Thermodynamic Binding Networks

David Doty[1], Trent A. Rogers[2], David Soloveichik[3](✉), Chris Thachuk[4],
and Damien Woods[5]

[1] University of California, Davis, Davis, USA
`doty@ucdavis.edu`
[2] University of Arkansas, Fayetteville, USA
`tar003@email.uark.edu`
[3] University of Texas at Austin, Austin, USA
`david.soloveichik@utexas.edu`
[4] California Institute of Technology, Pasadena, USA
`thachuk@caltech.edu`
[5] Inria, Paris, France
`damien.woods@inria.fr`

Abstract. Strand displacement and tile assembly systems are designed to follow prescribed kinetic rules (i.e., exhibit a specific time-evolution). However, the expected behavior in the limit of infinite time—known as thermodynamic equilibrium—is often incompatible with the desired computation. Basic physical chemistry implicates this inconsistency as a source of unavoidable error. Can the thermodynamic equilibrium be made consistent with the desired computational pathway? In order to formally study this question, we introduce a new model of molecular computing in which computation is driven by the thermodynamic driving forces of enthalpy and entropy. To ensure greatest generality we do not assume that there are any constraints imposed by geometry and treat monomers as unstructured collections of binding sites. In this model we design Boolean AND/OR formulas, as well as a self-assembling binary counter, where the thermodynamically favored states are exactly the desired final output configurations. Though inspired by DNA nanotechnology, the model is sufficiently general to apply to a wide variety of chemical systems.

D. Doty—Supported by NSF grant CCF-1619343.

T.A. Rogers—Supported by the NSF Graduate Research Fellowship Program under Grant No. DGE-1450079, NSF Grant CAREER-1553166, and NSF Grant CCF-1422152.

D. Soloveichik—Supported by NSF grants CCF-1618895 and CCF-1652824.

C. Thachuk—Supported by NSF grant CCF-1317694.

D. Woods—Part of this work was carried out at California Institute of Technology. Supported by Inria (France) as well as National Science Foundation (USA) grants CCF-1219274, CCF-1162589, CCF-1317694.

R. Brijder and L. Qian (Eds.): DNA 23 2017, LNCS 10467, pp. 249–266, 2017.
DOI: 10.1007/978-3-319-66799-7_16

1 Introduction

Most of the models of computing that have come to prominence in molecular programming are essentially kinetic. For example, models of DNA strand displacement cascades and algorithmic tile assembly formalize desired interaction rules followed by certain chemical systems over time [8,12]. Basing molecular computation on kinetics is not surprising given that computation itself is ordinarily viewed as a *process*. However, unlike electronic computation, where thermodynamics holds little sway, chemical systems operate in a Brownian environment [2]. If the desired output happens to be a meta-stable configuration, then thermodynamic driving forces will inexorably drive the system toward error. For example, *leak* in most strand displacement systems occurs because the thermodynamic equilibrium of a strand displacement cascade favors incorrect over the correct output, or does not discriminate between the two [11]. In DNA tile assembly, we typically must find and exploit kinetic barriers to unseeded growth to enforce that growth happens only from seed assemblies, otherwise thermodynamically favored assemblies will quickly form that are not the intended self-assembly program execution from the seed/input [1,10].

We introduce the Thermodynamic Binding Networks (TBN) model, where information processing is due entirely to the thermodynamic tradeoff between *entropy* and *enthalpy*, and not any particular reaction pathway. In most experimental systems considered in DNA nanotechnology, thermodynamic favorability is determined by a tradeoff between: (1) the number of base pairs formed or broken (all else being equal, a state with more base pairs bound is more favorable); (2) the number of separate complexes (all else being equal, a state with more free complexes is more favorable). We use the terms enthalpy and entropy to describe (1) and (2) respectively (although this use does not perfectly align with their physical definitions, see Sect. 2). Intuitively, the entropic benefit of configurations with more separate complexes is due to additional microstates, each describing the independent three-dimensional positions of each complex. Although the general case of a quantitative trade-off between enthalpy and entropy is complex, we develop an elegant formulation based on the limiting case in which enthalpy is infinitely more favorable than entropy. Intuitively, this limit corresponds to increasing the strength of binding, while diluting (increasing the volume), such that the ratio of binding to unbinding rate goes to infinity. Systems studied in molecular programming can in principle be engineered to arbitrarily approach this limit. Indeed, this is the regime previously studied in the context of leak reduction for strand displacement cascades [11]. Figure 1 shows a simple TBN, which can exist in 9 possible binding configurations. The favored (stable) configuration is the one that, among the maximally bound ones (bottom row), maximizes the number of separate complexes (bottom right).

As a central choice in seeking a general theory, we *dispense with geometry*: formally, we treat monomers simply as multisets of binding sites (domains). Viewed in the context of strand displacement, this abstracts away secondary structure (the order of domains on a strand), allowing us to represent arbitrary molecular arrangements such as pseudoknots [4], and handle non-local error

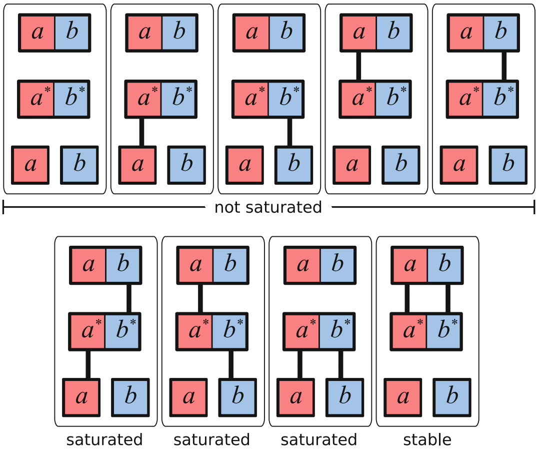

Fig. 1. An example TBN $\mathcal{T} = (\mathcal{D}, \mathcal{M})$. $\mathcal{D} = \{a, b\}$ and $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4\}$, where monomers $\mathbf{m}_1 = \{a, b\}, \mathbf{m}_2 = \{a^*, b^*\}, \mathbf{m}_3 = \{a\}$, and $\mathbf{m}_4 = \{b\}$. Note that the order of domains does not matter (thus, $\{a, b\} = \{b, a\}$). There are nine distinct configurations for the monomer collection $\vec{\mathbf{c}} = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4\}$ consisting of a single copy of each of these monomers. The five in the top row are *not saturated* meaning that they do not maximize the number of bound domains, whereas the four configurations in the bottom row are all *saturated*. In addition to being saturated, the configuration in the bottom right is *stable* as it maximizes the number of separate complexes (3) among all saturated configurations (the other saturated configurations have 2).

modes such as spurious remote toeholds [5]. In the context of tile self-assembly, we consider configurations in which binding does not follow the typical regular lattice structure. Since the TBN model does not rely on geometric constraints to enforce correct behavior, showing that specific undesired behavior is prevented by enthalpy and entropy alone leads to a stronger guarantee. Thus, for example proving leaklessness in this model would imply that even if pseudoknots, or other typically disallowed structures form, we would still have little leak. Indeed, by casting aside the vagaries of DNA biophysics (e.g., persistence length, number of bases per turn, sequence dependence on binding strength, etc.), our aim is to develop a general theory of programmable systems based on molecular bonds, a theory that will apply to bonds based on other substrates such as proteins, base stacking, or electric charge.

After introducing the TBN model in Sect. 2, we give results on Boolean circuit-based and self-assembly-based computation. In Sect. 3 we show how to construct AND and OR gates where the thermodynamically favored configurations encode the output. We develop provable guarantees on the entropic penalty that must be overcome to produce an incorrect 1 output, showing how the logic gates can be designed to make the penalty arbitrarily large. Although completely modular reasoning seems particularly tough in this model, we develop a proof technique based on logically excising domains to handle the composition of Boolean gates—specifically trees of AND gates. Further work is needed to generalize these results to arbitrary circuits.

In Sect. 4 we look at self-assembly, beginning with questions about large assemblies. On the one hand we exhibit a class of TBNs with thermodynamical stable assemblies (with simple 'tree' connectivity) of size exponential in the number of constituent monomer types. On the other hand, we show that this bound is essentially tight by giving an exponential size upper bound on the size of stable assemblies in general. These self-assembly results, along with the binary counter result below, tell us that *monomer-efficient* self-assembly is indeed possible within this model, but that (somewhat surprisingly for a model that favors enthalpy infinitely over entropy) super-exponential size polymers are necessarily unstable, even if they are self-assemblable in kinetic-based models.

For clarity of thought in separating the computational power of thermodynamics and kinetics, throughout much of this paper we do not identify any particular kinetic pathway leading to the desired TBN stable state. Of course real-world physical systems do not operate at thermodynamic equilibrium, and might take longer than the lifetime of the universe to get there. Thus, for such 'kinetically trapped' systems, encoding desired output in thermodynamic equilibrium is not enough by itself. To address this, in the full version of this paper we give a kinetically and thermodynamically favoured binary counter that assembles in both the abstract Tile Assembly Model and the TBN model. Similarly, the strand displacement AND gate from Ref. [11] can be shown to compute correctly in the TBN model [3]. Nonetheless, more work is needed to come up with TBN schemes that have fast kinetic pathways, in addition to the provable thermodynamic guarantees.

2 Model

Let $\mathbb{N}, \mathbb{Z}, \mathbb{Z}^+$ denote the set of nonnegative integers, integers, and positive integers, respectively. A key type of object in our definitions is a multiset, which we define in a few different ways as convenient. Let $\mathcal{A}$ be a finite set. We can define a multiset over $\mathcal{A}$ using the standard set notion, e.g., $\mathbf{c} = \{a, a, c\}$, where $a, c \in \mathcal{A}$. Formally, we view multiset $\mathbf{c}$ as a vector assigning counts to $\mathcal{A}$. Letting $\mathbb{N}^{\mathcal{A}}$ denote the set of functions $f : \mathcal{A} \to \mathbb{N}$, we have $\mathbf{c} \in \mathbb{N}^{\mathcal{A}}$. We index entries by elements of $a \in \mathcal{A}$, calling $\mathbf{c}(a) \in \mathbb{N}$ the *count of* a *in* $\mathbf{c}$. Fixing some arbitrary ordering on the elements of $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$, we may equivalently view $\mathbf{c}$ as an element of $\mathbb{N}^k$, where for $i \in \{1, 2, \ldots, k\}$, $\mathbf{c}(i)$ denotes $\mathbf{c}(a_i)$. Let $\|\mathbf{c}\| = \|\mathbf{c}\|_1 = \sum_{a \in \mathcal{A}} \mathbf{c}(a)$ denote the *size* of $\mathbf{c}$. For any vector or matrix $\mathbf{c}$, let $\mathrm{amax}(\mathbf{c})$ denote the largest absolute value of any component of $\mathbf{c}$.

We model molecular bonds with precise binding specificity abstractly as binding "domains", designed to bind only to other, specific binding domains. Formally, consider a finite set $\mathcal{D}$ of *primary domain types*. Each primary domain type $a \in \mathcal{D}$ is mapped to a *complementary domain type* (a.k.a., *codomain type*) denoted a^*. Let $\mathcal{D}^* = \{a^* \mid a \in \mathcal{D}\}$ denote the set of codomain types of $\mathcal{D}$. The mapping is assumed 1-1, so $|\mathcal{D}^*| = |\mathcal{D}|$. We assume that domains of type a bind only to those of type a^* and vice versa.[1] The set $\mathcal{D} \cup \mathcal{D}^*$ is the set of *domain types*.

We assume a finite set $\mathcal{M}$ of *monomer types*, where a monomer type $\mathbf{m} \in \mathbb{N}^{\mathcal{D} \cup \mathcal{D}^*}$ is a non-empty multiset of domain types, e.g., $\mathbf{m} = \{a, b, b, c^*, a^*\}$, where primary domain types $a, b, c \in \mathcal{D}$. A *thermodynamic binding network* (TBN) is a pair $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ consisting of a finite set $\mathcal{D}$ of primary domain types and a finite set $\mathcal{M} \subset \mathbb{N}^{\mathcal{D} \cup \mathcal{D}^*}$ of monomer types. A *monomer collection* $\vec{\mathbf{c}} \in \mathbb{N}^{\mathcal{M}}$ of $\mathcal{T}$ is multiset of monomer types; intuitively, $\vec{\mathbf{c}}$ indicates how many of each monomer there are, but not how they are bound.[2]

Since one monomer collection usually contains more than one copy of the same domain type, we use the term *domain* to refer to each copy separately.[3] We similarly reserve the term *monomer* to refer to a particular instance of a monomer type if a monomer collection has multiple copies of the same monomer type.

A single monomer collection $\vec{\mathbf{c}}$ can take on different configurations depending on how domains in monomers are bound to each other. To formally model configurations, we first need the notion of a *bond assignment* M, which is simply a matching[4] on the bipartite graph (U, V, E) describing all possible bonds, where U is the set of all primary domains on all monomers in $\vec{\mathbf{c}}$, V is the set of all codomains on all monomers in $\vec{\mathbf{c}}$, and E is the set of edges between primary domains and their complements $\{\{u, v\} \mid u \in U, v \in V, v = u^*\}$. A *configuration* α of monomer collection $\vec{\mathbf{c}}$ is then the (multi)graph $(U \cup V, E_M)$, where the edges E_M describe both the association of domains within the same monomer, and the bonding due to M. Specifically, for each pair of domains $d_i, d_j \in \mathcal{D} \cup \mathcal{D}^*$ that are part of the same monomer in $\vec{\mathbf{c}}$, let $\{d_i, d_j\} \in E_M$, calling this a *monomer edge*, and for each edge $\{d_i, d_i^*\}$ in the bond assignment M, let $\{d_i, d_i^*\} \in E_M$, calling this a *binding edge*. Let $[\vec{\mathbf{c}}]$ be the set of all configurations of a monomer collection $\vec{\mathbf{c}}$. We say the size of a configuration, written $|\alpha|$, is simply the number of monomers in it.

Another graph that will be useful in describing the connectivity of the monomers, independent of which exact domains are bound, is the *monomer binding graph* $G_\alpha = (V_\alpha, E_\alpha)$, which is obtained by contracting each monomer edge of α. In other words, V_α is the set of monomers in α, with an edge between monomers that share at least one pair of bound domains.

---

[1] That is, we assume *like-unlike* binding such as that found in DNA Watson-Crick base-pairing, as opposed to *like-like* binding such as hydrophobic molecules with an affinity for each other in aqueous solution, or base stacking between the blunt ends of DNA helices [6,13]. It is not clear the extent to which this choice affects the computational power of our model.

[2] Because a monomer collection is a multiset of monomer types, each of which is itself a multiset, we distinguish them typographically with an arrow.

[3] For instance, the monomer collection shown in Fig. 1 has 2 domains of type a, 2 domains of type b, and 1 domain of type a^* and b^* each.

[4] A matching of a graph is a subset of edges that share no vertices in common. In our case this enforces that a domain is bound to at most one other domain.

Which configurations are thermodynamically favored over others depends on two properties of a configuration: its bond count and entropy. The *enthalpy* $H(\alpha)$ of a configuration is the number[5] of binding edges (i.e., the cardinality of the matching M). The *entropy* $S(\alpha)$ of a configuration is the number of connected components of α.[6] Each connected component is called a *polymer*.[7] Note that a polymer is itself a configuration, but of a smaller monomer collection $\vec{\mathbf{c}}' \leq \vec{\mathbf{c}}$. As with all configurations, the size of a polymer is the number of monomers in it.

Intuitively, configurations with higher enthalpy $H(\alpha)$ (more bonds formed) and higher entropy $S(\alpha)$ (more separate complexes) are thermodynamically favored. What happens if there is a conflict between the two? One can imagine capturing a tradeoff between enthalpy and entropy by some linear combination of $H(\alpha)$ and $S(\alpha)$. In DNA nanotechnology applications, the tradeoff can be controlled by increasing the number of nucleotides constituting a binding domain (increasing the weight on $H(\alpha)$), or by decreasing concentration (increasing the weight on $S(\alpha)$).[8]

In the rest of this paper, we study the particularly interesting limiting case in which enthalpy is *infinitely* more favorable than entropy.[9] We say a configuration α is *saturated* if it has no pair of domains d and d^* that are both unbound; this

---

[5] We are assuming bonds are of equal strength (although the definition can be naturally generalized to bonds of different strength).

[6] Our use of the terms "enthalpy" and "entropy", and notation H and S is meant to evoke the corresponding physical notions. Note, however, that there are other contributions to physical entropy besides the number of separate complexes. Indeed, the free energy contribution of forming additional bonds typically contains substantial enthalpic and entropic parts.

[7] We are generalizing the convention for the word "polymer" in the chemistry literature. We have no requirement that a polymer be linear, nor that it consist of repeated subunits. We chose "polymer" rather than "complex" to better contrast with "monomer".

[8] In typical DNA nanotechnology applications, the Gibbs free energy $\Delta G(\alpha)$ of a configuration α can be estimated as follows. Bonds correspond to domains of length l bases, and forming each base pair is favorable by $\Delta G^\circ_{\mathrm{bp}}$. Thus, the contribution of $H(\alpha)$ to $\Delta G(\alpha)$ is $(\Delta G^\circ_{\mathrm{bp}} \cdot l)H(\alpha)$. At 1 M, the free energy penalty due to decreasing the number of separate complexes by 1 is $\Delta G^\circ_{\mathrm{assoc}}$. At effective concentration C M, this penalty increases to $\Delta G^\circ_{\mathrm{assoc}} + RT\ln(1/C)$. As the point of zero free energy, we take the configuration with no bonds, and all monomers separate. Thus, the contribution of $S(\alpha)$ to $\Delta G(\alpha)$ is $(\Delta G^\circ_{\mathrm{assoc}} + RT\ln(1/C))(|\alpha| - S(\alpha))$, where $|\alpha|$ is the total number of monomers. To summarize,

$$\Delta G(\alpha) = (\Delta G^\circ_{\mathrm{bp}} \cdot l)H(\alpha) + (\Delta G^\circ_{\mathrm{assoc}} + RT\ln(1/C))(|\alpha| - S(\alpha)).$$

Note that, as expected, this is a linear combination of $H(\alpha)$ and $S(\alpha)$, and that increasing the length of domains l weighs $H(\alpha)$ more heavily, while decreasing the concentration C weighs $S(\alpha)$ more heavily. Typically $G^\circ_{\mathrm{bp}} \approx -1.5$ kcal/mol, and $G^\circ_{\mathrm{assoc}} \approx 1.96$ kcal/mol [9].

[9] Note that the other limiting case, where entropy is infinitely more favorable, is degenerate: the most favorable configuration in that case always has every monomer unconnected to any other.

is equivalent to stating that α has maximal bonding among all configurations in $[\vec{\mathbf{c}}]$. We say a configuration $\alpha \in [\vec{\mathbf{c}}]$ is *stable* (aka thermodynamically favored) if it is saturated and maximizes the entropy among all saturated configurations, i.e., every saturated configuration $\alpha' \in [\vec{\mathbf{c}}]$ obeys $S(\alpha') \leq S(\alpha)$. Let $[\vec{\mathbf{c}}]_\square$ denote the set of stable configurations of monomer collection $\vec{\mathbf{c}}$. See Fig. 1 for an example thermodynamic binding network that has a single stable configuration. We note that, consistent with our model, in strand displacement cascades "long" domains are assumed to always be paired, and systems can be effectively driven by the formation of more separate complexes [14].

3 Thermodynamic Boolean Formulas

Figure 2 shows an example of a TBN that performs AND computation, based on the CRN strand displacement gate from Ref. [11]. Realized as a strand displacement system, it has a kinetic pathway taking the untriggered (left) to the triggered (right) configuration. The inputs are specified by the presence (logical value 1) or absence (logical value 0) of the input monomers $\mathbf{i}_1$ and $\mathbf{i}_2$. The output convention followed is the following. The output is 1 if and only if *some* stable configuration has the output monomer $\mathbf{o}$ unbound to any other monomer (free). This can be termed the *weak* output convention. Alternatively, in the *strong* output convention, output 1 implies *every* stable configuration has the output monomer $\mathbf{o}$ free, and output 0 implies *every* stable configuration has the output

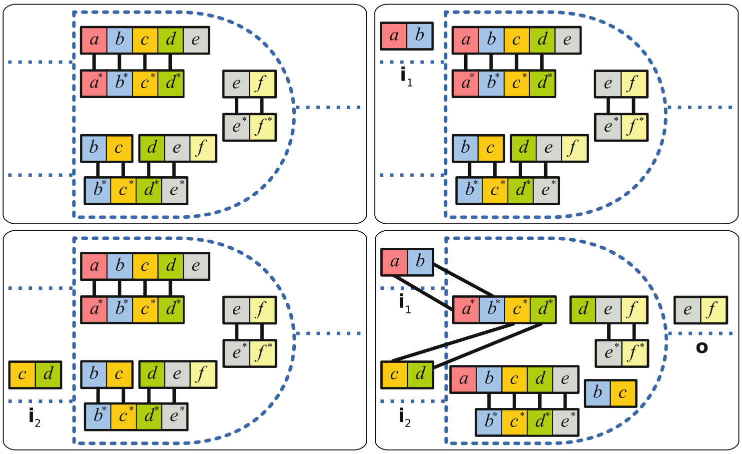

Fig. 2. Basic AND gate: Monomers $\mathbf{i}_1 = \{a, b\}$ and $\mathbf{i}_2 = \{c, d\}$ represent the input, $\mathbf{o} = \{e, f\}$ represents the output, while the remainder are intermediate monomers to implement the logic relating the input to the output. If either or both inputs are missing, then the only stable configuration has the present input monomers free (unbound) and the output monomer $\mathbf{o}$ not free (bound). If both input monomers are present, then there are two stable configurations: one with inputs free (and $\mathbf{o}$ not free), or the one depicted with $\mathbf{o}$ free and both inputs bound.

monomer **o** bound to some other monomer. More complex AND gate designs are compatible with the strong output convention (not shown).

Note that even the weak output convention, coupled with a kinetic pathway releasing the output given the correct inputs, can be used to argue that: (1) if the correct inputs are present the output will be produced (via kinetic argument), (2) if the correct inputs are not present then ultimately little output will be free (thermodynamic argument). In the context of strand displacement cascades, TBNs can explore arbitrary structures (pseudoknots, remote toeholds, etc.) since we do not impose any ordering on domains in a monomer, nor any geometry. This strengthens the conclusion of (2), showing that arbitrary (even unknown) kinetic pathways must lead to a thermodynamic equilibrium with little output.

While individual AND gates can be proven correct with respect to the above output conventions (e.g., through the SAT solver of Ref. [3]), it remains to be shown that these components can be safely composed into arbitrary Boolean circuits. Note that the input and output monomers have orthogonal binding sites. This is important for composing AND gates, where the output of one acts as an input to another. As is typical for strand displacement logic, OR gates can be trivially created when multiple AND gates have the same output. Dual-rail AND/OR circuits are sufficient to compute arbitrary Boolean functions without explicit NOT gates. Nonetheless it is not obvious that the input convention (complete presence or absence of input monomers) matches the output convention (weak or strong). It is also not clear how statements about the stable configurations of the whole circuit can be made based on the stable configurations of the individual modules.

We now show that correct composition can be proven in certain cases. Although we believe that the gate shown in Fig. 2 is composable, the argument below relies on a different construction. We further consider a restricted case of AND gate formulas (trees).

An important concept in the argument below is the notion of "distance to stability". This refers to the difference between the entropy of the stable configurations and the largest entropy of a saturated configuration with incorrect output. The larger the distance to stability, the larger the entropy penalty to incorrectly producing the output. Unlike the simple AND gate from Fig. 2, the constructions below can be instantiated to achieve arbitrary desired distance to stability (by increasing the redundancy parameter n).

Many open questions remain. Can our techniques be generalized to arbitrary circuits, rather than just trees of AND gates? Can we prove these results for logic gates that have a corresponding kinetic pathway (like the AND gates in Fig. 2 which can be instantiated as strand displacement systems)? Finally, in our Boolean gate constructions, we assume that the monomer collection has exactly one copy of certain monomers. It remains open whether these schemes still work if there are many copies of all monomers.

3.1 Translator Cascades

We begin with the simplest of circuits, *translator cascades* ($x_1 \rightarrow x_2 \rightarrow ... \rightarrow x_{k+1}$), which simply propagate signal through k layers when the input signal x_1 is present. Logically a translator gate is simply a repeater gate. The input is the presence or absence of the input monomer consisting of n copies of domain x_1. Our analysis below implies that if and only if the input is present, there is a stable configuration with n copies of x_{k+1} domain in the same polymer. The *terminator gadget* converts this output to the weak output convention defined above (whether or not the monomer consisting of n copies of domain x_{k+1} is free). The following Lemma shows that we can exactly compute the distance from stability of a translator cascade shown in Fig. 3. Besides being a "warm-up" for AND gate cascades, the Lemma is used in the proof of Theorem 2.

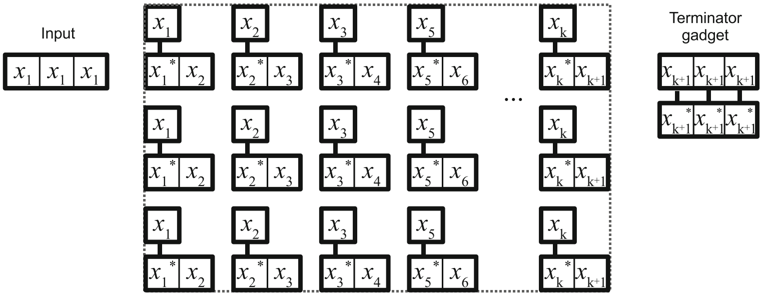

Fig. 3. A cascade of k translator gates discussed in Sect. 3.1, with redundancy parameter $n = 3$. We say that a configuration of a formula has output 1 if the terminator monomer $\{x_{k+1}, \ldots, x_{k+1}\}$ is free, and has output 0 otherwise. Redundancy parameter n specifies the number of copies of monomers and domains as shown.

Observation 1. *The intended configuration* α *of a monomer collection representing a depth* k*, redundancy* n *translator cascade, without input, and with output* 0*, is saturated and has* $S(\alpha) = nk + 1$*. (See Fig. 3.)*

Lemma 1. *If* γ *is a saturated configuration of a monomer collection representing a depth* k*, redundancy* n *translator cascade, without input, and with output* 1*, then* $S(\gamma) = n(k-1) + 2$*.*

The proof of Lemma 1 appears in the full version of this paper. Taken together, Observation 1 and Lemma 1 imply that the redundancy parameter (n) guarantees the distance to stability ($n - 1$) for a translator cascade of any length.

3.2 Trees of AND Gates

In this section we motivate how Boolean logic gates can be composed such that the overall circuit has a guaranteed distance to stability, relative to a redundancy

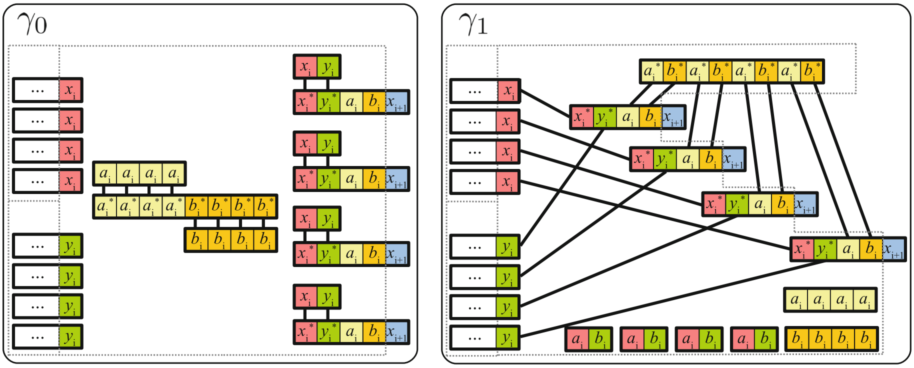

Fig. 4. AND gates used in Sect. 3.2, with redundancy parameter $n = 4$. Two saturated configurations are shown: γ_0 is the intended configuration corresponding to output of 0. γ_1 is the intended configuration corresponding to output of 1. Input domains are x_i y_i, and output domains are x_{i+1}. The output is considered to be 1 in any configuration where all n output domains are in the same polymer, 0 otherwise. Dashed boxes represent that any domain type appearing inside of a box does have have a complement appearing outside of the box.

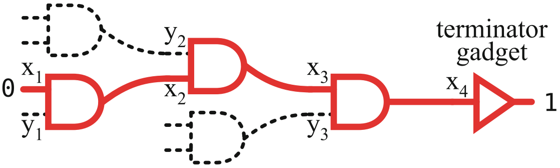

Fig. 5. Shown highlighted is a leak path through a tree of AND gates from a missing input ("0") to erroneous output ("1").

parameter n. Specifically, we start with the AND gate design of Fig. 4, and we give a concrete argument for a tree of these AND gates (e.g., Fig. 5).

Theorem 2. *Consider a TBN for AND gates, with redundancy n, composed into a tree of depth k. If at least one of the inputs is not present, the distance to stability for any saturated configurations with output* 1 *is at least $n - 2k - 1$.*

Proof. Let γ be any saturated configuration of the TBN with output 1. Consider the missing input and define the *leak path* to be the linear sequence of AND gates from the missing input to and including the terminator gadget. For convenience we imagine relabelling all the domains in the leak path indexed by the position of the AND gate in the leak path. For example, Fig. 5 highlights the leak path through the tree from a missing input ("0") to erroneous output ("1"). Specifically, the domain names as shown in Fig. 4 appear in the ith AND gate (for $1 \leq i \leq k$), where x_{k+1} feeds into the terminator gadget. Domains y_i connect the leak path to the rest of the tree.

Definition 1. *Given a configuration α of a monomer collection $\vec{\mathbf{c}}$, we say we* excise *a domain d if we create a new configuration α' by removing the node corresponding to d and all incident edges. (Note that α' is a configuration of a monomer collection of a different TBN.)*

Manipulation 1. Excise all domains of type y_i and codomains of type y_i^* on monomers of the leak path involved in fan-in, $1 \leq i \leq k$, yielding the new configuration γ'. Note that if domain y_i is on a monomer other than the leak path, then it is not excised.

The leak path in γ' now has no domains in common with the rest of the tree (and thus no bonds). Let γ'_L be the subconfiguration of the leak path, and let γ'_R be the subconfiguration of the rest of the system. (Note $\gamma' = \gamma'_L \cup \gamma'_R$.)

Observation 3. *Given a saturated configuration α, if you excise all domains or codomains of a particular type (or both its domains and codomains) yielding α', then α' is saturated.*

By Observation 3 γ' is saturated since for every domain type y_i and codomain type y_i^*, every instance of y_i^* is excised; $1 \leq i \leq k$. This implies γ'_L and γ'_R are also saturated.

Manipulation 2. Excise all domains of type a_i and b_i and all codomains of type a_i^* and b_i^* in γ'_L, $1 \leq i \leq k$, yielding the new configuration γ''_L. By Observation 3, γ''_L is saturated.

Claim A. $S(\gamma') \geq S(\gamma)$.

Proof of the claim. Entropy can only be decreased via excision if an entire monomer is excised. Since Manipulation 1 only excised domain and codomain types from the set $\mathcal{D}' = \bigcup_{i=1}^{k} \{y_i, y_i^*\}$, and those domain types only appear on monomers which also have domain instances with types not in $\mathcal{D}'$, then no entire monomer was excised. ■

Claim B. $S(\gamma''_L) \geq S(\gamma'_L) - 3k$.

Proof of the claim. For every layer i, $1 \leq i \leq k$, there are 3 monomers that only contain domain and codomain types in the set $\{a_i, b_i, a_i^*, b_i^*\}$. Therefore, γ''_L contains at most 3 fewer monomers than γ'_L, for each of the k layers. ■

Claim C. $S(\gamma''_L) = n(k-1) + 2$.

Proof of the claim. Recognize that γ''_L is a saturated configuration of a monomer collection representing a depth k, redundancy n translator cascade, without input, and with output 1. The claim follows by Lemma 1. ■

Claim D. $S(\gamma) \leq n(k-1) + 2 + S(\gamma'_R) + 3k$.

Proof of the claim.

$$
\begin{aligned}
S(\gamma) &\leq S(\gamma') && \text{by Claim A} \\
&= S(\gamma'_L) + S(\gamma'_R) && \\
&\leq S(\gamma''_L) + S(\gamma'_R) + 3k && \text{by Claim B} \\
&\leq n(k-1) + 2 + S(\gamma'_R) + 3k && \text{by Claim C}
\end{aligned}
$$

■

Now, take the monomers from the leak path in γ, and configure them into the "untriggered configuration" (see Fig. 4, left), yielding subconfiguration β. Let $\alpha = \beta \cup \gamma'_R$. Note that β is saturated, and therefore α is a saturated configuration of the entire tree (*i.e.,* the same TBN as γ).

Observation 4. $S(\alpha) = S(\gamma'_R) + k(n+1) + 1$.

Finally, consider the entropy gap between α and γ.

$$
\begin{aligned}
S(\alpha) - S(\gamma) &\geq S(\gamma'_R) + k(n+1) + 1 - S(\gamma) && \text{by Observation 4} \\
&\geq S(\gamma'_R) + k(n+1) + 1 && \\
&\qquad - (n(k-1) + 2 + S(\gamma'_R) + 3k) && \text{by Claim D} \\
&= n - 2k - 1
\end{aligned}
$$

Therefore, there exists a saturated configuration with output 0 over the same TBN as γ, but with entropy at least $n - 2k - 1$ larger, thus establishing the theorem. □

Theorem 2 seems to suggest that in order to maintain the bound on distance to stability for incorrect computation, the redundancy parameter n should increase to compensate for an increase in circuit depth k. However, a more sophisticated argument shows that manipulations 1 and 2 can decrease entropy by at most $k + 1$. Following the above argument, the distance to stability is found to be $n - 2$. This is optimal because a single AND gate with redundancy $n = 2$ can be shown to have no entropy gap between output 0 and output 1 configurations.

4 Thermodynamic Self-assembly: Assembling Large Polymers

TBNs can not only exhibit Boolean circuit computation, but they can also be thought of as a model of self-assembly. Here we begin to explore this connection by asking a basic question motivated by the abstract Tile Assembly Model (aTAM) [12]: how many different monomer *types* are required to assemble a large polymer?

Favoring enthalpy infinitely over entropy, on its face, *appears* to encourage large polymers. Perhaps we can imagine designing a single TBN $\mathcal{T}$ that can assemble arbitrarily large polymers where for each $n \in \mathbb{N}$, $\mathcal{T}$ has a stable polymer α composed of at least n monomers. In this section we show that this is impossible: every TBN $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ has stable polymers of size at most exponential in the number of domain types $|\mathcal{D}|$ and monomer types $|\mathcal{M}|$ (Theorem 9). The proof shows that any polymer ρ larger than the bound can be partitioned into at least two saturated (maximally bound) polymers, which implies that ρ is not stable. Figure 6 gives an example. We also show that this upper bound is essentially tight by constructing a family of systems with exponentially large stable polymers (Theorem 5). Taken together, the exponential lower bound of Theorem 5 and upper bound of Theorem 9 give a relatively tight bound on the maximum size achievable for stable TBN polymers.

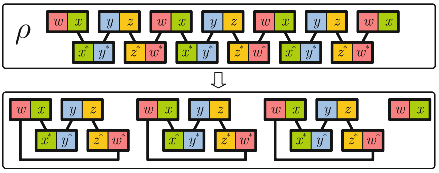

Fig. 6. A polymer ρ composed of several copies of four monomer types, which is not stable since it can be broken into several smaller polymers (bottom panel) such that all domains are bound.

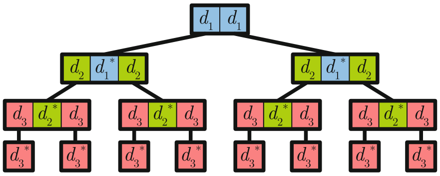

Fig. 7. An example of a TBN from Theorem 5 for $n = 4$ and $k = 2$.

Is it possible to construct algorithmically *interesting* TBN polymers that are stable? In the full version of this paper, we show that a typical binary counter construction from the aTAM model is not stable, but can be modified to become stable in our model. Importantly, this TBN binary counter demonstrates that in principle algorithmically complex assemblies could have effective assembly pathways (aTAM) as well as be thermodynamically stable (TBN).

4.1 Superlative Trees: TBNs with Exponentially Large Stable Polymers

The next theorem shows that there are stable polymers that are exponentially larger than the number of domain types and monomer types required to assemble them.

Theorem 5. *For every* $n, k \in \mathbb{Z}^+$, *there is a TBN* $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ *with* $|\mathcal{D}| = n-1$ *and* $|\mathcal{M}| = n$, *having a stable polymer of size* $\frac{k^n-1}{k-1}$.

Proof. An example of $\mathcal{T}$ for $n = 4$ and $k = 2$ is shown in Fig. 7. Let $\mathcal{D} = \{d_1, \ldots, d_n\}$ and $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_n\}$, where, for each $j \in \{2, \ldots, n-1\}$, $\mathbf{m}_j = \{d^*_{j-1}, k \cdot d_j\}$ (i.e., 1 copy of d^*_{j-1} and k copies of d_j), $\mathbf{m}_1 = \{k \cdot d_1\}$, and $\mathbf{m}_n = \{d^*_{n-1}\}$. Define $\vec{\mathbf{c}} \in \mathbb{N}^{\mathcal{M}}$ by $\vec{\mathbf{c}}(\mathbf{m}_j) = k^{j-1}$ for $j \in \{1, \ldots, n\}$. Then $\|\vec{\mathbf{c}}\| = \sum_{j=1}^{n} k^{j-1} = \frac{k^n-1}{k-1}$. Observe that $[\vec{\mathbf{c}}]$ has a unique (up to isomorphism) saturated configuration α (which is therefore stable), described by a complete k-ary tree: level $j \in \{1, \ldots, n-1\}$ of the tree is composed of k^{j-1} copies of $\mathbf{m}_j$, each bound to k children of type $\mathbf{m}_{j+1}$ in level $j+1$. $\square$

The remainder of Sect. 4 is devoted to proving that no stable polymer ρ can have size *more* than exponential in $|\mathcal{D}|$ and $|\mathcal{M}|$.

4.2 A Linear Algebra Framework

We prove Theorem 9, the main result of Sect. 4, by viewing TBNs from a linear algebra perspective. Let $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ be a TBN, with $\mathcal{D} = \{d_1, \ldots, d_d\}$ and $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_m\}$. For a matrix $\mathbf{A}$, let $\mathbf{A}(i, j)$ denote the entry in the i'th row and j'th column. Define the $d \times m$ *positive monomer matrix* $\mathbf{M}^+_{\mathcal{T}}$ of $\mathcal{T}$ by $\mathbf{M}^+_{\mathcal{T}}(i, j) = \mathbf{m}_j(d_i)$. Define the $d \times m$ *negative monomer matrix* $\mathbf{M}^-_{\mathcal{T}}$ of $\mathcal{T}$ by $\mathbf{M}^-_{\mathcal{T}}(i, j) = \mathbf{m}_j(d^*_i)$. Define the $d \times m$ *monomer matrix* $\mathbf{M}_{\mathcal{T}}$ of $\mathcal{T}$ to be $\mathbf{M}_{\mathcal{T}} = \mathbf{M}^+_{\mathcal{T}} - \mathbf{M}^-_{\mathcal{T}}$. Note that $\mathbf{M}^+_{\mathcal{T}}$ and $\mathbf{M}^-_{\mathcal{T}}$ are matrices over $\mathbb{N}$, but $\mathbf{M}_{\mathcal{T}}$ is over $\mathbb{Z}$.

The rows of the monomer matrix $\mathbf{M}_{\mathcal{T}}$ correspond to domain types and the columns correspond to monomer types. The mapping from a TBN $\mathcal{T}$ to a monomer matrix $\mathbf{M}_{\mathcal{T}}$ is not 1-1: $\mathbf{M}_{\mathcal{T}}(i, j)$ is the number of d_i domains minus the number of d^*_i domains in monomer type $\mathbf{m}_j$, which would be the same, for instance, for monomer types $\mathbf{m}_1 = \{d_1, d_3\}$ and $\mathbf{m}_2 = \{d_1, d_1, d^*_1, d_3\}$. Let $\vec{\mathbf{c}}$ be a monomer collection and let $\mathbf{d} = \mathbf{M}_{\mathcal{T}} \vec{\mathbf{c}} \in \mathbb{N}^d$; for $i \in \{1, \ldots, d\}$, $\mathbf{d}(i)$ is the number of d_i domains minus the number of d^*_i domains in the whole monomer collection $\vec{\mathbf{c}}$.

Let $\alpha \in [\vec{\mathbf{c}}]$ be saturated; α can only have a domain d_i unbound if all copies of its complement d^*_i are bound, and vice versa. If $\mathbf{d}(i) > 0$, in α there is an excess of d_i domains, and all d^*_i domains are bound. If $\mathbf{d}(i) < 0$, in α there is an excess of d^*_i domains, and all d_i domains are bound. This leads to the following observation.

Observation 6. *Let $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ be a TBN and $\vec{\mathbf{c}} \in \mathbb{N}^{\mathcal{M}}$ a monomer collection. Let $\mathbf{d} = \mathbf{M}_{\mathcal{T}} \vec{\mathbf{c}}$. Then for every configuration $\alpha \in [\vec{\mathbf{c}}]$, α is saturated if and only if, for all $i \in \{1, \ldots, d\}$, if $\mathbf{d}(i) \geq 0$ (respectively, if $\mathbf{d}(i) \leq 0$), then $\mathbf{d}(i)$ is the number of unbound d_i (resp., d^*_i) domains in α.*

Let $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ and $\mathcal{T}' = (\mathcal{D}, \mathcal{M}')$ be TBNs with the same set of domain types. Then we call $\mathcal{T}'$ a *relabeling* of $\mathcal{T}$ if there exists a subset $D \subseteq \mathcal{D}$ such that $\mathcal{M}'$ can be obtained from $\mathcal{M}$ by starring any instance of $d_i \in D$ in $\mathcal{M}$ and unstarring any instance of d^*_i in $\mathcal{M}$. Since this corresponds to negating the i'th row of $\mathbf{M}_{\mathcal{T}}$, which negates the i'th entry of the vector $\mathbf{d} = \mathbf{M}_{\mathcal{T}} \vec{\mathbf{c}}$, this gives the following observation.

Observation 7. *Let* $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ *be a TBN and* $\vec{\mathbf{c}} \in \mathbb{N}^{\mathcal{M}}$ *a monomer collection. There exists a relabeling* $\mathcal{T}'$ *of* $\mathcal{T}$ *so that* $\mathbf{M}_{\mathcal{T}'}\vec{\mathbf{c}} \geq 0$*.*

Combining Observations 6 and 7 results in the following observation, which essentially states that for any given monomer collection $\vec{\mathbf{c}}$, we may assume without loss of generality that domains unbound in saturated configurations $\alpha \in [\vec{\mathbf{c}}]$ are all primary domain types.

Observation 8. *Let* $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ *be a TBN and* $\vec{\mathbf{c}} \in \mathbb{N}^{\mathcal{M}}$ *a monomer collection. There exists a relabeling* $\mathcal{T}'$ *of* $\mathcal{T}$ *so that, letting* $\mathbf{d} = \mathbf{M}_{\mathcal{T}'}\vec{\mathbf{c}}$*, for all configurations* $\alpha \in [\vec{\mathbf{c}}]$*,* α *is saturated if and only if, for all* $i \in \{1, \ldots, d\}$*,* $\mathbf{d}(i) \in \mathbb{N}$ *is the number of unbound primary domains of type* $d_i \in \mathcal{D}$ *in* α*.*

The following lemma is a key technical tool for showing that a polymer is not stable (or equivalently that a stable configuration has entropy greater than 1 and therefore cannot be a single polymer). It generalizes the idea shown in Fig. 6 that if one can find a monomer subcollection $\vec{\mathbf{c}}_1$ in a larger collection $\vec{\mathbf{c}}$, and $\vec{\mathbf{c}}_1$ has a saturated configuration with *no* bonds left unbound, then one can create a saturated configuration $\gamma \in [\vec{\mathbf{c}}]$ with no bonds between $\vec{\mathbf{c}}_1$ and $\vec{\mathbf{c}} - \vec{\mathbf{c}}_1$. (Thus γ has at least two polymers.)

More generally, given a monomer collection $\vec{\mathbf{c}}$ with at least as many d_i as d_i^* domains (under appropriate relabeling this holds for each i by Observation 7), if we can partition $\vec{\mathbf{c}}$ into subcollections $\vec{\mathbf{c}}_1$ and $\vec{\mathbf{c}}_2$, and each of them *also* has at least as many d_i as d_i^* domains for each $i \in \{1, \ldots, d\}$, then every stable configuration $\alpha \in [\vec{\mathbf{c}}]_\square$ has at least two polymers, since there is a saturated configuration of $\vec{\mathbf{c}}$ in which there are no bonds between $\vec{\mathbf{c}}_1$ and $\vec{\mathbf{c}}_2$.[10]

Lemma 2. *Let* $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ *be a TBN, let* $\vec{\mathbf{c}} \in \mathbb{N}^{\mathcal{M}}$ *be a monomer collection of* $\mathcal{T}$ *such that* $\mathbf{M}_{\mathcal{T}}\vec{\mathbf{c}} \geq \mathbf{0}$*, and let* $\alpha \in [\vec{\mathbf{c}}]_\square$ *be a stable configuration. If there exist nonempty subcollections* $\vec{\mathbf{c}}_1, \vec{\mathbf{c}}_2 \in \mathbb{N}^{\mathcal{M}}$ *where 1)* $\vec{\mathbf{c}}_1 + \vec{\mathbf{c}}_2 = \vec{\mathbf{c}}$ *and 2)* $\mathbf{M}_{\mathcal{T}}\vec{\mathbf{c}}_1 \geq 0$ *and* $\mathbf{M}_{\mathcal{T}}\vec{\mathbf{c}}_2 \geq 0$*, then* $S(\alpha) > 1$*.*

The proof of Lemma 2 appears in the full version of this paper.

4.3 Exponential Upper Bound on Polymer Size

We now show a converse to Theorem 5, namely Theorem 9, showing that stable polymers have size at most exponential in the number of domain and monomer types. The proof of Theorem 9 closely follows Papadimitriou's proof that integer programming is contained in NP [7]. That proof shows, for any linear system

---

[10] Observations 6, 7, and 8 are not really *necessary* for our technique, but simplify the description of the conditions under which $\vec{\mathbf{c}}_1$ and $\vec{\mathbf{c}}_2$ would be saturated: specifically, that if $\mathbf{d} = \mathbf{M}_{\mathcal{T}}\vec{\mathbf{c}}$ is in the nonnegative orthant, then so are $\mathbf{d}_1 = \mathbf{M}_{\mathcal{T}}\vec{\mathbf{c}}_1$ and $\mathbf{d}_2 = \mathbf{M}_{\mathcal{T}}\vec{\mathbf{c}}_2$. If we did not use relabeling (thus could not guarantee that $\mathbf{d}$ is in the nonnegative orthant) then the requisite condition to apply Lemma 2 would be that $\mathbf{d}$, $\mathbf{d}_1$, and $\mathbf{d}_2$ all occupy the same orthant; i.e., for all $i \in \{1, \ldots, d\}$, if any of $\mathbf{d}(i)$, $\mathbf{d}_1(i)$, or $\mathbf{d}_2(i)$ are negative, then the other two are not positive.

$\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is a given $n \times m$ integer matrix, $\mathbf{b} \in \mathbb{Z}^n$ is a given integer vector, and $\mathbf{x}$ represents the m unknowns, that if the system has a solution $\mathbf{x} \in \mathbb{N}^m$, then it has a "small" solution $\mathbf{x}' \in \mathbb{N}^m$. "Small" means that $\mathrm{amax}(\mathbf{x}')$ is at most exponential in $n + m + \mathrm{amax}(\mathbf{A}) + \mathrm{amax}(\mathbf{b})$. The technique of [7] proceeds by showing that any sufficiently large solution $\mathbf{x} \in \mathbb{N}^m \setminus \{\mathbf{0}\}$ can be split into two vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{N}^m \setminus \{\mathbf{0}\}$ such that $\mathbf{x}_1 + \mathbf{x}_2 = \mathbf{x}$, where $\mathbf{Ax}_1 = \mathbf{0}$, so $\mathbf{x}_2$ is also a solution: $\mathbf{Ax}_2 = \mathbf{A}(\mathbf{x} - \mathbf{x}_1) = \mathbf{Ax} - \mathbf{Ax}_1 = \mathbf{Ax} = \mathbf{b}$. This is useful because $\mathbf{x}_1$ and $\mathbf{x}_2$ satisfy the hypothesis of Lemma 2, which tells us that all stable configurations $\alpha \in [\mathbf{x}]$ obey $S(\alpha) > 1$, so any single-polymer configuration of $\mathbf{x}$ is not stable.

We include the full proof for three reasons: (1) self-containment, (2) it requires a bit of care to convert our inequality $\mathbf{Ax} \geq \mathbf{0}$ into an equality as needed for the technique,[11] and (3) although the proof of [7] is sufficiently detailed to prove our theorem, the statement of the theorem in [7] hides the details about splitting the vector, which are crucial to obtaining our result.

We require the following discrete variant of Farkas' Lemma, also proven in [7].

Lemma 3 **(*[7]*).** *Let $a, d, l \in \mathbb{Z}^+$, $\mathbf{v}_1, \ldots, \mathbf{v}_l \in \{0, \pm 1, \ldots, \pm a\}^d$, and $K = (ad)^{d+1}$. Then exactly one of the following statements holds:*

1. *There exist l integers $n_1, \ldots, n_l \in \{0, 1, \ldots, K\}$, not all 0, such that $\sum_{j=1}^{l} n_j \mathbf{v}_j = \mathbf{0}$.*
2. *There exists a vector $\mathbf{h} \in \{0, \pm 1, \ldots, \pm K\}^d$ such that, for all $j \in \{1, \ldots, l\}$, $\mathbf{h}^\mathsf{T} \cdot \mathbf{v}_j \geq 1$.*

Intuitively, statement (1) of Lemma 3 states that the vectors can be added to get $\mathbf{0}$ (they are "directions of balanced forces" [7]). This is false if and only if statement (1) holds: the vectors all lie on one side of some hyperplane, whose orthogonal vector $\mathbf{h}$ would then have positive dot product with each of the vectors $\mathbf{v}_j$ (thus adding any of them would move positively in the direction $\mathbf{h}$ and could never cancel to get $\mathbf{0}$).

Intuitively, Theorem 9 states that the size of polymers in stable configurations is upper bounded by a function which is exponential in d. We prove this by first defining a constant K which is exponential in d. If each of the m individual monomer counts is less than K, then we are done since no polymer in the configuration can have size bigger than mK. If some of the monomer counts are greater than K (call these *large-count monomers*), we consider two cases.

For the first case, we consider the scenario where the vectors which describe the monomer types with large monomer counts are such that they can "balance" each other out with relatively small linear combination coefficients. If this is

---

[11] In particular, the proof of [7] upper bounds the size of $\mathbf{x}$ in terms of the entries of both $\mathbf{A}$ and $\mathbf{b}$. However, the naïve way to solve a linear inequality $\mathbf{Ax} \geq \mathbf{0}$ using an equality, by introducing slack variables $\mathbf{b}$ and asking for solutions $\mathbf{x} \in \mathbb{N}^m$, $\mathbf{b} \in \mathbb{N}^n$ such that $\mathbf{Ax} = \mathbf{b}$, allows for the possibility that $\|\mathbf{b}\|$ is very large compared to $\|\mathbf{A}\|$, in which case upper bounding $\|\mathbf{x}\|$ in terms of both $\mathbf{A}$ and $\mathbf{b}$ does not help to bound $\|\mathbf{x}\|$ in terms of $\mathbf{A}$ alone.

the case, then we can make a saturated subconfiguration which has at least one polymer using these small linear combination coefficients and large-count monomer types since the domains and codomains completely "balance" each other out. We can then use the rest of the counts of the configuration to make another saturated subconfiguration which has at least one polymer. This is shown mathematically by applying Lemma 3 to show that the monomer counts in the polymer can be split to find a configuration consisting of two separate saturated polymers. This means that there is a saturated configuration that has at least two polymers which contradicts the assumption α is a single stable polymer.

If there exist no such linear combination to "balance out" out the vectors describing the large-count monomers, then Lemma 3 tells us all of these vectors lie on the same side of some hyperplane. In this case, we show that counts of the small-count monomers play a role in bounding the counts of the large-count monomers. Intuitively, if all of the vectors describing the large-count monomers lie on the same side of some hyperplane, they are missing domains and codomains which will allow them to bind together. The domains and codomains they need in order to bind together, then must be found on the small-count monomer. Consequently, this means the size of polymers will be bound by the counts of small-count monomers (which is exponential in K). The proof appears in the full version of this paper.

Theorem 9. *Let* $\mathcal{T} = (\mathcal{D}, \mathcal{M})$ *be a TBN with* $d = |\mathcal{D}|$ *and* $m = |\mathcal{M}|$. *Let* $a = \max_{\mathbf{m} \in \mathcal{M}, d_i \in \mathcal{D} \cup \mathcal{D}^*} \mathbf{m}(d_i)$ *be the maximum count of any domain in any monomer. Then all polymers of every stable configuration* α *of* $\mathcal{T}$ *have size at most* $2(m + d)(ad)^{2d+3}$.

References

1. Barish, R.D., Schulman, R., Rothemund, P.W.K., Winfree, E.: An information-bearing seed for nucleating algorithmic self-assembly. Proc. Natl. Acad. Sci. **106**(15), 6054–6059 (2009)
2. Bennett, C.H.: The thermodynamics of computation—a review. Int. J. Theor. Phys. **21**(12), 905–940 (1982)
3. Breik, K., Prakash, L., Thachuk, C., Heule, M., Soloveichik, D.: Computing properties of stable configurations of thermodynamic binding networks (2017, in preparation)
4. Dirks, R.M., Bois, J.S., Schaeffer, J.M., Winfree, E., Pierce, N.A.: Thermodynamic analysis of interacting nucleic acid strands. SIAM Rev. **49**(1), 65–88 (2007)
5. Genot, A.J., Zhang, D.Y., Bath, J., Turberfield, A.J.: Remote toehold: a mechanism for flexible control of DNA hybridization kinetics. J. Am. Chem. Soc. **133**(7), 2177–2182 (2011)
6. Gerling, T., Wagenbauer, K.F., Neuner, A.M., Dietz, H.: Dynamic DNA devices and assemblies formed by shape-complementary, non-base pairing 3D components. Science **347**(6229), 1446–1452 (2015)
7. Papadimitriou, C.H.: On the complexity of integer programming. J. ACM (JACM) **28**(4), 765–768 (1981)

8. Phillips, A., Cardelli, L.: A programming language for composable DNA circuits. J. R. Soc. Interface **6**(Suppl 4), S419–S436 (2009)
9. SantaLucia Jr., J., Hicks, D.: The thermodynamics of DNA structural motifs. Annu. Rev. Biophys. Biomol. Struct. **33**, 415–440 (2004)
10. Schulman, R., Winfree, E.: Programmable control of nucleation for algorithmic self-assembly. SIAM J. Comput. **39**(4), 1581–1616 (2009)
11. Thachuk, C., Winfree, E., Soloveichik, D.: Leakless DNA strand displacement systems. In: Phillips, A., Yin, P. (eds.) DNA 2015. LNCS, vol. 9211, pp. 133–153. Springer, Cham (2015). doi:10.1007/978-3-319-21999-8_9
12. Winfree, E.: Algorithmic self-assembly of DNA. Ph.D. thesis, California Institute of Technology (1998)
13. Woo, S., Rothemund, P.W.K.: Programmable molecular recognition based on the geometry of DNA nanostructures. Nat. Chem. **3**, 620–627 (2011)
14. Zhang, D.Y., Turberfield, A.J., Yurke, B., Winfree, E.: Engineering entropy-driven reactions and networks catalyzed by DNA. Science **318**(5853), 1121–1125 (2007)