_{\mathcal{N}}} m(\gamma)$ for some $\alpha_{\mathcal{N}} \in \mathcal{R}_{\mathcal{N}}$, then $\beta \xrightarrow{\alpha_{\mathcal{N}}} \gamma$. (If $\alpha_{\mathcal{N}}$ can occur in formal configuration $m(\beta)$, then it can also be simulated in any configuration with the same interpretation. Note that since m is surjective, this implies that all formal reactions are simulated.)*

Theorem 48. *Let \mathcal{N} be a CRN that is binary, reversible, $\leq k$ -catalytic, singular, and catalytically non-competitive. Then for each $w \in \mathbb{N}$, $w \geq 2$, there is a TBN \mathcal{T} so that $\forall B \in \mathbb{N}$, $k \leq B \leq k + w - 2$, \mathcal{T} subject to global barrier B simulates \mathcal{N} .*

Proof. Note that if \mathcal{N} is ≤ 0 -catalytic or ≤ 1 -catalytic, then it is ≤ 2 -catalytic, so we may assume without loss of generality that $k \geq 2$. The statement then follows from the proofs of soundness (Lemma 52) and completeness (Lemma 56). \square

Setting $w = ck - k + 2$, any factor- c approximation algorithm computing TBN energy barriers could solve reachability for such CRNs, which have a PSPACE-complete reachability problem by Corollary 43, implying the main result of this paper:

Corollary 49. *It is PSPACE-complete to approximate within any constant factor the energy barrier between two configurations of a TBN.*

4.5.3 Grid gate

To demonstrate how CRNs satisfying the constraints of Corollary 43 can be implemented within the TBN model, we modify the construction of the “grid gate” discussed by Breik, et al. in [10]. That paper shows that for a given parameter n , a TBN can be constructed using n^2 site types and $2n+1$ monomers such that there are two stable configurations, and the barrier to transition between the configurations is n . An optional “catalyst” monomer can be added, which reduces this transition barrier to 1.

The grid gate is shown in Fig. 4.3. A monomer G forms the backbone of the arrangement, containing all of the starred sites (each one of the n^2 unique site types). The unstarred sites are divided into rows (alt. columns) to form the H_i (alt. V_i) monomers. In order to *cover* G (that is, bind each starred site on G in order to form a self-saturated polymer), it is necessary to either have all of the H monomers in a complex with G (denoted G_H), or to have all of the V monomers in a complex with G (denoted G_V). To transition between G_H and G_V requires an intermediate configuration in which all of the monomers are merged into a single complex (hence, barrier of n). However, in the presence of the catalyst C , a pathway of alternating merges and splits has a barrier of 1 (see Fig. 4.4). The bi-stable, reversible, and catalytic nature of this system is the basis for the binary, reversible, and catalytic constraints on CRNs simulated in Theorem 48.

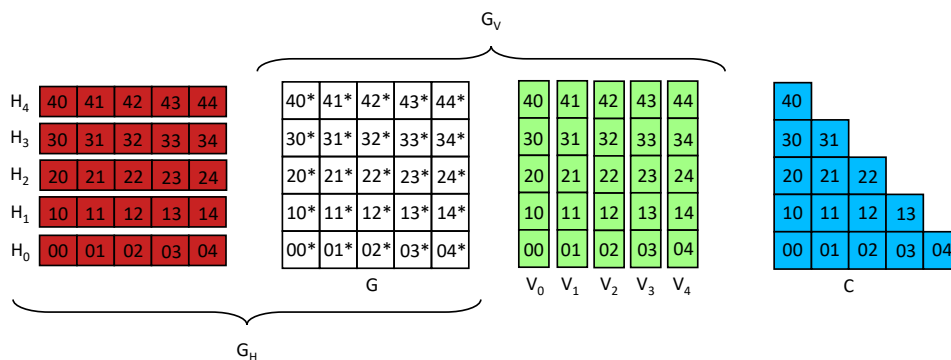


Figure 4.3: Example of a grid gate with $n = 5$. G_H is formed by taking the G and H monomers together; G_V by the G and V monomers together. G_H and G_V each contain $n + 1$ monomers, but to transition between G_H and G_V requires an intermediate configuration in which $2n + 1$ of these monomers are brought together into a single polymer, creating an energy barrier of n . This barrier can be lowered down to 1 in the presence of a catalyst C (shown at right). The 2D depiction is for intuition only and does not imply a particular physical geometry of the monomers.

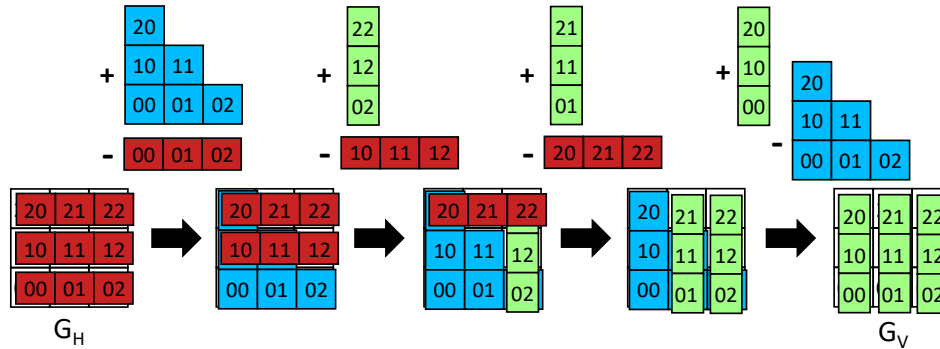


Figure 4.4: A catalyzed pathway with $n = 3$. The complex G_H (left) can transition through a series of merge/split pairs in order to become the complex G_V (right). The H monomers are shown in red (dark), the V monomers in green (light), and the catalyst C in blue (mid). At each stage in the pathway, one merge (shown with “+”) and one split (shown with “-”) occur without exposing any of the starred sites. In this way, all polymers remain self-saturated, and so each configuration in the pathway is saturated.

4.5.4 Modifications to grid gate for simulating CRNs

In our simulation we will use a TBN \mathcal{T} that is the union of many grid gates each with disjoint binding sites, and each grid will represent a different binary pair of formal species. We will refer to this TBN \mathcal{T} as an *amalgamated grid gate*. Let $\mathcal{N} = (\Lambda_{\mathcal{N}}, \mathcal{R}_{\mathcal{N}})$ be a binary, reversible, catalytic, singular, 1-safe CRN (that we will be simulating). For a given pair of formal species $\{Z^F, Z^T\} \subset \Lambda_{\mathcal{N}}$, let $\mathcal{T}|Z$ denote the Z module: a complete set of monomers $\{G, H_0, \dots, H_{n-1}, V_0, \dots, V_{n-1}\}$ as described in Section 4.5.3 and illustrated in Fig. 4.3 (without C), with domains disjoint from all other modules. \mathcal{T} is then constructed from the union of the modules for all formal species pairs. Note that our CRN is both singular and 1-safe, and so in the TBN simulation there is exactly one module for each formal species.

Within the context of a TBN configuration, the *configuration of a module* is the configuration of the TBN when restricted to the monomers of the module. For each formal (binary) species in the CRN, the corresponding module will represent false when it is in its H -configuration: $\{G_H, \{V_0\}, \dots, \{V_{n-1}\}\}$, and represent true when it is in its V -configuration: $\{G_V, \{H_0\}, \dots, \{H_{n-1}\}\}$. A *base configuration* is one in which no two monomers from separate modules are in the same polymer, and every module is either in its H -configuration or its V -configuration. Note that the base configurations are stable, and that there is a unique base configuration for every interpretation.

We allow the modules to interact through the following mechanism: rather than add an explicit catalyst C monomer directly as shown in Figs. 4.3 and 4.4, we distribute the sites of C (i.e. the *catalytic region*) as “overhangs” to monomers in a different module (see Fig. 4.5, left). For instance, to simulate $X^F \xrightarrow{A^F} X^T$, we create a grid module X and add its catalytic region as overhangs to the H monomers of the module A . In this way, the catalytic region for X is only “brought together” if module A is in its H -configuration (representing A^F). To implement reactions requiring more than one catalyst, we distribute the catalytic region across several modules (see Fig. 4.5, right). The distribution achieves several properties that are helpful in proving correctness, e.g., each overhang has intersection ≤ 1 with each H and V monomer of the module it catalyzes. We do not directly implement uncatalyzed reactions $X^F \leftrightarrow X^T$, but instead add a new “fixed species” X_{cat} to the CRN that begins in and remains in its V -configuration (representing X_{cat}^T), so that the 1-catalytic reaction $X^F \xrightarrow{X_{\text{cat}}^T} X^T$ correctly implements the desired reaction.⁹

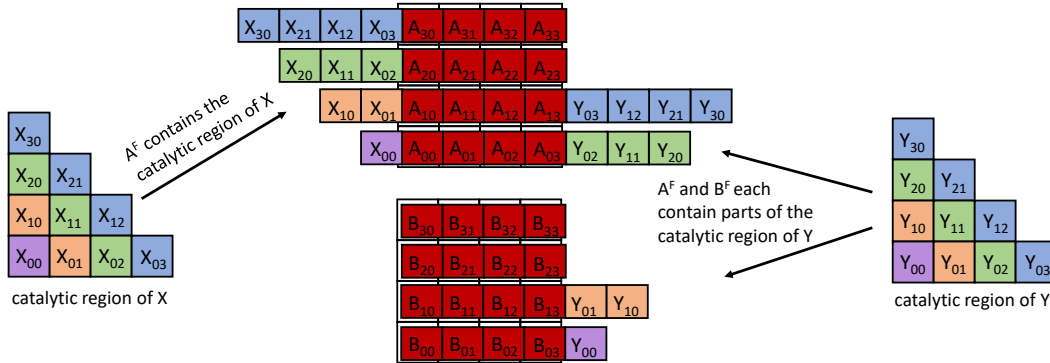


Figure 4.5: **Left:** an implementation of $X^F \xrightarrow{A^F} X^T$. The catalytic region of the grid representing X is distributed as “overhangs” (site types $X_{??}$) to the H monomers of the grid representing A , so that the catalyst is *formed* when A has all of its H monomers (representing A^F). **Right:** an implementation of $Y^F \xrightarrow{A^F+B^F} Y^T$ is shown. Here the overhangs (site types $Y_{??}$) are shared between the H monomers of the A and B grids, so that both A^F and B^F must cooperate to catalyze the transition of Y .

When a formal species is a catalyst in a formal reaction, we will say in the simulation that a catalyst has *formed* if its corresponding grid module is in the correct configuration

⁹We could equivalently simulate uncatalyzed reactions by adding an explicit monomer C to the system as in Fig. 4.3; however the proofs contained here are made more straightforward by not considering uncatalyzed reactions as a special case.

(H -configuration or V -configuration), as this will bring together the catalytic overhangs for that reaction (as shown in Fig. 4.5).

To formalize the construction, for a particular binary reaction, let k be the number of catalysts in the reaction. Choose $w \geq 2$ and let $n = kw$. Without loss of generality, let the catalysts be the G_H complexes of their respective modules (for one can replace each occurrence of H with V in the following discussion if required). Group the sites of the catalytic region based upon their L_1 distance from the lower-left corner (see Fig. 4.6). Distribute the sites of groups 0 to $w - 1$ as overhangs to monomers H_0, \dots, H_{w-1} in the module of the first catalyst (i.e. group 0 to monomer H_0 , group 1 to monomer H_1 , etc.) Distribute the next w groups (groups w through $2w - 1$) to monomers H_0, \dots, H_{w-1} in the second catalyst (i.e. group w to monomer H_0 , etc.). Continue in this fashion until all kw groups are distributed as overhangs to other modules.

This scheme can also implement many 1-catalytic reactions involving the same reactant (e.g. $X^F \xrightarrow{A^F} X^T$ and $X^F \xrightarrow{Z^T} X^T$). However, if overhangs are distributed in this fashion and there existed two competing reactions, say $X^F \xrightarrow{A^F+B^F} X^T$ and $X^F \xrightarrow{C^F+D^F} X^T$, then a new pathway would be inadvertently created: $X^F \xrightarrow{A^F+D^F} X^T$. For this reason, we will use Theorem 45, which shows that CRNs can be reduced to a form in which there is no such competition, i.e., the CRN is *catalytically non-competitive*.

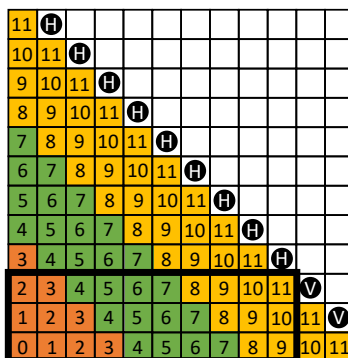


Figure 4.6: Example of the split catalyst scheme with $n = 12$, $w = 4$, and $k = 3$. The sites in the catalytic region are labelled with their groups (0 – 11) and colored according to which of the three catalysts that they will overhang. The off-diagonal sites are represented by filled circles and then marked with their representatives in an example HV-sequence $\{H_3, \dots, H_{11}, V_{11}, V_{10}\}$. The rectangle is the 3×10 portion not covered by the monomers of the HV-sequence.

Define the interpretation function m of a TBN configuration as follows: in the config-

uration of a particular module $\mathcal{T}|Z$, if G is in the same polymer as all of $H_{n-1}, \dots, H_{\lfloor n/2 \rfloor}$ (at least the “upper half” of the set of H s), then m maps that module to Z^F , else Z^T ; union the result over all modules. In this way every configuration can be interpreted, even if it is not (fully) in an H -configuration or V -configuration (for instance, if the configuration resembled one of the intermediate steps shown in Fig. 4.4).

We now proceed with a series of lemmas that serve to establish the two main conditions required of a simulation (stated in Definition 47 and required by Theorem 48): soundness (Lemma 52) and completeness (Lemma 56)

Lemma 50 below establishes a minimal set of monomers required in each module, and is used to prove the stronger Lemma 51 which states that all the modules are effectively in H -configurations, V -configurations, or else exist in a larger polymer that is in the process of simulating a formal reaction (and that the catalysts for the reaction are also in the same polymer).

Lemma 50. *Let $\beta \in \Gamma_{\mathcal{T}}^B$. In β , the configuration of each module $\mathcal{T}|Z$ must have one polymer that contains G and an HV -sequence: a set of $n - 1$ consecutive monomers of the form $\{H_t, \dots, H_{n-1}, V_{n-1}, \dots, V_{n-t+1}\} (1 \leq t \leq n)$.*

Proof. For contradiction, let us suppose that G is not in the same polymer with an HV -sequence. We consider the off-diagonal sites of the grid (black dots with letters in Fig. 4.6). Then for some $1 \leq i \leq n - 2$, H_i and V_{i+1} are not both present; and then the site in row i and column $i + 1$ would be exposed in configuration β . Then the polymer with G is not self-saturated, and so the configuration β is not saturated, hence not in $\Gamma_{\mathcal{T}}^B$, a contradiction. \square

Lemma 51. *Let \mathcal{T} be an amalgamated grid gate with parameter $w \geq 2$ that is constructed according to a binary, reversible, $\leq k$ -catalytic, singular, and catalytically non-competitive CRN $\mathcal{N} = (\Lambda_{\mathcal{N}}, \mathcal{R}_{\mathcal{N}})$ with $k \geq 2$. Let $B \leq k + w - 2$, and let $\beta \in \Gamma_{\mathcal{T}}^B$ be an implementation configuration as described in Definition 46. For every formal species $Z \in \Lambda_{\mathcal{N}}$, at least one of the following must be true:*

1. *In β , the configuration of $\mathcal{T}|Z$ has a polymer that contains G and all of the H monomers.*

2. In β , the configuration of $\mathcal{T}|Z$ has a polymer that contains G and all of the V monomers, and $Z^T \in m(\beta)$.
3. There is a reaction $\alpha_N \in \mathcal{R}_N$ of the form $Z^F \xleftrightarrow{K_1, \dots, K_t} Z^T$ ($t \geq 1$) and $\{K_1, \dots, K_t\} \subset m(\beta)$. Furthermore, in β , there is a single polymer that contains the G monomers of the modules $\mathcal{T}|K_1, \dots, \mathcal{T}|K_t$, and $\mathcal{T}|Z$.

Proof. If the module $\mathcal{T}|Z$ has a polymer that contains G and all of the V monomers, suppose for contradiction that $Z^T \notin m(\beta)$, i.e. $Z^F \in m(\beta)$. Then this polymer has all of the V s and $H_{n-1}, \dots, H_{\lfloor n/2 \rfloor}$. But in this case, the H 's were not necessary to cover G , and could have all been split from the polymer containing G , and so β has at least $\lfloor n/2 \rfloor \geq n/k = w \geq k + w - 1$ entropy less than another configuration in $\Gamma_{\mathcal{T}}^B$, hence certainly not reachable from a stable configuration within the global barrier of $B \leq k + w - 2$, therefore $\beta \notin \Gamma_{\mathcal{T}}^B$, a contradiction. Therefore if G is in a polymer with all of the V monomers, then $Z^T \in m(\beta)$.

The remainder of this proof then considers the case that G is not in a polymer with all of the H monomers, and also not in a polymer with all of the V monomers, and shows that condition 3 must apply (the module is *in transition* of some reaction α_N , and all related catalysts are both formed and in the same polymer as G).

Consider the G monomer of the $\mathcal{T}|Z$ module and how it is covered in the overall configuration β . By Lemma 50 there must be an HV -sequence – this will cover the off-diagonal sites (black dots with letters in Fig. 4.6).

Consider now the area left uncovered by the $n - 1$ monomers in the sequence just described. The uncovered area is a rectangle that contains n unique group labels. (Fig. 4.6 shows a 3×10 rectangle with group labels 0-11.) We now consider how to cover this rectangular region.

Beyond the HV -sequence, it is possible for these sites to be covered by additional H and V monomers (as up to this point we do not currently have any of the H s or V s that overlap the rectangle), but since we are under the assumption that we do not have all of the H s or all of the V s, merging optimal selections of H s and V s can at most increase the number of fully covered groups by 1 for each monomer recruited in this fashion. For

example, in Fig. 4.6, the only groups that can be covered by a single H or V monomer are of type 0, covered by H_0 or V_0 , or type 11, covered by H_2 or V_9 . (Note that H_1 would not fully cover any group, since its groups are $1, \dots, 10$, which exist elsewhere in the rectangle, and similarly V_1, \dots, V_8 would cover only group labels that have other copies in the rectangle.)

It is also possible to recruit the monomers that contain the catalytic overhangs. If the overhangs are not present in the same monomer as the G on their module (i.e. the catalyst is not formed), recruiting monomers with overhangs in this manner only increases the number of covered labels by 1 for each monomer recruited, because catalytic overhangs by definition have only a single group label. For example, in Fig. 4.6, only group 2 would be covered if we merged H_2 of the module that contains the group 2 overhangs.¹⁰

If there is one catalyst for this reaction, and it is formed, then it can cover all of the numeric groups. If there are at least two catalysts for this reaction, and if the catalysts for this reaction are formed, then w groups can be covered by one monomer (i.e. the catalyst covers the groups that it contains as overhangs)¹¹.

Thus, to cover all n groups without all of the catalysts formed and in the same polymer as the G of $\mathcal{T}|Z$, it is necessary to have at least $n = kw$ additional monomers/complexes in the polymer with G . Adding this to the $n - 1$ monomers of the HV -sequence, we find that it is necessary to recruit $n - 1 + kw$ monomers to cover the transitioning grid, which is $kw - 1$ more than the number of monomers required in the base configuration (which requires n monomers – i.e. H -configuration or V -configuration). Then since β has at least $kw - 1$ entropy less than a stable configuration, it is not reachable within the global barrier $B \leq k + w - 2$ and this would contradict that $\beta \in \Gamma_{\mathcal{T}}^B$, so we conclude that it is not possible in β to cover G without the correct catalysts both formed and present in the

¹⁰One might also merge the monomers with overhangs together first (with or without their corresponding G) and then merge the resulting complex onto the transitioning grid, but this is functionally the same as merging each onto the grid individually. One might also try to use a partially formed catalyst (i.e. another grid in transition), but an inductive argument would explain why this requires more merges than having only a single grid in transition.

¹¹Here it is important to note that the CRN is catalytically non-competitive, and so the overhangs must be provided by the correct catalytic species (and not, for example, provided by the two halves of the catalytic region that overhang the modules $\mathcal{T}|C_1$ and $\mathcal{T}|K_2$ in the competitive reactions $Z^F \xleftarrow{C_1+C_2} Z^T$ and $Z^F \xleftarrow{K_1+K_2} Z^T$)

polymer with G . □

Lemma 52 (simulation is sound). *Let $k \geq 2$. Let $B \leq k+w-2$. For every implementation configuration $\beta \in \Gamma_T^B$, if an operation $\beta \rightarrow \gamma$ is not silent, then for some $\alpha_{\mathcal{N}} \in \mathcal{R}_{\mathcal{N}}$, $\beta \xrightarrow{\alpha_{\mathcal{N}}} \gamma$ and $m(\beta) \xrightarrow{\alpha_{\mathcal{N}}} m(\gamma)$.*

Proof. There are two cases in which an operation can be non-silent.

Case 1: Split

In this case, in the configuration of some module $T|Z$, an H monomer was split from the polymer containing G (affecting a change in interpretation from $Z^F \rightarrow Z^T$). More specifically, in the context of β , the configuration of the module had a polymer that contained G as well as the monomers $H_{n-1}, \dots, H_{\lfloor n/2 \rfloor}$; in the context of γ , the configuration of the module contains no such polymer. Either the second or third condition of Lemma 51 applies here.

Suppose the second condition applies; but then $Z^T \in m(\beta)$, and it is not possible to change its interpretation via a split (it requires a merge of at least one H).

Then the third condition applies, and all of the necessary catalysts for the reaction $\alpha_{\mathcal{N}} = Z^F \xleftarrow{K_1, \dots, K_t} Z^T$ are contained in $m(\gamma)$, and so were also contained in $m(\beta)$ (because this was a split) and so the reaction was applicable to $m(\beta)$.

Case 2: Merge

This case follows symmetrically from the first case, for as the argument above was made by reasoning about a split from β to γ , so too could the same argument be made about a merge from γ to β (and hence by symmetry, about a merge from β to γ). □

We now establish certain reachability conditions that are necessary to prove completeness. Lemma 53 states that each configuration can silently reach its “correct” base configuration. Lemma 55 gives a path by which a base configuration can correctly transition to another base configuration according to a formal reaction while using only one non-silent step. Finally, Corollary 54 states that each base configuration can silently reach any configuration with the same interpretation. By concatenating these three paths, we establish the reachability sufficient to prove completeness in Lemma 56.

Lemma 53. *Let $B \leq k + w - 2$ and let $\beta \in \Gamma_{\mathcal{T}}^B$. Let $\beta_0 \in \Gamma_{\mathcal{T}}^B$ be the base configuration with $m(\beta) = m(\beta_0)$. Then $\beta \xrightarrow{1} \beta_0$.*

Proof. The inductive argument below will establish that β can reach β_0 by a path with barrier 1; however, it may be the case that β already has B fewer entropy than β_0 , in which case it would not be possible to execute a barrier 1 path without going over the global barrier B . To argue that such “headroom” exists, recall that $\beta \in \Gamma_{\mathcal{T}}^B$ and that by the definition of $\Gamma_{\mathcal{T}}^B$, β is reachable from some stable configuration of \mathcal{T} by a path \mathbf{p} with barrier at most B .

Consider the last operation in path \mathbf{p} . If that last operation was a split, then β is at most $B - 1$ entropy away from a stable configuration, and the new path of barrier 1 can be applied without going over the global barrier B . If instead the last operation in \mathbf{p} was a merge, if it did not change the interpretation, we can reverse the merge (with a split) to arrive at a configuration β^- that is at most $B - 1$ entropy away from a stable configuration, and then append the path of barrier 1. Finally, if the last operation in \mathbf{p} was a merge that changed the interpretation of a module $\mathcal{T}|Z$, then it must have done so by merging an H into the polymer with G (so that $Z^F \in m(\beta)$). By applying Lemma 51 to β , we see that the third case is the only case that can apply (and also applies to the configuration β^- that preceded it \mathbf{p}); that is, the catalysts are present to cover the numerically labelled sites in Fig. 4.6. But since this is the case, and we also have Z^F (indicating the presence of $H_{n-1}, \dots, H_{\lfloor n/2 \rfloor}$), we can split the singleton $V_{\lfloor n/2 \rfloor}$ from its monomer, and arrive at a configuration β^+ and the pathway of barrier 1 can now be appended.

We now make the inductive hypothesis that if β has m modules that are not in H -configurations or V -configurations, then β can reach a configuration β' that has $\max\{0, m - 1\}$ modules that are not in H -configurations or V -configurations, $m(\beta) = m(\beta')$, and β can reach β' by a path with barrier 1. We prove this by induction on m .

The base case $m = 0$ is evident; note that if β has 0 modules that are not in H -configurations or V -configurations, then greedily splitting the modules from each other results in the base configuration β_0 .

Consider now a configuration that has $m + 1$ modules that are not in H -configurations

or V -configurations. Let $\mathcal{T}|Z$ be such a module. One of the three conditions of Lemma 51 must apply to this module.

In the first case (all H s are in the same polymer as G), any V s in the same polymer as G can be split using a path of zero barrier, and now this module is in an H -configuration.

In the second case (all V s are in the same polymer as G), $Z^T \subset m(\beta)$, and any H s in the same polymer as G can be split using a path of zero barrier to arrive at the V -configuration.

In the third case, the module is in the “middle” of simulating a valid transition between Z^F and Z^T (or vice versa). All of the necessary catalysts are present (and furthermore, on the same polymer as the G monomer of the $\mathcal{T}|Z$ module. By alternately merging an H and splitting a V (as in Fig. 4.4), and then splitting off any remaining V s, the module can resolve to its H -configuration (or it can alternate merging a V and splitting an H to resolve to its H -configuration). We follow the path that does not change the interpretation. This silent pathway results in a new configuration β' in which $\mathcal{T}|Z$ is now in an H -configuration or a V -configuration, and β can reach β' by a path of barrier 1. \square

Corollary 54. *Let $B \in \mathbb{N}$ and let $\beta_0 \in \Gamma_{\mathcal{T}}^B$ be a base configuration. Let $\beta \in \Gamma_{\mathcal{T}}^B$ with $m(\beta) = m(\beta_0)$. Then $\beta_0 \xrightarrow{\perp} \beta$.*

Proof. TBN paths are reversible; we obtain the desired result by reversing the path of Lemma 53. \square

Lemma 55. *Let $B \geq k$. Let $\alpha_{\mathcal{N}} \in \mathcal{R}_{\mathcal{N}}$ be applicable to a formal configuration $\mathbf{c}_{\mathcal{N}} \in \mathbb{N}^{\Lambda_{\mathcal{N}}}$, resulting in formal configuration $\mathbf{d}_{\mathcal{N}} \in \mathbb{N}^{\Lambda_{\mathcal{N}}}$. Let $\gamma_0 \in \Gamma_{\mathcal{T}}^B$ be the base configuration with $m(\gamma_0) = \mathbf{c}_{\mathcal{N}}$ and let $\delta_0 \in \Gamma_{\mathcal{T}}^B$ be the base configuration with $m(\delta_0) = \mathbf{d}_{\mathcal{N}}$. Then $\gamma_0 \xrightarrow{\alpha_{\mathcal{N}}} \delta_0$.*

Proof. We prove this by the existence of a path; this path follows from the construction of the amalgamated TBN. If $\alpha_{\mathcal{N}}$ was applicable to $\mathbf{c}_{\mathcal{N}}$, then all of the necessary catalysts are present in the formal configuration, and so the corresponding modules in the implementation will be in the configuration (H -configuration or V -configuration) that brings together the correct catalytic overhangs as shown in Fig. 4.5. Without loss of generality,

assume that the species $R_{\mathcal{N}}$ converts from false to true in the binary reaction (i.e. from H -configuration to V -configuration in the TBN).

The path then proceeds as follows: merge the necessary catalysts onto the G -containing polymer that represents $R_{\mathcal{N}}^F$ in $\alpha_{\mathcal{N}}$ (there will be at most k polymers to merge, because the CRN is $\leq k$ -catalytic). Then (as in Fig. 4.4) split from this polymer the monomer H with the subscript of least value. Then merge the monomer V with the subscript of greatest value. Continue in this way until there are no H monomers left in the polymer containing G . Then split off the catalysts. At no point was barrier B exceeded because each intermediate configuration had no more than $k \leq B$ entropy less than the (stable) base configurations. \square

Lemma 56 (simulation is complete). *Let $B \geq k$. For every implementation configuration $\beta \in \Gamma_{\mathcal{T}}^B$, if $m(\beta) \xrightarrow{\alpha_{\mathcal{N}}} m(\gamma)$ for some $\alpha_{\mathcal{N}} \in \mathcal{R}_{\mathcal{N}}$, then $\beta \xrightarrow{\alpha_{\mathcal{N}}} \gamma$.*

Proof. By Lemma 53, there is a base configuration β_0 so that $\beta \xrightarrow{\perp} \beta_0$. By Corollary 54, there is a base configuration γ_0 so that $\gamma_0 \xrightarrow{\perp} \gamma$. By Lemma 55, $\beta_0 \xrightarrow{\alpha_{\mathcal{N}}} \gamma_0$. By concatenating these paths, we achieve a path $\beta \xrightarrow{\alpha_{\mathcal{N}}} \gamma$. \square

Chapter 5

Verification Software

The content of this chapter comes from Sections 2.1-4.3 of a preprint [30] and is currently in review for publication.

5.1 Preliminaries

We present now a number of definitions necessary for the exposition of this chapter. Some definitions (particularly those involving TBNs) will overlap those given in previous chapters, but here will be presented in a fashion that makes the connection between the concepts and their equivalents in integer vectors more apparent. This difference in presentation will be necessary to establish intuitive connections that lead up to the integer programming model.

5.1.1 Definitions

A *multiset* is an unordered collection of objects allowing duplicates (including countably infinite multiplicities), e.g., $\mathbf{v} = \{2 \cdot a, b, \infty \cdot d\}$. Equivalently, a multiset with elements from a finite set U is a vector $\mathbf{v} \in (\mathbb{N} \cup \{\infty\})^U$ describing the counts, indexed by U ; in the example of Fig. 1.1, if $U = \{a, b, c, d\}$, then $\mathbf{v}(a) = 2$, $\mathbf{v}(b) = 1$, $\mathbf{v}(c) = 0$, and $\mathbf{v}(d) = \infty$. The *cardinality* of a multiset $\mathbf{v} \in \mathbb{N}^U$ is $|\mathbf{v}| = \sum_{u \in U} \mathbf{v}(u)$; a *finite multiset* \mathbf{v} obeys $|\mathbf{v}| < \infty$. A *site type* is a formal symbol, such as a , representing a specific binding site on a molecule; in Fig. 1.1 the site types are a, a^*, b, b^* . Each site type has a corresponding *complement type* denoted by a star, e.g. a^* . Complementarity is an

involution: $(a^*)^* = a$. A site and its complement can form an attachment called a *bond*. We follow the convention that for any complementary pair of sites a, a^* , the total count of a^* across the whole TBN is at most that of a , i.e., the starred sites are *limiting*. A *monomer type* is a finite multiset of site types. When context implies a single instance of a monomer/site type, we may interchangeably use the term *monomer/site*.¹

A *thermodynamic binding network (TBN)* is a multiset of monomer types; equivalently, a vector in $(\mathbb{N} \cup \{\infty\})^m$, if m is the number of monomer types and we have fixed some standardized ordering of them. We allow some monomer counts to be infinite in order to capture the case where some monomers are added in “large excess” over others, a common experimental approach [47, 45]. A *polymer* is a finite multiset of monomer types, equivalently, a vector in \mathbb{N}^m , where m is the number of monomer types in a standardized ordering.² Note that despite the suggestive lines representing bonds in Fig. 1.1, this definition does not track which pairs of complementary sites are bound within a polymer.

Let $S_{\mathcal{T}}$ (respectively, $S_{\mathcal{T}}^*$) be the set of unstarred (resp., starred) site types of \mathcal{T} . For a monomer \mathbf{m} and site type $s \in S_{\mathcal{T}}$, let $\mathbf{m}(s)$ denote the count of s minus the count of s^* in \mathbf{m} (intuitively, $\mathbf{m}(s)$ is the “net count” of s in \mathbf{m} , negative if there are more s^* .) For $s^* \in S_{\mathcal{T}}^*$ let $\mathbf{m}(s^*) = -\mathbf{m}(s)$. The *exposed sites* of a polymer \mathbf{P} are a finite multiset of site types that results from removing as many (site, complement) pairs from a polymer as possible, described by the net count of sites when summed across all monomers in the polymer. For example, in the polymer $\{\{a^*, b^*\}, \{a, c\}, \{a, b, c\}, \{c, d^*\}\}$, the exposed sites are $\{a, 2 \cdot c, d^*\}$.

A *configuration* of a TBN is a partition of the TBN into polymers. A polymer is *self-saturated* if it has no exposed starred sites. A configuration is *saturated* if all of its polymers are self-saturated. Since we assume that, across the entire configuration, starred sites are limiting, this is equivalent to stipulating that the maximum number of bonds are formed. Write $\Gamma_{\mathcal{T}}$ to denote the set of all saturated configurations of the TBN \mathcal{T} . A

¹Concretely, in a DNA nanotech design, a monomer corresponds to a strand of DNA, whose sequence is logically partitioned into binding sites corresponding to long (5-20 base) regions, e.g., 5'-AAAGG-3', intended to bind to the complementary sequence 3'-TTTCC-5' that is part of another strand.

²The term “polymer” is chosen to convey the concept of combining many atomic objects into a complex, but it is not necessarily a linear chain of repeated units.

configuration is *stable* if it is saturated and has the maximum number of non-singleton polymers among all saturated configurations.

Since the number of polymers may be infinite, we will use the equivalent notion that stable configurations are those that can be constructed by starting with the “melted” configuration whose polymers are all singletons containing only one monomer, performing the minimum number of polymer merges necessary to reach a saturated configuration. For example, consider the TBN consisting of monomer types $\mathbf{t} = \{a\}$, $\mathbf{b} = \{a^*\}$, with counts $\infty \cdot \mathbf{t}$ and $2 \cdot \mathbf{b}$. The unique stable configuration has polymers $\{2 \cdot \{\mathbf{b}, \mathbf{t}\}, \infty \cdot \mathbf{t}\}$, since two merges of a \mathbf{b} and a \mathbf{t} are necessary and sufficient to create this configuration from the individual monomers.

5.1.2 Solvers

The problems addressed in this paper are NP-hard. To tackle this difficulty, we cast the problems as integer programs and use the publicly available IP solver SCIP [28].

We also use the open-source software OR-tools[44], which is a common front-end for SCIP[28], Gurobi [29], and a bundled constraint programming solver CP-SAT. Though we model our problems as IPs, we would also like to be able to solve for all feasible/optimal solutions rather than just one, which CP-SAT can do. This flexible front-end lets us switch seamlessly between the two types of solvers without significant alterations to the model.

We have found that the most efficient way to produce a full set of optimal solutions is to first use SCIP to find the optimal objective value, then to constrain the model to that objective value and produce the full set of solutions with CP-SAT. We believe that this is because SCIP more quickly establishes a bound on the objective value using the dual bound, whereas CP-SAT (using more of a SAT-based approach) must either explore or prune all possibilities that might lead to a better objective value. We use the open-source software package 4ti2[2] to calculate Hilbert Bases as described in Section 5.3.

5.2 Computing stable configurations of TBNs

Section 5.2.1 formally defines the stable configurations problem. Section 5.2.2 explains our IP formulation of the problem. Section 5.2.3 shows empirical runtime benchmarks.

5.2.1 Finding stable configurations of TBNs

We consider the problem of finding the stable configurations of a TBN. Given a TBN \mathcal{T} , let $\Gamma_{\mathcal{T}}$ denote the set of all saturated configurations of \mathcal{T} .

Recall that a configuration $\gamma \in \Gamma_{\mathcal{T}}$ is defined as a partition of the monomers of \mathcal{T} into polymers, so its elements $\mathbf{P} \in \gamma$ are polymers, i.e., multisets of monomers. For any $\gamma \in \Gamma_{\mathcal{T}}$, we define the corresponding *partial configuration* $\bar{\gamma} = \{\mathbf{P} \in \gamma : |\mathbf{P}| > 1\}$ that excludes polymers consisting of only a single monomer. Note that in the context of \mathcal{T} , the mapping $\gamma \mapsto \bar{\gamma}$ is one-to-one. We consider only partial configurations with finite-sized polymers. The notion of partial configuration will be useful in reasoning about TBNs with infinite monomer counts but finite size polymers, since all but finitely many monomers will be excluded from the partial configurations we consider.

Now we define the number of elementary merge operations required to reach a saturated configuration from the configuration of all singletons:

$$m(\gamma) = \left(\sum_{\mathbf{P} \in \bar{\gamma}} |\mathbf{P}| \right) - |\bar{\gamma}| \quad (5.1)$$

We can then define the stable configurations as those saturated configurations that minimize the number of merges required to reach them from the all-singletons configuration.

$$\text{STABLECONFIGS}(\mathcal{T}) = \{\gamma \in \Gamma_{\mathcal{T}} : (\forall \gamma' \in \Gamma_{\mathcal{T}}) m(\gamma) \leq m(\gamma')\}$$

Note that $m(\gamma) = m(\bar{\gamma})$. Thus the STABLECONFIGS problem may be equivalently posed as finding the set of partial configurations $\bar{\gamma}$ that minimize $m(\bar{\gamma})$.

We now describe how to handle infinite counts. A configuration is saturated if and only if none of its starred sites (elements of $S_{\mathcal{T}}^*$) are exposed. Thus we focus on the subset of monomers that contain starred sites: the *limiting monomers* $\mathcal{T}_L = \{\mathbf{m} \in \mathcal{T} : \mathbf{m} \cap S_{\mathcal{T}}^* \neq \emptyset\}$. Limiting monomers are required to have finite count, whereas nonlimiting monomers

(those with all unstarred sites) are allowed to be finite or infinite count. Our IP representation of a configuration explicitly accounts for *all* the limiting monomers, but only the nonlimiting monomers (in $\mathcal{T} \setminus \mathcal{T}_L$) in a polymer with some limiting monomer; implicitly every other nonlimiting monomer is unbound (i.e., in its own singleton polymer). This allows us to describe infinite configurations where all but finitely many of the infinite count monomers are unbound, guaranteeing that the number of merges counted in Eq. (5.1) is finite.

5.2.2 Casting StableConfigs as an IP

5.2.2.1 Finding a single stable configuration

We first describe how to find a single element from $\text{STABLECONFIGS}(\mathcal{T})$ by identifying its partial configuration in \mathcal{T} . We begin by fixing a bound B on the number of non-singleton polymers in any partial configuration. If no *a priori* bound for B is available, conservatively take $B = |\mathcal{T}_L|$, the total number of limiting monomers.

Integer variables Assume an arbitrary ordering of the m monomer types $\mathbf{m}_1, \mathbf{m}_2, \dots$. Our IP formulation uses $B \cdot m + B$ integer-valued variables describing the solution via its partial configuration:

- $\text{Count}(\mathbf{m}, j)$: count of monomer type $\mathbf{m} \in \mathcal{T}$ in polymer \mathbf{P}_j where $j \in \{1, 2, \dots, B\}$
- $\text{Exists}(j)$: false (0) if polymer \mathbf{P}_j is empty, possibly true (1) otherwise, $j \in \{1, 2, \dots, B\}$

Example 57. Recall the TBN of Fig. 1.1. Suppose the TBN has 1 each of $\mathbf{m}_1 = \{a^*, b^*\}$, $\mathbf{m}_2 = \{a, b\}$, $\mathbf{m}_3 = \{a\}$, $\mathbf{m}_4 = \{b\}$, with upper bound $B = 2$ on the number of non-singleton polymers. $\mathcal{T}_L = \{\mathbf{m}_1\}$ since \mathbf{m}_1 is the only monomer with starred sites. The linear constraints (see below for details) do not require all copies of $\mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4 \in \mathcal{T} \setminus \mathcal{T}_L$ to be included in a polymer. The stable configuration on the right of Fig. 1.1 (partition $\{\mathbf{m}_1, \mathbf{m}_2\}, \{\mathbf{m}_3\}, \{\mathbf{m}_4\}$) is represented in the IP by setting $\text{Count}(\mathbf{m}_1, 1) = \text{Count}(\mathbf{m}_2, 1) = 1$, $\text{Count}(\mathbf{m}_3, 1) = \text{Count}(\mathbf{m}_4, 1) = 0$ (monomers 1 and 2 are in polymer 1, but monomers 3 and 4 are not), and setting $\text{Count}(\mathbf{m}_i, 2) = 0$ for $i = 1, 2, 3, 4$ (no monomers are in polymer 2), $\text{Exists}(1) = 1$ (polymer 1 is non-empty), and $\text{Exists}(2) = 0$ (polymer 2 is empty).

Example 58. Suppose a TBN with the same monomer types as Example 57 has 3 of \mathbf{m}_1 and infinitely many of the remaining monomers (allowed since they are not limiting), with $B = 4$. The partial configuration where two copies of \mathbf{m}_1 are each bound to a single \mathbf{m}_2 (forming two polymers with two monomers each, as in the stable configuration of Fig. 1.1), and the third \mathbf{m}_1 is bound to an \mathbf{m}_3 and an \mathbf{m}_4 (forming one polymer with three monomers, as in the rightmost non-stable saturated configuration of Fig. 1.1) is represented in the IP by setting $\text{Exists}(1) = \text{Exists}(2) = \text{Exists}(3) = 1$ and $\text{Exists}(4) = 0$, and

$$\begin{aligned} \text{Count}(\mathbf{m}_1, 1) &= 1, & \text{Count}(\mathbf{m}_2, 1) &= 1, & \text{Count}(\mathbf{m}_3, 1) &= 0, & \text{Count}(\mathbf{m}_4, 1) &= 0, \\ \text{Count}(\mathbf{m}_1, 2) &= 1, & \text{Count}(\mathbf{m}_2, 2) &= 1, & \text{Count}(\mathbf{m}_3, 2) &= 0, & \text{Count}(\mathbf{m}_4, 2) &= 0, \\ \text{Count}(\mathbf{m}_1, 3) &= 1, & \text{Count}(\mathbf{m}_2, 3) &= 0, & \text{Count}(\mathbf{m}_3, 3) &= 1, & \text{Count}(\mathbf{m}_4, 3) &= 1, \\ \text{Count}(\mathbf{m}_1, 4) &= 0, & \text{Count}(\mathbf{m}_2, 4) &= 0, & \text{Count}(\mathbf{m}_3, 4) &= 0, & \text{Count}(\mathbf{m}_4, 4) &= 0. \end{aligned}$$

The constraints described below allow $\text{Exists}(j) = 0$ even if polymer j is nonempty, even though the variables ultimately aim to count exactly the number of nonempty polymers (as $\sum_{j=1}^m \text{Exists}(j)$). A false negative undercounts the number of polymers, overcounting the number of merges in Eq. (5.1). However, the number of merges is being *minimized* by the IP. For a given setting of **Count** variables, the minimum is achieved (subject to the constraints) by setting each $\text{Exists}(j) = 1$ *if and only* if polymer j is nonempty.

Linear constraints Let $\mathcal{T}(\mathbf{m})$ denote the number of monomers of type \mathbf{m} in the TBN \mathcal{T} . Recall that $\mathbf{m}(s)$ is the net count of site type $s \in S_{\mathcal{T}}$ in monomer type \mathbf{m} (negative if \mathbf{m} has more s^* than s).

$$\sum_{j=1}^B \text{Count}(\mathbf{m}, j) = \mathcal{T}(\mathbf{m}) \quad \forall \mathbf{m} \in \mathcal{T}_L \quad (5.2)$$

$$\sum_{j=1}^B \text{Count}(\mathbf{m}, j) \leq \mathcal{T}(\mathbf{m}) \quad \forall \mathbf{m} \in \mathcal{T} \setminus \mathcal{T}_L \quad (5.3)$$

$$\sum_{\mathbf{m} \in \mathcal{T}} \text{Count}(\mathbf{m}, j) \cdot \mathbf{m}(s) \geq 0 \quad \forall j \in \{1, 2, \dots, B\}, \forall s \in S_{\mathcal{T}} \quad (5.4)$$

$$\sum_{\mathbf{m} \in \mathcal{T}_L} \text{Count}(\mathbf{m}, j) \geq \text{Exists}(j) \quad \forall j \in \{1, 2, \dots, B\} \quad (5.5)$$

Constraints Eq. (5.2) and Eq. (5.3) intuitively establish “monomer conservation” in the partial configuration. Constraint Eq. (5.2) enforces that we account for every limiting monomer in \mathcal{T} . Constraint Eq. (5.3) establishes that for *non*-limiting monomers, we cannot exceed their supply (trivially satisfied for any infinite-count monomer); any leftovers are assumed to be in singleton polymers in the full configuration, but are not explicitly described by `Count` variables. Constraint Eq. (5.4) ensures that all polymers are self-saturated. Specifically, the count of site $s \in S_{\mathcal{T}}$ within polymer j must meet or exceed that of s^* . Lastly, constraint Eq. (5.5) enforces that nonempty polymers contain at least one limiting monomer. Ideally, this constraint should enforce that if a polymer contains no monomers at all, then it cannot be part of the nonempty polymer tally; however, if the constraint were modeled in this way, the formulation would admit invalid partial configurations that include explicit singleton polymers.

Linear objective function Subject to the above constraints, we minimize the number of merges needed to go from a configuration where all monomers are separate to a saturated configuration. For finite count TBNs, this is the number of monomers minus the number of polymers in the partial configuration. Equivalently (and applying to infinite TBNs), this is the sum over all nonempty polymers of its number of monomers minus 1. Formally, the IP minimizes Eq. (5.6):

$$\sum_{j=1}^B \left[\left(\sum_{\mathbf{m} \in \mathcal{T}} \text{Count}(\mathbf{m}, j) \right) - \text{Exists}(j) \right] \quad (5.6)$$

If polymer j is empty ($\sum_{\mathbf{m} \in \mathcal{T}} \text{Count}(\mathbf{m}, j) = 0$), then constraint Eq. (5.4) forces $\text{Exists}(j) = 0$; otherwise $\text{Exists}(j) = 1$ minimizes Eq. (5.6). Thus the outer sum is over the nonempty polymers.

5.2.2.2 Finding all stable configurations

While an IP formulation for finding a single stable configuration is well-defined above, without modification it is ill-suited as a formulation to find *all* stable configurations. In addition, tightening the available constraints (e.g., enforcing $\text{Exists}(j) \iff$ polymer j is nonempty, described below) provides a more robust framework to which to add custom constraints (e.g. specifying a fixed number of polymers).

IP to find optimal objective value, CP to enumerate optimal solutions One straightforward improvement is to solve for the optimal value of the objective function using a dedicated IP solver such as SCIP, whose primal-dual methods exploit the underlying real-valued geometry of the search space to find an objective value more efficiently than Constraint Programming (CP) solvers such as CP-SAT. Then, use this optimal value to bootstrap the CP formulation, which is better suited to enumerating all solutions with a given objective value. This works particularly well in our experiments: use SCIP to solve the optimization problem (but SCIP has no built-in ability to enumerate all feasible solutions), then use CP-SAT (which takes longer than SCIP to find the objective value) to locate all feasible solutions to the IP obtaining the objective value found by SCIP.

Enforcing that Exists variables exactly describe nonempty polymers

Constraint Eq. (5.4) enforces that $\text{Exists}(j) = 0$ if polymer \mathbf{P}_j is empty, but it does not enforce the converse. However, when using CP-SAT with a *fixed* objective value, we can no longer rely on the minimization of Eq. (5.6) to enforce that $\text{Exists}(j) = 1 \iff \mathbf{P}_j$ is nonempty.

We add a new constraint to handle this. Let

$$C = 1 + \sum_{s \in S_{\mathcal{T}}} \sum_{\mathbf{m} \in \mathcal{T}} \mathcal{T}(\mathbf{m}) \cdot \mathbf{m}(s^*). \quad (5.7)$$

C is an upper bound on the largest number of monomers in a polymer in any valid partial configuration of \mathcal{T} minimizing Eq. (5.6). This corresponds to the worst case in which a single polymer contains *every* limiting monomer, and each starred site is bound to its own unique monomer. The following constraint enforces that if $\text{Exists}(j) = 0$, then polymer \mathbf{P}_j contains no monomers:

$$\sum_{\mathbf{m} \in \mathcal{T}_L} \text{Count}(\mathbf{m}, j) \leq C \cdot \text{Exists}(j) \quad \forall j \in \{1, 2, \dots, B\} \quad (5.8)$$

Eliminating symmetries due to polymer ordering In the formulation of Section 5.2.2, many isomorphic solutions exist in the feasible region. For instance, one could obtain a “new” solution by swapping the compositions of polymers \mathbf{P}_1 and \mathbf{P}_2 . The number of isomorphic partial configurations grows factorially with the number of polymers.

Before asking the solver to enumerate all solutions, we must add constraints that eliminate isomorphic solutions. We achieve this by using the (arbitrary) ordering of the monomer types to induce a lexicographical ordering on the polymers, then add constraints ensuring that any valid solution contains the polymers in sorted order.

Sorting non-binary vectors in an IP is generally a difficult task (for instance, see [56]). The primary reason for this difficulty is that encoding the sorting constraints involves logical implications ($p \implies q$), which, being a type of disjunction ($\neg p \text{ OR } q$), are difficult to encode into a convex formulation described as a conjunction (AND) of several constraints. However, we do have an upper bound C on the values that the **Count** variables can take, making certain “large-number” techniques possible.

Intuitively, when comparing two lists of scalars (i.e., vectors) to verify that they are correctly sorted, one must proceed down the list of entries until one of the entries is larger than its corresponding entry in the other list. For as long as the numbers are the same, they are considered “tied”. When one entry exceeds the corresponding other, the tie is considered “broken”, after which no further comparisons need be conducted between the two vectors.

We $B \cdot m$ new Boolean (0/1-valued) variables ($\text{Tied}(\mathbf{m}_i, j)$ for each $i = 1, \dots, m$ and $j = 1, \dots, B$), that reason about consecutive pairs of polymers $\mathbf{P}_{j-1}, \mathbf{P}_j$. We describe constraints enforcing that for each $h \leq i$, $\text{Tied}(\mathbf{m}_i, j) = 1 \iff \text{Count}(\mathbf{m}_h, j-1) = \text{Count}(\mathbf{m}_h, j)$.

Let C be defined as in Eq. (5.7). For simplicity of notation below, define the constants $\text{Tied}(\mathbf{m}_0, j) = 1$ for all $j = 1, \dots, B$. The meaning of the sorting variables is then enforced by the following constraints, which we define for $i \in \{1, 2, \dots, m\}$ and $j \in \{2, 3, \dots, B\}$:

$$\text{Tied}(\mathbf{m}_i, j) \leq \text{Tied}(\mathbf{m}_{i-1}, j) \tag{5.9}$$

$$\text{Count}(\mathbf{m}_i, j-1) - \text{Count}(\mathbf{m}_i, j) \leq C \cdot (1 - \text{Tied}(\mathbf{m}_i, j)) \tag{5.10}$$

$$\text{Count}(\mathbf{m}_i, j-1) - \text{Count}(\mathbf{m}_i, j) \geq -C \cdot (1 - \text{Tied}(\mathbf{m}_i, j)) \tag{5.11}$$

$$\text{Count}(\mathbf{m}_i, j-1) - \text{Count}(\mathbf{m}_i, j) \geq 1 - C \cdot (1 + \text{Tied}(\mathbf{m}_i, j) - \text{Tied}(\mathbf{m}_{i-1}, j)) \tag{5.12}$$

Intuitively, Eq. (5.9) enforces $\text{Tied}(\mathbf{m}_i, j) \implies \text{Tied}(\mathbf{m}_{i-1}, j)$: a tie in the current entry

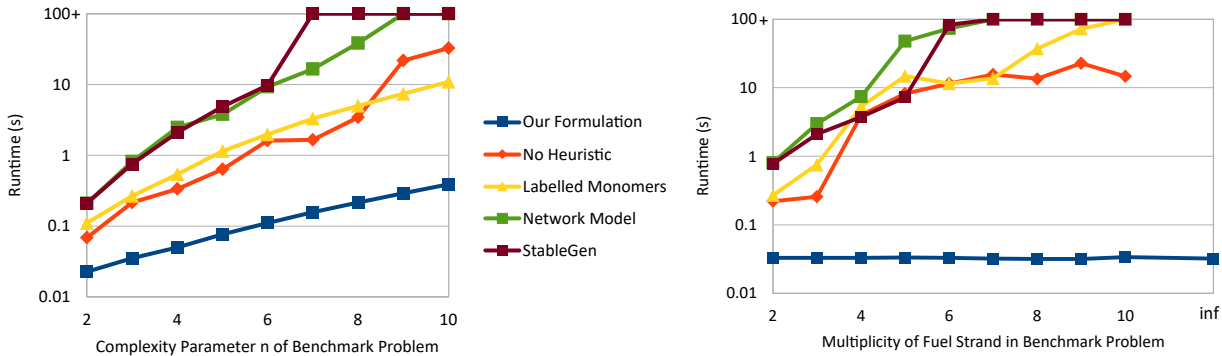


Figure 5.1: Empirical tests solving STABLECONFIGS for our benchmark problem based upon its complexity parameter n (left), and the multiplicity of the unstarred “fuel” strands (right). Our formulation is tested against several variations on the approach (which are described in the text) and the StableGen algorithm from [11]. The TBN is parameterized by n and contains the monomers $G_n = \{x_{ij}^* : 1 \leq i, j \leq n\}$, $H_i = \{x_{ij} : 1 \leq j \leq n\}$ for all $1 \leq i \leq n$, and $V_j = \{x_{ij} : 1 \leq i \leq n\} \cup \{x_{ij} : j \leq i \leq n\}$ for all $1 \leq j \leq n$. See Fig. 6 from [10] for a detailed explanation of this TBN and its operation. The vertical axis is log scale. Points at the top of the scale timed out after 100 seconds. The alternate formulations cannot solve the instance in the case of infinite fuel strands.

is only relevant if the tie was not resolved before. Eq. (5.10) and Eq. (5.11) together enforce $\text{Tied}(\mathbf{m}_i, j) \implies (\text{Count}(\mathbf{m}_i, j - 1) = \text{Count}(\mathbf{m}_i, j))$: ties can only continue for as long as the corresponding entries are equal. Eq. (5.12) enforces $\text{Tied}(\mathbf{m}_{i-1}, j) \wedge \neg \text{Tied}(\mathbf{m}_i, j) \implies (\text{Count}(\mathbf{m}_i, j - 1) > \text{Count}(\mathbf{m}_i, j))$: ties can only be broken if the tie was not broken previously and the current entries are ordered correctly. Thus any solution satisfying these constraints must sort the polymers.

5.2.3 Empirical running time measurements

For our empirical tests we use as a benchmark the autocatalytic TBN described in [10, Section 4.2.2 and Fig. 6]. This TBN features two large monomers of size n^2 in which n is a parameter in the design, as well as a variable number of additional monomers (“fuels”) intended to be present in large excess quantities.

In addition to the formulation we give in this paper, we also tested a number of formulation variants, including the StableGen algorithm originally posed in [11] for solving the STABLECONFIGS problem, justifying some of our design choices. “No Heuristic” performs a thorough accounting of all monomers (not just those needed to achieve saturation against the limiting monomers). “Labelled Monomers” assumes that the monomers are provided as a set, rather than a multiset. “Network Model” is a modification of StableGen with an alternate saturation constraint which does not require the explicit invocation of

site-level bonds.

Each data point represents the average of three runs, and the solver was allowed to run for up to 60 seconds before a timeout was forced. Fig. 5.1 (left) shows the runtimes as they increase with the parameter n , holding the count of each fuel at 2. Fig. 5.1 (right) fixes $n = 3$ and shows the runtimes as they increase with the multiplicity of the fuel monomers. Note that our formulation can solve the case when fuels are in unbounded excess, while the variant formulations require bounded counts of all monomers.

Our formulation solves all of the benchmark problems in under one second, suggesting that it is suitable for much larger/more complex problems than were approachable previously.

5.3 Computing bases of locally stable configurations of TBNs

We now shift attention to *locally* stable configurations: those in which no polymer can be split without breaking a bond. Such a configuration may not be stable, but the only paths to create more polymers, without breaking any bonds, require first merging existing polymers (i.e., going uphill in energy before going down). The saturated configurations are precisely those obtained by merging polymers starting from some locally stable configuration. In this section we describe a technique for computing what we call the *polymer basis*: the (finite) set of polymers that can exist in locally stable configurations. In Section 5.3.1, we show that an algebraic concept called the *Hilbert basis* [24] characterizes the polymer basis. In Sections 5.3.2 and 5.3.3 we show how the polymer basis can be used to reason about TBN behavior.

5.3.1 Equivalence of polymer bases and Hilbert bases

We note that the connection between Hilbert bases and polymer bases is not particularly deep and does not require clever techniques to prove. Once the definitions are appropriately set up, the equivalence follows almost immediately. (Though we provide a self-contained proof.) The primary insight of this section is that casting TBNs in our IP formulation sets up the connection with Hilbert bases. Since highly optimized soft-

ware exists for computing Hilbert bases [2], this software can be deployed to automate reasoning about TBNs.

Let M be a set of monomer types with $m = |M|$. Let \mathcal{S}_M denote the *TBN schema of M* , the set of all TBNs containing only monomers from M , such that starred sites are limiting (i.e., such that saturated configurations have all starred sites bound). Let A_M be the *matrix representation* of the monomer types in \mathcal{S}_M , describing the contents of each monomer type: formally, the row- i , column- j entry of A_M is $\mathbf{m}_j(s_i)$, the net count of site type s_i in monomer type \mathbf{m}_j (as an example, $\{a^*, b, a, a, a, c, c^*, c^*\}$ has net count 2 of a , 1 of b , and -1 of c). Formally, a TBN $\mathcal{T} \in \mathcal{S}_M$ if and only if $A_M \mathcal{T} \geq \mathbf{0}$.

Recall that $\Gamma_{\mathcal{T}}$ is the set of saturated configurations of the TBN \mathcal{T} , and that a polymer \mathbf{P} is *self-saturated* if it has no exposed starred sites, i.e., $A_M \mathbf{P} \geq \mathbf{0}$. Define the *polymer basis* $\mathcal{B}_{\mathcal{S}_M}$ to be the set of all polymers \mathbf{P} with the following properties:

- $(\exists \mathcal{T} \in \mathcal{S}_M)(\exists \alpha \in \Gamma_{\mathcal{T}}) \mathbf{P} \in \alpha$ (i.e., \mathbf{P} appears in some saturated configuration of a TBN using only the monomer types from M .)
- There is no partition of \mathbf{P} into two (or more) self-saturated polymers.

For example, consider the monomers $G = \{a^*, b^*, c^*, d^*\}$, $H_1 = \{a, b\}$, $H_2 = \{c, d\}$, $V_1 = \{a, c\}$, $V_2 = \{b, d\}$ and let $M = \{G, H_1, H_2, V_1, V_2\}$. The polymer basis $\mathcal{B}_{\mathcal{S}_M}$ is $\{G, H_1, H_2\}, \{G, V_1, V_2\}, \{H_1\}, \{H_2\}, \{V_1\}, \{V_2\}$. All other self-saturated polymers are unions of these.

To show that polymer bases can be characterized by Hilbert bases, we must first define some additional terms. A *conical combination* of a set of vectors is a linear combination of the vectors using only nonnegative coefficients. An *integer conical combination* of a set of vectors is a conical combination of the vectors using only integer coefficients. A *(polyhedral) convex cone* $C = \{\lambda_1 \mathbf{a}_1 + \dots + \lambda_n \mathbf{a}_n : \lambda_1, \dots, \lambda_n \geq 0\}$ is the space of all conical combinations of a finite set of vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ (and is said to be *generated* by $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$). C is *pointed* if $C \cap (-C) = \{\mathbf{0}\}$. A set of the form $\{\mathbf{x} \in \mathbb{R}^m : A\mathbf{x} \geq \mathbf{0} \text{ and } \mathbf{x} \geq \mathbf{0}\}$ is always a pointed convex cone [24].

A set is *inclusion-minimal* with respect to a property if it has no proper subset that satisfies the same property. The *Hilbert basis* of a pointed convex cone C is the unique

inclusion-minimal set of integer vectors such that every integer vector in C is an integer conical combination of the vectors in the Hilbert basis. For example, the Hilbert basis of the convex cone generated (with nonnegative real coefficients) by $(1, 3)$ and $(2, 1)$ is $\{(1, 1), (1, 2), (1, 3), (2, 1)\}$; note that $\frac{2}{5} \cdot (1, 3) + \frac{4}{5} \cdot (2, 1) = (2, 2)$, which is not an integer combination of $(1, 3)$ and $(2, 1)$, but $2 \cdot (1, 1) = (2, 2)$.

Recall that the matrix-vector product $A_M \mathbf{P}$ gives the number of exposed sites of each type in the polymer, so that $A_M \mathbf{P} \geq \mathbf{0}$ iff the polymer is self-saturated (i.e. none of the starred sites are exposed).

We are then interested in vectors contained in $\{\mathbf{P} \in \mathbb{N}^m : A_M \mathbf{P} \geq \mathbf{0}\}$. Noting that $\mathbb{N}^m = \{\mathbf{P} \in \mathbb{R}^m : \mathbf{P} \geq \mathbf{0}\} \cap \mathbb{Z}^m$, we can equivalently state that we are interested in all integer vectors contained in the pointed convex cone $\{\mathbf{P} \in \mathbb{R}^m : A_M \mathbf{P} \geq \mathbf{0} \text{ and } \mathbf{P} \geq \mathbf{0}\}$.

Theorem 59. *Let \mathcal{S}_M be a TBN schema and let A_M be the matrix representation of its monomer types. Then the polymer basis $\mathcal{B}_{\mathcal{S}_M}$ of \mathcal{S}_M is the Hilbert basis of $\{\mathbf{P} \in \mathbb{R}^m : A_M \mathbf{P} \geq \mathbf{0} \text{ and } \mathbf{P} \geq \mathbf{0}\}$.*

Proof. Note that the integer vectors in $\{\mathbf{P} \in \mathbb{R}^m : A_M \mathbf{P} \geq \mathbf{0} \text{ and } \mathbf{P} \geq \mathbf{0}\}$ are precisely the polymers that appear in saturated configurations of TBNs in \mathcal{S}_M , since $A_M \mathbf{P} \geq \mathbf{0} \iff$ polymer \mathbf{P} is self-saturated, and \mathcal{S}_M is defined to have starred sites limiting, so that a configuration is saturated if and only if each of its polymers is self-saturated.

We must show two properties to establish that $\mathcal{B}_{\mathcal{S}_M}$ is the Hilbert basis. First we must show that every polymer in saturated configurations of \mathcal{S}_M is a nonnegative integer combination of polymers in $\mathcal{B}_{\mathcal{S}_M}$. Next, to establish inclusion-minimality, we must show that no polymer can be removed from $\mathcal{B}_{\mathcal{S}_M}$ while satisfying the first property.

To see the first property, consider a polymer \mathbf{P} in a saturated configuration of some TBN in \mathcal{S}_M . If it cannot be split into multiple self-saturated polymers, then we are done since it is in $\mathcal{B}_{\mathcal{S}_M}$ (it is the integer combination consisting of one copy of itself). Otherwise, we can iteratively split \mathbf{P} into polymers $\mathbf{P}_1, \dots, \mathbf{P}_k$ that themselves cannot be split into self-saturated polymers. Then $\mathbf{P} = \mathbf{P}_1 + \dots + \mathbf{P}_k$.

To see the second property, consider a self-saturated polymer $\mathbf{P} \in \mathcal{B}_{\mathcal{S}_M}$ that can be removed while maintaining the first property. Since \mathbf{P} is an integer vector in $\{\mathbf{P} \in \mathbb{R}^m :$

$A_M \mathbf{P} \geq \mathbf{0}$ and $\mathbf{P} \geq \mathbf{0}$, \mathbf{P} is the nonnegative integer sum of some polymers remaining in $\mathcal{B}_{S_M} \setminus \{\mathbf{P}\}$. However, all polymers in \mathcal{B}_{S_M} are self-saturated, so \mathbf{P} can be partitioned into multiple self-saturated polymers. Thus \mathbf{P} is not an element of \mathcal{B}_{S_M} to begin with. \square

5.3.2 Using the polymer basis to reason about TBN behavior

The complexity of computing the polymer basis in general can be very large; however, once it is calculated, reasoning about the stable configurations becomes a simpler task. For instance, in a previous example we had $\mathcal{B}_{S_M} = \{ \{G, H_1, H_2\}, \{G, V_1, V_2\}, \{H_1\}, \{H_2\}, \{V_1\}, \{V_2\} \}$. We can see from the above basis that in saturated configurations, G can only be present one of two unsplittable polymer types: $\{G, H_1, H_2\}$ or $\{G, V_1, V_2\}$, and we can optimize the number of polymers in a configuration by taking the other monomers as singletons (which is allowed, as these singletons are in the polymer basis). More generally, reasoning about stable configurations amounts to determining the number of each polymer type to use from the polymer basis so that the union of all polymers is the TBN, while using the maximum number of polymers possible. Our software can also solve for stable configurations in this way; specifically, for a TBN \mathcal{T} , it can calculate the polymer basis (abbreviated here as \mathcal{B}) and then solve for the stable configurations using the following IP:

$$\max_{\mathbf{c} \in \mathbb{N}^{|\mathcal{B}|}} \|\mathbf{c}\|_1 \text{ subject to } \sum_{i=1}^{|\mathcal{B}|} c_i \mathcal{B}_i = \mathcal{T}$$

Alternately, one can solve for the stable systems via an augmentation approach (see [24]).

If the goal is simply to solve the STABLECONFIGS problem, we do not expect that solving for the stable configurations in this way will be more efficient than the previous formulation, as the time spent computing the Hilbert basis alone can require a great deal longer than solving via the formulation of the previous section. Instead, the true value of the basis is in its ability to describe *all* saturated configurations of a TBN.

For instance, in [10], the authors define an augmented TBN model in which a system can move between saturated configurations by two atomic operations: polymers can be pairwise merged (with an energetic penalty, i.e., higher energy) or they can be split into

two so long as no bonds are broken (with an energetic benefit, i.e., lower energy; for instance $\{a, b\}, \{a^*, b^*\}, \{a\}, \{a^*\}$ can be split into $\{a, b\}, \{a^*, b^*\}$ and $\{a\}, \{a^*\}$, whereas $\{a\}, \{a^*\}$ cannot be split). Any saturated polymer not in the basis can split into its basis components without breaking any bonds. Thus the polymer basis contains all polymers that can form in a *local* minimum energy configuration, i.e., one where no polymer can split.

When designing a TBN, the designer will typically have a sense for which polymers are to be “allowed” in local energy minima. Proving that the system observes this behavior was not previously straightforward, but we can now observe that the TBN will behave ideally when its expected behavior matches its polymer basis.

5.3.3 A case example: Circular Translator Cascade

We now discuss an example of using the polymer basis to reason about a TBN’s kinetic behavior, studying a TBN known as a *circular translator cascade*, first defined in [10]:

$$\begin{aligned} & \{\{a, b, c\}, \{b, c, d\}, \{c, d, e\}, \{d, e, f\}, \{e, f, a\}, \{f, a, b\}, \\ & \{a^*, b^*\}, \{b^*, c^*\}, \{c^*, d^*\}, \{d^*, e^*\}, \{e^*, f^*\}, \{f^*, a^*\}\} \end{aligned}$$

There are two stable configurations of this TBN, shown in Fig. 5.2.

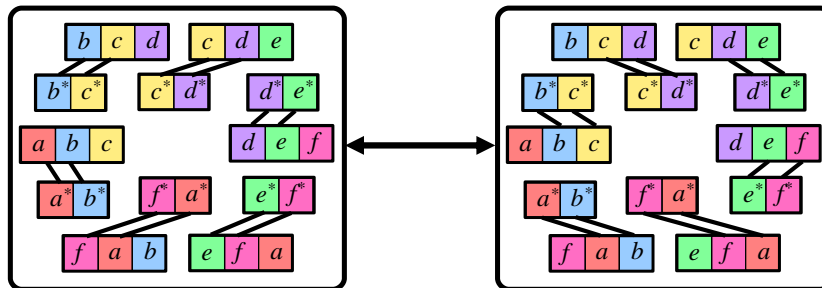


Figure 5.2: The two stable configurations of a variant of the circular translator cascade described in [10]. In the left configuration, the unstarred monomers are bound to their “left-side” companions (e.g. $\{a, b, c\}$ is bound to $\{a^*, b^*\}$), and in the right configuration, the unstarred monomers are bound to their “right-side” companions (e.g. $\{a, b, c\}$ is bound to $\{b^*, c^*\}$).

We consider now the “pathways” by which one of the stable configurations can “transition” to another. This process is described formally in [10]; here we give an intuitive description. Informally, we admit as atomic operations the ability for two polymers to merge or for one polymer to split into two polymers, so long as the resulting configuration

remains saturated. In essence, these operations are modelling the physical phenomenon of solutes colocalizing in solution before reactions occur, specifically in dilute solutions in which enthalpic bond rearrangements occur on a timescale much faster than the timescale for entropic colocalization. If many polymers must be merged in some intermediate configuration to transition between stable configurations, then since each merge is individually unlikely, the successive merges required are exponentially unlikely i.e., a large *energy barrier* exists to transition between the configurations.

The design intention of this TBN is to have two stable configurations with a large energy barrier to transition between them. For the largest possible energy barrier, the transition should require the simultaneous merging of *all* of the polymers into a single polymer as an intermediate step. However, this is not the case for the TBN of Fig. 5.2; the polymer basis gives insight into why. See [10, Section A.2] for an argument why more domain types and monomer types are required. We interpret this as a design error (in fact it actually *was* a design error in an early draft of [10]). We now explain how the error can be detected by reasoning about the polymer basis of the system, justifying that the automated computation of the polymer basis by our software enables one to automate some reasoning about the correct behavior of TBNs.

If it were true that the polymer basis contained only the 12 polymer types that are present in the two stable configurations of Fig. 5.2, then that would be sufficient to prove the high energy barrier. To see why this is true, suppose there were a locally stable intermediate configuration that is part of a lower barrier transition. Since the configuration is locally stable, it is saturated, and none of its polymers can be partitioned into self-saturated polymers. By definition, the polymer basis should then contain all of the polymers present in this intermediate configuration. However, all of the polymers in the basis have exactly two monomers, and so there must be 6 polymers in the intermediate configuration. The stable configurations also have 6 polymers, and so the intermediate configuration is also stable, but this contradicts that there are only two stable configurations.

In fact, the polymer basis for this TBN has 57 entries (determined via our software),

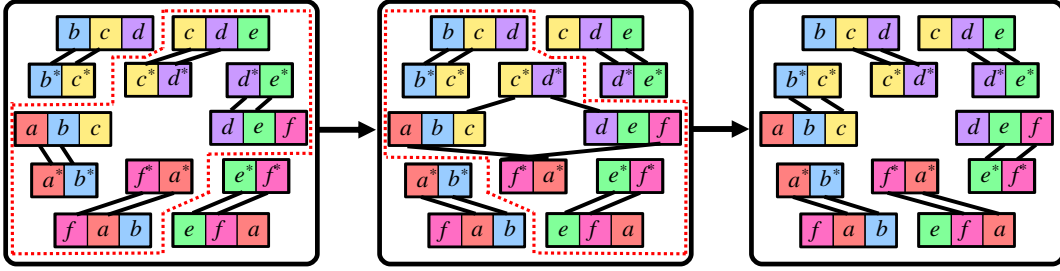


Figure 5.3: A counterexample to the claim that transitioning between stable configurations of this TBN requires the simultaneous merger of all monomers into a single polymer. Starting from the stable configuration on the left, by merging the four polymers within the red dotted outline, it is possible to re-arrange bonds and then split to the middle configuration. Then from the middle configuration, by merging the three polymers in the red dotted online, it is possible to re-arrange bonds and then split to the stable configuration on the right. Such intermediate configurations are evident by examining the elements of the polymer basis.

not 12, and we can use this basis to disprove the high energy barrier, i.e, to show that there is a sequence of merges and splits that transitions between the two stable configurations, without all monomers ever being merged into a single polymer. To discover a pathway that demonstrates the lower energy barrier, consider one unexpected entry in the polymer basis: $\mathbf{P} = \{\{a, b, c\}, \{d, e, f\}, \{c^*, d^*\}, \{f^*, a^*\}\}$. Its existence in the polymer basis tells us that there must be some saturated configuration that contains it. If we examine where these monomers were in one of the original stable configurations (Fig. 5.2, left), we see that these were originally in polymers

$$\{\{a, b, c\}, \{a^*, b^*\}\}, \quad \{\{c, d, e\}, \{c^*, d^*\}\}, \quad \{\{d, e, f\}, \{d^*, e^*\}\}, \quad \{\{f, a, b\}, \{f^*, a^*\}\}.$$

From the starting configuration, if only these four polymers were merged, then they could then iteratively split into \mathbf{P} , $\{\{c, d, e\}, \{d^*, e^*\}\}$, and $\{\{f, a, b\}, \{a^*, b^*\}\}$. Since the latter two polymers are part of the target configuration, one could now greedily merge all polymers except for these latter two and then split into the target configuration. At no point in the interim were all polymers merged together into a single polymer. The resulting pathway is illustrated in Fig. 5.3.

The difference between intended and actual barrier in this design becomes more pronounced if it is scaled up to include more site types and monomers. In [10] it is shown that by modifying the design, it is possible to achieve a linear energy barrier by using a quadratic number of site types.

Chapter 6

Grid Gate Experiment

In this section I describe the design and partial results of a lab experiment attempting to implement a 2×2 grid gate (as in Section 3.2) using DNA. This experiment was conducted at UT Austin in March of 2020; however, the experiment was cut short when the university ordered a stop to visiting student research in an attempt to prevent the spread of Covid-19. This section will then contain the results that were possible to obtain before the experiment’s end. In addition to the summary of design principles and results given here, detailed procedures and all raw data are available at <https://github.com/drhaley/Exp-2020-Mar>.

6.1 Designs

In creating an experiment that had the properties of the 2×2 grid gate, we opted to introduce geometry into the DNA design that reflects the (essentially) square positioning of the idealized grid gate. We opted to implement the large G monomer as a single strand which is bound at one (Fig. 6.2) or two positions (Fig. 6.1) by staples to prevent the large strand from opening, thus creating an inner “region” that acts as the 2×2 grid. We also designed a “zero-staple” variant (Fig. 6.3) which uses a different method to keep the large strand from opening up.

Each design has many of the same characteristics: one large *scaffold* strand which represents the G monomer in the TBN model, as well as strands representing H_1 , H_2 , V_1 , V_2 , and a catalyst. The domains are located at sites that are the intersection of an H

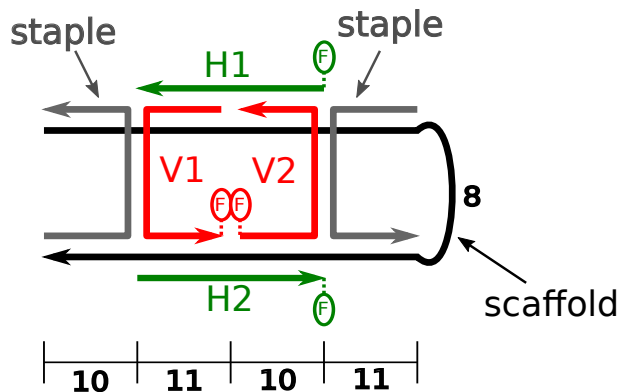


Figure 6.1: The two-staple grid gate design. A long scaffold (in black) has a center 2x2 region that can bind to either a pair of H strands (green) or a pair of V strands (red). The region is flanked on either side by staples which prevent the region from opening up during displacement. Fluorophores (circles with F) are placed in locations that will be in close proximity if the corresponding strands are on the scaffold. The numbers listed indicate the length of each section in nucleotides; the 8 on the right side indicates the length of the single-stranded section of scaffold.

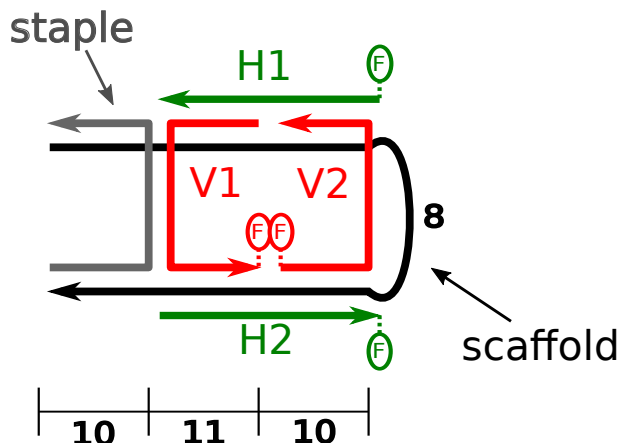


Figure 6.2: The one-staple grid gate design. A long scaffold (in black) has a center 2x2 region that can bind to either a pair of H strands (green) or a pair of V strands (red). The region is flanked on one side (left) by a staple which prevents the region from opening up on that side during displacement. The other side (right) is prevented from completely opening by the Fluorophores (circles with F) are placed in locations that will be in close proximity if the corresponding strands are on the scaffold. The numbers listed indicate the length of each section in nucleotides; the 8 on the right side indicates the length of the single-stranded section of scaffold.

monomer with a V monomer. Domains were chosen with lengths that placed crossovers as close as possible to their natural positions based upon an estimate of 21 bases for two full turns of the DNA. Single-stranded sections labelled with the numeral 8 are poly-T segments of length eight and serve to connect sections that are not adjacent in space. In each figure, lines of a single color are DNA strands with 5'-3' orientation, and each pair of adjacent strands are bound in a double helix. At any given time, either the pair of green

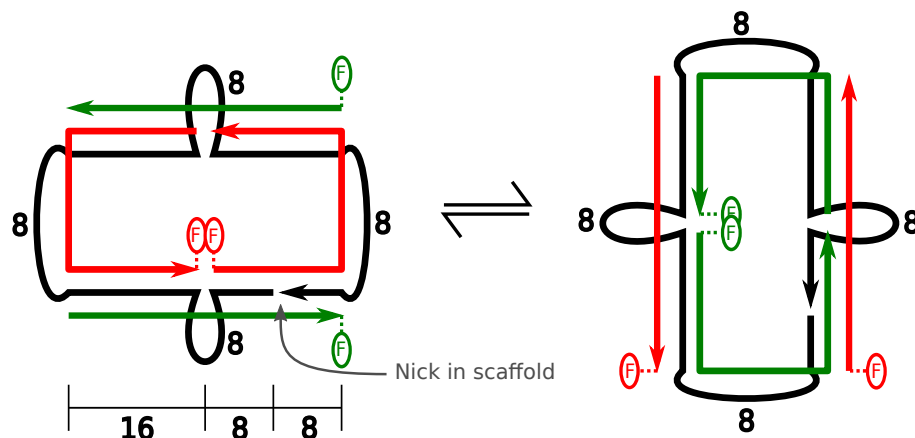


Figure 6.3: The zero-staple grid gate design. The scaffold is a long single strand (beginning and ending at the “nick” noted in the figure). The center 2x2 region that can bind to either a pair of H strands (green) or a pair of V strands (red). Because the scaffold is bound at the nick either to a V strand or H strand, the scaffold cannot open, except during displacement when we rely on the energetic penalties of the TBN to promote its closure. Here there are several loopouts of length 8. In the figure on the left, the leftmost and rightmost loopouts are necessary geometrically because of distance between the top and bottom of the scaffold in that configuration. In the figure on the right, the topmost and bottommost loopouts are necessary for the same reason. This design does not prescribe a preference between the two geometric configurations shown.

H strands is present, or the pair of red V strands is present.

The two staple and one staple designs share the same sequences – that is, the sequences used for the domains are identical, and so many of the same strands can be re-used for experiments using either design. Indeed, the only difference in sequences is in the rightmost side of their respective figures, with the one staple scaffold having a poly-T segment connecting the top and bottom helices, while the two staple scaffold has domains that complement the staple strand (and no poly-T is necessary because the twisting of the helix causes the local geometry on the right side of the figure to be in close proximity).

The zero staple design does not use any sequences borrowed from its siblings and has domains of different sizes which admit two specific geometries (shown left and right). These symmetric, competing geometries are designed specifically so that that there is no energetic favorability that biases towards one configuration or the other (that is, for the H strands or V strands to be attached to the scaffold). Ideally, for all domains to have equivalent kinetic and energetic properties, the scaffold would be one contiguous, circular strand; however, since we cannot manufacture a truly circular strand, we position a nick in the center of one of the domains. On account of its positioning, the scaffold can only

separate the ends of the nick if it binds the half-domains at its ends with separate strands – but such an arrangement is entropically disfavored.

To measure the “togetherness” of the H and V strands, we used Cy3 and Cy5 fluorophore pairs which together can produce a signal via fluorescence resonance energy transfer (FRET) [31]. Their locations are denoted on the diagrams by an F with a circle; the fluorophore at the 3' end of its strand was always Cy3 and at the 5' end was always Cy5. By exciting the combination of fluorophores at 532nm and then reading at 665nm, it is possible to measure whether the fluorophores are in close proximity.¹ Some of the experiments would use H strands without fluorophores and V strands with fluorophores, or vice versa, so that there would only be two types of strands with fluorophores present in any single experiment. However, our early experiments suggested that while FRET is useful for studying proximity 10 bases apart or more[32], we hypothesize that either this phenomenon is less pronounced when only one base apart, or else there is some rigidity to our designs that holds these fluorophores at a relative angle that prevents FRET signal propagation (as FRET is sensitive to relative angle of the fluorophores). We expect that using fluorophore/quencher pairs in the future could provide a more robust readout method than FRET.

In many DNA displacement designs, short segments of exposed, single-stranded DNA called “toeholds” are used to provide a location at which kinetic behavior can be initiated. In our designs, there was no apparent manner in which to create a toehold-facilitated design that was enthalpically neutral at each stage (without changing some of the other design constraints). For this reason, it is intended in each design that displacement should occur without toeholds. While we anticipated that while this would increase the temperature required to perform the displacement, we expected that there would still be a large energetic gap between the catalyzed pathway and the leak reaction.

All strands were ordered dry from IDT. Scaffold strands and strands with fluorophores were ordered PAGE-purified; other strands were ordered unpurified.

¹Note that despite the suggestive straight lines in our figures, DNA characteristically and predictably twists, and so the fluorophores on the H strands are in close proximity in the configurations shown.

6.2 Designing Orthogonal Domains

Central to the TBN model is the concept of orthogonal domains. In theory, a particular domain a should have strong affinity to bind with domain a^* and zero affinity to bind with any other domain (e.g. to b or to b^*). However, in DNA this ideal cannot be realized. DNA strands are chains of nucleotides with an identifying nitrogenous base typically abbreviated as A , T , C , or G . Watson-Crick complementarity states that the strongest bonds are formed between $A - T$ pairs and $C - G$ pairs, but in practice, nearly all pairs of domains will have enough of these pairs to be able to form at least some bonds, because the physical strands and the possible locations of bonds are not fixed in space. For example, if we identify a subsequence of the DNA with a TBN domain (e.g. $a = 5' - AATCCGTCTT - 3'$), a will have a strong affinity to its Watson-Crick complement $a^* = 5' - AAGACGGATT - 3'$, and a will have significant affinity to bind with other domains (e.g. $b = 5' - ACCTTTACCA - 3'$). Furthermore, in this example, a has a significant affinity to bind even with other a domains!

These and many more considerations must be taken into account when designing sequences so that their behavior matches the TBN model as closely as possible. The sequence designer was coded in Python with callouts to the packages ViennaRNA [35] and NUPACK [61]. The source code for the designer is available at <https://github.com/drhaley/SequenceDesigner>.

6.2.1 Constraints

We use a number of heuristics, which were selected with input given to us by Boya Wang and David Soloveichik. Each is chosen to greatly lessen the chance that the chosen sequences will form a strand with a strong self-structure.

- The unstarred domains use only the three-letter code A, T, C (and thus, the starred domains only use A, T, G). This greatly reduces the affinity between pairs of unstarred domains (e.g. $a - b$) or starred domains (e.g. $a^* - b^*$). It also has the effect of reducing secondary structure in the eventual strands, if those strands are constructed from entirely from unstarred domains (or entirely from starred domains).

- Substrings of the following forms are not allowed: $\{C, G\}^4$ (for instance, no *CGGG* or *GGGG*), $\{A, T\}^5$ (for instance, no *ATTAA*), *AAAA*, *TTTT*.
- Each domain must have a *C* or a *G* within three bases of each end.

The sequence designer generates random sequences obeying the heuristics and adds them to a growing collection of sequences that will be used as TBN domains in the final design if they obey the energetic constraints specified below (as verified by ViennaRNA). When enough domains are generated in this way, the designer concatenates the domains into the final strands according to the design and does a final check to ensure that these final strands have minimal self-structure.

In the below charts, affinity is the negative of free energy. All affinity values are given in kcal/mol. When two strands are compared against each other, the value considered is the difference between their combined free energy and the the sum of their free energies when separate.

The threshold values given below were selected manually by tightening and relaxing constraints until the algorithm could find the correct number of domains within an overnight time frame. In practice, most thresholds exist only to prevent the choosing of qualitative outliers. The bulk of computing work is spent enforcing an upper bound on the affinity of a prospective domain to all pairs of domains that have already been added to the set (see `undesirable_affinity_max` below). Improvements to the selection algorithm in order to speed up the optimization of this bound would be a significant boon to the sequence design software.

All designs assume that the sequence of all *T*s (the *poly-T domain*) is fixed as a first choice, and so all domains must obey the same relative energetics towards the poly-T domain as they would towards any other domain. For example, a design needing eight orthogonal domains would be calculated by finding a set of nine domains (one of which is the poly-T domain). This was done so that our designs could explicitly include poly-T segments (e.g. the 8 nucleotide loopouts) without these segments being "sticky" to the other domains.

The energetic/affinity thresholds are as follows:

- **hairpin_threshold**: upper bound for a domain's affinity to itself (self-structure of a strand with only the one domain). This check is repeated for the complemented domain.
- **desired_affinity_min**: lower bound on a domain's affinity to its complement (e.g. $a - a^*$)
- **desired_affinity_max**: upper bound on a domain's affinity to its complement (e.g. $a - a^*$)
- **undesirable_affinity_max**: upper bound on a domain's affinity to either one or two domains that are not its complement (e.g. $a - a$, $a - b$, $a - b^*$, $a - b^*c^*$, $a - aa$). This check is repeated for the complement domain.
- **strand_hairpin_threshold**: upper bound on a finished strand's self-affinity. This is performed for every strand except for the long-scaffold strand.
- **strand_hairpin_threshold**: upper bound on the finished scaffold strand's self-affinity. This strand requires a relaxed threshold on account of its extended length, because a great deal more self-structure is possible with long strands.

In the following subsections, we give the precise sequences used in the designs. Each strand is given in the 5'-3' orientation, and whitespace is added to show breaks in the TBN-level domains (note that these breaks are for conceptual purposes only and the breaks are not explicitly represented in the physical strands). Above each section representing a TBN-level domain, a suggestive name has been added to aid in the visual mapping of the sequences to the individual designs.

6.2.1.1 Grid Gate with Staples

In the lab notebook and sequence orders, this is design 8408.

Catalyst: x12 x11 x21
5'-TCCAATCCCT AACTTAACCAT ACACACAACA-3'

H1: x12 x11
5'-TCCAATCCCT AACTTAACCAT-3'

H2: x21 x22
5'-ACACACAACA TCTACTACTCA-3'

V1: x11 x21
5'-AACTTAACCAT ACACACAACA-3'

V2: x22 x12
5'-TCTACTACTCA TCCAATCCCT-3'

Left Staple: x2L x1L
5'-TTCCTTTACATATCCT TCAAATCAAATCACTT-3'

Right Staple: x1R x2R
5'-TCTTCATACTCTTTCA TCTATTCTCACCTCAA-3'

Scaffold (one-staple):

 x1L* x11* x12* poly-T
5'-AAGTGATTTGATTTGA ATGGTTAAGTT AGGGATTGGA TTTTTTTT
 x22* x21* xL2*
TGAGTAGTAGA TGTTGTGTGTA GGATATGTAAAGGAA-3'

Scaffold (two-staple):

 x1L* x11* x12* x1R*
5'-AAGTGATTTGATTTG AATGGTTAAGTT AGGGATTGGA TGAAAGAGTATGAAGA
 x2R* x22* x21* xL2*
TTGAGGTGAGAATAGA TGAGTAGTAGA TGTTGTGTGTA GGATATGTAAAGGAA-3'

This design has domains of size 10, 11 and 16. The domains of size 16 are referred to

here as *staple domains* owing to their use in the design, and the domains of size 10 and 11 are referred to here as *long domains*. The same domains were used for the 1-staple and 2-staple designs, as the two designs differ only in the concatenation of domains into the scaffold strand.

```
# thresholds for the long domains against themselves
```

```
hairpin_threshold = 0.05
```

```
desired_affinity_min = 13.0
```

```
desired_affinity_max = 1.1 * desired_affinity_min
```

```
undesirable_affinity_max = 5.5
```

```
# thresholds for the staple domains against themselves
```

```
staple_hairpin_threshold = 0.05
```

```
staple_desired_affinity_min = 20.0
```

```
staple_desired_affinity_max = 1.25 * staple_desired_affinity_min
```

```
staple_undesirable_affinity_max = 10.0 # staple against staple and/or long
```

```
# staples should not be sticky to the long domains
```

```
staple_to_long_domain_undesirable_affinity_max = 7.5
```

```
# threshold for the strands' self structure
```

```
strand_hairpin_threshold = 1.0
```

```
scaffold_hairpin_threshold = 3.0
```

6.2.1.2 Grid Gate without Staples

In the lab notebook and sequence orders, this is design 5948.

```

                x12                x11                x21
Catalyst: 5'-ACACTCTCCACCTACT CAATCAACAACCTCAT TCCCTTACTTCTACCA-3'
                x12                x11
H1:      5'-ACACTCTCCACCTACT CAATCAACAACCTCAT-3'
                x21                x22
H2:      5'-TCCCTTACTTCTACCA ACAAATCCTCCTATCC-3'
                x11                x21
V1:      5'-CAATCAACAACCTCAT TCCCTTACTTCTACCA-3'
                x22                x12
V2:      5'-ACAAATCCTCCTATCC ACACTCTCCACCTACT-3'
```

Scaffold:

```

        x22L*  poly-T        x21*        poly-T        x11*
5'-GGATTTGT TTTTTTTT TGGTAGAAGTAAGGGA TTTTTTTT ATGAGGTTGTTGATTG
        poly-T        x12*        poly-T        x22R*
TTTTTTTT AGTAGGTGGAGAGTGT TTTTTTTT GGATAGGA-3'
```

This design has domains of size 16. However, there is a nick in the scaffold strand in the middle of one of the domains, and so this effectively creates two domains each of size 8 in that region. The sequence design extends this principle by designing an appropriate number of *half-domains* of size 8 that must obey the prescribed energetics relative to the other half-domains.

```
# thresholds for the half domains against themselves
hairpin_threshold = 0.05
desired_affinity_min = 9.5
desired_affinity_max = 1.1 * desired_affinity_min
undesirable_affinity_max = 6.5
```

```
# threshold for the strands' self structure
strand_hairpin_threshold = 1.0
scaffold_hairpin_threshold = 3.0
```

6.3 Results

This section is separated into two parts to highlight discoveries as well as design-related difficulties in different stages of the experiment. Section 6.3.1 addresses the assay technique (FRET). Section 6.3.2 discusses results related to the formation of the initial grid gate complexes. All data shown is raw values; no normalization is applied.

6.3.1 Assay technique

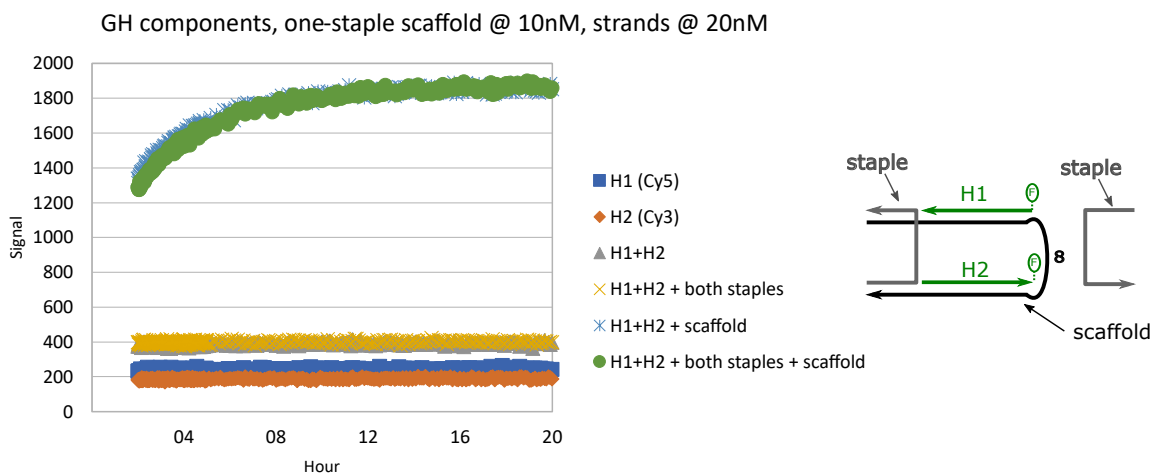


Figure 6.4: Formation experiments for the one-stapled G_H complex. 665nm emission signal is on the vertical axis, time on the horizontal axis. The full G_H complex consists of a scaffold monomer, a single staple, and both the H_1 and H_2 monomers. An additional staple with orthogonal domains (borrowed from the two-staple experiment) was also added as part of this test to ensure that none of the staples interfered with the assay technique. The legend on the right (along with the illustration at right of figure) provides a breakdown of the subsets of strands that produced which signal, with controls being expectedly low, the G_H signal being expectedly high, and the unstapled experiment (“H1+H2+scaffold”) being unexpectedly high.

Fig. 6.4 gives empirical insight into how our assay technique is intended to work. When only fluorophore Cy3 is present (indicated by “H2 Cy3”), the fluorophore is excited by input at 532nm, but the direct emission signal from Cy3 at 665nm is low. Similarly, when

Cy5 is present (indicated by “H1 Cy5”), the fluorophore does not excite easily with input at 532nm, and so the direct emission signal from Cy5 at 665nm is also low. Even when the two are in solution together (“H1+H2”, with or without the staples), the signal output is very close to the sum of signal that can be produced by the strands individually.

Instead, FRET is designed to detect if the Cy3 and Cy5 strands are in *close proximity*, as they would be in a formed complex (as in Fig. 6.2). Indeed, for the one-staple G_H version of the grid design (“H1+H2+both staples+scaffold” in Fig. 6.4), this appears quite plainly in the signal results – the formed grid produces a signal that is over four times that of the controls (at 20nM concentration of the fuel strands). Interestingly, it also produces a similarly high signal even without the staple, which geometrically would result in an “opened” configuration of the gate in which Cy3 and Cy5 were 8 bases apart. This data is promising in that it suggests that there is a great deal of leniency in “how close” the fluorophores need to be for detection, but it turns out that there are complications with very close proximity (0 bases apart) which we will discuss further below. A close inspection of the Fig. 6.2 also reveals that there seem to be some sort of non-trivial kinetics happening even as late as eight hours after the initial complex formation – we will discuss the possible reasons and some different annealing schedules below.

Unfortunately, such strong signal differences in the formed complexes were not universal, as seen in Fig. 6.5 and Fig. 6.6. The G_H complex with one staple (orange data series in Fig. 6.5) far exceeds its controls in signal output, but the remainder of the G_H designs produced signals that were only 40%-70% above control. The situation is arguably worse with the G_V complexes, for while there was some success with the no-staple design, many signals did not appreciably exceed that of the controls.

With only the signals from formation attempts, it would not have been possible to definitively conclude that our assay technique was responsible for these discrepancies, for it could be that the intended complexes were simply not forming (or impartially forming), in which case they should not produce a strong signal. To troubleshoot this, we ordered new strands denoted “ComplementH” and “ComplementV” in their respective figures. The ComplementH (alt. ComplementV) strands are each one strand that is the

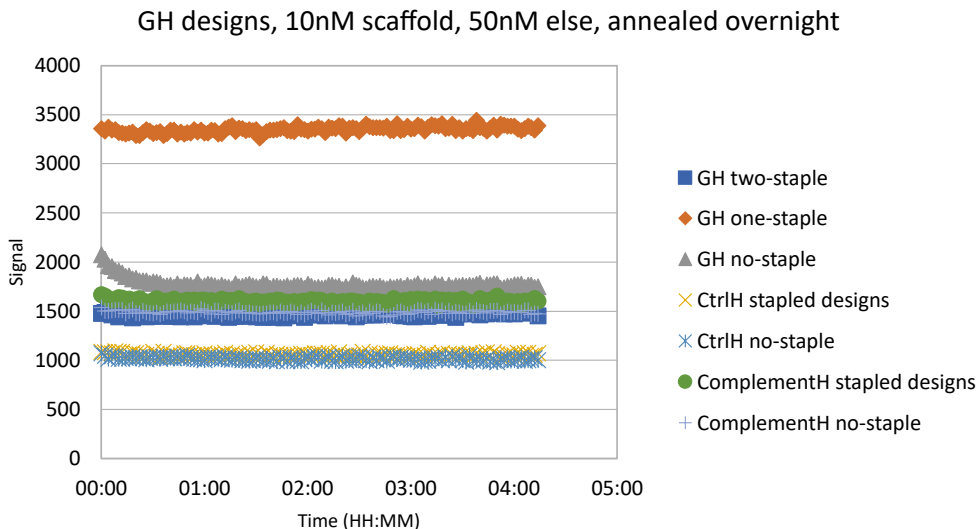


Figure 6.5: Formation experiments for each of the G_H complex designs. The one-staple G_H design produces an expectedly high signal, but many of the other designs produced signals that only marginally exceeded the controls (“CtrlH”). Two additional experiments appear here using strands that bind to bring H_1 and H_2 into close proximity (with their fluorophores one base apart), these are labelled as “ComplementH”, and are unexpectedly low.

exact Watson-Crick complement that binds to H_1 and H_2 (alt. V_1 and V_2) of the design, placing the Cy3 and Cy5 fluorophores at a distance of one base apart (as they would be in the overall design). Our expectation was that this would produce a maximal signal, but instead the signal was similarly lackluster, indicating that FRET does not produce a large signal when the fluorophores are close together, contrary to our initial expectations from Fig. 6.4. Our conclusion was that FRET as an assay technique was insufficient for our purposes, and the likely cause was that the especially close proximity of the fluorophores in the design was somehow forcing them into a geometric arrangement in which they were (or were close to) perpendicular. We ordered replacement strands that would assay via a fluorophore/quencher method instead, which is not sensitive to relative angle, but these new strands did not arrive before the experiment was brought to its untimely close.

6.3.2 Stable formation of Grid Gate

An early concern in the experiment was the appearance of kinetic artifacts in the data – changes in signal after the supposed initial formation of the complex. There were two likely contributors to this phenomenon – the annealing schedule and the relative concentration

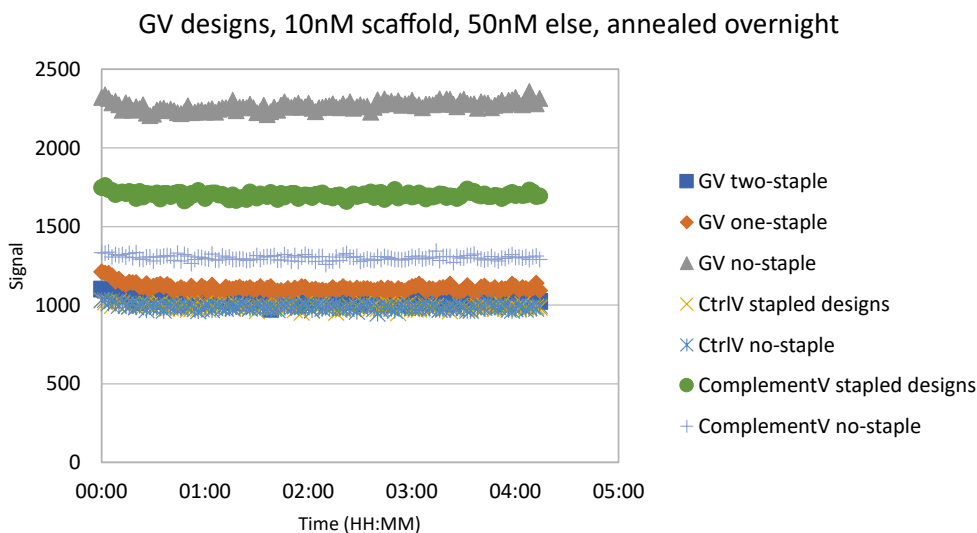


Figure 6.6: Formation experiments for each of the G_V complex designs. The zero-staple G_V design produces an expectedly high signal, although not as high as G_H in the one-staple design. Many of the other designs produced signals that only marginally exceeded the controls (“CtrlV”). Two additional experiments appear here using strands that bind to bring V_1 and V_2 into close proximity (with their fluorophores one base apart), these are labelled as “ComplementV”, and are unexpectedly low.

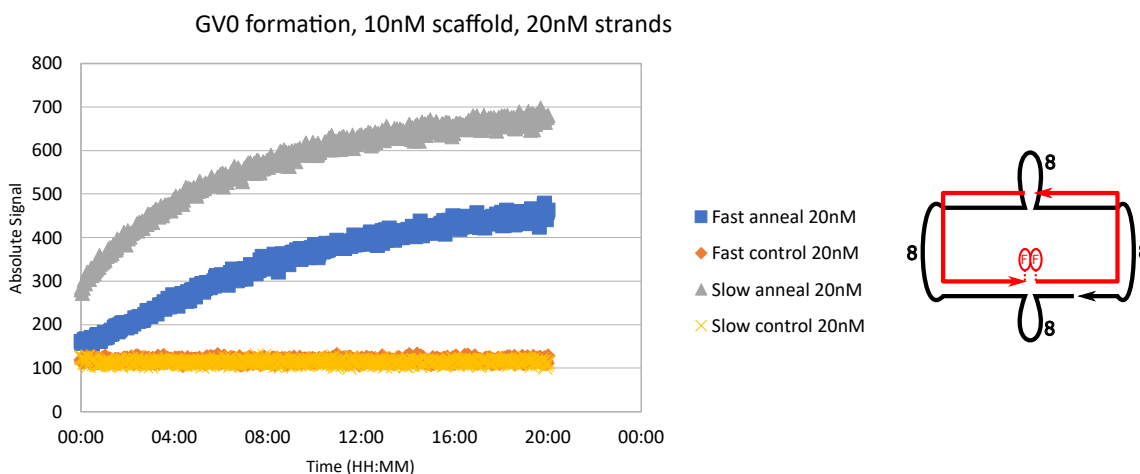


Figure 6.7: Formation experiments for the zero-staple G_V design. The V_1 and V_2 strands were kept at a 2:1 concentration ratio relative to the scaffold. Two annealing schedules were tested – a “fast” anneal over approximately one hour, and a “slow” anneal over approximately nine hours. The data shows a great deal of change (evidence of kinetic behavior) over many hours after the initial anneal is finished.

of the scaffold strand to the fuel strands.

In Fig. 6.7 and Fig. 6.8, we see the effect of both annealing schedule and relative concentration on the early kinetics. We chose to focus on the zero-staple G_V design

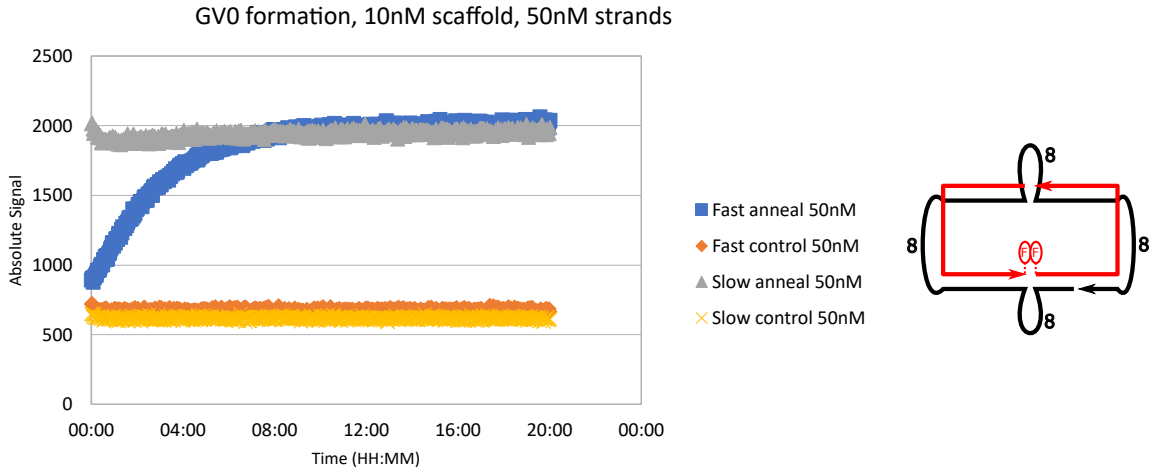


Figure 6.8: Formation experiments for the zero-staple G_V design. The V_1 and V_2 strands were kept at a 5:1 concentration ratio relative to the scaffold. Two annealing schedules were tested – a “fast” anneal over approximately one hour, and a “slow” anneal over approximately nine hours. The data from the fast anneal shows a great deal of change (evidence of kinetic behavior) over many hours after the initial anneal is finished, but the slow anneal shows very little change, suggesting that the formation of the complex was able to complete during the slower annealing period.

because while the signal was not as strong as G_H one-staple, it had more visible kinetics that we wanted to investigate. The data points labelled “fast” used the following annealing schedule: heat to 90C, hold for 10 minutes, then reduce by 0.1C every 6 seconds until 20C (this fast anneal takes slightly longer than one hour). The data points labelled “slow” also heated to 90C and held for 10 minutes, but reduced the temperature by 0.1C every 48 seconds until 20C (this slow anneal takes approximately $9\frac{1}{2}$ hours, and we would typically run this anneal overnight between lab sessions).

In each graph, we see that the annealing schedule and concentration has a pronounced effect on early kinetics, and the control signals remain steady in all cases. The slow anneal shows a marked reduction in early kinetics, and keeping the fuel strands at a ratio of 5:1 over the scaffold strand also gives a strong reduction in kinetic behavior as compared to its 2:1 counterpart.

While the effect of kinetics became quite muted in the zero-staple G_V experiment, many other complexes in the various designs still showed significant early kinetics, although this was harder to troubleshoot as these experiments had very low signal-to-noise ratio. Given more time, more experiments might have given more insight to the phe-

nomenon; however, we did have time to try two things during the time of kinetic activity that both proved to have no discernable effect on the kinetics: (1) using a clean pipette to vigorously agitate the contents of the well (by repeatedly removing and re-adding solution to the same well), and (2) removing the plate from the reader and vigorously shaking it before placing it back in the reader.

We have also hypothesized that there may be some connection between the kinetics and the post-annealing treatment of the solution. After annealing, the solution is not guaranteed to be well-mixed (for instance, evaporated liquid can condense on the inside of the tube lid). Post-anneal, we vortexed these solutions to re-mix their contents. It is possible, although we think it unlikely, that such vortexing may disturb the delicate structures of the gate complex, and this would explain a portion of the kinetics observed. An alternate treatment might include tapping tubes on the bench to dislodge the condensed fluid and/or using a few repeated pulses via a clean pipette to more gently mix the contents.

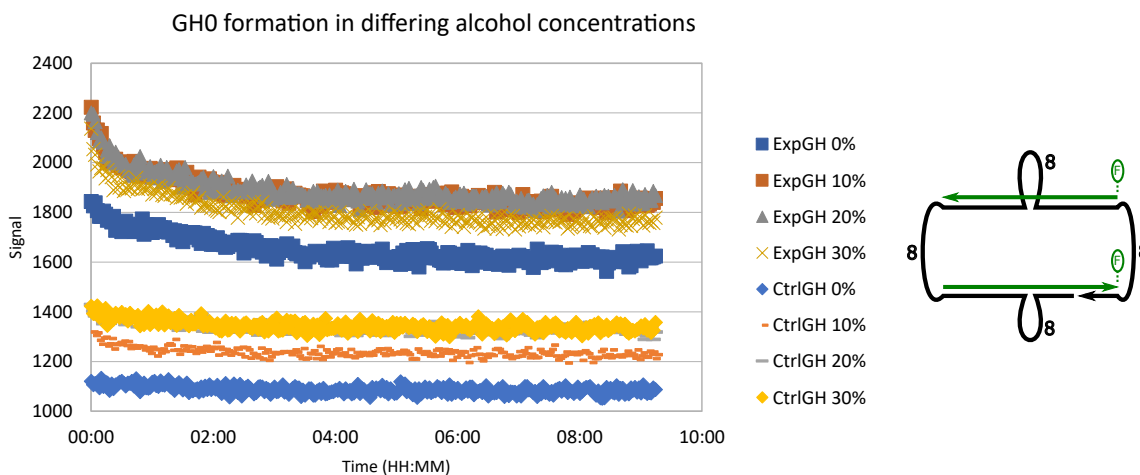


Figure 6.9: Formation experiments for the zero-staple G_H design in varying concentrations of ethanol. The H_1 and H_2 strands were kept at a 5:1 concentration ratio relative to the scaffold and the solution was annealed overnight. The presence of ethanol increases both the control and experimental signals.

As our intention with the design was to have a toeholdless displacement pathway, we expected that in the post-formation experiments, the structure of the formed complexes would need to be robust to increased temperature and/or increased ethanol concentration. This was because while this toeholdless version of displacement can occur, it is much

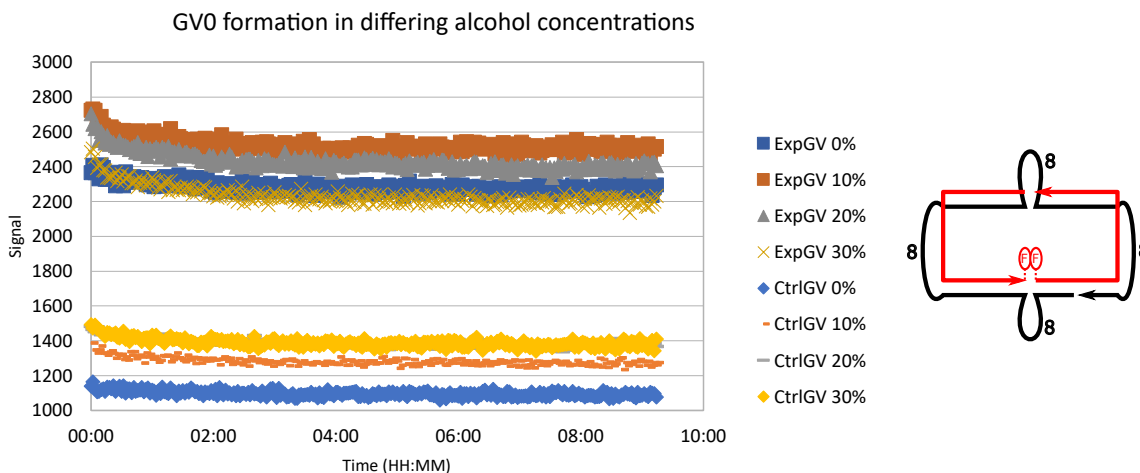


Figure 6.10: Formation experiments for the zero-staple G_V design in varying concentrations of ethanol. The V_1 and V_2 strands were kept at a 5:1 concentration ratio relative to the scaffold and the solution was annealed overnight. The presence of ethanol increases the control signal, and at lower concentration also increase the experimental signal. At higher ethanol concentration the difference between experimental signal and control signal is measurably lessened.

less likely to occur than displacement using toeholds, and so we did not enter with the expectation that it would occur at close to room temperature with typical salt concentrations. Higher temperatures have the relative effect of weakening the bonds between DNA nucleotides; ethanol can also be used to accomplish this.

Fig. 6.9 and Fig. 6.10 show attempts to form the zero-staple versions of G_H and G_V (respectively) in differing concentrations of ethanol. While at first it appears that a small amount of ethanol in solution causes a slightly larger signal (suggesting a more stable complex), a further inspection shows that the control signals are also increased by a similar amount. Indeed, for ethanol concentrations tested (0% to 30%), increasing amounts of ethanol did not improve the signal-to-noise ratio, and at concentrations closer to 30%, demonstrably reduced it. In further experiments it might have been possible to try to narrow in on a concentration range in which the signal-to-noise ratio was largely undisturbed, but as the ethanol was clearly affecting our assay technique via a mechanism that was not immediately clear, the potential benefit of using ethanol seemed offset by it being a potential source of error in future experiments.

6.4 Next Steps

A solid assay technique being necessary evidence to proper complex formation, we recommend that future attempts use fluorophore/quencher pairs rather than FRET. In particular, FRET is sensitive to relative angle of fluorophores, and there is a more limited selection of fluorophores for which FRET can be effective. The intention of using FRET in this experiment was to measure proximity of the strands to each other, and thus evidence the formation of the complex by a stronger signal; however, in fluorophore/quencher pair proximity is evidenced by an absence of signal. Our initial thoughts in including FRET in the design were to maximize signal-to-noise, and so we entered with the expectation that Cy3 and Cy5 control signals would be very low, so that the FRET signal would be relatively high. While this was the case in some of our designs (one-staple G_H in particular), the sensitivity of the technique to geometry makes it unpredictable enough that it does not serve well in its intended purpose. Furthermore, FRET signals using Cy3 and Cy5 are strongly temperature dependent [34], and so it should not be expected that these signals should behave consistently at the different temperatures required in later stages of the experiment. On the other hand, by using fluorophore/quencher pairs, we expect a high control signal and a much lower experimental signal, and if the signal fidelity is improved by this alternate assay technique, then it may not be necessary to run experiments in concentration ratios as high as 5:1, meaning that less fuel strands are needed, improving the signal-to-noise ratio substantially.

With an improved assay technique established, and the early experiments re-run to confirm the formation of the complexes, the next step in the experiment would be to analyze temperature robustness of the complexes, by first forming them and then testing the temperature at which they disassemble (the melting temperature). This establishes the “high barrier” that the grid gate is designed originally to have. This temperature is likely to vary across the different designs due to intuitive geometric stability provided by the staples as well as the differing domain lengths between the stapled and zero-staple designs.

With a uncatalyzed melting temperature established, the next step would be to demon-

strate that displacement can occur at a temperature below the melting temperature. As designed, the experiment would test this by trying to displace strands with fluorophores with versions that do not have fluorophores (and vice versa). For instance, an ideal experiment would begin with a formed grid (say G_H) and a positive signal indicating the initial presence of the correct complex, and then be followed by the addition of H_1 strands that did not have a fluorophore. If displacement occurs, then the experimental signal should begin to approach the control. Similarly, if starting from a version of the complex using H_1 without the fluorophore and then adding H_1 strands with fluorophores, the experimental signal should move away from the control signal. The time scale of the displacement is evidenced by the convergence of these two experiments.

Lastly, once these experiments have been conducted, the final experiments can be attempted, specifically adding all of the strands and seeing at what temperature a G_H to G_V conversion is affected, with and without the catalyst monomer.

REFERENCES

- [1] <http://web.archive.org/web/20050308115423/http://www.icparc.ic.ac.uk/~wh/golf/> Our case is the diagonal of the matrix shown on this page. See also: <http://mathworld.wolfram.com/SocialGolferProblem.html>, https://en.wikipedia.org/wiki/Graeco-Latin_square, <https://math.stackexchange.com/questions/186298/equal-sized-partitions-without-overlap>.
- [2] 4ti2 team. 4ti2 – a software package for algebraic, geometric and combinatorial problems on linear spaces. <https://4ti2.github.io>.
- [3] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006.
- [4] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.
- [5] Charles H Bennett. Time/space trade-offs for reversible computation. *SIAM Journal on Computing*, 18(4):766–776, 1989.
- [6] Michael Blondin, Javier Esparza, and Stefan Jaax. Large flocks of small birds: On the minimal size of population protocols. In Rolf Niedermeier and Brigitte Vallée, editors, *STACS 2018: 35th Symposium on Theoretical Aspects of Computer Science*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [7] Michael Blondin, Javier Esparza, and Stefan Jaax. Peregrine: A tool for the analysis of population protocols. In *CAV: International Conference on Computer Aided Verification*, pages 604–611. Springer, 2018.
- [8] Michael Blondin, Javier Esparza, Stefan Jaax, and Philipp J Meyer. Towards efficient verification of population protocols. In *PODC: Proceedings of the ACM Symposium on Principles of Distributed Computing*, pages 423–430, 2017.
- [9] Keenan Breik, Cameron Chalk, David Doty, David Haley, and David Soloveichik. Programming substrate-independent kinetic barriers with thermodynamic binding networks. In *International Conference on Computational Methods in Systems Biology*, pages 203–219. Springer, 2018.
- [10] Keenan Breik, Cameron Chalk, David Haley, David Doty, and David Soloveichik. Programming substrate-independent kinetic barriers with thermodynamic binding networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019. to appear. Special issue of invited papers from CMSB 2018.
- [11] Keenan Breik, Chris Thachuk, Marijn Heule, and David Soloveichik. Computing properties of stable configurations of thermodynamic binding networks. *Theoretical Computer Science*, 785:17–29, 2019.

- [12] Keenan Breik, Chris Thachuk, Marijn Heule, and David Soloveichik. Computing properties of stable configurations of thermodynamic binding networks. *Theoretical Computer Science*, 785:17–29, 2019.
- [13] Luca Cardelli, Milan Češka, Martin Fränzle, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, and Max Whitby. Syntax-guided optimal synthesis for chemical reaction networks. In *CAV: International Conference on Computer Aided Verification*, pages 375–395. Springer, 2017.
- [14] Luca Cardelli, Marta Kwiatkowska, and Luca Laurenti. Stochastic analysis of chemical reaction networks using linear noise approximation. *Biosystems*, 149:26–33, 2016.
- [15] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Forward and backward bisimulations for chemical reaction networks. In *CONCUR 2015*, 2015.
- [16] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Syntactic markovian bisimulation for chemical reaction networks. In *Models, Algorithms, Logics and Tools*, pages 466–483. Springer, 2017.
- [17] Cameron Chalk, Jacob Hendricks, Matthew J Patitz, and Michael Sharp. Thermodynamically favorable computation via tile self-assembly. In *International Conference on Unconventional Computation and Natural Computation*, pages 16–31. Springer, 2018.
- [18] Ioannis Chatzigiannakis, Othon Michail, and Paul G Spirakis. Algorithmic verification of population protocols. In *SSS: Symposium on Self-Stabilizing Systems*, pages 221–235. Springer, 2010.
- [19] Yuan-Jyue Chen, Benjamin Groves, Richard A. Muscat, and Georg Seelig. DNA nanotechnology from the test tube to the cell. *Nature Nanotechnology*, 10:748–760, 2015.
- [20] Julien Clément, Carole Delporte-Gallet, Hugues Fauconnier, and Mihaela Sighireanu. Guidelines for the verification of population protocols. In *ICDCS: 31st International Conference on Distributed Computing Systems*, pages 215–224. IEEE, 2011.
- [21] Charles J Colbourn. *CRC handbook of combinatorial designs*. CRC press, 2010.
- [22] Anne Condon, Bonnie Kirkpatrick, and Ján Maňuch. Design of nucleic acid strands with long low-barrier folding pathways. *Natural computing*, 16(2):261–284, 2017.
- [23] Wojciech Czerwiński, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for Petri nets is not elementary. In *STOC 2019: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 24–33, 2019.
- [24] Jesús A De Loera, Raymond Hemmecke, and Matthias Köppe. *Algebraic and geometric ideas in the theory of discrete optimization*. SIAM, 2012.

- [25] David Doty, Trent A. Rogers, David Soloveichik, Chris Thachuk, and Damien Woods. Thermodynamic binding networks. In Robert Brijder and Lulu Qian, editors, *DNA Computing and Molecular Programming: 23rd International Conference*, pages 249–266. Springer, 2017.
- [26] Javier Esparza, Pierre Ganty, Rupak Majumdar, and Chana Weil-Kennedy. Verification of immediate observation population protocols. In Sven Schewe and Lijun Zhang, editors, *29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018, Beijing, China*, volume 118 of *LIPICs*, pages 31:1–31:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [27] Javier Esparza, Mikhail Raskin, and Chana Weil-Kennedy. Parameterized analysis of immediate observation Petri nets. In *International Conference on Applications and Theory of Petri Nets and Concurrency*, pages 365–385. Springer, 2019.
- [28] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 7.0. Technical report, Optimization Online, March 2020.
- [29] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020. <http://www.gurobi.com>.
- [30] David Haley and David Doty. Computing properties of thermodynamic binding networks: An integer programming approach. *arXiv preprint arXiv:2011.10677*, 2020.
- [31] Asif Iqbal, Sinan Arslan, Burak Okumus, Timothy J Wilson, Gerard Giraud, David G Norman, Taekjip Ha, and David MJ Lilley. Orientation dependence in fluorescent energy transfer between cy3 and cy5 terminally attached to double-stranded nucleic acids. *Proceedings of the National Academy of Sciences*, 105(32):11176–11181, 2008.
- [32] Asif Iqbal, Sinan Arslan, Burak Okumus, Timothy J Wilson, Gerard Giraud, David G Norman, Taekjip Ha, and David MJ Lilley. Orientation dependence in fluorescent energy transfer between cy3 and cy5 terminally attached to double-stranded nucleic acids. *Proceedings of the National Academy of Sciences*, 105(32):11176–11181, 2008.
- [33] Robert Johnson, Qing Dong, and Erik Winfree. Verifying chemical reaction network implementations: A bisimulation approach. *Theoretical Computer Science*, 765:3–46, 2019.
- [34] Xun Li, Yandong Yin, Xinxing Yang, Zeyong Zhi, and Xin Sheng Zhao. Temperature dependence of interaction between double stranded dna and cy3 or cy5. *Chemical Physics Letters*, 513(4-6):271–275, 2011.

- [35] Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdisen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(1), 2011.
- [36] Ján Maňuch, Chris Thachuk, Ladislav Stacho, and Anne Condon. NP-completeness of the energy barrier problem without pseudoknots and temporary arcs. *Natural Computing*, 10(1):391–405, 2011.
- [37] Leigh-Anne Mathieson and Anne Condon. On low energy barrier folding pathways for nucleic acid sequences. In *International Conference on DNA Computing and Molecular Programming*, pages 181–193. Springer, 2015.
- [38] Ernst W Mayr and Jeremias Weihmann. A framework for classical Petri net problems: Conservative Petri nets as an application. In *International Conference on Applications and Theory of Petri Nets and Concurrency*, pages 314–333. Springer, 2014.
- [39] Robin Milner. *Communication and concurrency*, volume 84. Prentice hall Englewood Cliffs, 1989.
- [40] Cristopher Moore and Stephan Mertens. *The Nature of Computation*. OUP Oxford, 2011.
- [41] Steven R Morgan and Paul G Higgs. Barrier heights between ground states in a model of RNA secondary structure. *Journal of Physics A: Mathematical and General*, 31(14):3153, 1998.
- [42] Niall Murphy, Rasmus Petersen, Andrew Phillips, Boyan Yordanov, and Neil Dalchau. Synthesizing and tuning stochastic chemical reaction networks with specified behaviours. *Journal of The Royal Society Interface*, 15(145):20180283, 2018.
- [43] Jun Pang, Zhengqin Luo, and Yuxin Deng. On automatic verification of self-stabilizing population protocols. In *TASE: 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*, pages 185–192, 2008.
- [44] Laurent Perron and Vincent Furnon. OR-tools. <https://developers.google.com/optimization>.
- [45] Lulu Qian and Erik Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196, 2011.
- [46] Mikhail A. Raskin, Chana Weil-Kennedy, and Javier Esparza. Flatness and complexity of immediate observation Petri nets. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPICs*, pages 45:1–45:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

- [47] Paul W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.
- [48] Seung Woo Shin, Chris Thachuk, and Erik Winfree. Verifying chemical reaction network implementations: A pathway decomposition approach. *Theoretical Computer Science*, 765:67–96, 2019.
- [49] Richard P. Stanley. *Enumerative Combinatorics: Volume 1*. Cambridge University Press, New York, NY, USA, 2nd edition, 2011. see <http://www-math.mit.edu/~rstan/ec/ec1.pdf>.
- [50] Jun Sun, Yang Liu, Jin Song Dong, and Jun Pang. Pat: Towards flexible verification under fairness. In *CAV: International Conference on Computer Aided Verification*, pages 709–714. Springer, 2009.
- [51] Gaston Tarry. Le problème des 36 officiers. *Compte Rendu de l'Association Française pour l'Avancement des Sciences*, 1:122–123, 1900. see https://en.wikipedia.org/wiki/Thirty-six_officers_problem.
- [52] Chris Thachuk. *Space and energy efficient molecular programming and space efficient text indexing methods for sequence alignment*. PhD thesis, University of British Columbia, 2013.
- [53] Chris Thachuk and Anne Condon. Space and energy efficient computation with DNA strand displacement systems. In *International Conference on DNA Computing and Molecular Programming*, pages 135–149. Springer, 2012.
- [54] Chris Thachuk, Ján Maňuch, Arash Rafiey, Leigh-Anne Mathieson, Ladislav Stacho, and Anne Condon. An algorithm for the energy barrier problem without pseudoknots and temporary arcs. In *Biocomputing 2010*, pages 108–119. World Scientific, 2010.
- [55] Chris Thachuk, Erik Winfree, and David Soloveichik. Leakless DNA strand displacement systems. In *International Workshop on DNA-Based Computers*, pages 133–153. Springer, 2015.
- [56] Andrew C Trapp and Oleg A Prokopyev. Solving the order-preserving submatrix problem via integer programming. *INFORMS Journal on Computing*, 22(3):387–400, 2010.
- [57] Boya Wang, Chris Thachuk, Andrew D Ellington, Erik Winfree, and David Soloveichik. Effective design principles for leakless strand displacement systems. *Proceedings of the National Academy of Sciences*, 115(52):E12182–E12191, 2018.
- [58] Boya Wang, Chris Thachuk, Andrew D Ellington, Erik Winfree, and David Soloveichik. Effective design principles for leakless strand displacement systems. *Proceedings of the National Academy of Sciences*, 115(52):E12182–E12191, 2018.

- [59] Damien Woods[†], David Doty[†], Cameron Myhrvold, Joy Hui, Felix Zhou, Peng Yin, and Erik Winfree. Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature*, 567(7748):366–372, 2019. [†]joint first authors.
- [60] Peng Yin, Harry MT Choi, Colby R Calvert, and Niles A Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451(7176):318, 2008.
- [61] Joseph N. Zadeh, Conrad D. Steenberg, Justin S. Bois, Brian R. Wolfe, Marshall B. Pierce, Asif R. Khan, Robert M. Dirks, and Niles A. Pierce. NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173, 2011.
- [62] David Yu Zhang, Andrew J. Turberfield, Bernard Yurke, and Erik Winfree. Engineering entropy-driven reactions and networks catalyzed by DNA. *Science*, 318(5853):1121–1125, 2007.