Interface Focus

Interface Focus (2012) **2**, 512–521
doi:10.1098/rsfs.2011.0106
Published online 15 February 2012

Less haste, less waste: on recycling and its limits in strand displacement systems

Anne Condon*, Alan J. Hu, Ján Maňuch and Chris Thachuk

Department of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z4

We study the potential for molecule recycling in chemical reaction systems and their DNA strand displacement realizations. Recycling happens when a product of one reaction is a reactant in a later reaction. Recycling has the benefits of reducing consumption, or waste, of molecules and of avoiding fuel depletion. We present a binary counter that recycles molecules efficiently while incurring just a moderate slowdown compared with alternative counters that do not recycle strands. This counter is an n-bit binary reflecting Gray code counter that advances through 2^n states. In the strand displacement realization of this counter, the waste—total number of nucleotides of the DNA strands consumed—is polynomial in n, the number of bits of the counter, while the waste of alternative counters grows exponentially in n. We also show that our n-bit counter fails to work correctly when many ($\Theta(n)$) copies of the species that represent the bits of the counter are present initially. The proof applies more generally to show that in chemical reaction systems where all but one reactant of each reaction are catalysts, computations longer than a polynomial function of the size of the system are not possible when there are polynomially many copies of the system present.

Keywords: DNA computing; strand displacement systems; strand recycling

1. INTRODUCTION

DNA strand displacement is a form of chemical reaction in which one or more single-stranded DNA molecules—the reactants—bind to a multi-stranded complex, thereby displacing other single-stranded molecules—the products. DNA strand displacements are important and versatile reactions that have already supported wet-lab simulation of logic circuits and DNA walkers and can in principle support general purpose computation in an energy-efficient manner [1–9]. Such DNA strand displacement reactions, and chemical reactions more generally, typically consume strands or reactants at all reaction steps. Catalyst strands are an exception in that they are not consumed during the course of a reaction, but are recycled to perform the same operation multiple times.

Can chemical reactions and their strand displacement system realizations recycle strands in more general ways? We show that the answer is yes: we describe chemical reaction system computations and their strand displacement realizations in which recycling of strands significantly reduces waste and avoids fuel depletion while incurring just a moderate slowdown relative to comparable computations that do not recycle strands. Thus our title: less haste, less waste. Our recycling computations are binary counters—simple and yet fundamental constructs in computation. A new feature of our strand displacement constructions is a mutex synchronization primitive, which ensures that reactions proceed atomically in the sense that all products of one reaction have been released before the next starts. The second contribution of the paper is to demonstrate a limit to recycling: recycling is not possible in certain classes of chemical reaction systems, which include certain classes of strand displacement systems that should work correctly even when many copies of the initial state of the system are present in the same environment.

The rest of this introduction illustrates the concept of strand recycling and gives an overview of our results and related work. Sections 2 and 3 then provide the technical details of the strand-recycling counters and the limits of recycling.

*Author for correspondence (condon@cs.ubc.ca).

One contribution of 13 to a Theme Issue 'Computability and the Turing centenary'.

1.1. On the potential for strand recycling

We illustrate the concept of recycling using a 3-bit counter that is specified as a chemical reaction system—details of a strand displacement implementation are in §2. This binary reflecting Gray code counter [10] follows the sequence of bit values shown in the left column of figure 1*a*. It advances in such a way that exactly one bit changes at each step and gets its name from the following property: if the states of the counter are written in a column starting from $0_n 0_{n-1} \ldots 0_1$ and a line is drawn just below row 2^{i-1}, where bit i changes from 0_i to 1_i, then in the next 2^{i-1} rows the values of the low order $i-1$ bits are

Received 31 October 2011
Accepted 20 January 2012

(*a*)

b_3	b_2	b_1	reaction	consumed	produced
0_3	0_2	0_1	(1-for)	$\mathcal{T}_1^f$	$\mathcal{T}_1^r$
0_3	0_2	1_1	(2-for)	$\mathcal{T}_2^f$	$\mathcal{T}_2^r$
0_3	1_2	1_1	(1-rev)	$\mathcal{T}_1^r$	$\mathcal{T}_1^f$
0_3	1_2	0_1	(3-for)	$\mathcal{T}_3^f$	$\mathcal{T}_3^r$
1_3	1_2	0_1	(1-for)	$\mathcal{T}_1^f$	$\mathcal{T}_1^r$
1_3	1_2	1_1	(2-rev)	$\mathcal{T}_2^r$	$\mathcal{T}_2^f$
1_3	0_2	1_1	(1-rev)	$\mathcal{T}_1^r$	$\mathcal{T}_1^f$
1_3	0_2	0_1			

(*b*)
(1) $\mathbf{0}_1 \rightleftharpoons \mathbf{1}_1$
(2) $\mathbf{0}_2 + 1_1 \rightleftharpoons \mathbf{1}_2 + 1_1$
(3) $\mathbf{0}_3 + 1_2 + 0_1 \rightleftharpoons \mathbf{1}_3 + 1_2 + 0_1$

Figure 1. (*a*) Enumeration of counter states (left three columns), reaction that advances the state from one row to the next and its direction—forward (for) or reverse (rev)—and sets of transformer molecules consumed and produced. (*b*) The chemical reaction system for a 3-bit binary reflecting Gray code counter. Species that appear on just one side of the reaction are shown in boldface, which correspond to the bit that changes value as a result of the reaction. To ensure correctness, additional 'catalyst' species appear on both sides and the corresponding bits are unchanged. At any step, only one reaction is applicable to advance the counter, although since the reactions are reversible the counter could also retreat to its previous value.

the reflection of those above the line. For example, consider bits b_2 and b_1 of the 3-bit sequence in figure 1*a*: the last four rows are a reflection of the first four rows. We call the resulting sequence of states the *Gray code sequence*.

Figure 1*b* gives the chemical reaction system for this counter, which we call GRAY. The state of the 3-bit GRAY counter is determined by three *signal* molecules, one per bit. The presence of a single copy of signal molecule 0_i denotes that the ith bit has value 0, while the presence of a single copy of 1_i denotes that the ith bit has value 1. The initial counter state is $0_30_20_1$ and the reactions ensure that exactly one of 0_i and 1_i is present at any time. The counter advances through application of the three reversible chemical reactions (1–3) of figure 1*b*. Each row of the table in figure 1*a* lists the reaction needed to produce the subsequent row; for example, the counter advances from $0_30_21_1$ to $0_31_21_1$ via reaction (2) in the forward direction (2-for).

In realizing these reactions with strand displacement systems (see §2), additional *transformer* strands that are not shown in the chemical reaction system are consumed and produced. For example, transformers might serve to ensure that all reactants are available before any product is produced, or may be side-products of a reaction. Suppose that a set of strands $\mathcal{T}_i^f$ is consumed and a set $\mathcal{T}_i^r$ is produced when reaction (i) takes place in the forward direction; conversely $\mathcal{T}_i^r$ is consumed and $\mathcal{T}_i^f$ is produced when reaction (i) takes place in the reverse direction.

The key point is that in most of the rows, the signal molecules and transformer strands that are consumed were produced by reactions of earlier rows and are thus recycled. For example, in the third step the counter advances from state $0_31_21_1$ to $0_31_20_1$, uses reaction (1) in the reverse direction (1-rev) and consumes the signal molecule 0_1 and the set of transformer strands $\mathcal{T}_1^r$ that were produced in the first step, thereby recycling $\mathcal{T}_1^r$. Only in the three rows $0_30_20_1$, $0_30_21_1$ and $0_31_20_1$—precisely those rows when a reaction occurs for the first time—the molecules consumed are not produced in earlier rows. In contrast, a chemical reaction system for a standard binary counter produces waste molecules at every step and these waste molecules are never recycled in subsequent steps.

Table 1. Comparison of n-bit counter implementations. The GRAY and GRAY-FO counters are described in §2. The Qian–Soloveichik–Winfree (QSW) counter is based on the simulation of stack machines by strand displacement reactions of Qian *et al.* [11].

properties	GRAY	GRAY-FO	QSW [11]
reaction order	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$
consumption (waste)	$\Theta(n^3)$	$\Theta(n^3)$	$\Theta(2^n)$
expected time (haste)	$\Theta(n^32^{2n})$	$\Theta(n^32^{2n})$	$\Theta(2^{2n})$

Recycling in DNA strand displacement systems offers the potential of supporting energy-efficient DNA computations in which the waste, or number of strands consumed, is logarithmic in the length of the computation. Systems that recycle molecules do not use fuel, i.e. large concentrations of certain transformer species that bias reactions in one direction, and so are not prone to problems of fuel depletion or fuel leakage. However, such advantages come at a price: our counter proceeds somewhat more slowly—is less hasty—than comparable fuel-driven strand displacement counters. The slowdown is due in part to the fact that reactions are used in both directions. Thus, our GRAY counter is not biased to advance towards the final state but rather performs an unbiased random walk, both advancing and retreating, ultimately reaching the final state. We also describe a counter (GRAY-FO) that uses reactions of fixed order, i.e. the maximum number of reactants and products in any reaction are fixed, independent of the number of counter bits.

Table 1 summarizes properties of our counters and compares with another counter, which we call Qian–Soloveichik–Winfree (QSW), based on the work of Qian *et al.* [11] (see §1.3). The properties considered are (i) *order* or max number of reactants or products of chemical reactions that describe the counter, (ii) *waste* or total number of nucleotides needed to implement the counter, and (iii) *haste* or expected time for the counter to reach a designated final state from its initial state when the volume equals the waste. We describe how order, waste and haste grow as a function of n, the number of counter bits. Here and throughout the paper, we use O-notation and

Θ-notation for this purpose. Formally, if $f(n)$ and $g(n)$ are non-negative functions, we say that $f(n)$ is $O(g(n))$ if for all n above some size, $f(n)$ is bounded above by a constant times $g(n)$, i.e. $f(n) \leq cg(n)$ for some constant c. A function $f(n)$ is $\Theta(g(n))$ if $f(n)$ is $O(g(n))$ and in addition, $f(n)$ is bounded below by a (typically different) constant times $g(n)$. That is, $f(n)$ is $\Theta(g(n))$ if $c'g(n) \leq f(n) \leq cg(n)$ for some constants c and c' and all sufficiently large n. Our use of Θ-notation enables us to distinguish between large differences in the growth of two functions without getting bogged down in details. For example, we will show that the waste of the GRAY counter is $\Theta(n^3)$, meaning that the number of nucleotides of waste molecules needed to implement an n-bit counter grows roughly proportionally to n^3, which is markedly less than the exponential ($\Theta(2^n)$) growth of waste molecules for the QSW counter.

Our n-bit binary reflecting Gray code counter, GRAY, uses reactions of maximum order $\Theta(n)$, generates only $\Theta(n^3)$ waste and uses expected time $\Theta(n^3 2^{2n})$ to reach the final state. Our GRAY-FO counter improves on the GRAY counter in that the reaction order is $\Theta(1)$. The QSW counter also has reaction order $\Theta(1)$ and has expected time $\Theta(2^{2n})$, which is somewhat better than the expected time needed by our counters. However, the QSW counter generates $\Theta(2^n)$ waste, exponentially worse than our counters. All three counters are deterministic in that they advance and retreat through a predetermined linear ordering of states.

1.2. On the limits of strand recycling

Our n-bit GRAY counter advances correctly through 2^n states because only single copies of initial species are present. In §3, we show that if $\Theta(n)$ copies of the initial species are present, then the counter does not advance properly in a very strong sense: the final state of the counter can be reached in just $O(n^2)$ chemical reactions, rather than using the intended sequence of 2^n reactions. We also prove more general limits on molecule recycling when multiple copies of the initial species are present, under some restrictions on the allowable chemical reaction systems. In particular, if the waste of such a chemical reaction system is logarithmic in the length of a valid computation, the system does not work correctly when polynomially many copies of the initial reactants are present.

1.3. Related work

Qian *et al.* [11] showed how to simulate a stack machine using strand displacement systems. A binary counter can be implemented via a stack machine; we call such a counter a QSW counter and we compare its properties and resources with our counters in table 1. We compare our results to a fuel-biased QSW counter as the unbiased version is slower and we also assume that all fuel must be initially present in the reaction volume—the counter operates in a closed system.

Cardelli [12,13] has shown how primitives that support concurrent models of computation, such as fork and join gates, can be implemented using strand displacement systems. Many of our techniques are similar to those of Cardelli's constructions: for example, our signal strands share a common toehold while the long domains are distinct, and we do not use branched structures. To effect an abstract chemical reaction with i reactants and i products, we use cascading of strand exchanges whereby the reactants are first absorbed (by transformer molecules) and products are then released by further strand exchanges. This order of events is similar to an i-way join followed by an i-way fork of Cardelli; it is similar also to the strand displacement realizations of i-way molecular reactions of Qian *et al.* [11]. A new feature of our constructions is the use of a *mutex* strand to ensure that the $(k+1)$th reaction of a deterministic computation does not proceed until all products of the kth reaction have been produced.

Building on models of Winfree and Rothemund [14,15], Reif *et al.* [16] studied a tile-based graph assembly model in which tiles may both adhere to and be removed from a tile assembly. In their self-destructible graph assembly model, the removal of tiles allows for the possibility of tile reuse. The authors demonstrate that tile reuse is possible in an abstract tile model, via a PSPACE-hardness result. Doty *et al.* [17] showed a negative result on tile reuse for an irreversible variant of the model of Reif *et al.*

Kharam *et al.* [18] describe a DNA binary counter in which bit values are represented using relative concentrations of two molecule species. This is very different than our work in this paper, where the values of bits (0 and 1) are represented by the absence (or presence) of certain signal molecules.

2. GRAY: A BINARY REFLECTING GRAY CODE COUNTER

Here, we describe the chemical reaction system and strand displacement implementation of our GRAY counter, provide a proof of its correctness and analyse its expected time (haste) and space usage (waste). We show how it can be modified to use only bi-molecular reactions, resulting in our fixed-order GRAY counter: GRAY-FO.

2.1. Chemical reaction system for the GRAY counter

We generalize the 3-bit GRAY counter in §1.1 to n-bits. The counter state is represented by n *signal* molecules, one per bit. Presence of signal molecule b_i denotes that the ith bit has value b_i, for $b = 0$ or $b = 1$. Initially, the state is $0_n \ldots 0_2\, 0_1$. Each possible state of the counter represents a value in the Gray code sequence. The counter is described abstractly by the following chemical reactions:

$$\text{(gc-1)} \quad \mathbf{0_1} \rightleftharpoons \mathbf{1_1}$$

$$\text{(gc-}i\text{)} \quad \mathbf{0_i} + 1_{i-1} + 0_{i-2} + \cdots + 0_1 \rightleftharpoons \mathbf{1_i} + 1_{i-1} + 0_{i-2} + \cdots + 0_1, \quad 2 \leq i \leq n.$$

Lemma 2.1. *This chemical reaction system ensures that the GRAY counter, when in state v, can only advance to state v_{next}, or retreat to state v_{prev}, corresponding to the next or previous value in the Gray code sequence,*

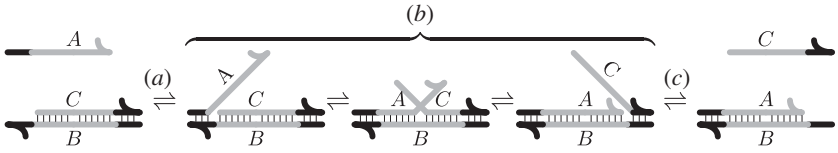

Figure 2. Strand displacement. (*a*) Toehold (black subsequence) of molecule *A* binds with its unpaired complement on molecule *B*. (*b*) Domain (grey subsequence) δ of *A* competes via a random walk process with the δ domain of *C* to bind with the complementary domain δ^* of *B* until all bases of *A* are bound to *B*. (*c*) Toehold of *C* detaches from *B*, at which point it has been displaced.

respectively, if each reaction is atomic,[1] *and all initial signal molecules exist as single copies.*

Proof. First, observe that at any state of the system, for each i, exactly one of the signal molecules 0_i and 1_i is present (in an unbound state). Hence, at any state of the system, only two reactions can be applied: (gc-1) and (gc-i), where i is the smallest index such that signal molecule 1_{i-1} is present. Indeed, the reactions (gc-j), where $2 \leq j < i$, cannot be applied as, by the definition of i, signal molecule 0_{j-1} is present, and hence, 1_{j-1} is not present. Similarly, the reactions (gc-j), where $j > i$, cannot be applied as signal molecule 1_{i-1} is present, and hence, 0_{i-1} is not present. It follows that at any state of the system, the system can only progress in the forward or the backward directions. ■

2.2. Strand displacement implementation of the GRAY counter

Soloveichik *et al.* [7] showed that arbitrary chemical reaction systems could be realized by using DNA molecules as the chemical species and DNA strand displacement reactions to implement reactions among those species. We illustrate a simple, reversible version of strand displacement in figure 2. First, a subsequence of a single-stranded molecule *A* binds to a duplex *B*-*C* by forming base pairs with a short complementary subsequence of unpaired bases of *B*. The short subsequences are called *toeholds* and are the black segments of the molecules in figure 2. Then, in a random walk process (often referred to as branch migration), the remaining bases of *A* compete with *C* to form base pairs with *B* because both *A* and *C* contain an identical subsequence δ that is complementary to a subsequence of *B*. We refer to sequence δ as a *domain*. Once the δ domain of *A* has bound to its complement δ^* of *B*, *C* remains bound to *B* by just a short toehold. The toehold bonds can break, thereby releasing *C*. (Of course, *A* may detach from the duplex *B*-*C* before *C* is released, in which case the reaction does not happen.) The reaction is reversible because *C* can bind to duplex *A*-*B* to displace *A* via the same principles. We refer to *A* and *C*, i.e. the molecules that are bound and released, as *signal molecules* and we refer to the duplexes *B*-*C* and *A*-*B* as transformer molecules. Abstractly, the overall process can be viewed as the reaction $\boldsymbol{A} \rightleftharpoons \boldsymbol{C}$ where, abusing notation somewhat, we use *A* and *C* to denote abstract molecular species, rather than individual DNA signal molecules that realize these species and we ignore the transformers. Consistent with Soloveichik *et al.* [7], we assume throughout that if an environment contains DNA strands with different domains, say δ and δ', then domains can be designed to be sufficiently different that a strand with domain δ never displaces a strand with domain δ'.

A strand displacement implementation of the GRAY counter requires a means to simulate its chemical reaction equations, which involve multiple reactants and products. Furthermore, the correctness of the counter is predicated on the assumption that each chemical reaction is *atomic*.

Qian *et al.* [11] proposed a construction—hereafter called the QSW construction—that is capable of simulating bi-molecular, and higher order, chemical reactions. Specifically, the construction can exchange a set of signal molecules (the reactants) for another set of signal molecules (the products) through a sequence of strand displacement events. Unfortunately, the construction does not simulate the higher order reaction atomically, because some product signal molecules can start initiating other reactions before all product signal molecules are produced. However, the strand displacements do occur in a fixed order and all reactant signal molecules are consumed before any product signal molecule is produced. We exploit this fact to simulate atomicity.

In particular, we borrow the concept of *transactions* from database and concurrency theory—a group of operations that either completes or does not complete in its entirety, and does not interfere with any other transaction. We implement transactions using a simple synchronization primitive: a *mutex*. A transaction must acquire the mutex in order to start, and releases it only when it completes. The state of our counter is defined only when the mutex is available. More precisely, let μ denote a single copy of a signal species representing the mutex. In any sequence of strand displacements representing a chemical reaction, μ is the first reactant to be consumed and the last product to be produced. Therefore, only one reaction (transaction) can be in progress at any given time. When μ is next available, either all strand displacements in the sequence took place and the counter is in a new state—the transaction succeeded—or the counter is in the same state and the configuration of all molecules is exactly the same prior to the reaction beginning—the transaction failed. Because each reaction is implemented as a transaction, it appears atomic and cannot interfere with other reactions.

[1]'Atomic' is standard computer science terminology for something that occurs as if all at once, hearkening back to the original Greek etymology of an atom as an indivisible unit. Reasoning about chemical reactions as computational processes can unfortunately result in clashes in terminology.

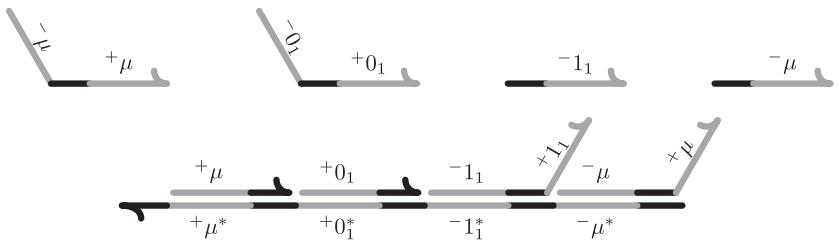

Figure 3. An example of signal molecules (top two left strands) and the transformer, consisting of auxiliary strands (top two right strands) and a saturated template strand (bottom complex) associated with the forward direction of $\mathbf{0_1 \rightleftharpoons 1_1}$. In this and later figures, the Watson–Crick complement of a domain x is denoted by x^* . The state of the system shown is 0_1 .

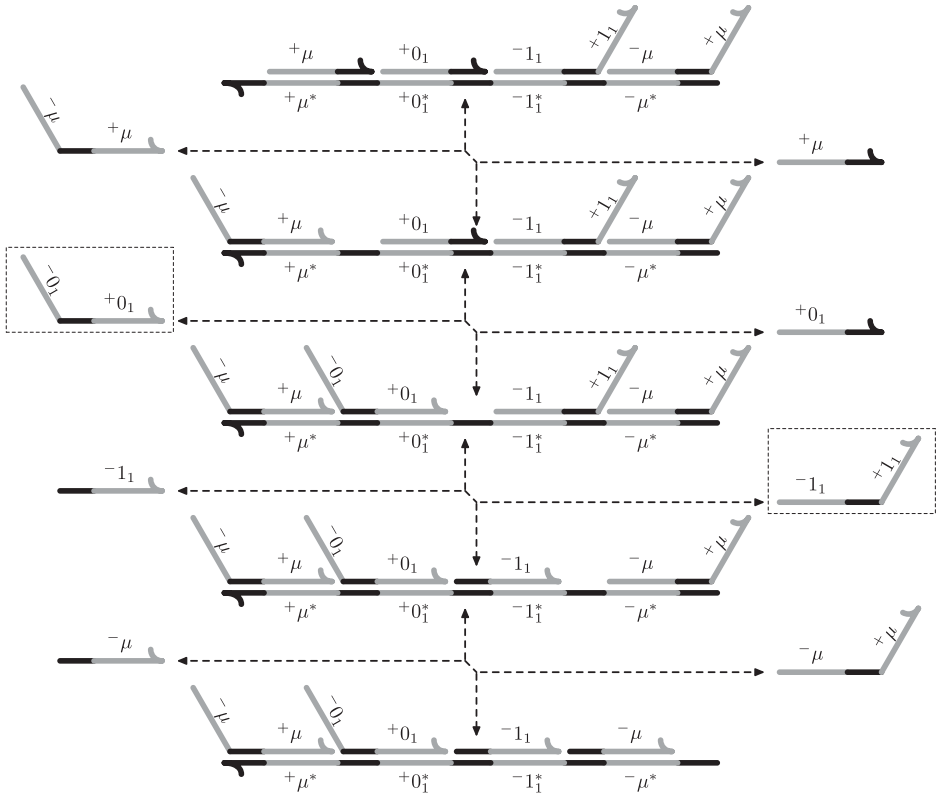

Figure 4. The sequence of strand displacement events for the reaction $\mathbf{0_1 \rightleftharpoons 1_1}$.

We will use only one type of toehold, and therefore we will not label toeholds in the figures below. All signal molecules in the QSW construction are of the same form: a negative recognition domain ^{-}d, followed by a toehold t and by a positive recognition domain ^{+}d. The construction also uses auxiliary strands consisting of a single domain and a single toehold, and one template strand initially bound to signal molecules and auxiliary strands in such a way that no domain of the template is uncovered. We call a template strand with all domains bound to other strands, *saturated*. We refer to the saturated template complex and associated auxiliary strands, collectively, as a transformer. An example of the signal molecules and the transformer associated with the forward direction of the reaction $\mathbf{0_1 \rightleftharpoons 1_1}$ is given in figure 3.

As previously discussed, the reaction can only initiate if the signal molecule $\boldsymbol{\mu}$ is present, and can only complete if all other reactants—in this case 0_1, assuming a forward reaction—are available. An example of the sequence of strand displacements for the reaction $\mathbf{0_1 \rightleftharpoons 1_1}$ is given in figure 4. The reaction proceeds from top to bottom in the forward direction and from bottom to top in the backward direction.

The transformers that implement the ith reaction (gc-i) are a straightforward generalization of the first reaction. As before, the signal molecule $\boldsymbol{\mu}$ must initiate the first strand displacement, and is not produced until the last strand displacement. The number of required intermediate strand displacement reactions is dependent on the number of reactants and products. Specifically, the ith reaction requires $2i + 2$ strand displacements to complete. An example of the transformer for the ith reaction is given in figure 5.

2.3. Correctness

In the reactions of our counter, strand displacement should only happen when the toehold of the invading strand first binds to a free toehold, following which

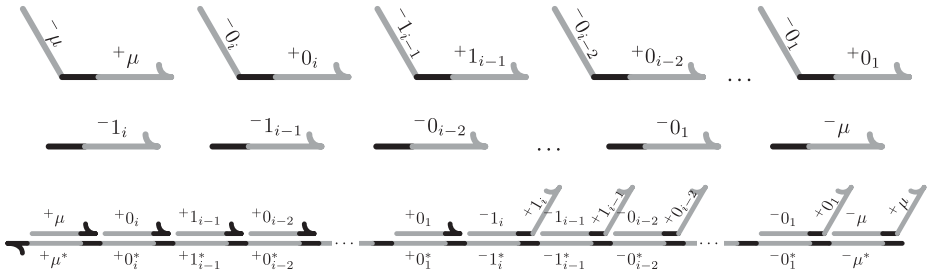

Figure 5. An example of the signal molecules and the transformer molecules for the ith reaction. The counter is in state $b_n \dots b_{i+1}$ $0_i\ 1_{i-1}\ 0_{i-2} \dots 0_1$.

a domain of the invading strand displaces the bound domain of the strand being released. The invading and released domains should be identical. We say that such a strand displacement is *legal*. Illegal strand displacements can arise when the invading domain is different from the released domain; we call such displacements *mismatched domain displacements*. Illegal strand displacements can also arise due to blunt-end displacement, i.e. displacements where invading and released domains are identical but domain displacement is not preceded by the binding of a free toehold, or when more than one invading domain strand displaces the strand being released. Designing strands to ensure only legal displacements occur with high probability is beyond the scope of this paper. We will assume that only legal displacements occur.

Lemma 2.2. *The earlier mentioned strand displacement implementation of the GRAY counter ensures that all chemical reactions occur as transactions* (*and therefore appear atomic*), *assuming all strand displacements are legal.*

Proof. We argue by induction on the sequence of chemical reactions. Prior to any chemical reaction beginning, we require the following invariant to hold: (i) all template strands of all transformers are saturated, and require the mutex signal molecule $\boldsymbol{\mu}$ to initiate the first strand displacement and (ii) there is exactly one available copy of $\boldsymbol{\mu}$. The invariant is trivially satisfied for the base case, when no reaction has yet occurred. Suppose the first $i-1$ reactions appear atomic, and the invariant is satisfied. Without loss of generality, suppose the next attempted reaction involves the kth transformer.

Because we assume that all strand displacements are legal, no auxiliary strand or signal molecule representing the value of a digit can displace any strand in any transformer. Since there is exactly one available copy of the mutex signal species $\boldsymbol{\mu}$, that strand alone can initiate a reaction. Suppose the reaction is in the forward direction, as the reverse direction is symmetric. The signal molecule $\boldsymbol{\mu}$ must initiate the first strand displacement by binding to the left end of the kth transformer's template strand. This begins the transaction. Note that there is another copy of $\boldsymbol{\mu}$ sequestered at the right end of the template. When the signal molecule $\boldsymbol{\mu}$ is once again produced, there are two cases to consider.

Case 1. If the copy on the right end of the transformer is released, then the transaction succeeded and the counter is in a new state. Furthermore, the invariant is preserved as (i) the kth transformer is saturated, and only a signal strand $\boldsymbol{\mu}$ can initiate a new reaction on the right end of the template and (ii) exactly one signal molecule $\boldsymbol{\mu}$ was produced as the final strand displacement.

Case 2. Otherwise, the original copy of $\boldsymbol{\mu}$ was released, the transaction failed, and the counter is in the same state, satisfying the invariant, as any intermediate strand displacements must have been reversed prior to the original $\boldsymbol{\mu}$ signal molecule being released.

Importantly, whether or not a transaction succeeds, while one is in progress no other reaction can be initiated because no other copy of signal species $\boldsymbol{\mu}$ is available. Thus, all reactions are implemented as transactions and appear atomic. ■

2.4. Waste and haste analysis of the GRAY counter

Here, we analyse the waste—the total number of nucleotide bases of all species consumed and haste—expected time—of the GRAY counter as it advances from initial to final states. We assume single copies of the initial signal, transformer and mutex species. To analyse waste, we first count the number of bases required for all initial signal, transformer and mutex molecules.

Lemma 2.3. *The total number of nucleotide bases needed for a single copy of each initial signal, transformer and mutex species of the n-bit GRAY counter is* $\boldsymbol{\Theta}(n^3)$.

Proof. Each signal molecule 0_i and the initial mutex molecule $\boldsymbol{\mu}$ is composed of a toehold and two long domains. The same is true of the molecules for states 1_i and the sequestered signal molecules $\boldsymbol{\mu}$ that are part of the initial transformer species. There is an auxiliary transformer strand molecules consisting of one toehold and one long domain for each type of signal species. Similar to previous strand displacement implementations [7], we assume the $\boldsymbol{\Theta}(n)$ domains of the signal species can be designed to have length $\boldsymbol{\Theta}(n)$ to ensure only legal displacements occur for the duration of the counter. We choose the toehold length to be $\boldsymbol{\Theta}(1)$ Because the domain length dominates the toehold length, the total number of bases in all signal species and auxiliary strands is $\boldsymbol{\Theta}(n^2)$.

The template strands in the sets $\mathcal{T}_i^f$ and $\mathcal{T}_i^r$ have $\boldsymbol{\Theta}(i)$ domains, which dominate their length, and thus the template strands have length $\boldsymbol{\Theta}(in)$. Thus, the total number of bases in all transformer molecules in the system is $\sum_{i=1}^{n} \boldsymbol{\Theta}(in) = \boldsymbol{\Theta}(n^3)$.

Next, consider the expected time (haste) for the counter to progress from its initial to final states.

Lemma 2.4. *Assuming a single copy of each initial signal, transformer and mutex species, and that all strand displacements are legal and all reactions occur*

as transactions (*appear atomic*), *the GRAY counter advances through the* 2^n *states of the Gray code sequence in* $\Theta(n^3 2^{2n})$ *expected time* (*haste*).

Proof. We assume that reactions occur in a volume of size $\Theta(n^3)$, because this is the total number of bases of species in the system. Each strand displacement step involves interaction between two species and thus the rate of each strand displacement step is $1/\Theta(n^3)$.

First, consider the *shortest path* from the initial state to the final state. On this path, each order-i reaction is applied 2^{n-i} times and involves $\Theta(i)$ strand displacements. Thus, the total number of strand displacement steps along the shortest path is $\sum_{i=1}^{n} \Theta(i)2^{n-i} = \Theta(2^n)$.

Because each reaction is reversible, the system does not strictly follow the shortest path but rather proceeds as an unbiased random walk along this path. The expected number of steps for a random walk to reach one end of a length-$\Theta(2^n)$ path from the other is $\Theta((2^n)^2) = \Theta(2^{2n})$ [19]. Therefore, the expected number of strand displacement steps is $\Theta(2^{2n})$. Since each strand displacement step occurs at a rate of $1/\Theta(n^3)$, the overall expected time—the haste—is $\Theta(n^3 2^{2n})$. Note that the haste is polynomial in the $\Omega(2^n)$ steps required to proceed through all 2^n unique states of an n-bit binary Gray code counter. ■

Combining lemmas 2.1–2.4, we have our first main result.

Theorem 2.5. *An n-bit binary reflecting Gray code counter can be implemented as a DNA strand displacement system that proceeds through the* 2^n *unique states of the Gray code sequence in* $\Theta(n^3 2^{2n})$ *expected time* (*haste*) *and uses only* $\Theta(n^3)$ *nucleotides* (*waste*).

2.5. A fixed-order implementation of the GRAY counter

An n-digit GRAY counter can perform a computation having length exponential in n, while only generating waste polynomial in n. However, it relied on template strands containing $O(n)$ domains, each of length $O(n)$, resulting in an overall length of $O(n^2)$ bases. Synthesis of long nucleic acid strands is challenging, and the fidelity of synthesized strands generally decreases as sequence length increases. For this reason, it is desirable to bound the length of all strands in the system to $O(n)$ bases. We now briefly describe how a template strand from the GRAY counter consisting of $2i+2$ domains can be split into $i+1$ template strands requiring four domains each, for any $i > 1$. The overall waste will only be increased by a constant, resulting in the same volume, and thus the same haste.

To simplify the description, we introduce some notation. Consider the (gc-i) reaction of the GRAY counter that has i reactants and i products:

$$\mathbf{0_i} + 1_{i-1} + 0_{i-2} + \cdots + 0_1 \rightleftharpoons \mathbf{1_i} + 1_{i-1} + 0_{i-2} + \cdots + 0_1.$$

The previous implementation demonstrated that by using the QSW construction and introducing a mutex species μ—thus creating an order $i+1$ reaction—chemical reactions occur as transactions and therefore appear atomic. Specifically, μ is first consumed, then 0_i, then 1_{i-1} and so on. Likewise, after all reactants are consumed, 1_i is first produced, then 1_{i-1} and so on, until finally μ is produced. We denote a strand displacement implementation supporting a transaction of this type, which is initiated by consuming a mutex α, and terminated when producing a mutex β, by

$$[\alpha + 0_i + 1_{i-1} + 0_{i-2} + \cdots + 0_1 \rightleftharpoons 1_i + 1_{i-1} + 0_{i-2} + \cdots + 0_1 + \beta].$$

In the case of the GRAY counter, $\alpha = \beta = \mu$. Our goal is to convert this order $i+1$ reaction into a cascade of $i+1$ bi-molecular reactions, while preserving the appearance of atomicity. Using the earlier mentioned notation, we implement the following reaction cascade:

$$\left.\begin{array}{l} [\mu + 1_{i-1} \rightleftharpoons 1_{i-1} + \alpha_i^1] \\ [\alpha_i^1 + 0_{i-2} \rightleftharpoons 0_{i-2} + \alpha_i^2] \\ \quad\vdots \\ [\alpha_i^{i-2} + 0_1 \rightleftharpoons 0_1 + \alpha_i^{i-1}] \end{array}\right\} \text{catalysts checked in sequence}$$

$$\left.\begin{array}{l} [\alpha_i^{i-1} + 0_i \rightleftharpoons 1_i + \alpha_i^{i-1}] \\ [\alpha_i^{i-1} + 1_i \rightleftharpoons 1_i + \mu] \end{array}\right\} \text{the } i\text{th bit flipped and the mutex signal molecule released.}$$

The overall transaction has been split into a cascade of sub-transactions. Each sub-transaction is implemented as a bi-molecular reaction using the QSW construction (figure 4). The first $i-1$ sub-transactions check, in sequence, that all $i-1$ catalysts are present. The mutex signal molecule μ is consumed during the first check. The last two transactions will first perform the bit flip and then releases the mutex signal molecule. Every sub-transaction, except the $(i-1)$st and the ith, produces a unique mutex molecule that is required to initiate the next sub-transaction in the cascade. Upon successful completion of the first i sub-transactions in the cascade, the final sub-transaction occurs, producing the original mutex molecule μ, and thus finalizing the overall transaction. The implementation works in the reverse direction in a similar way with the exception that the bit is flipped first and the mutex signal molecule μ is released only after the presence of all catalysts have been verified. Using the earlier mentioned transformation for all higher order reactions in the original GRAY counter implementation results in a new, fixed-order counter, GRAY-FO.

3. LIMITS ON MOLECULE RECYCLING IN GENERAL CHEMICAL REACTION SYSTEMS

In this section, we show that certain classes of chemical reaction systems that efficiently recycle strands, or that can perform useful computations for time that significantly exceeds the number of signal molecules, cannot work properly when multiple copies of the initial signal molecules are present. In particular, our GRAY counters do not work in a multi-copy setting.

The underlying problem is the representation of the state of the system as specific *combinations* of signal molecules. If there are multiple copies of the system in

the same reaction vessel—as would typically occur in a laboratory setting—then the states of the different copies may interfere with one another. To illustrate this point, we again consider the 3-bit GRAY counter. Initially, in a single copy of the construction, the signal molecules $\{0_3, 0_2, 0_1\}$ denote the state $0_3 0_2 0_1$. Consider a two-copy system where the initial set of present signal molecules is duplicated, yielding the multiset $\{0_3, 0_3, 0_2, 0_2, 0_1, 0_1\}$. As in the single copy case, assume reaction (1) occurs in the forward direction, followed by reaction (2) in the forward direction. The resulting multiset of signal molecules is $\{0_3, 0_3, 0_2, 1_2, 0_1, 1_1\}$. In the single copy case, we intend that reaction (1) in the reverse direction will occur next; however, given the current set of present signal molecule in the two-copy case, reaction (3) in the forward direction could instead occur, resulting in the multiset $\{0_3, 1_3, 0_2, 1_2, 0_1, 1_1\}$. At this point, a copy of every signal molecules is present, and any reaction can occur, in either direction. Furthermore, the single copy case required *at least* seven reactions to produce the final state $1_3 0_2 0_1$, whereas the two-copy case can reach it in three. Crosstalk between the copies has broken the counter.

In the remainder of this section, we treat this problem formally. We define a *chemical reaction system* to be a tuple $\mathbf{C} = \{S, \mathcal{R}, S_0, s_{\text{end}}\}$, where

- $\mathcal{S}$ is a set of signal species.
- $\mathcal{R}$ is a set of *reaction equations*, where each $R \in \mathcal{R}$ is an ordered pair of multisets of signal molecules. Intuitively, a reaction equation $R = (I, P)$ consumes the signal molecules in I as *inputs* and produces the signal molecules in P as *products*. Our formalism is directional to allow modelling non-reversible reactions; a reversible chemical reaction is modelled as two separate elements of $\mathcal{R}$, i.e. (I, P) and (P, I).
- S_0 is a multiset of signal molecules initially present.
- $s_{\text{end}} \in S$ is a signal molecule denoting the end of computation.[2]

Example 3.1. Let us describe the 3-bit GRAY counter as a chemical reaction system. The set of signal species is $S = \{0_1, 0_2, 0_3, 1_1, 1_2, 1_3, s_{\text{end}}\}$, where s_{end} denotes the end of computation. The initial multiset is $S_0 = \{0_1, 0_2, 0_3\}$. Finally, we have the following set of reaction equations:

$$\begin{gathered}\{R_{\text{1-for}} = (\{0_1\}, \{1_1\}), R_{\text{1-rev}} = (\{1_1\}, \{0_1\}),\\ R_{\text{2-for}} = (\{1_1, 0_2\}, \{1_1, 1_2\}),\\ R_{\text{2-rev}} = (\{1_1, 1_2\}, \{1_1, 0_2\}),\\ R_{\text{3-for}} = (\{0_1, 1_2, 0_3\}, \{0_1, 1_2, 1_3\}),\\ R_{\text{3-rev}} = (\{0_1, 1_2, 1_3\}, \{0_1, 1_2, 0_3\})\\ R_{\text{end}} = (\{0_1, 0_2, 1_3\}, \{s_{\text{end}}\})\}.\end{gathered}$$

These reactions, with the exception of the last one, formally define the reactions shown in figure 1*b*.

An *x-copy* version of $\mathbf{C}$, denoted $\mathbf{C}^{(x)}$, is obtained by duplicating the initial multiset S_0 x times, i.e. $\mathbf{C}^{(x)} = \langle S, \mathcal{R}, S_0^{(x)}, s_{\text{end}}\rangle$, where $S_0^{(x)}$ is a multiset consisting of x copies of S_0.

[2]A computation may have multiple final states. To model this situation, we can let s_{end} be produced in all final reactions, in addition to any other signal molecules that may indicate the result of the computation.

We formalize computations in $\mathbf{C}$ in the natural manner: let $\boldsymbol{\rho}$ be a sequence of reactions $R_1, R_2, \ldots, R_m$ from $\mathcal{R}$, where each $R_i = (I_i, P_i)$. We define $\boldsymbol{\rho}$ to be a *trace* of $\mathbf{C}$ if $\boldsymbol{\rho}$ induces a corresponding sequence of multisets $S_0, S_1, \ldots, S_m$, with S_0 being the multiset of initial signal molecules in $\mathbf{C}$, and for all $1 \le i \le m$, we have both $I_i \subseteq S_{i-1}$ and $S_i = S_{i-1} - I_i + P_i$. (We use '$-$' and '$+$' to denote multiset subtraction and union.)

Example 3.1 (continued). The shortest trace producing s_{end} is the sequence of reactions

$$R_{\text{1-for}}, R_{\text{2-for}}, R_{\text{1-rev}}, R_{\text{3-for}}, R_{\text{1-for}}, R_{\text{2-rev}}, R_{\text{1-rev}}, R_{\text{end}},$$

which induces the following sequence of multisets:

$$\{0_1, 0_2, 0_3\}, \{1_1, 0_2, 0_3\}, \{1_1, 1_2, 0_3\}, \{0_1, 1_2, 0_3\}$$

and

$$\{0_1, 1_2, 1_3\}, \{1_1, 1_2, 1_3\}, \{1_1, 0_2, 1_3\}, \{0_1, 0_2, 1_3\}, \{s_{\text{end}}\}.$$

We use B_s to denote the *bandwidth* of signal species s, i.e. the maximum number of copies of s that appears in a multiset I of any reaction $(I, P) \in \mathcal{R}$, and we use $B_{\mathbf{C}}$ to denote the *bandwidth* of $\mathbf{C}$, i.e. the sum of bandwidths of all signal species in S.

The next definitions help delineate the class of chemical reaction systems for the main result of this section. For a reaction equation $R = (I, P)$, consider the signal molecules in $I - P$. We dub these input signal molecules *proper*. The other signal molecules, in $I \cap P$, function as catalysts—they are necessary for the reaction, but not consumed. We define a set of reactions to be *k-proper* if k is the maximum number of proper inputs of all reactions in the set. (Since I is a multiset, each proper copy of the same input molecule in I contributes one to the overall count of proper inputs.) Note that the GRAY counter system is 1-proper. Finally, we use $|X|$ to denote the number of elements in X with multiplicities for any multiset X.

Theorem 3.2. *Let $\mathbf{C} = \langle S, \mathcal{R}, S_0, s_{\text{end}}\rangle$ be a 1-proper chemical reaction system. If there exists a trace that produces s_{end} in $\mathbf{C}$, then for the x-copy chemical reaction system $\mathbf{C}^{(x)}$ with $x \ge B_{\mathbf{C}} + 1$, there exists a computation that produces s_{end} in at most $(B_{\mathbf{C}} + 1)B_{\mathbf{C}}/2 + 1$ steps.*

Proof. Let $\boldsymbol{\rho} = R_1, \ldots, R_m$ be a trace for a computation that produces s_{end} in the last step in the (single-copy) system $\mathbf{C}$ and let $S_0, \ldots, S_m$ be the corresponding sequence of multisets of signal molecules. Let S' be the multiset of signal molecules obtained by including w_s copies of each $s \in S$, where $w_s \ge 0$ is the maximum number of copies of signal molecule s that appears in a multiset I of any reaction (I, P) of the sequence $\boldsymbol{\rho}$. Note that $|S'| = B_{\mathbf{C}}$. Let $k = |S' - S_0|$. Note that $k \le B_{\mathbf{C}}$. The goal is to produce all signal molecules in the multiset $S' - S_0$, so that we can apply the last reaction of $\boldsymbol{\rho}$ and produce s_{end}.

We construct a trace of the appropriate length for the multi-copy system from the trace $\boldsymbol{\rho}$ for the single-copy system. The high-level structure of the proof is as follows. First, we project out from $\boldsymbol{\rho}$ the k reactions, in order, that first produce each of the molecules in the multiset $S' - S_0$. From that sequence, we build a trace of the multi-copy system that is the concatenation of k phases. Each phase adds one more signal molecule to the multiset of

signal molecules present, preserves the presence of all signal molecules previously produced and 'consumes' one copy of the initial signal molecules in S_0. We will show that the jth phase is at most j reactions long; so the total length of the trace producing $S' - S_0$ is bounded by $\sum_{j=1}^{k} j = (k+1)k/2 \leq (B_{\mathbf{C}}+1)B_{\mathbf{C}}/2$.

We now formalize the construction of the k phases. Define the first appearance of the cth copy of signal molecule s to be in S_i if there are at least c copies of s in multiset S_i and less than c copies of s in each of S_0, $S_1, \ldots, S_{i-1}$. Let $s_1, \ldots, s_k$ be the sequence of signal molecules (with multiplicities) from $S' - S_0$ in order of their first appearances in $S_1, \ldots, S_m$ and let $R_{\text{index}}(s_j)$ be the reaction in $\boldsymbol{\rho}$ that first produced this copy of s_j. In other words, $R_{\text{index}}(s_j)$ is the reaction that produced the first appearance of s_j (where s_j is the cth copy of some signal molecule s, for some c). The k phases will produce the signal molecules in $S' - S_0$ exactly in this order: signal molecule s_j will be produced in phase j. Each phase j will consist of several reactions: 0 to j which will produce s_j without removing any other signal molecule from set $S' - S_0$, but they can remove one signal molecule from S_0, which is replenished by adding one new copy of S_0 at the beginning of this phase. To find the sequence of these reactions, we will work backwards. Assuming s_j is not already present in the current multiset of signal molecules, we use reaction $R_{\text{index}(s_j)}$ to produce s_j. As a result, we might have removed one of the signal molecules $s_1, \ldots, s_{j-1}$, say s_i. If that is the case, we repeat the process of producing signal molecule s_i, i.e. repeating reactions of phase i.

The k phases are constructed to maintain three invariants.

— After the jth phase, the multiset of signal molecules contains the multiset $\{s_1, \ldots, s_j\}$.
— The trace constructed so far has not relied on the existence of more than j copies of the initial signal molecules S_0.
— For each $i \leq j$, the ith phase has used at most i reactions.

The invariants are vacuously true initially (before any phases). Assuming they are true after $j-1$ phases, we construct the jth phase as follows. If s_j is already present in the current multiset of molecules, we do nothing. Otherwise, the first reaction in the phase is $R_{\text{index}}(s_j)$, the reaction that produced s_j for the first time. We know this reaction can be applied because all of $\{s_1, \ldots, s_{j-1}\}$ are available, as well as the jth copy of S_0. This guarantees that the multiset now contains s_j, and we have relied on only j copies of S_0. However, because the system is 1-proper, the reaction consumed at most 1 input signal molecule. If the reaction consumed 0 molecules, or if the 1 molecule is in S_0, the invariant is maintained and the phase ends. Otherwise, the reaction consumed some s_i, where $i < j$. To restore s_i to the multiset, we repeat the sequence of reactions of the ith phase. Note that this is valid because the new copy of S_0 was not yet used and all signal molecules required for phase i are still present. The number of reactions of phase j can be bounded by the number of reactions of phase i plus one. By the induction invariant, the ith phase requires at most i reactions and since $i < j$, the jth phase requires at most j reactions.

Concatenating the k phases produces a trace for the k-copy chemical reaction system $\mathbf{C}^{(k)}$, which produces all of $\{s_1, \ldots, s_k\}$ within $(k+1)k/2$ reactions. If s_{end} is not in $S' - S_0 = \{s_1, \ldots, s_k\}$, then S' contains all inputs needed for the last reaction in $\boldsymbol{\rho}$ that produces s_{end}. Thus, to produce s_{end}, we might need one additional copy of the initial set and one additional step. Since $k \leq B_{\mathbf{C}}$, the result follows. ■

The following two examples show that theorem 3.2 cannot be improved without altering the definition of chemical reactions systems. The first example shows that the bound on the number of steps is tight.

Example 3.3. Consider the following 1-proper chemical reaction system $\mathbf{E}_1$:

$$\begin{gathered} s_0 \rightleftharpoons s_1 \\ s_1 \rightleftharpoons s_2 \\ \vdots \\ s_{k-2} \rightleftharpoons s_{k-1} \\ s_{k-1} + s_1 + s_2 + \cdots + s_{k-1} \rightleftharpoons s_k + s_1 + s_2 + \cdots + s_{k-1} \end{gathered}$$

with the initial set containing k copies of s_0. Consider the x-copy chemical reaction system $\mathbf{E}_1^{(x)}$, where x is arbitrarily large. To produce s_k, we need one copy of signal molecules $s_1, \ldots, s_{k-2}$ and two copies of signal molecule s_{k-1}. The number of steps needed to produce signal molecule s_i from s_0, $i < k$, is i. Hence, the number of steps needed to produce all required signal molecules is $1 + 2 + \ldots + k - 1 + k - 1 = (k+1)k/2 - 1$. Thus, the number of steps needed to produce s_k is $(k+1)k/2$, which meets the bound of theorem 3.2 because $k = |S' - S_0|$.

The second example shows that the assumption that the system is 1-proper is crucial.

Example 3.4. Consider the following 2-proper chemical reaction system $\mathbf{E}_2$:

$$s_i + s_i \rightleftharpoons s_{i+1} + s_0, \quad \text{for } i = 0, \ldots, k-1.$$

with the initial set containing k copies of s_0 . Consider the x-copy chemical reaction system $\mathbf{E}_2^{(x)}$, where x is arbitrarily large. We will show that the number of steps needed to produce s_k is exponential in k.

Initially, we have $n = xk$ copies of s_0. Let n_i be the number of copies of s_i. Hence, initially, $n_0 = n$ and $n_1 = \cdots = n_k = 0$. Note that the chemical reaction system preserves the number of signal molecules present, i.e. at any time, the total number of signal molecules is n. Consider the function F of n_i's defined as follows:

$$F(n_0, \ldots, n_k) = \sum_{i=0}^{k} 2^i n_i.$$

Note that

— initially, the value of F is n;
— each forward reaction increases the value of F by 1 (and each backward reaction decreases it by 1); and
— if a state containing at least one copy of s_k is reached then the value of F is at least $n - 1 + 2^k$.

Hence, the number of steps needed to produce s_k is at least $2^k - 1$.

Note that theorem 3.2 is much stronger than our intuitive notion of crosstalk short-circuiting a computation. It states that with only a linear number of copies, any signal molecule can be produced in at most a quadratic length computation.

We can formalize the intuitive notion of short-circuiting. A system $\mathbf{C}$ is *x-copy-tolerant* if, for all $s \in S$, the length of the shortest trace to produce s in $\mathbf{C}$ and in $\mathbf{C}^{(x)}$ is the same. A system is *copy-tolerant* if it is *x-copy-tolerant* for all x.

With that definition, we have the following corollary based on the fact that if a 1-proper chemical reaction system is $(B_{\mathbf{C}} + 1)$-copy-tolerant, then s_{end} can be computed in $\mathbf{C}$ in the same number of steps as in $\mathbf{C}^{(B_{\mathbf{C}}+1)}$, which is polynomial in $B_{\mathbf{C}}$ by theorem 3.2.

Corollary 3.5. *For any 1-proper chemical reaction system* $\mathbf{C} = \langle S, \mathcal{R}, S_0, s_{\text{end}} \rangle$ *that is* $(B_{\mathbf{C}} + 1)$*-copy-tolerant, if there is a computation that produces a given signal molecule* s_{end} *in* $\mathbf{C}$ *then there is a computation that produces* s_{end} *in* $\mathbf{C}$ *in* $O((B_{\mathbf{C}})^2)$ *steps.*

Informally, this implies that any chemical reaction system that is robust in a multi-copy setting cannot have signal molecules whose production requires a computation length exponential in the size of the system.

4. CONCLUSIONS

In this paper, we have introduced the concept of recycling, or molecule reuse, in strand displacement systems and chemical reaction systems. Our n-bit GRAY counters effectively use recycling to step through 2^n states while consuming, or wasting, molecules whose total number of bases is $O(n^3)$. Our GRAY counter strand displacement constructions also introduce the use of a *mutex* strand to ensure that higher level chemical reactions are executed atomically. Finally, we show limits to recycling: for example, signal molecules representing the final state of our n-bit counter can be generated using just $O(n^2)$ reactions when $\Theta(n)$ copies of the initial signal molecules share the same volume.

One weakness of our counter construction is that the number of distinct domains needed is polynomial in n, the number of bits of the counter. In contrast, a QSW binary counter that is implemented via the stack machine of Qian *et al.* [11] uses just a constant number of domains independent of n. Is it possible to construct an n-bit counter that combines the best of the GRAY and QSW counters, i.e. generates waste that is polynomial in n and uses $O(1)$ distinct domains? More generally, can *all* computation be realized by strand displacement systems whose waste and haste are within a (small) polynomial factor of the space and time of the computation? Our negative result raises the question as to whether there are alternative strand displacement realizations of certain chemical reaction system classes that generate little waste, say logarithmic in the computation length, and that also behave correctly in the multi-copy setting. We will investigate these questions in future work.

REFERENCES

1 Hongzhou, G., Chao, J., Xiao, S. J. & Seeman, N. C. 2010 A proximity-based programmable DNA nanoscale assembly line. *Nature* **465**, 202–205. (doi:10.1038/nature09026)

2 Lund, K. *et al.* 2010 Molecular robots guided by prescriptive landscapes. *Nature* **465**, 206–210. (doi:10.1038/nature09012)

3 Omabegho, T., Sha, R. & Seeman, N. C. 2009 A bipedal DNA Brownian motor with coordinated legs. *Science* **324**, 67–71. (doi:10.1126/science.1170336)

4 Qian, L. & Winfree, E. 2011 A simple DNA gate motif for synthesizing large-scale circuits. *J. R. Soc. Interface* **8**. (doi:10.1098/rsif.2010.0729)

5 Seelig, G., Soloveichik, D., Zhang, D. Y. & Winfree, E. 2006 Enzyme-free nucleic acid logic circuits. *Science* **314**, 1585–1588. (doi:10.1126/science.1132493)

6 Shin, J. S. & Pierce, N. A. 2004 A synthetic DNA walker for molecular transport. *J. Am. Chem. Soc.* **126**, 10 834–10 835. (doi:10.1021/ja047543j)

7 Soloveichik, D., Seelig, G. & Winfree, E. 2010 DNA as a universal substrate for chemical kinetics. *Proc. Natl Acad. Sci. USA* **107**, 5393–5398. (doi:10.1073/pnas.0909380107)

8 Venkataraman, S., Dirks, R. M., Rothemund, P. W. K., Winfree, E. & Pierce, N. A. 2007 An autonomous polymerization motor powered by DNA hybridization. *Nat. Nanotech.* **2**, 490–494. (doi:10.1038/nnano.2007.225)

9 Zhang, D. Y. & Seelig, G. 2011 Dynamic DNA nanotechnology using strand displacement reactions. *Nat. Chem.* **3**, 103–113. (doi:10.1038/nchem.957)

10 Savage, C. 1997 A survey of combinatorial Gray codes. *SIAM Rev.* **39**, 605–629. (doi:10.1137/S0036144595295272)

11 Qian, L., Soloveichik, D. & Winfree, E. 2011 Efficient Turing-universal computation with DNA polymers. In *Proc. Sixteenth Annual Conf. on DNA Computing and Molecular Programming, Hong Kong, 14–17 June 2010.* Lecture Notes in Computer Science, no. 6518, pp. 123–140. Berlin, Germany: Springer.

12 Cardelli, L. 2010 Two-domain DNA strand displacement. In *Proc. Developments in Computational Models (DCM 2010), Edinburgh, UK, 9–10 July 2010*, vol. 26, pp. 47–61. Electronic Proceedings in Theoretical Computer Science. See http://eptcs.org/content.cgi?DCM2010.

13 Cardelli, L. 2011 Strand algebras for DNA computing. *Nat. Comput.* **10**, 407–428. (doi:10.1007/s11047-010-9236-7)

14 Winfree, E. 1998 *Algorithmic self-assembly of DNA*. Pasadena, CA: Caltech.

15 Rothemund, P. W. K. & Winfree, E. 2000 The program-size complexity of self-assembled squares. In *Proc. Thirty-Second Annual ACM Symp. on Theory of Computing, Portland, OR, 21–23 May 2000*, pp. 459–468. New York, NY: Association for Computing Machinery.

16 Reif, J. H., Sahu, S. & Yin, P. 2006 Complexity of graph self-assembly in accretive systems and self-destructible systems. In *Proc. Eleventh Annual Conf. on DNA-Based Computing, London, Canada, 6–9 June 2005.* Lecture Notes in Computer Science, no. 3892, pp. 257–275. Berlin, Germany: Springer.

17 Doty, D., Kari, L. & Masson, B. 2011 Negative interactions in irreversible self-assembly. In *Proc. Sixteenth Annual Conf. on DNA Computing and Molecular Programming, Hong Kong, 14–17 June 2010.* Lecture Notes in Computer Science no. 6518, pp. 37–48. Berlin, Germany: Springer.

18 Kharam, A., Jiang, H., Riedel, M. & Parhi, K. 2011 Binary counting with chemical reactions. In *Proc. 2011 Pacific Symp. on Biocomputing, Hawaii, 4–8 January 2011.* pp. 302–313. Singapore: World Scientific Publishing.

19 Feller, W. 1971 *An introduction to probability theory and its applications*, vol. 1. New York, NY: Wiley.