Speed faults in computation by chemical reaction networks*

Ho-Lin Chen[1], Rachel Cummings[2], David Doty[2], and David Soloveichik[3]

[1] National Taiwan University, Taipei, Taiwan holinc@gmail.com
[2] California Institute of Technology, Pasadena, California, USA
rachelc@u.northwestern.edu,ddoty@caltech.edu
[3] University of California, San Francisco, San Francisco, CA, USA
david.soloveichik@ucsf.edu

Abstract. Chemical reaction networks (CRNs) formally model chemistry in a well-mixed solution. Assuming a fixed molecular population size and bimolecular reactions, CRNs are formally equivalent to population protocols, a model of distributed computing introduced by Angluin, Aspnes, Diamadi, Fischer, and Peralta (PODC 2004). The challenge of fast computation by CRNs (or population protocols) is to ensure that there is never a bottleneck "slow" reaction that requires two molecules (agent states) to react (communicate), both of which are present in low ($O(1)$) counts. It is known that CRNs can be fast in expectation by avoiding slow reactions with high probability. However, states may be reachable (with low probability) from which the correct answer may only be computed by executing a slow reaction. We deem such an event a *speed fault*. We show that the problems decidable by CRNs guaranteed to avoid speed faults are precisely the *detection problems*: Boolean combinations of questions of the form "is a certain species present or not?". This implies, for instance, that no speed fault free CRN could decide whether there are at least two molecules of a certain species, although a CRN could decide this in "fast" expected time — i.e. speed fault free CRNs "can't count."

1 Introduction

Understanding the principles of molecular computation is essential to making sense of information processing in biological cellular regulatory networks. Further, in engineering of life-like devices (e.g. "wet robots" that can patrol the blood for cancer cells) we are rapidly approaching the point where we are limited by conceptual understanding: How molecular networks can be programmed

* The third, and fourth authors were supported by the Molecular Programming Project under NSF grants 0832824 and 1317694, the first author was supposed by NSC grant number 101-2221-E-002-122-MY3, the second author was supported by NSF grants CCF-1049899 and CCF-1217770, the third author was supported by a Computing Innovation Fellowship under NSF grant 1019343, NSF grants CCF-1219274 and CCF-1162589, and the fourth author was supported by NIGMS Systems Biology Center grant P50 GM081879.

to process information and carry out computation subject to the natural constraints of aqueous chemistry is still not well-understood.

A foundational model of chemistry commonly used in natural sciences is that of chemical reaction networks (CRNs), i.e., (finite) sets of chemical reactions such as $A + B \rightarrow A + C$. Subject to discrete semantics (integer number of molecules) the model corresponds to a continuous time, discrete state, Markov process [12]. A state of the system is a vector of non-negative integers specifying the molecular counts of the species (e.g., A, B, C), a reaction can occur only when all its reactants are present, and transitions between states correspond to reactions (i.e., when the above reaction occurs the count of B is decreased by 1 and the count of C increased by 1). The transition rate is proportional to the product of the counts of the reactants. CRNs are widely used to describe natural biochemical systems such as the intricate cellular regulatory networks responsible for the information processing within cells. With recent advances in synthetic biology, CRNs are a promising language for the design of artificial biochemical networks. For example, the physical primitive of nucleic-acid strand displacement cascades provides concrete chemical implementations of arbitrary CRNs [4, 8, 16]. Thus, since in principle any CRN can be built, hypothetical CRNs with interesting behaviors are becoming of more than theoretical interest.

The importance of the CRN model is underscored by the observation that intimately related models repeatedly arise in theoretical computer science under different guises: e.g. vector addition systems [13], Petri nets [14], population protocols [1]. The connection to distributed computing models, in turn, resulted in novel insights regarding natural cellular regulatory networks [5].

Parallelism is a basic attribute of chemistry, and one that is of central importance in understanding molecular information processing. This kind of parallelism is both a blessing and a curse: it can be used to speed up computation, but we must be careful to avoid "race conditions" (reactions happening in an unintended order) which may lead to error.

Consider a very basic task: a chemical system (e.g. cell) responding to molecular signals present in very small quantities. More specifically, say the computation is to produce at least one molecule of Y if and only if there is at least one molecule of species A_1 and at least one molecule of species A_2. Consider the strategy shown in Fig 1(b). Intuitively, this corresponds to having receptors F that in order to activate need to bind both A_1 and A_2. By having n receptors F we can increase the rate of the first reaction, but if there is only one molecule of A_1, there will be at most one molecule of F' and thus the second reaction occurs at a rate independent of the amount of receptor. Thus this scheme is "not parallelizeable".[4]

[4] Bimolecular reaction rates scale inversely with the total volume, and it is impossible to fit arbitrarily many molecules in a fixed volume. While for large enough molecular counts we will run into this finite density constraint, we study the scaling of speed with molecular count before that point is reached. An alternate perspective is that our task is to compute as quickly as possible in volume sufficient to allow molecules of F to fill the volume with constant density [15].

A better strategy is to amplify the signal before taking the conjunction: e.g. Fig 1(c). Here the receptors release A_1 back upon interacting with it, and a single A_1 can interact with many receptors (converting them from F to F'). Intuitively, the more receptors F we have, the faster we'll get a large number of F''s, and the faster the Y will get produced via the second reaction. More specifically, observe that starting with $n > 0$ molecules of F, and a molecule of A_1 and A_2 each, the reachable states without Y are: for $0 \leq m \leq n$, $((n - m)\ F,\ m\ F',\ 1\ A_1,\ 1\ A_2)$. From any reachable state without Y, we can reach Y through a sequence of reaction executions where one of the reactants is present in at least $\lfloor n^{1/2} \rfloor$ count,[5] and under stochastic chemical kinetics, the expected time to traverse this path is $O(1/n^{1/2})$ — decreasing with n.[6] Scheme Fig 1(d) is even faster: it can be shown that from any reachable state, the expected time to produce Y scales as $O(\log n/n)$.

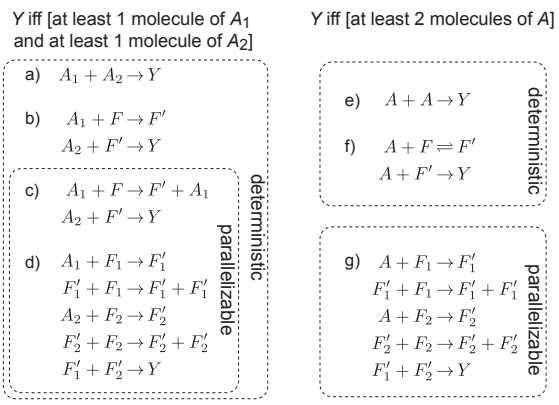

Fig. 1. Two molecular computation tasks: predicates "Is there at least 1 molecule of A_1 and at least one molecule of A_2?" (left), and "Are there at least 2 molecules of A?" (right). CRNs (a)-(d) compute the first predicate (left), and CRNs (e)-(g) compute the second (right). Parameter n is the initial amount of F, or F_1 and F_2 species which help in the computation. Informally the parallelizeable CRNs are those that produce the output faster with increasing n. Deterministic CRNs are those that compute correctly no matter what order the reactions happen to occur in. Other strategies (not shown) involve producing Y but consuming it if the predicate is not satisfied.

Now consider a slightly different computational task: produce at least one molecule of Y if and only if there are at least 2 molecules of species A. The natural analog of Fig 1(b) fails to be deterministic: the reactions $A + F \to F'$,

[5] If $m < \lfloor n^{1/2} \rfloor$, execute the first reaction $\lfloor n^{1/2} \rfloor - m$ times (resulting in $\lfloor n^{1/2} \rfloor$ molecules of F'), and then execute the second reaction. If $m \geq \lfloor n^{1/2} \rfloor$, execute the second reaction.

[6] The rate of a bimolecular reaction is proportional to the product of the counts of the reactants. Thus the expected time from the state with $m < \lfloor n^{1/2} \rfloor$ molecules of F' to reach the state with $\lfloor n^{1/2} \rfloor$ molecules of F' is proportional to $\sum_{i=m}^{\lfloor n^{1/2} \rfloor} 1/(n-i) \leq n^{1/2} \cdot 1/(n - n^{1/2}) = O(1/n^{1/2})$. Finally the rate of the second reaction when there are $\lfloor n^{1/2} \rfloor$ molecules of F' is proportional to $n^{1/2}$ and thus the expected time for it to fire is $O(1/n^{1/2})$ for a total expected time of $O(1/n^{1/2})$. Note that power $n^{1/2}$ was chosen in the analysis to ensure the optimal tradeoff between the rates of individual reaction executions and the total number of reaction executions.

$A + F' \to Y$ suffer from a "race condition" where Y is never produced if both molecules of A happen to react with F. This can be fixed by having the receptor F bind A reversibly[7] as in Fig. 1(f). However, this scheme is not parallelizeable for the same reason as (b).

The natural analog of the parallelizeable reaction scheme Fig 1(c) will not solve this task correctly at all: With reactions $A+F \to F'+A$, $A+F' \to Y$, even a single molecule of A will always lead to a Y.

Also problematic is the scheme shown in Fig 1(g) based on (d). While it is parallelizeable, it also suffers from a race condition that can result in an error. If the two molecules of A happen to react with different receptor types (F_1 and F_2) then Y will be produced. However, if both A's react with the same receptor type, Y will never be produced.

Informally, our main result is that no CRN is deterministic and parallelizeable at the same time for the "2 A problem" (or any computation that involves counting, rather than simply detecting the presence or absence of input species). Thus deterministic and parallelizeable must be disjoint in Fig. 1(right). Unlike the examples above, we allow a broader range of schemes that could produce and consume Y repeatedly but eventually converge on the presence or absence of Y as the output. In order to define "parallelizeable" formally, we introduce the notion of a "speed fault". A speed fault occurs if a state is reached such that to stabilize to the correct output from that state requires using a bimolecular reaction with both reactants bounded independently of n. Thus "deterministically parallelizeable" corresponds to speed fault free. Our main result is that the problems decidable by speed fault free CRNs are precisely the *detection problems*: Boolean combinations of questions of the form "is a certain species present or not?". Thus speed fault free CRNs "can't count."

The current work stems from the desire to understand fast deterministic computation in CRNs and population protocols. While sophisticated chemical algorithms and protocols have been developed to compute a large class of functions quickly and without error (see next section), most constructions are not deterministically fast in the same strong sense as they are deterministic. Indeed, deterministic computation is a worst case notion that intuitively ensures correctness no matter what unlucky sequence of reactions occurs. However, fast computation is defined with respect to large probability reaction sequences. Our definition captures the natural worst case notion of speed.[8]

Our positive result shows how any detection problem can be decided by a speed fault free CRN, and further shows that this computation is fast in the standard stochastic chemical kinetics model [12]. The largest part of this paper

[7] A reversible reaction $A + F \rightleftharpoons F'$ is simply syntactic sugar for two irreversible reactions $A + F \to F'$ and $F' \to A + F$.

[8] We observe that in the literature on computation in CRNs and population protocols it is almost never the case that computation is slow because the necessary sequence of reactions is too long – rather, slowdown is dominated by reaction bottlenecks where two low count species must react. Thus in this work we focus on this essential type of delay, captured in our notion of speed faults.

concerns the negative result that only detection problems can be computed by speed fault free CRNs (Section 4.2). The proof of the negative result consists of finding a worst-case reaction sequence that leads to a speed fault, assuming a non-detection problem is computed.

Absent speed-faults, the $O(1)$-count species must initiate cascades through intermediary large count species in order to "communicate." Consider the above "$2A$ problem." We can imagine isolating the two copies of A in "separate test tubes" and then use the symmetry between the two A molecules to make the system think that it's communicating with just one A (and thereby fail to detect the second A). To make this argument precise we develop a pumping technique which formally distinguishes species that can get arbitrarily large with increasing n from species whose counts are bounded by a constant[9]. We show that all large count species that can be encountered along a trajectory can be pumped to be *simultaneously* large. We then show that in the context of large counts of all pumpable species, reaction sequences can be decomposed into separate test tubes (parallel decomposition). A key part of the argument involves showing that the speed fault free CRN cannot detect small changes to pumpable species; for this we develop a new technique for performing surgery on reaction sequences.

2 Previous work and future directions

Much related work in the distributed computing community is phrased in the language of population protocols rather than CRNs (e.g. [2]). While population protocols are equivalent to CRNs with exactly two reactants and two products, and thus a fixed population size, CRNs can naturally describe reactions that consume or produce net molecules. As a result CRNs can potentially explore an unbounded state space, and certain questions that are not natural for population protocols become germane for CRNs (for example: Turing universality). Because our negative result naturally applies to a changing population size, we phrase this paper in the language of CRNs.

CRNs have a surprisingly rich computational structure. If we allow the number of species and reactions to scale with the size of the input (i.e. we view CRNs as a non-uniform model of computation), then $\log s$ species can deterministically simulate space s-bounded Turing machines [6]. (These results are presented in a model called vector addition systems [13], but easily carry over.) Thus CRNs are a very powerful model of non-uniform computation. On the other hand, we ask what functions can be computed by a fixed CRN (i.e. fixed number of species and reactions, with input encoded in the initial molecular counts, which corresponds to a uniform model). In this setting, CRNs are not Turing universal, unless we allow for some probability of error [3, 15]. In attempting Turing universal computation, there will provably always be "race conditions" that lead to error if certain reactions occur in a (maybe unlikely but possible) malicious order. The

[9] Note that our pumping lemma is very different from a similarly called "pumping lemma" of ref. [2], which shows that how input can be increased without changing the output (thus pumping "input")

fact that even such Turing universal computation is possible, and indeed can be made "fast" is surprising since finite CRNs necessarily must represent binary data strings in a unary encoding, since they lack positional information to tell the difference between two molecules of the same species.

Deterministic computation of both predicates and functions has been exactly characterized, and corresponds to semilinear sets and functions [2, 7]. Angluin, Aspnes, and Eisenstat [2] showed that all semilinear predicates can be deterministically computed in expected $O(n \text{ polylog } n)$ "interactions" (molecules bumping into each other). In a volume of fixed size, with n molecules, there are an expected $\Theta(n^2)$ such interactions per unit time, which yields expected time $O((1/n)\text{polylog } n)$ — decreasing with n. Our results imply that when computing semilinear predicates other then the detection problems, it is always possible to reach a state (speed fault) from which the expected time to finish the computation is $\Omega(1)$ — independent of n. It is easy to reconcile the two results: in the construction of ref. [2], the probability that a speed fault is reached decreases with n, and thus the total expected time decreases with n as well. Our result implies that this is a necessary feature of any such construction, and is not simply due to insufficient cleverness of the researchers to avoid speed faults.

Other work showing the challenges in parallelizing CRNs include the investigation of running multiple copies of networks in parallel [9], and the inability of networks starting with only large count species to delay the production of any species [11].

While in this work we focused on parallelizable predicates, it remains to explore the class of parallelizable functions. For example, if the initial amount of A is the input and the final amount of B is the output, then we can think of the reaction $F + A \to 2B$ as deterministically computing $f(x) = 2x$. Clearly as the amount of F increases, the computation converges faster. On the other hand, we believe that computing division by 2 should not be possible without speed faults, although that remains to be shown.

Since the occurrence of a speed fault leads to a slow computational bottleneck, speed faults affect the tail bounds on the distribution of the computation time. Indeed, two CRNs may compute with the same fast expected time, but the one susceptible to speed faults will likely have a larger probability of taking significantly longer. It remains to rigorously draw out the connection between tail bounds and speed faults.

3 Preliminaries

3.1 Chemical reaction networks

If Λ is a finite set (in this paper, of chemical species), we write $\mathbb{N}^\Lambda$ to denote the set of functions $f : \Lambda \to \mathbb{N}$. Equivalently, we view an element $\mathbf{c} \in \mathbb{N}^\Lambda$ as a vector of $|\Lambda|$ nonnegative integers, with each coordinate "labeled" by an element of Λ. Given $S \in \Lambda$ and $\mathbf{c} \in \mathbb{N}^\Lambda$, we refer to $\mathbf{c}(S)$ as the *count of* S *in* $\mathbf{c}$. Let $|\mathbf{c}| = \|\mathbf{c}\|_\infty = \max_{S \in \Lambda} \mathbf{c}(S)$. We write $\mathbf{c} \leq \mathbf{c}'$ to denote that $\mathbf{c}(S) \leq \mathbf{c}'(S)$ for all $S \in \Lambda$,

and $\mathbf{c} < \mathbf{c}'$ if $\mathbf{c} \leq \mathbf{c}'$ and $\mathbf{c} \neq \mathbf{c}'$. Since we view vectors $\mathbf{c} \in \mathbb{N}^\Lambda$ equivalently as multisets of elements from Λ, if $\mathbf{c} \leq \mathbf{c}'$ we say $\mathbf{c}$ is a *subset* of $\mathbf{c}'$. Given $\mathbf{c}, \mathbf{c}' \in \mathbb{N}^\Lambda$, we define the vector component-wise operations of addition $\mathbf{c} + \mathbf{c}'$, subtraction $\mathbf{c} - \mathbf{c}'$, and scalar multiplication $n\mathbf{c}$ for $n \in \mathbb{N}$. For a set $\Delta \subset \Lambda$, we view a vector $\mathbf{c} \in \mathbb{N}^\Delta$ equivalently as a vector $\mathbf{c} \in \mathbb{N}^\Lambda$ by assuming $\mathbf{c}(S) = 0$ for all $S \in \Lambda \setminus \Delta$. Write $\mathbf{c} \upharpoonright \Delta$ to denote the vector $\mathbf{d} \in \mathbb{N}^\Delta$ such that $\mathbf{c}(S) = \mathbf{d}(S)$ for all $S \in \Delta$. Given $S_1, \ldots, S_k \in \Lambda$, $\mathbf{c} \in \mathbb{N}^\Lambda$, and $n_1, \ldots, n_k \in \mathbb{Z}$,we write $\mathbf{c} + \{n_1 S_1, \ldots, n_k S_k\}$ to denote vector addition of $\mathbf{c}$ with the vector $\mathbf{v} \in \mathbb{Z}^{\{S_1, \ldots, S_k\}}$ with $\mathbf{v}(S_i) = n_i$.

Given a finite set of chemical species Λ, a *reaction* over Λ is a triple $\alpha = \langle \mathbf{r}, \mathbf{p}, k \rangle \in \mathbb{N}^\Lambda \times \mathbb{N}^\Lambda \times \mathbb{R}^+$, specifying the stoichiometry (amount consumed/produced) of the reactants and products, respectively, and the *rate constant* k. A reaction is *unimolecular* if it has one reactant and *bimolecular* if it has two reactants. For simplicity, in this paper we use $k = 1$ and the rate constant is omitted. For instance, given $\Lambda = \{A, B, C\}$, the reaction $A + 2B \to A + 3C$ is the pair $\langle (1, 2, 0), (1, 0, 3) \rangle$. A *(finite) chemical reaction network (CRN)* is a pair $N = (\Lambda, R)$, where Λ is a finite set of chemical *species*, and R is a finite set of reactions over Λ. A *state* of a CRN $N = (\Lambda, R)$ is a vector $\mathbf{c} \in \mathbb{N}^\Lambda$.

Given a state $\mathbf{c}$ and reaction $\alpha = \langle \mathbf{r}, \mathbf{p} \rangle$, we say that α is *applicable* to $\mathbf{c}$ if $\mathbf{r} \leq \mathbf{c}$ (i.e., $\mathbf{c}$ contains enough of each of the reactants for the reaction to occur). If α is applicable to $\mathbf{c}$, then write $\alpha(\mathbf{c})$ to denote the state $\mathbf{c} + \mathbf{p} - \mathbf{r}$ (i.e., the state that results from applying reaction α to $\mathbf{c}$). A finite or infinite sequence of reactions (α_i), where each $\alpha_i \in R$, is a *reaction sequence*. Given an initial state $\mathbf{c}_0$ and a reaction sequence (α_i), the induced *execution sequence* (or *path*) q is a finite or infinite sequence of states $q = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \ldots)$ such that, for all $\mathbf{c}_i \in q$ $(i \geq 1)$, $\mathbf{c}_i = \alpha_i(\mathbf{c}_{i-1})$. If a finite execution sequence q starts with $\mathbf{c}$ and ends with $\mathbf{c}'$, we write $\mathbf{c} \Longrightarrow_q \mathbf{c}'$. We write $\mathbf{c} \Longrightarrow \mathbf{c}'$ if such an execution sequence exists and we say that $\mathbf{c}'$ is *reachable* from $\mathbf{c}$. We often abuse terminology and refer to reaction sequences and execution sequences (paths) interchangeably.

We will find ourselves frequently dealing with infinite sequences of states. The following technical lemma elucidates certain convenient properties of any such sequence and will be used repeatedly.

Lemma 3.1 (Dickson's Lemma [10]). *The set of states $\mathbb{N}^k$ is well-quasi-ordered. In particular, every infinite sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots$ of states has an infinite nondecreasing subsequence $\mathbf{x}_{i_0} \leq \mathbf{x}_{i_1} \leq \ldots$, where $i_0 < i_1 < \ldots \in \mathbb{N}$, and every set $U \subseteq \mathbb{N}^k$ has a finite number of minimal elements.*

3.2 Stable decidability of predicates

We now review the definition of stable decidability of predicates introduced by Angluin, Aspnes, and Eisenstat [2]. Intuitively, some species "vote" for a YES/NO answer, and a CRN N is a stable decider if N is guaranteed to reach a consensus vote.

A *chemical reaction decider* (CRD) is a tuple $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon, \phi, \mathbf{s})$, where (Λ, R) is a CRN, $\Sigma \subseteq \Lambda$ is the *set of input species*, $\Upsilon \subseteq \Lambda$ is the set of *voters*, $\phi : \Upsilon \to \{\text{NO}, \text{YES}\}$ is the *(Boolean) output function*, and $\mathbf{s} \in \mathbb{N}^{\Lambda \setminus \Sigma}$ is the *initial*

context. For the input vector $(n_1, \ldots, n_k) \in \mathbb{N}^k$, where $k = |\Sigma|$, we write the initial state as $\mathbf{i}(n_1, \ldots, n_k) \in \mathbb{N}^\Lambda$ defined by: $\mathbf{i}(n_1, \ldots, n_k) \upharpoonright \Sigma = (n_1, \ldots, n_k)$ and $\mathbf{i}(n_1, \ldots, n_k) \upharpoonright (\Lambda \setminus \Sigma) = \mathbf{s}$. We extend ϕ to a partial function on states $\Psi : \mathbb{N}^\Lambda \dashrightarrow \{\text{NO}, \text{YES}\}$ as follows. $\Psi(\mathbf{c})$ is undefined if either $\mathbf{c}(X) = 0$ for all $X \in \Upsilon$, or if there exist $X_0, X_1 \in \Upsilon$ such that $\mathbf{c}(X_0) > 0$, $\mathbf{c}(X_1) > 0$, $\phi(X_0) = NO$ and $\phi(X_1) = YES$. Otherwise, there exists $b \in \{NO, YES\}$ such that $(\forall X \in \Upsilon)(\mathbf{c}(X) > 0$ implies $\phi(X) = b)$; in this case, the *output* $\Psi(\mathbf{c})$ of state $\mathbf{c}$ is b.

A state $\mathbf{o}$ is *output stable* if $\Psi(\mathbf{o})$ is defined and, for all $\mathbf{c}$ such that $\mathbf{o} \Longrightarrow \mathbf{c}$, $\Psi(\mathbf{c}) = \Psi(\mathbf{o})$. We call a whole CRD $\mathcal{D}$ *stable* if, for any initial state $\mathbf{i}$, there exists $b \in \{NO, YES\}$ such that, for every state $\mathbf{x}$ reachable from $\mathbf{i}$, there is an output stable state $\mathbf{o}$ reachable from $\mathbf{x}$ such that $\Psi(\mathbf{o}) = b$. If $\mathcal{D}$ is stable, then for some unique subset $S_0 \subseteq \mathbb{N}^k$ of inputs it always converges to output 0 and stays with that output, and for the remainder $S_1 = \mathbb{N}^k \setminus S_0$ it always converges to output 1 and stays with that output. We say that $\mathcal{D}$ *stably decides* the set S_1, or that $\mathcal{D}$ *stably decides* the predicate $\psi : \mathbb{N}^k \to \{0, 1\}$ defined by $\psi(\mathbf{x}) = 1$ iff $\mathbf{x} \in S_1$.

A set $A \subseteq \mathbb{N}^k$ is *linear* if $A = \{\ \mathbf{b} + \sum_{i=1}^{p} n_i \mathbf{u}_i \mid n_1, \ldots, n_p \in \mathbb{N}\ \}$ for some constant vectors $\mathbf{b}, \mathbf{u}_1, \ldots, \mathbf{u}_p \in \mathbb{N}^k$. A is *semilinear* if it is a finite union of linear sets. The following theorem is due to Angluin, Aspnes, and Eisenstat [2]:

Theorem 3.2 ([2]). *A set $A \subseteq \mathbb{N}^k$ is stably decidable by a CRD if and only if it is semilinear.*

If a YES voter (or any other species, for that matter) cannot be produced by any sequence of reactions from a state $\mathbf{y}$, then it cannot be produced from any subset $\mathbf{y}' \leq \mathbf{y}$. The following lemma is useful when we want to argue the other way: that for certain species, beyond a certain value, *increasing* their counts cannot affect the ability or inability of the state to produce a YES voter. We say that a state $\mathbf{c}$ is *committed* if, for all states $\mathbf{z}$ such that $\mathbf{c} \Longrightarrow \mathbf{z}$, $\mathbf{z}(S) = 0$ for all YES-voting species S. In particular, all output-stable NO states are committed, and for stable CRDs, committed states are reachable only from inputs on which the predicate is false.[10]

Lemma 3.3. *For each CRD, there is a constant c such that, for all committed states $\mathbf{c}$, if $\mathbf{c}(S) > c$ for some $S \in \Lambda$, then for all $n \in \mathbb{Z}$, $\mathbf{c} + \{nS\}$ is also committed.*

4 Speed fault free CRDs

In this section we show our main result that speed fault free CRDs decide only "detection problems," i.e., detecting the presence or absence of a species, but

[10] A committed state is not be output-stable NO if a state without any voters is reachable from it. The distinct notion of "committed" is useful because (unlike for output NO stability) the negation of committed is closed under superset (see the proof of Lemma 3.3), yet (like for output NO stability) reaching a committed state implies that the predicate value must be false.

not distinguishing between two different positive counts of it. To allow for "parallelization" of the computation, we introduce a "fuel" species F, whose count is allowed to start arbitrarily large.[11] Increasing the amount of fuel species is analogous to increasing the amount of "receptor" in the introduction. We then formalize the concept of "speed fault free" discussed informally in the introduction. Briefly, a CRN experiences a speed fault if it reaches a state from which all paths to a correct state execute some reaction when the counts of all of its reactants are bounded by a constant (a "slow" reaction). Note that in the stochastic model, the expected time for such a reaction to occur is bounded below by a constant (independent of the amount of fuel).

Let $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon, \phi, \mathbf{s})$ be a stable CRD, where $\Sigma = \{A_1, \ldots, A_k\}$ are the input species and $\Lambda \setminus \Sigma$ contains a special "fuel" species F, with variable initial count n. The initial count of every other species in $\Lambda \setminus (\Sigma \cup \{F\})$ is $\mathbf{s}$ (unchanging with respect to n). Write the initial state of $\mathcal{D}$ with some number n_i of each input A_i and n molecules of F as $\mathbf{i}_n(n_1, \ldots, n_k)$.

Let $f \in \mathbb{N}$, let $\alpha \in R$ be a reaction and $\mathbf{x} \in \mathbb{N}^\Lambda$ be a state. We say that α occuring in state $\mathbf{x}$ is *f-fast* if at least one reactant has count at least f in $\mathbf{x}$. An execution sequence is called *f-fast* if all reactions in it are f-fast. [12]

Definition 4.1. *A stable CRD $\mathcal{D}$ is* speed fault free *if for all $n_1, \ldots, n_k$ and all $f \in \mathbb{N}$, for all sufficiently large n, for any state $\mathbf{x}$ such that $\mathbf{i}_n(n_1, \ldots, n_k) \Longrightarrow \mathbf{x}$, there is an output stable state $\mathbf{y}$ (which has the correct answer with respect to $n_1, \ldots, n_k$ by the stability of $\mathcal{D}$) such that $\mathbf{x} \Longrightarrow \mathbf{y}$ by an f-fast execution sequence.*

Definition 4.2. *A set $S \subseteq \mathbb{N}^k$ is a* simple detection set *if there is a $1 \leq i \leq k$ such that $S = \left\{ (x_1, \ldots, x_k) \in \mathbb{N}^k \;\middle|\; x_i > 0 \right\}$. A set is a* detection set *if it is expressible as a combination of finite unions, intersections, and complements of simple detection sets.*

In other words, the predicate corresponding to a simple detection set S is a finite Boolean combination of questions of the form "is a certain species present?". The following theorem is the main result of this paper. We show each direction in two separate lemmas, Lemma 4.4 and Lemma 4.10.

Theorem 4.3. *The sets decidable by speed fault free CRDs are precisely the detection sets.*

[11] Allowing multiple fuel species $F_1, F_2, \ldots$ does affect our results since one of our reactions can be $F \to F_1 + F_2 \ldots$.

[12] It is worth noting that fast reaction sequences are not necessarily fast in the standard sense of stochastic kinetics, since although each reaction occurs quickly, it could be that there are a huge number of reactions in the sequence. Since our main result is a lower bound, this does not hurt the argument (and our upper bound result also shows that it is possible to decide detection problems quickly under the standard stochastic model).

4.1 Detection problems are decidable by speed fault free CRDs

This is the easier direction of Theorem 4.3. We give the intuition behind the proof here, and we do not formally define the model of stochastic chemical kinetics used to prove the expected running time. See the full version of this paper for detailed definitions and the proof.

Lemma 4.4. *Every detection set is decidable by a speed fault free CRD. This CRD takes expected time $O(\log n/n)$ expected time to stabilize under the standard model of stochastic chemical kinetics with constant volume.*

Proof (sketch). To detect whether a species A is present or not, we may use "epidemic" reactions $A + F \to F_a$ and $F_a + F \to 2F_a$, where F votes NO and F_a votes YES. That is, if A encounters an F, then F changes state to F_a, and this information is "broadcast" throughout the population of F's. Since the sum $\mathbf{c}(F_a) + \mathbf{c}(F) = n$ is constant in any reachable state $\mathbf{c}$, the second bimolecular reaction always has a reactant with count $\geq n/2$ (hence that reaction is always $\frac{n}{2}$-fast), and the output-stable YES state is reached when all F's are converted to F_a. The extension to k input species just means that each F must store k bits, one for each input species. □

4.2 Speed fault free CRDs decide only detection problems

Before proceeding to the main argument, we need to develop some technical machinery. We first show that if a fast execution sequence is used to decrease the count of some species, then we can identify certain reactions that must necessarily occur (reaction extraction). We then develop a notion of pumping, which is used to identify species that can get arbitrarily large with increasing fuel. Finally, we show that reaction sequences in which one reactant is always pumpable can be decomposed into separate "test-tubes" (parallel decomposition). Finally we stitch these notions together to show that speed fault free CRDs cannot compute more than detection problems.

Reaction extraction lemma Intuitively, the lemma below states that a fast reaction sequence that decreases certain species from high counts to low counts must contain reactions of a certain restricted form. These reactions will later be used to do "surgery" on fast reaction sequences, because they give a way to alter the count of certain species, by inserting or removing those reactions, while carefully controlling the effect these insertions and removals have on counts of other species.

Lemma 4.5. *Let $c_1, c_2 \in \mathbb{N}$ such that $c_2 > |\Lambda| \cdot c_1$, let $\mathbf{x}, \mathbf{y} \in \mathbb{N}^\Lambda$ such that $\mathbf{x} \Longrightarrow \mathbf{y}$ via c_2-fast reaction sequence q. Let $\Delta = \{D \in \Lambda | \mathbf{x}(D) \geq c_2, \mathbf{y}(D) \leq c_1\}$. Then there is an order on Δ, so that we may write $\Delta = \{D_1, D_2, \ldots, D_l\}$, such that, for all $i \in \{1, \ldots, l\}$, there is a reaction α_i of the form $D_i \to P_1 + \ldots + P_k$ or $D_i + S \to P_1 + \ldots + P_k$, such that $S, P_1, \ldots, P_k \notin \{D_1, \ldots, D_i\}$, and α_i occurs at least $\frac{c_2 - |\Lambda| \cdot c_1}{|R|}$ times in q in states $\mathbf{c}$ in which $\mathbf{c}(S) \geq c_2$.*

Lemma 4.5 is formally proved in the full version of this paper. Intuitively, to see such an ordering exists, it helps to think in reverse, first defining the last element D_l of the ordering. Consider the potential function $\Phi(\mathbf{c}) = \sum_{D \in \Delta} \mathbf{c}(D)$; then $\Phi(\mathbf{x})$ is large (at least $|\Delta| \cdot c_2$) and $\Phi(\mathbf{y})$ is small (at most $|\Delta| \cdot c_1$). On the path from $\mathbf{x}$ to $\mathbf{y}$, when Φ is between c_2 and $|\Delta| \cdot c_1$, it cannot get smaller by reactions of the form $D_i + D_j \to \ldots$, since $D_i, D_j \in \Delta$, or that reaction would not be c_2-fast. Therefore to get Φ down requires reactions with at most one reactant in Δ. Furthermore, if any product were in Δ, this would not decrease the value of Φ, hence some reaction must be of the desired form: consuming exactly one element of Δ. This element is D_l, the last in the ordering. Inductively defining an ordering on $\Delta \setminus \{D_l\}$ gives the entire ordering.

Pumpable sets of species This section defines *pumpable* sets of species: species whose counts can be made arbitrarily large by increasing the amount of fuel (species F, see Definition 4.1) and proves some basic properties about them. For example, the fuel species F is trivially pumpable. If there is a reaction $F + A \to F' + A$, then F' is pumpable (if there is an A), because F can be arbitrarily large. To get a handle on the notion of speed fault free, we define pumping to enforce a certain kind of self-consistency (Π-friendly): you can pump without requiring any reactions where all reactants are not pumpable.

Let $\Pi \subseteq \Lambda$. If a reaction has at least one reactant in Π, say the reaction is *Π-friendly*. If $\mathbf{x} \Longrightarrow \mathbf{y}$ via a reaction sequence in which all reactions are Π-friendly, then we write $\mathbf{x} \Longrightarrow^{\Pi} \mathbf{y}$. Let $Z = (\mathbf{z}_1 \leq \mathbf{z}_2 \leq \mathbf{z}_3 \ldots)$, where each $\mathbf{z}_n \in \mathbb{N}^{\Lambda}$, be an infinite nondecreasing sequence of states. A set of species $\Pi \subseteq \Lambda$ is *Z-pumpable* if there exists a sequence of states $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots)$ such that: (1) for all $P \in \Pi$ and $m \in \mathbb{N}$, $\mathbf{x}_m(P) \geq m$, and (2) for all $m \in \mathbb{N}$, there exists $n \in \mathbb{N}$ such that $\mathbf{z}_n \Longrightarrow^{\Pi} \mathbf{x}_m$.[13] Call such a sequence $(\mathbf{x}_m)$ a *pumping sequence* for Π. Π is *maximal Z-pumpable* if it is Z-pumpable and no strict superset of Π is Z-pumpable.

The next proposition shows that after pumping a maximal Π, all other species have bounded counts in all states reachable by Π-friendly paths. It is proven in the full version of this paper. Intuitively, it holds because if any other species $S \notin \Pi$ could get large via some reaction sequence r, then we could make the species in Π so large that we are able to hold some in reserve, then execute r, and then we would have S and all of Π large at the same time, contradicting the maximality of Π. We will use Proposition 4.6 repeatedly, but its most important consequence, intuitively, is that that the only way to get something outside of Π "large" is by executing a "slow" reaction (between two reactants not in Π).

Proposition 4.6. *Let $Z = (\mathbf{z}_1 \leq \mathbf{z}_2 \leq \ldots)$ be a infinite nondecreasing sequence of states, and let $\Pi \subseteq \Lambda$ be maximal Z-pumpable, with pumping sequence $(\mathbf{x}_m)$. Then there is a constant c such that, for all states $\mathbf{y}$ and $m, n \in \mathbb{N}$ such that $\mathbf{x}_m \Longrightarrow^{\Pi} \mathbf{y}$, then for all $S \in \Lambda \setminus \Pi$, $\mathbf{y}(S) < c$.*

[13] We can assume that $n \to \infty$ as $m \to \infty$. This is because $(\mathbf{z}_n)$ is a nondecreasing sequence, and so if $\mathbf{z}_n \Longrightarrow^{\Pi} \mathbf{x}_m$ for some $n, m \in \mathbb{N}$, then for all $n' > n$, there is a superset $\mathbf{x}'_m \geq \mathbf{x}_m$ such that $\mathbf{z}_{n'} \Longrightarrow^{\Pi} \mathbf{x}'_m$, and $\mathbf{x}'_m(S) \geq m$ for all $S \in \Pi$.

Parallel decomposition Intuitively, the following lemma shows that systems reacting by Π-friendly reactions can be effectively decomposed into separate non-interacting "test tubes" (in the context of a large excess of Π).[14] The following lemma is proved in the full version of this paper.

Lemma 4.7. *Suppose* $\mathbf{x}+\mathbf{y} \Longrightarrow^{\Pi} \mathbf{z}$. *Then there are* $\mathbf{p}, \mathbf{p}', \mathbf{p}'' \in \mathbb{N}^{\Pi}$, *and* $\mathbf{z}', \mathbf{z}'' \in \mathbb{N}^{\Lambda}$ *such that* $\mathbf{p}+\mathbf{x} \Longrightarrow^{\Pi} \mathbf{p}'+\mathbf{z}'$ *and* $\mathbf{p}+\mathbf{y} \Longrightarrow^{\Pi} \mathbf{p}''+\mathbf{z}''$, *where* $\mathbf{z}'+\mathbf{z}''=\mathbf{z}$ *and* $\mathbf{p}'+\mathbf{p}''=2\mathbf{p}$.

Main proof Throughout this section, let $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon, \phi, \mathbf{s})$ be an arbitrary speed fault free CRD with $\Sigma = \{A_1, \ldots, A_k\}$ and fuel species F as in Definition 4.1. Supposing for the sake of contradiction that $\mathcal{D}$ decides some non-detection set, then there must exist some species A_i (assume without loss of generality that $i = 1$), and an input value $(n_1, n_2, \ldots, n_k) \in \mathbb{N}^k$, where $n_1 \geq 1$, with answer NO (without loss of generality) but input value $(n_1+1, n_2, \ldots, n_k)$ with answer YES. Let $\mathbf{i}_n$ be the above initial state with n_1 molecules of A_1, having n fuel molecules. We will show that for sufficiently large n, $\mathbf{i}_n + \{A_1\}$ is able to reach a state without YES-voting species, from which the only way to produce a YES voter is to execute a slow bimolecular reaction.

We now define two infinite sequences of states $(\mathbf{x}_m)$ and $(\mathbf{y}_m)$ used in the rest of the argument. Intuitively $(\mathbf{x}_m)$ makes "large" all species than can get large from $(\mathbf{i}_n)$, while $(\mathbf{y}_m)$ is a sequence of committed states reachable from $(\mathbf{x}_m)$ (but they have to be defined in a rather exacting way.) Let sequence $I = (\mathbf{i}_n)$ and let $\Pi \subseteq \Lambda$ be maximal I-pumpable with pumping sequence $(\mathbf{x}_m)$. In the full version of the paper we show that there is a $\mathbf{d} \in \mathbb{N}^{\Pi}$ such that $\mathbf{x}_m = \mathbf{x}_{m-1} + \mathbf{d}$. Define the sequence of output-stable NO states $(\mathbf{y}_m)$ inductively as follows. For the base case, let $\mathbf{y}_1$ be any output-stable NO state such that $\mathbf{x}_1 \Longrightarrow_{r_1} \mathbf{y}_1$; such a path r_1 must exist because $\mathcal{D}$ is stable. Inductively assume that $\mathbf{x}_{m-1} \Longrightarrow_{r_{m-1}} \mathbf{y}_{m-1}$. Then $\mathbf{x}_m = \mathbf{x}_{m-1} + \mathbf{d} \Longrightarrow_{r_{m-1}} \mathbf{y}_{m-1} + \mathbf{d}$. Let $f_m \in \mathbb{N}$ be the largest number such that there is a f_m-fast path p_m from $\mathbf{y}_{m-1} + \mathbf{d}$ to an output-stable NO state $\mathbf{y}_m$. Then let r_m be r_{m-1} followed by p_m.[15] By Proposition 4.6, once f is sufficiently large, any f-fast reaction sequence from $\mathbf{x}_m$ to $\mathbf{y}_m$ must be Π-friendly. Thus by reindexing $(\mathbf{x}_m)$ to start with a sufficiently large member of the sequence, we have that for all m, $\mathbf{x}_m \Longrightarrow^{\Pi} \mathbf{y}_m$.

By Dickson's Lemma there is an infinite nondecreasing subsequence $Y = (\mathbf{y}_{s_1}, \mathbf{y}_{s_2}, \ldots)$. Let $\Gamma = \{\ S \in \Lambda \mid \lim_{n\to\infty} \mathbf{y}_{s_n}(S) = \infty\ \}$. By Proposition 4.6, $\Gamma \subseteq \Pi$ since $\mathbf{x}_{s_n} \Longrightarrow^{\Pi} \mathbf{y}_{s_n}$. Let $\Delta = \Pi \setminus \Gamma$. These are the species that are "large" in $(\mathbf{x}_m)$ but are bounded in Y. By further taking appropriate subsequences, we can ensure that each $\mathbf{y}_{s_n}(S) = \mathbf{y}_{s_{n+1}}(S)$ if $S \in \Lambda \setminus \Gamma$ and $\mathbf{y}_{s_n}(S) < \mathbf{y}_{s_{n+1}}(S)$ if $S \in \Gamma$.

[14] Note that in this way Π-friendly bimolecular reactions act somewhat analogously to unimolecular reactions: if $\mathbf{x}+\mathbf{y} \Longrightarrow \mathbf{z}$ by a sequence of unimolecular reactions, then $\mathbf{x} \Longrightarrow \mathbf{z}'$ and $\mathbf{y} \Longrightarrow \mathbf{z}''$ such that $\mathbf{z}'+\mathbf{z}''=\mathbf{z}$.

[15] By the definition of speed fault free, $\lim_{m\to\infty} f_m = \infty$, since $\mathbf{x}_m$ and $\mathbf{y}_m$ for increasing m are reachable from input states $\mathbf{i}_n$ with increasing amounts of fuel.

Recall that a state is *committed* if it cannot produce a YES voter. The next lemma, formally proved in the full version of the paper, shows that changing counts of pumpable species (Π) by a "small" amount in $\mathbf{x}_m$, so long as m is sufficiently large, cannot change the ability of $\mathbf{x}_m$ to reach a committed state. Intuitively, later on $\mathbf{e}$ will represent a change in counts due to "processing" the extra copy of A_1 (the one that changes the correct answer in state $\mathbf{i}_n(n_1, \ldots, n_k)$ from NO to YES), and the following lemma will help us to derive a contradiction because the extra copy of A_1 should enable the production of a YES voter.

Lemma 4.8. *Let sequences $(\mathbf{x}_m)$ and $(\mathbf{y}_m)$ be as defined above. For all $\epsilon \in \mathbb{N}$, there exists $\epsilon' \in \mathbb{N}$ such that the following holds. For all $\mathbf{e} \in \mathbb{Z}^\Pi$ with $|\mathbf{e}| \leq \epsilon$, for infinitely many m, there exists $\mathbf{e}_m \in \mathbb{Z}^\Gamma$ with $|\mathbf{e}_m| \leq \epsilon'$, and $m_2 < m$ such that $\mathbf{x}_m + \mathbf{e} \Longrightarrow^\Pi \mathbf{y}_{m_2} + \mathbf{e}_m$ and $\mathbf{y}_{m_2} + \mathbf{e}_m$ is committed.*

Proof (sketch). We know that $\mathbf{x}_m \Longrightarrow_{r_m} \mathbf{y}_m$. Consider applying r_m to $\mathbf{x}_m + \mathbf{e}$ to get $\mathbf{y}_m + \mathbf{e}$. This may not work because it could drive some species negative, and the final state may not be committed. We use Lemma 4.5 to obtain an ordering $\Delta = \{D_1, \ldots, D_l\}$ such that we can add or remove from r_m reactions of the form $\alpha_i : D_i + S \to P_1 + \ldots + P_k$ where $S, P_1 \ldots, P_k$ are in $\Gamma \cup \{D_{i+1}, \ldots, D_l\}$. This gives a way to "fix" the count of D_i to make its count equal to its count in $\mathbf{y}_m$ by either removing α_i (to increase) or adding extra instances of α_i (to decrease), while affecting only species in Γ or "after" D_i (hence their counts will be fixed later). The counts of $D_1, \ldots, D_{i-1}$, which have already been fixed, are unaffected by the surgery to fix D_i, because they do not appear in α_i. When we are done, we have increased the "error" in species in Γ (corresponding to $\mathbf{e}_m \in \mathbb{Z}^\Gamma$ in the lemma statement), but by Lemma 3.3, $\mathbf{y}_m + \mathbf{e}_m$ is still committed. Unfortunately, we may be taking some species negative in the middle of the fixed path. To handle this, the full argument essentially relies on the definition of $\mathbf{y}_m$ iteratively defined by adding $\mathbf{d}$ to $\mathbf{y}_{m-1}$, and ends by reaching committed state $\mathbf{y}_{m_2} + \mathbf{e}_m$, for a smaller $m_2 < m$ (see full paper). □

The next lemma uses Lemma 4.8 to show that, from state $\mathbf{x}_m + \mathbf{e}$, with $\mathbf{e} \in \mathbb{Z}^\Lambda$ "small," we can reach a committed state in which every species that can be "large", is actually large.

Lemma 4.9. *Let sequence $(\mathbf{x}_m)$ be as defined above. For all $\epsilon \in \mathbb{N}$, there exists $c \in \mathbb{N}$ and $\Omega \subseteq \Lambda$ such that the following holds. For all $\mathbf{e} \in \mathbb{Z}^\Pi$ such that $|\mathbf{e}| \leq \epsilon$, there exists an infinite sequence $W_\mathbf{e} = (\mathbf{w}_n)$ of states such that, for all $n \in \mathbb{N}$, there exists $m_n \in \mathbb{N}$, such that the following is true: (1) $\mathbf{x}_{m_n} + \mathbf{e} \Longrightarrow^\Pi \mathbf{w}_n$, (2) $\mathbf{w}_n$ is committed, (3) for all $S \in \Omega$, $\mathbf{w}_n(S) \geq n$, (4) for all $S \in \Lambda \setminus \Omega$ and all $\mathbf{u}$ such that $\mathbf{w}_n \Longrightarrow^\Omega \mathbf{u}$, $\mathbf{u}(S) \leq c$, and (5) $\mathbf{w}_n$ are nondecreasing.*

Lemma 4.9 is proven in the full version of this paper. Intuitively (albeit imprecisely), it follows by letting Ω be a maximal Y-pumpable set of species, where Y is the infinite sequence of committed states of the form $\mathbf{y}_{m_2} + \mathbf{e}_m$ shown to exist in Lemma 4.8. That is, while Π contains species that can simultaneously get large in state $\mathbf{x}_m$ starting from the initial state, and Γ contains species that

happen to be large in the committed states $\mathbf{y}_{m_2} + \mathbf{e}_m$ reachable from $\mathbf{x}_m + \mathbf{e}$, Ω contains possibly more species than Γ: those that can get large, starting from states $\mathbf{y}_{m_2} + \mathbf{e}_m$.

The next lemma shows that speed fault free CRDs decide only detection problems. Lemma 4.10 is formally proved in the full version of this paper.

Lemma 4.10. *$\mathcal{D}$ is not speed fault free.*

Proof (sketch). Recall initial states $\mathbf{i}_n$ encode an input value making the predicate false, and $\mathbf{i}_n + \{A_1\}$ encode an input value making the predicate true. Let $\mathbf{e} = \mathbf{0}$ and consider the corresponding $W_{\mathbf{0}} = (\mathbf{w}_n)$. By Lemma 4.9 we have $\mathbf{i}_{n'} \Longrightarrow^{\Pi} \mathbf{x}_m \Longrightarrow^{\Pi} \mathbf{w}_n$. We can rewrite this path as $(\mathbf{i}_{n'} \setminus \{A_1\}) + \{A_1\} \Longrightarrow^{\Pi} \mathbf{w}_n$, and applying Lemma 4.7 obtain that there are $\mathbf{p}, \mathbf{p}' \in \mathbb{N}^{\Pi}$ such that: $\mathbf{p} + \{A_1\} \Longrightarrow^{\Pi} \mathbf{p}' + \mathbf{b}$, where $\mathbf{b} \leq \mathbf{w}_n$. Call this path r. Since $\mathbf{b} \leq \mathbf{w}_n$, it must be that $\mathbf{b}$ is committed even if any amount of Ω is added to it.

Let $\mathbf{e} = \mathbf{p}' - \mathbf{p} \in \mathbb{Z}^{\Pi}$ and consider the (different) sequence $W_{\mathbf{e}} = (\mathbf{w}_n)$ obtained using this $\mathbf{e}$ from Lemma 4.9. For all n, there is m such that $\mathbf{x}_m + \mathbf{e} \Longrightarrow^{\Pi} \mathbf{w}_n$ by some path p_n. Now, choose n large enough (so $\mathbf{x}_m \geq \mathbf{p}$) and add the extra molecule of A_1: $\mathbf{x}_m + \{A_1\} \Longrightarrow^{\Pi}_{r} \mathbf{x}_m + (\mathbf{p}' - \mathbf{p}) + \mathbf{b} = \mathbf{x}_m + \mathbf{e} + \mathbf{b} \Longrightarrow^{\Pi}_{p_n} \mathbf{w}_n + \mathbf{b}$. Because this state is reachable from a valid initial state with one extra molecule of A_1, we must be able to produce a YES voter from it. By assumption of a speed fault free CRD, this must be a fast path: for all f, there is an n such that $\mathbf{w}_n + \mathbf{b} \Longrightarrow \mathbf{z}_n$ by an f-fast path q_n, and $\mathbf{z}_n$ contains a YES voter. Is q_n Ω-friendly? If q_n is Ω-friendly then by Lemma 4.7 we can reach a YES voter solely from $\mathbf{w}_n$ or $\mathbf{b}$ given enough extra of species in Ω. This is a contradiction since both $\mathbf{w}_n$ and $\mathbf{b}$ are committed, even if any amount of Ω is added (by Lemma 3.3).

Thus q_n cannot be entirely Ω-friendly. Let α_n be the first reaction that is not Ω-friendly, and let $\mathbf{u}_n$ be the state immediately before this reaction occurs. If for all f, there is a q_n that is f-fast, it must be that $\mathbf{u}_n$ contains count f of some species X_n that is not in Ω (otherwise, α_n would be Ω-friendly). Consider $f > 2c$ where c is the constant from Lemma 4.9. Since the initial portion of q_n that leads to $\mathbf{u}_n$ is Ω-friendly, we have $\mathbf{w}_n + \mathbf{b} \Longrightarrow^{\Omega} \mathbf{u}_n$ and Lemma 4.7 applies. Consequently, $\exists \mathbf{o}, \mathbf{o}', \mathbf{o}'' \in \mathbb{N}^{\Omega}$ and $\mathbf{u}', \mathbf{u}'' \in \mathbb{N}^{\Lambda}$ such that $\mathbf{o} + \mathbf{w}_n \Longrightarrow^{\Omega} \mathbf{o}' + \mathbf{u}'_n$ and $\mathbf{o} + \mathbf{b} \Longrightarrow^{\Omega} \mathbf{o}'' + \mathbf{u}''_n$ and $\mathbf{u}'_n + \mathbf{u}''_n = \mathbf{u}_n$. Thus either $\mathbf{u}'_n$ or $\mathbf{u}''_n$ must contain at least $f/2$ of X_n. Since $\mathbf{w}_{n'}$ are nondecreasing and are larger than n' on Ω, for large enough n', $\mathbf{w}_{n'}$ from $W_{\mathbf{e}}$ exceeds $\mathbf{o} + \mathbf{w}_n$ and $\mathbf{w}_{n'}$ from $W_{\mathbf{0}}$ exceeds $\mathbf{o} + \mathbf{b}$. But then we obtain a contradiction of condition (4) in Lemma 4.9. □

Acknowledgements. We thank Damien Woods, Anne Condon, Chris Thachuk, Bonnie Kirkpatrick, Monir Hajiaghayi, and Ján Maňuch for useful discussions.

References

[1] Dana Angluin, James Aspnes, Zoë Diamadi, Michael Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18:235–253, 2006. Preliminary version appeared in PODC 2004.

[2] Dana Angluin, James Aspnes, and David Eisenstat. Stably computable predicates are semilinear. In *PODC 2006: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 292–299, New York, NY, USA, 2006. ACM Press.

[3] Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199, September 2008. Preliminary version appeared in DISC 2006.

[4] Luca Cardelli. Strand algebras for DNA computing. *Natural Computing*, 10(1):407–428, 2011.

[5] Luca Cardelli and Attila Csikász-Nagy. The cell cycle switch computes approximate majority. *Scientific Reports*, 2, 2012.

[6] E. Cardoza, Richard J. Lipton, and Albert R. Meyer. Exponential space complete problems for Petri nets and commutative semigroups (preliminary report). In *STOC 1976: Proceedings of the 8th annual ACM Symposium on Theory of Computing*, pages 50–54. ACM, 1976.

[7] Ho-Lin Chen, David Doty, and David Soloveichik. Deterministic function computation with chemical reaction networks. *Natural Computing*, 2013. to appear. Preliminary version appeared in DNA 2012.

[8] Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik, and Georg Seelig. Programmable chemical controllers made from DNA. *Nature Nanotechnology*, 8(10):755–762, 2013.

[9] Anne Condon, Alan Hu, Ján Maňuch, and Chris Thachuk. Less haste, less waste: On recycling and its limits in strand displacement systems. *Journal of the Royal Society Interface*, 2:512–521, 2012. Preliminary version appeared in DNA 2011.

[10] Leonard E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, October 1913.

[11] David Doty. Timing in chemical reaction networks. In *SODA 2014: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 772–784, January 2014.

[12] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[13] Richard M Karp and Raymond E Miller. Parallel program schemata. *Journal of Computer and system Sciences*, 3(2):147–195, 1969.

[14] Carl A Petri. Communication with automata. Technical report, DTIC Document, 1966.

[15] David Soloveichik, Matthew Cook, Erik Winfree, and Jehoshua Bruck. Computation with finite stochastic chemical reaction networks. *Natural Computing*, 7(4):615–633, 2008.

[16] David Soloveichik, Georg Seelig, and Erik Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393, 2010. Preliminary version appeared in DNA 2008.