Speed faults in computation by chemical reaction networks *

Ho-Lin Chen† Rachel Cummings‡ David Doty§ David Soloveichik¶

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 not rely on 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 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 predicates stably decidable by CRNs guaranteed to avoid speed faults are precisely the *detection predicates*: Boolean combinations of questions of the form "is a certain species present or not?". This implies, for instance, that no speed fault free CRN decides whether there are at least two molecules of a certain species — i.e., speed fault free CRNs "can't count."

1 Introduction

Background. Understanding the principles of molecular computation is essential to making sense of information processing in biological cellular regulatory networks. Further, we are rapidly approaching the limit of our conceptual understanding in engineering of artificial regulatory networks, whether to be inserted into biology to rewire behavior, or for completely synthetic life-like systems. The theory of computation has proven invaluable in realizing information processing in electronic systems, and much-studied algorithms underly the behavior of everything from the internet to video games. However, a deep understanding of the computational principles underlying much of chemical regulation is still lacking. How molecular networks can be programmed to process information and carry out computation subject to the natural constraints of aqueous chemistry is 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 $A + B \to A + C$. Subject to discrete semantics (integer number of molecules) the model corresponds to a continuous time,

*A preliminary version of this article appeared as [10]; the current version has been significantly revised for clarity, and includes several omitted proofs. The first, 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, and NSF grants CCF-1219274 and CCF-1162589, and the fourth author was supported by NIGMS Systems Biology Center grant P50 GM081879.

†National Taiwan University, Taipei, Taiwan, holinc@gmail.com

‡California Institute of Technology, Pasadena, CA, USA, rachelc@u.northwestern.edu.

§University of California – Davis, Davis, CA, USA, doty@ucdavis.edu

¶University of Texas, Austin, Austin, TX, USA, david.soloveichik@utexas.edu

discrete state, Markov process [20]. 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 [6, 12, 26]. 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 [22], Petri nets [24], population protocols [1]. The connection to distributed computing models, in turn, resulted in novel insights regarding natural cellular regulatory networks. For example the "approximate majority" population protocol [4] has been connected to a number of biological networks [7, 8].

Motivation: parallelism in chemical computation. Parallelism is a basic attribute of chemistry, and one that is of central importance in understanding molecular information processing. Intuitively, the more molecules are concentrated in a fixed volume, the more interactions per unit time can occur. 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) that may lead to error.

As motivation consider a few very basic tasks in which a chemical system (e.g., cell) responds to molecular signals present in very small quantities. Fig 1 contains a number of examples of chemical computation of a predicate over the initial molecular counts of the input species A (or $A_1, \ldots, A_k$ for multiple inputs). The truth value of the predicate is output by the species Y: if the predicate is true (YES) the system eventually reaches a state with Y permanently present, while if the predicate is false (NO) the system eventually reaches a state with Y permanently absent. In all cases in the figure, the initial state contains exactly the input molecules and n molecules of *fuel* species F.[1] Chemically the fuel species can be thought of as receptors that mediate the interactions. In order to ascertain whether the computation speeds up with greater "parallelism", we look at how the expected time of the computation scales with n.

Consider the CRN shown in Fig 1(b) for the predicate "there is at least 1 molecule of species A_1 and at least 1 molecule of species A_2". Intuitively, this strategy 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 G and thus the second reaction occurs at a rate independent of the amount of receptor. This "bottleneck" reaction makes this scheme not parallelizeable.

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 G). Intuitively, the more receptors F we have, the faster we'll get a large number of G's, and the faster Y will get produced via the second reaction. More specifically, observe that starting with $n > 0$ molecules of F, and one molecule of A_1 and A_2 each, the reachable

[1] Some CRNs in Figure 1 require n to exceed a positive constant lower bound for correctness: 1 in cases (b),(c),(f),(g), 2 in cases (d),(h), and 3 in case (i).

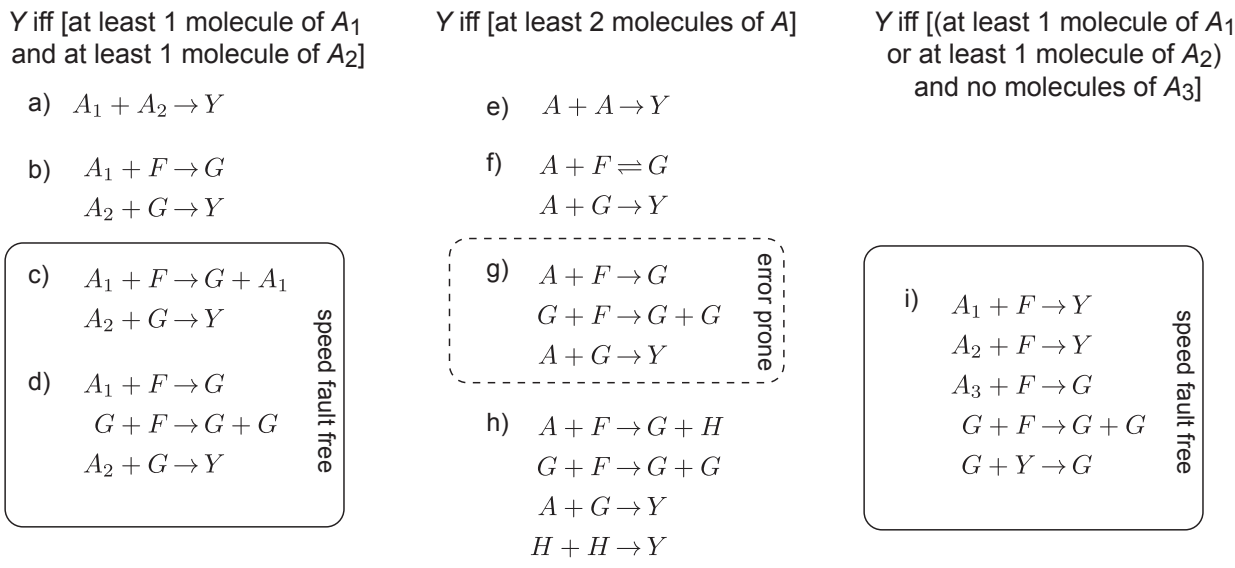

Figure 1: Motivating examples of computation and parallelism with chemical reaction networks. The CRNs in each column compute a different predicate, which is shown on top. The initial state contains exactly the input molecules and n molecules of fuel species F. The CRN outputs YES (respectively NO) when it reaches a state in which Y is present (resp. absent) and every reachable state has Y present (resp. absent). (Note that the formal definitions in this paper adhere to a different output convention that is symmetric with respect to NO and YES outputs; see Remark 3.1.) **a.** A_1 and A_2 directly interact. **b.** F acts as a receptor that needs to react with A_1 and A_2 to activate and produce Y. **c.** Same as (b) but receptors F release A_1 back upon interacting. **d.** Same as (b) but receptors F upon reacting with A_1 autocatalytically amplify the signal. **e.** The natural analog of (a) for the second predicate. **f.** An analog of (b) for the second predicate. Note that the first reaction is made reversible (a reversible reaction is simply syntactic sugar for two irreversible reactions) to avoid error if both A molecules react with F. **g.** The natural analog of (d) for the second predicate. However this CRN is error prone: if both A molecules react with F then Y will not be produced. **h.** We can eliminate the error possibility in (g) by adding species H that is a "backup copy" of A, and the fourth reaction that produces Y in case both A react with F. **i.** In the previous examples, once Y is produced it can never be consumed and thus the system has stabilized to an output. For "non-monotonic" predicates, such as that in the third column, it is necessary to consume Y as well as to produce it.

states without Y are: for $0 \leq m \leq n$, $((n-m)\ F,\ m\ G,\ 1\ A_1,\ 1\ A_2)$. From any reachable state without Y, we can reach a state with a Y through a sequence of reaction executions where one of the reactants is present in at least $\lfloor\sqrt{n}\rfloor$ count,[2] and under stochastic chemical kinetics, the expected time to produce a Y is $O(1/\sqrt{n})$ — decreasing with n.[3] Scheme Fig 1(d) is even faster: it can be shown that from any reachable state, the expected time to produce Y scales as $O(\log(n)/n)$.

[2]If $m < \lfloor\sqrt{n}\rfloor$, execute the first reaction $\lfloor\sqrt{n}\rfloor - m$ times (resulting in $\lfloor\sqrt{n}\rfloor$ molecules of G), and then execute the second reaction. If $m \geq \lfloor\sqrt{n}\rfloor$, execute the second reaction.

[3]Section 4.1 gives the formal model required to derive this expected time; here we briefly justify the claimed expected time. To simplify the analysis we assume the second reaction does not happen until the first reaction has produced at least $\sqrt{n}$ copies of G; otherwise the expected time is even lower than that derived below. The rate of a bimolecular reaction is proportional to the product of the counts of the reactants. After i instances of the first reaction have occurred, there are $n-i$ copies of F, so the expected time for the next occurrence of the first reaction is $\frac{1}{n-i}$, so by linearity of expectation the expected time from the state with $m < \lfloor\sqrt{n}\rfloor$ molecules of G to reach the state with $\lfloor\sqrt{n}\rfloor$ molecules of G is proportional to $\sum_{i=m}^{\lfloor\sqrt{n}\rfloor} 1/(n-i) \leq \sqrt{n} \cdot 1/(n-\sqrt{n}) = O(1/\sqrt{n})$. Finally the rate of the second reaction when there are $\lfloor\sqrt{n}\rfloor$ molecules of G is proportional to $\sqrt{n}$ and thus the expected time for it to fire is $O(1/\sqrt{n})$ for a total expected time of $O(1/\sqrt{n})$. The threshold $\sqrt{n}$ used in the analysis was chosen to ensure the optimal tradeoff between the rates of individual reaction executions and the total number of reaction executions.

Thus both (c) and (d) are parallelizeable.

Now consider a slightly different predicate: "there are at least 2 molecules of species A". Consider the CRN in Fig 1(g) based on (d). While it is parallelizeable in the sense that with increasing n is more quickly converges to *an* answer, the answer might be *wrong*. Indeed, it suffers from a race condition: if both A's react with F, Y will never be produced. We can fix this system by adding species H that "backup" A as shown in Fig. 1(h). Then if both A happen to react with F, the last reaction $H + H \to Y$ can still produce Y. Although the last reaction corrects the error, it acts as a bottleneck if both A happen to react with F.

When a CRN reaches a state from which output can never change again, we say that the CRN has *stabilized*; stabilization is a well-established notion of output for population protocols [2] and naturally extends to CRNs [11]. Note that systems (c), (d), and (i) have the property that the expected time to stabilize to the correct answer from *any reachable* state decreases with n. This property can be thought of as a certain kind of "self-stabilization" (note the different use of the word stabilization), in which a distributed system is expected to work as desired even after worst-case transient faults [5]. Is it possible to construct a CRN for the predicate "there are at least 2 molecules of species A" that has this property? (Our negative result shows that this is impossible.)

Speed faults. We formalize the notion of unavoidable bottleneck reactions by defining *speed faults*. A speed fault occurs if a state is reached such that to stabilize to the correct output from that state requires using a reaction where the counts of all reactants are bounded by a constant independent of n. Thus in Fig. 1(h), a speed fault occurs if both A molecules react with F in the first reaction and we have to rely on the last reaction.[4]

Note that the utility of the notion of speed faults is strongest for proving negative results. Although the occurrence of a speed fault in a particular execution sequence implies that it will take a long time to stabilize to the correct output, the absence of a speed fault does not imply fast stabilization. This is because an execution sequence might take a long time if it consists of a lot of reaction executions, whose number may increase with n.[5] Thus for our positive result we compute the expected time from any reachable state and show that it indeed decreases with n (Lemma 4.4).

We also emphasize that the possibility of a speed fault does not imply the CRN requires a large expected time *from the initial state*, because the speed fault may only be encountered with very low probability (that may decrease with n).

Our definition of speed fault free CRNs considers the paths to stabilization and not convergence. We say that an execution sequence *converged* to a particular output value at the time point when the output is produced and never changes again, although a different output might be reachable for a while longer (because the CRN has not yet stabilized). For example, consider the CRN in Fig. 1(i), and suppose that we start with 1 molecule of A_1, A_2 and A_3 each (and n fuel molecules F). Further suppose that the following (unlikely) sequence of events happens: Y is produced in the first reaction, G is produced in the third reaction, Y is consumed in the fifth reaction, and then the fourth reaction converts all of F to G before reaction 2 has a chance to occur. Note that in this case, the CRN quickly converged to the correct output (NO) and never changed the answer

[4] By "speed fault" we do not mean the event "a reaction between two low count species" (a slow reaction), but rather the event "enter a state from which a slow reaction is necessary to reach the correct output." Some CRNs experience the fault immediately, such as $A_1 + A_2 \to Y$, which suffers from "original sin": even from the initial state, a slow reaction is required to produce the output.

[5] 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.

again, but it takes many more reaction executions before the second reaction became impossible (all F was consumed) and the system stabilized.

Results. To discuss the generality of our results, we need to consider the input convention more carefully. In the examples above, the initial state contains only input species and F. In general starting with fixed amounts of other species as an *initial context* (for example a single leader molecule L) allows for a greater variety of CRNs.[6] Unless otherwise specified, we allow an arbitrary initial context. Our main result (Theorem 4.3) says that the predicates decidable by speed fault free CRNs are precisely the *detection predicates*: Boolean combinations of questions of the form "is a certain species present or not?". Thus speed fault free CRNs "can't count."

A simpler-to-prove version of the negative direction (Lemma 4.13) shows that speed fault free CRNs without an initial context (so-called *leaderless* CRNs [18]) cannot compute a predicate ψ unless it is closed under doubling: $\psi(\mathbf{x}) = \psi(2\mathbf{x})$ for all inputs $\mathbf{x}$. The "2A predicate" in Fig. 1 is not closed under doubling ($\psi(1) = 0$ but $\psi(2) = 1$), thus this result immediately implies that it is not computable by any speed fault free leaderless CRN. As Remark 4.3 indicates, the proof's conclusion is actually stronger than the negative direction of Theorem 4.3, because it excludes fast convergence, not just fast stabilization.

High level intuition for the negative results. Disallowing speed-faults, the $O(1)$-count species must initiate cascades through intermediary large count species in order to "communicate." Consider the above "$2A$ predicate." 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 that formally distinguishes species that can get arbitrarily large with increasing n from species whose counts are bounded by a constant.[7] 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 extending the argument to allow an initial context 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.

Finite density constraint. It is physically impossible to fit arbitrarily many molecules in a fixed physical volume. While for large enough molecular counts we will run into this finite density constraint [25], we study the scaling of speed with molecular count before that point is reached. A complementary perspective is that our task is to compute as quickly as possible in volume $\Omega(n)$ with $O(n)$ total molecules, where the n molecules of F represent the "other" molecules. The more of these other molecules there are, the slower our computation will be (since the volume scales with n), unless we involve F in the computation. In this perspective, a speed fault corresponds to reaching a state from which we require an $\Omega(n)$ time ("slow") reaction, while our positive result implies that all detection predicates can be computed in $O(\log n)$ time ("fast") from any reachable state. In the context of time complexity of population protocols [3], these respectively correspond to the notion of $\Omega(n)$ versus $O(\log n)$ "parallel time," or equivalently, $\Omega(n^2)$ versus $O(n \log n)$ expected pairwise interactions between agents (on the assumption that $\approx n$ interactions happen per "unit time").

[6] For example, the simplest CRN computing the predicate "there is an odd count of A" may be: $A + L \to Y$, $A + Y \to L$, which starts with 1 copy of L as the leader. (Of course, this CRN is not parallelizeable.)

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

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).

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., view CRNs as a non-uniform model of computation), then there is a certain sense in which $\log s$ species can deterministically simulate space s-bounded Turing machines, albeit the simulation is grossly inefficient [9]. These results are presented in a model called vector addition systems [22], but carry over. On the other hand, we can ask — as we do here — 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 (i.e., view CRNs as a uniform model of computation). In this setting, CRNs are not Turing universal, unless we allow for some probability of error [3, 25] (but see [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 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[8] has been exactly characterized, and corresponds to semilinear sets and functions [2, 11].[9] Angluin, Aspnes, and Eisenstat [2] define the formal notion of determinism that we use here ("stable" computation, see Section 3.3). The authors also showed that all semilinear predicates can be deterministically computed in expected $O(n \text{ polylog } n)$ "interactions" (molecules bumping into each other) from the initial state. 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 $O((1/n)\text{polylog } n)$ in our setting. Since semilinear predicates are a much larger class than detection predicates, their construction is necessarily susceptible to speed faults. Indeed, the overall expected time to complete the computation starting from the initial state decreases with n only because it becomes less and less *likely* that a speed fault occurs.

A number of fundamental questions related to the speed of deterministic computation in CRNs remain unanswered. The first concerns the gap between two key time points in stable computation. A stably computing CRN must eventually reach a state with the correct output such that no sequence of reactions can change the output (stabilization). However, as we've seen, there might be a delay between when the output changes for the last time (convergence), and states with a different output become unreachable (stabilization). Indeed, in the construction of ref. [2], there is a factor of n difference between the expected times to convergence and stabilization from the initial state. Our result implies that for all non-detection predicates, there is a reachable state from which the expected time to stabilize doesn't decrease with n. But we conjecture a stronger statement: *for all non-detection predicates, there is a reachable state such that the expected time to converge is bounded below by a constant independent of* n. Here, we prove a special case of this conjecture for leaderless CRNs stably computing predicates not closed under doubling (see Remark 4.3).

[8] In function computation, unlike predicate computation, the exact count of output species represents the output value.

[9] However, the computational power of predicates with asymmetric output remains to be shown, although it is likely to be semilinear as well.

We also conjecture that there isn't a way to make stabilization fast, even in expectation from the initial state: *for all non-detection predicates, the expected time to reach an output stable state from the initial state is bounded below by a constant independent of n.* This does not contradict the above-mentioned result [2] that all semilinear predicates can be deterministically computed in expected time $O((1/n)\text{polylog } n)$, since this was shown for convergence rather than stabilization.

In our definition of speed fault free CRNs, we have introduced an auxiliary "fuel" species, primarily to avoid conflating the questions "What is the input to the computation?" and "How many molecules are available to help parallelize the computation?" For each fixed input $\mathbf{x}$, the amount of fuel n is allowed to be arbitrarily large compared to $\|\mathbf{x}\|$, so the input molecules contribute negligibly to the parallelization. However, it is also natural to define speed fault free CRNs without the fuel species, so that the molecular count is potentially dominated by the size of the input. The resulting characterization may be more aligned with the expected time results from the literature on population protocols and CRNs [1,3,4,11,14,17–19,25].

While in this work we focus on parallelizeable predicates, it remains to explore the class of parallelizeable functions. For example, if the initial amount of A is the input and the final amount of Y is the output, then we can think of the reaction $F + A \to 2Y$ as deterministically computing $f(x) = 2x$. Clearly as the amount of F increases, the computation converges and stabilizes faster. On the other hand, we believe that computing the function $f(x) = \lfloor x/2 \rfloor$ is not possible without speed faults, although that remains to be shown. (It is computable by the reaction $A + A \to Y$, but this does not speed up with F.)

The current work stems from an effort to develop lower bounds for computation time for CRNs and population protocols. We hope that the techniques that we have developed will also prove useful in showing lower bounds on the computation time for other tasks than those considered here. For example, recent work adapts the techniques herein to prove an expected time lower bound on leader election [19]. The conclusion is that the naïve leader elimination CRN ($L + L \to L$) has optimal expected time to stabilize to a single leader from a "uniform" initial state.

Other work showing the challenges in parallelizing CRNs include the investigation of running multiple copies of networks in parallel [13], and the inability of networks starting with only large count species to delay the production of any species [17]. Although using more molecules in the same volume can increase the number of interactions per unit time, our results and these citations indicate that it can be sometimes nontrivial or impossible to exploit these interactions for computation.

3 Preliminaries

3.1 Chemical reaction networks

For $k \in \mathbb{Z}^+$, we write $\mathbb{N}^k$ to denote the set of all vectors of k nonnegative integers. A *predicate* is a Boolean-valued function $\psi : \mathbb{N}^k \to \{0,1\}$; for notational convenience we identify a predicate ψ equivalently with the set $\psi^{-1}(1) = \{ \mathbf{x} \in \mathbb{N}^k \mid \psi(\mathbf{x}) = 1 \}$. 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 $\mathbf{c} - \mathbf{c}'$ (which can have negative entries), 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 $S \in \Lambda \setminus \Delta$. 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 $\mathbf{v}(S_i) = n_i$. (Note that we will sometimes use negative coefficients in the notation $\{n_1 S_1, \ldots, n_k S_k\}$.)

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. We use no higher-order reactions in this paper.[10] Since the results of this paper hold no matter the rate constants, without loss of generality, in this paper we use $k = 1$ and the rate constant is omitted from the notation. For instance, given $\Lambda = \{A, B, C\}$, the reaction $A + B \to A + 3C$ is the pair $\langle (1,1,0), (1,0,3) \rangle$. 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 $\mathbf{c}_i \in q$ $(i \geq 1)$, $\mathbf{c}_i = \alpha_i(\mathbf{c}_{i-1})$.[11] 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 $\mathbf{c}$.

3.2 Algebra

A few concepts from vector algebra have proven useful in describing the reachable states of CRNs, as well as characterizing their computational power (see Section 3.3). 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. A is a *monoid* if $\mathbf{0} \in A$ and $A + A \subseteq A$, i.e., A is closed under addition. A is a *monoid coset (a.k.a. monoid offset)* if $A = \mathbf{b} + M$ for some constant vector $\mathbf{b} \in \mathbb{N}^k$ and monoid $M \subseteq \mathbb{N}^k$.

A linear set is a natural generalization of the notion of a periodic subset of $\mathbb{N}$ to higher dimensions. For example, the "slope 1 line" $\{ (x_1, x_2) \in \mathbb{N}^2 \mid x_1 = x_2 \}$ is a linear set. All linear sets are monoid cosets (and they are monoids if $\mathbf{b} = 0$ in the definition of linear), but the converse does not hold. For example, the set $\{ (x_1, x_2) \in \mathbb{N}^2 \mid x_2 \leq x_1 \leq 2^{x_2} - 1 \}$ is a monoid, but it is not linear (or even semilinear).

A powerful result due to Leroux [23] helps to restrict the complexity of the set of reachable states. It has long been known that the set of states reachable by a CRN from a given initial state is not necessarily semilinear [21], but recently Leroux showed that it is representable as a finite union of monoid cosets. (Leroux actually proves a more powerful result involving the first-order definability of certain sets, but the following implication is sufficient for our purposes.) For any CRN $C = (\Lambda, R)$ and set $X \subseteq \mathbb{N}^\Lambda$, let $\text{post}^C(X) = \{ \mathbf{y} \in \mathbb{N}^\Lambda \mid (\exists \mathbf{x} \in X)\ \mathbf{x} \Longrightarrow \mathbf{y} \}$ be the set of states reachable from some state in X.

Theorem 3.1 ([23]). *If $X \subseteq \mathbb{N}^\Lambda$ is semilinear, then $\text{post}^C(X)$ is a finite union of monoid cosets.*

[10] Sometimes the CRN model is extended to higher order reactions, but the kinetic model is hard to justify as more than two molecules are not likely to directly interact. Usually, higher order reactions are used simply as an approximation of a sequence of unimolecular and bimolecular elementary reactions.

[11] When the initial state to which a reaction sequence is applied is clear from context, we often overload terminology and refer to a reaction sequence and an execution sequence interchangeably as paths. The possibility that different reactions could result in identical state change (e.g., $A \to B$ and $A + C \to B + C$ when C is present) is immaterial to the arguments in this paper.

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

Lemma 3.2 (Dickson's Lemma [16]). *Every infinite sequence* $\mathbf{x}_0, \mathbf{x}_1, \ldots \in \mathbb{N}^\Lambda$ *has an infinite nondecreasing subsequence* $\mathbf{x}_{i_0} \leq \mathbf{x}_{i_1} \leq \ldots$, *where* $i_0 < i_1 < \ldots \in \mathbb{N}$.

3.3 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 is a stable decider if it 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 $\mathbf{x} = (x_1, \ldots, x_k) \in \mathbb{N}^k$, where $k = |\Sigma|$, we write the initial state as $\mathbf{n}(\mathbf{x}) = \{x_1 A_1, \ldots, x_k A_k\} + \mathbf{s} \in \mathbb{N}^\Lambda$ where $\Sigma = \{A_1, \ldots, A_k\}$ are the input species, and $\mathbf{s} \in \mathbb{N}^{\Lambda \setminus \Sigma}$ is the initial context. The initial context represents the molecules (such as a "leader") that can be assumed to be initially present independent of the input — which may assist in the computation. We extend ϕ to a partial function on states $\Phi : \mathbb{N}^\Lambda \to \{\text{NO}, \text{YES}\}$ as follows. $\Phi(\mathbf{c})$ 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* $\Phi(\mathbf{c})$ of state $\mathbf{c}$ is b.

A state $\mathbf{y}$ is *output stable* if $\Phi(\mathbf{y})$ is defined and, for all $\mathbf{c}$ such that $\mathbf{y} \Longrightarrow \mathbf{c}$, $\Phi(\mathbf{c}) = \Phi(\mathbf{y})$. We say that $\mathcal{D}$ *stably decides* the set $X \subseteq \mathbb{N}^k$, or that $\mathcal{D}$ *stably decides* the predicate $\psi_X : \mathbb{N}^k \to \{0, 1\}$ defined by $\psi_X(\mathbf{x}) = 1$ iff $\mathbf{x} \in X$, if, for all $\mathbf{x} \in X$, for all $\mathbf{c} \in \mathbb{N}^\Lambda$ such that $\mathbf{n}(\mathbf{x}) \Longrightarrow \mathbf{c}$, there exists an output stable $\mathbf{y} \in \mathbb{N}^\Lambda$ such that $\mathbf{c} \Longrightarrow \mathbf{y}$ and $\Phi(\mathbf{y}) = \psi_X(\mathbf{x})$.

Remark 3.1. For simplicity, the examples presented in Fig. 1 adhere to a different, *asymmetric* output convention wherein stabilization to a NO output occurs when every reachable state has no Y (i.e., no YES voter). It is trivial to convert most of those examples to the *symmetric* output convention defined above and used elsewhere in this paper: e.g., in Fig. 1(a)-(h), let all species $S \in \Lambda \setminus \{Y\}$ vote NO and add the reactions $Y + S \to 2Y$. In general, less is known about the asymmetric output convention than the symmetric one. For example, it is still not clear whether Theorem 3.3 applies. Remark 4.2 shows that we can nonetheless prove the impossibility of speed fault free computation of the "2A predicate," without an initial context, in the asymmetric case.

Remark 3.2. The above definition of stable decidability may seem weak since it does not actually require that the CRD *will* reach the output stable state, merely that it *could* from any reachable state. However, the definition is sufficient for a negative result: if a CRD does not satisfy it, then there is a reachable state from which the correct output stable state cannot be reached, and thus computation is not "deterministic". Further, our positive result (Section 4.1) is shown in the stochastic model of chemical kinetics where this weak definition of stable decidability implies that with probability 1 a correct output stable state will be reached (since in our construction there are finitely many distinct states that are reachable from any initial state). (The examples in Fig. 1, the examples below, as well as the construction for our positive result, actually satisfy a stronger combinatorial criterion: any sufficiently long reaction sequence will reach an output stable state.)

The following theorem due to Angluin, Aspnes, and Eisenstat [2] delineates the computational power of stable decidability. Recall the definition of semilinearity from Section 3.2.

Theorem 3.3 ([2]). *A set is stably decidable by a CRD if and only if it is semilinear.*

Example. The following CRD $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon, \phi, \mathbf{s})$ where $\Lambda = \{A_1, A_2, Y, N\}$, $\Sigma = \{A_1, A_2\}$, $\Upsilon = \{Y, N\}$, $\phi(Y) = \text{YES}$, $\phi(N) = \text{NO}$, $\mathbf{s} = \{1Y\}$, and R is the set of reactions below, stably decides the semilinear predicate $\psi(x_1, x_2) = 1$ iff $x_1 = x_2$ (recall that the set $(x_1, x_2) \in \mathbb{N}^2$ where $\psi(x_1, x_2) = 1$ is linear, and therefore semilinear):

$$\begin{aligned} A_1 + A_2 &\to Y \\ Y + N &\to Y \\ A_1 + Y &\to A_1 + N \\ A_2 + Y &\to A_2 + N \end{aligned}$$

The intuition is that the input species A_1 and A_2 cancel each other, and once either of them runs out Y can no longer be produced. If any A_1 or A_2 is left over, all Y will eventually be converted to N. The second reaction ensures that if $x_1 = x_2$, and we are left with a mixture of Y and N, all N degrades. (Note that this CRD satisfies a strong definition of stable decidability in that every sufficiently long reaction sequence eventually reaches the correct output stable state.)

4 Speed fault free CRDs

We first formally define speed fault free CRDs and then show our main result that speed fault free CRDs decide exactly the "detection predicates," i.e., detecting the presence or absence of a species, but 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.[12] Increasing the amount of fuel species is analogous to increasing the amount of "receptor" in the introduction. We now 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 output 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 $f \in \mathbb{N}$, let $\mathbf{c} \in \mathbb{N}^\Lambda$ be a state, and let $\alpha \in R$ be a reaction applicable to $\mathbf{c}$. We say that α occurring in state $\mathbf{c}$ is *f-fast* if at least one reactant has count at least f in $\mathbf{c}$. An execution sequence is called *f-fast* if all reactions in it are f-fast. It is worth noting that f-fast reaction sequences are not necessarily fast in the standard sense of stochastic kinetics (Section 4.1): even if each reaction occurs quickly, it could be that there are a huge number of reactions in the sequence. Also it is possible that an f-fast reaction is actually slower than one that is not f-fast: e.g., reaction $X + Y \to \ldots$ with f copies of X and 1 copy of Y is much slower than reaction $U + V \to \ldots$ with $f - 1$ copies of U and $f - 1$ copies of V. However, the expected time of a reaction that is *not* f-fast is bounded as a function of f (unimolecular: at least $1/(f-1)$, bimolecular: at least $v/(f-1)^2$; see Section 4.1) — and that is all that our negative result relies on. Our positive result shows not only the existence of desired f-fast paths, but also that the CRNs stabilize quickly under the standard stochastic model from any reachable state.

Definition 4.1. A *fueled* CRD is a 7-tuple $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon, F, \phi, \mathbf{s})$, where $F \in \Lambda \setminus \Sigma$ is the *fuel species*, and $(\Lambda, R, \Sigma, \Upsilon, \phi, \mathbf{s})$ is a CRD with input alphabet $\Sigma = (A_1, \ldots, A_k)$. For all $n, x_1, \ldots, x_k \in \mathbb{N}$ let $\mathbf{n}_n(x_1, \ldots, x_k)$ denote the initial state $\{nF, x_1A_1, \ldots, x_kA_k\} + \mathbf{s}$. We say $\mathcal{D}$ *stably computes* the predicate $\psi : \mathbb{N}^k \to \{0, 1\}$ if, for all $x_1, \ldots, x_k \in \mathbb{N}$ and all $n \in \mathbb{N}$, for any state $\mathbf{c}$ such that

[12] Allowing multiple fuel species $F_1, F_2, \ldots$ does not affect our results since one reaction could be $F \to F_1 + F_2 \ldots$.

$\mathbf{n}_n(x_1, \ldots, x_k) \Longrightarrow \mathbf{c}$, there is an output stable state $\mathbf{y}$ with $\phi(\mathbf{y}) = \psi(x_1, \ldots, x_k)$ such that $\mathbf{c} \Longrightarrow \mathbf{y}$. We say $\mathcal{D}$ is *speed fault free* if, for all $x_1, \ldots, x_k \in \mathbb{N}$ and all $f \in \mathbb{N}$, for all sufficiently large n, for any state $\mathbf{c}$ such that $\mathbf{n}_n(x_1, \ldots, x_k) \Longrightarrow \mathbf{c}$, there is an output stable state $\mathbf{y}$ with $\phi(\mathbf{y}) = \psi(x_1, \ldots, x_k)$ such that $\mathbf{c} \Longrightarrow \mathbf{y}$ by an f-fast execution sequence.

In other words, from any reachable state, there is always a sequence of fast reactions that reaches the correct answer. Note that our definition of speed fault free decidability naturally parallels the form of the definition of stable decidability: from any reachable state, there is always a sequence of reactions that reaches the correct answer.

Remark 4.1. The definition of speed-fault free requires the CRD to maintain an f-fast execution sequence to the output from any reachable state, *even those reached through a non-f-fast execution sequence*. Our definition is meant to capture the intuitive notion of "fast even in the worst case," and thus we do not limit the "adversary" to stay on fast paths. (It is open whether Lemma 4.15 holds in the case that the definition of speed-fault free is modified to disallow the adversary from executing reactions that are not f-fast.)

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 (predicate) is a *detection set (predicate)* if it is expressible as a combination of finite unions, intersections, and complements of simple detection sets.

In other words, a detection predicate 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 4.15.

Theorem 4.3. *A predicate is stably decidable by a speed fault free CRD if and only if it is a detection predicate.*

Example. The following CRD $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon, \phi, \mathbf{s})$ where $\Lambda = \{A_1, A_2, F, G\}$, $\Sigma = \{A_1, A_2\}$, $\Upsilon = \{A_1, A_2, F, X\}$, $\phi(A_1) = \text{YES}$, $\phi(A_2) = \text{NO}$, $\phi(F) = \text{NO}$, $\phi(X) = \text{NO}$, $\mathbf{s} = \{\}$, and R is the set of reactions below, stably decides the detection predicate $\psi(x_1, x_2) = 1$ iff $x_1 > 0$ and $x_2 = 0$. Further it is speed fault free.

$$\begin{aligned} F + A_2 &\to 2A_2 \\ F + A_1 &\to 2A_1 \\ F + X &\to 2X \\ A_2 + A_1 &\to 2X \\ A_2 + X &\to 2X \\ A_1 + X &\to 2X \end{aligned}$$

To see that this CRD stably decides ψ, it can be shown that from any state reachable from the initial state $\{x_1 A_1, x_2 A_2, nF\}$, we can reach one of four output stable states, depending on which of A_1, A_2 were present initially. Indeed, for this CRD any sufficiently long sequence of reactions reaches one of these output stable states. These four output stable states are $\mathbf{y}_0 = \{n'F\}$, $\mathbf{y}_1 = \{n'A_1\}$, $\mathbf{y}_2 = \{n'A_2\}$, $\mathbf{y}_3 = \{n'X\}$, where $n' = n + x_1 + x_2$ (i.e., n' is the total molecular count).[13] If $x_1 > 0$ and $x_2 = 0$ then $\mathbf{y}_1$ is reachable (output YES) and the others are not. Otherwise,

[13]There are other output stable states, for example $\{n_1 F, n_2 A_2\}$ for $n_1 + n_2 = n + x_2$, but for the purpose of showing the CRN is speed fault free, it is sufficient to show that some output stable state is reachable by a fast path.

if $x_1 = x_2 = 0$ then the only reachable state is $\mathbf{y}_0$ (output NO), if $x_1 = 0$ and $x_2 > 0$ then only $\mathbf{y}_2$ is reachable (output NO), if $x_1 > 1$ and $x_2 > 1$ then only $\mathbf{y}_3$ is reachable (output NO).

To see that this CRD is speed fault free, note that in any reachable state there are n' molecules (i.e., total count is preserved). Since there are 4 species, at least one of them must be present in count $n'/4$. Note that any two species can react. Thus, unless we are in one of the above output stable states, some $n'/4$-fast reaction can happen.

4.1 Positive results: detection predicates are stably decidable by speed fault free CRDs

In this section we show by construction that any detection predicate can be stably decided by a speed fault free CRD. We will also argue that the CRD is fast under the standard stochastic time model of chemical kinetics [20]. First, we define this model (since all rate constants in this paper are 1, we present a simplified model without rate constants).

Let *volume* $v \in \mathbb{R}^+$ represent the size of the physical system in which the reactions are occurring. Intuitively, the rates of the reactions scale as the number of ways in which the reacting combination of molecules can be chosen in the current state. Further, bimolecular reactions become slower when diluted in a larger volume. Formally, suppose the CRD is in state $\mathbf{c}$. The *propensity* of a unimolecular reaction $\alpha : X \to \ldots$ in state $\mathbf{c}$ is $\rho(\mathbf{c}, \alpha) = \mathbf{c}(X)$. The propensity of a bimolecular reaction $\alpha : X + Y \to \ldots$, where $X \neq Y$, is $\rho(\mathbf{c}, \alpha) = \frac{\mathbf{c}(X)\mathbf{c}(Y)}{v}$. The propensity of a bimolecular reaction $\alpha : X + X \to \ldots$ is $\rho(\mathbf{c}, \alpha) = \frac{1}{2}\frac{\mathbf{c}(X)(\mathbf{c}(X)-1)}{v}$. The propensity function determines the kinetics of the CRD as follows. The time until the next reaction occurs is an exponential random variable with rate $\rho(\mathbf{c}) = \sum_{\alpha \in R} \rho(\mathbf{c}, \alpha)$ (therefore expected value $1/\rho(\mathbf{c})$). The probability that next reaction will be a particular α_{next} is $\frac{\rho(\mathbf{c}, \alpha_{\text{next}})}{\rho(\mathbf{c})}$. In other words, the system is a continuous-time, discrete-state Markov process in which transitions between states correspond to reactions, with transition rate equal to the reaction propensity.

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

Proof. Let $A_1, \ldots, A_k$ be the input species. For each $a \in \{0,1\}^k$, let there be species X_a, which will represent a particular combination of inputs that has been detected. Then for each $a, b \in \{0,1\}^k$ such that $a \neq b$, we have reactions

$$X_a + X_b \quad \to \quad 2X_c$$

where c is the bit-wise OR of a and b. We identify X_{0^k} with the fuel species F (recall its initial count is n), and input species A_i with $X_{a(i)}$ where $a(i)$ is a vector of all zeros except in the ith position. Further, letting $f : \{0,1\}^k \to \{0,1\}$ describe the detection predicate, species X_a vote YES if $f(a) = 1$ and vote NO otherwise. The CRD starts in the state with exactly the input species and n fuel species X_{0^k}.

To analyze this CRD, note that for no further reactions to be possible, there can be exactly one species present. The only reachable state where there is only one species present is $\mathbf{y} = \{n' X_a\}$, where a is the bitwise OR of all the input $a(i)$'s, and n' is the sum of n and the counts of the input molecules (i.e., n' is the total molecular count). Note that by construction this state has the correct output. Since every reaction instance increases the count of 1's of a in some X_a, at most $n'k$ reaction instances can occur until all n' molecules become of identical species. This implies that

the CRD stably computes the detection predicate, and indeed satisfies a stronger combinatorial criterion that all sufficiently long reaction sequences eventually reach a correct output stable state.

Further, it is easy to see that from any reachable state, there is an $n'/2^k = \Omega(n)$-fast path to $\mathbf{y}$, and thus the CRD is speed fault free (k is a constant). In any reachable state, for some $a \in \{0,1\}^k$, the molecular count of X_a is at least $n'/2^k$ (since there are always n' total molecules). So as long as not all molecules are of the same species, an $(n'/2^k)$-fast reaction is possible.

Finally we analyze the expected time to reach $\mathbf{y}$ from any reachable state. If none of the input species is present then the output is correct at the start. Otherwise, for any bit i present in the initial state, let $n_i(t)$ be the total count of molecules with bit i set to 1 at time t. This function is monotone non-decreasing. In any state with $n_i(t) = l$, the sum of the propensities of reactions that increase $n_i(t)$ is $(n' - l)l/v$ regardless of the other bits. Therefore, the expected time for $n_i(t)$ to increase from l to $l+1$ is less than $\frac{2v}{n'l}$ if $l < n'/2$, and at most $\frac{2v}{(n'-l)n'}$ if $l \geq n'/2$. By linearity of expectation, the total expected time for $n_i(t)$ to go from 1 to n' is at most: $2v\left(\sum_{l=1}^{\frac{n'}{2}-1} \frac{1}{n'l} + \sum_{l=\frac{n'}{2}}^{n'-1} \frac{1}{(n'-l)n'}\right) = \frac{2v}{n'}\, O(\log(n'))$. For constant volume v, and since $n' = O(n)$ (we consider increasing the amount of fuel n for fixed input), this is $O(\log(n)/n)$. Finally, because there is a constant number of bits, the overall time to reach an output stable state is $O(\log(n)/n)$ and we obtain the statement of the lemma. □

4.2 Negative results: speed fault free CRDs stably decide only detection predicates

In this subsection we prove our main negative result, that speed fault free CRDs can decide only detection predicates. Subsection 4.2.1 develops a notion of pumping that allows us to reason about "large count" species (those that can increase with more fuel) versus "small count" species (those that are bounded no matter the amount of fuel). Subsection 4.2.2 shows a way in which reaction sequences could be decomposed into separate "test-tubes" (parallel decomposition). Subsection 4.2.3 uses this machinery to show a simpler result, Lemma 4.13, that *leaderless* speed fault free CRDs (those with initial context $\mathbf{0}$) stably decide only a limited class of predicates (those closed under doubling, which contain the detection predicates but also others such as $\psi(x_1, x_2) = 1 \iff x_1 = x_2$). In fact, for this class of predicates and CRDs, our proof shows the stronger conclusion that, when the answer is (w.l.o.g.) YES, there is a reachable state $\mathbf{x}$ such that every path from $\mathbf{x}$ to any state with positive count of a YES voter is slow.[14] To prove our main theorem, we develop additional technical machinery in Subsection 4.2.4 showing that certain fast paths that reduce the count of some species from "large" to "small" implies that the reactions obey a certain structure that will be useful for doing "surgery" on paths to slightly alter the count of those species in a controllable way. Finally, Subsection 4.2.5 proves our main result, Theorem 4.3.

4.2.1 Pumpable sets of species and Π-friendly paths

This subsection 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 if there is an A, $\{F'\}$ is pumpable, because F' can be arbitrarily large by executing the reaction sufficiently many times. From state $\{1A, nF\}$, the set $\{F, F'\}$ is pumpable

[14] Our main result, applied to general CRDs deciding a non-detection predicate, concludes only that every path from $\mathbf{x}$ to a *stable* YES state is slow.

since one can execute the reaction $n/2$ times to obtain counts $n/2$ of F and $n/2$ of F', and $n/2$ grows unboundedly with n.

This subsection also introduces the notion of Π-friendly paths for a subset of species $\Pi \subseteq \Lambda$, which is essentially (for careful choices of Π) a proxy for f-fast paths that is easier to work with. Intuitively, a path is Π-friendly if every reaction in it has at least one reactant in Π, and thus if Π are the only pumpable species (i.e., the only species that can get "large"), then f-fast paths must be Π-friendly. This notion will feature prominently in our arguments, in particular as a result of the parallel decomposition lemma on Π-friendly paths (shown in the next subsection). First we use it in this subsection to enforce a self-consistency of pumping: you can pump without requiring a reaction where no reactant is pumpable. This self-consistency will be necessary to ensure that the entire path we consider is Π-friendly and that the parallel decomposition lemma of the next subsection applies.

For a condition $\phi : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$, we write *for all* $u \in \mathbb{N}$, *there exists* $n \in \mathbb{N}$ *such that* $\phi(n, u) = 1$ *and* $n \to \infty$ *as* $u \to \infty$ to mean that for all $u \in \mathbb{N}$, there exists $n(u) \in \mathbb{N}$ depending on u such that $\phi(n(u), u) = 1$ and $\lim_{u\to\infty} n(u) = \infty$.

Let $N = (\mathbf{n}_n)_{n\in\mathbb{N}}$ and $U = (\mathbf{u}_u)_{u\in\mathbb{N}}$ be two infinite sequences of states, i.e., each $\mathbf{n}_n, \mathbf{u}_u \in \mathbb{N}^\Lambda$. (From now on the index specification will be implicit and we will simply write $(\mathbf{n}_n)$ instead of $(\mathbf{n}_n)_{n\in\mathbb{N}}$.) We write $N \Longrightarrow U$ if, for all $u \in \mathbb{N}$, there exists n such that $\mathbf{n}_n \Longrightarrow \mathbf{u}_u$, and $n \to \infty$ as $u \to \infty$.[15][16]

Let $\Pi \subseteq \Lambda$. If a reaction has at least one reactant in Π, say the reaction is Π-*friendly*. If $\mathbf{n} \Longrightarrow \mathbf{u}$ via a reaction sequence in which all reactions are Π-friendly, then we write $\mathbf{n} \Longrightarrow^\Pi \mathbf{u}$. We write $N \Longrightarrow^\Pi U$ if, for all $u \in \mathbb{N}$, there exists n such that $\mathbf{n}_n \Longrightarrow^\Pi \mathbf{u}_u$ and $n \to \infty$ as $u \to \infty$.

Let $N = (\mathbf{n}_0 \leq \mathbf{n}_1 \leq \dots)$ be an infinite nondecreasing sequence of states. A set of species $\Pi \subseteq \Lambda$ is N-*pumpable* if there exists a non-decreasing sequence of states $U = (\mathbf{u}_0 \leq \mathbf{u}_1 \leq \dots)$ such that for all $P \in \Pi$ and $u \in \mathbb{N}$, $\mathbf{u}_u(P) \geq u$, and $N \Longrightarrow^\Pi U$. Call U a *pumping sequence* for Π. Π is *maximal* N-*pumpable* if it is N-pumpable and no strict superset of Π is N-pumpable.

Example. Consider the sequence $N = (\mathbf{n}_n)$ defined by $\mathbf{n}_n = \{1A, nF\}$ for each $n \in \mathbb{N}$, and reactions $A \to B, A \to C, B + F \to 2B, C + F \to 2C$. The sets of species $\{B\}$ and $\{C\}$ are individually N-pumpable. Since the presence of B is mutually exclusive with that of C, the set $\{B, C\}$ is not N-pumpable. (However, for initial states N' defined by $\mathbf{n}'_n = \{2A, nF\}$, the set $\{B, C\}$ *is* N'-pumpable.) Neither $\{B\}$ nor $\{C\}$ is maximal N-pumpable, however, since the states $\{\frac{n}{2}B, \frac{n}{2}F\}$ and $\{\frac{n}{2}C, \frac{n}{2}F\}$ are also reachable from $\mathbf{n}_n$: the maximal N-pumpable sets are thus $\{B, F\}$ and $\{C, F\}$.

First, we observe that any species unbounded in N (such as F in the previous example) is not only trivially N-pumpable, but is contained in any maximal N-pumpable set.

Proposition 4.5. *Let N be an infinite nondecreasing sequence of states, let Π be maximal N-pumpable with pumping sequence U, and let $\Gamma \subseteq \Lambda$ be the unbounded species in N, i.e., for each $S \in \Gamma$, $\lim_{n\to\infty} \mathbf{n}_n(S) = \infty$.*[17] *Then $\Gamma \subseteq \Pi$.*

Proof. Since U is a pumping sequence for Π, for each $u \in \mathbb{N}$, there is an $n \in \mathbb{N}$ such that $\mathbf{n}_n \Longrightarrow^\Pi \mathbf{u}_u$ and $\mathbf{u}_u(S) \geq u$ for all $S \in \Pi$. Define $U' = (\mathbf{u}'_{u'})$ as follows. For each u', let $n'(u')$ be the smallest

[15] The requirement that $n \to \infty$ as $u \to \infty$ ensures that if we consider larger and larger states $\mathbf{u}_u$, we can start from larger and larger states $\mathbf{n}_n$ to get there.

[16] Note that this relation is transitive; in particular, if $N \Longrightarrow U \Longrightarrow Y$ for sequences $N = (\mathbf{n}_n), U = (\mathbf{u}_u), Y = (\mathbf{y}_y)$, then for each y, there is an n such that $\mathbf{n}_n \Longrightarrow \mathbf{y}_y$ and $n \to \infty$ as $y \to \infty$.

[17] Note that since the $\mathbf{n}_n$'s are nondecreasing, this implies that $\lim_{n\to\infty} \min_{S\in\Gamma} \mathbf{n}_n(S) = \infty$, i.e., all $S \in \Gamma$ are *simultaneously* growing with n.

integer such that for some $n < n'(u')$, writing $\mathbf{p} = \mathbf{n}_{n'(u')} - \mathbf{n}_n$ (note $\mathbf{p} \geq \mathbf{0}$ since N is nondecreasing), 1) $\mathbf{n}_n \Longrightarrow^{\Pi} \mathbf{u}_u$ such that $u \geq u'$, and 2) $\mathbf{p}(S) \geq u'$ for all $S \in \Gamma$. Then $\mathbf{n}_{n'(u')} = \mathbf{n}_n + \mathbf{p} \Longrightarrow^{\Pi} \mathbf{u}_u + \mathbf{p}$. Define $\mathbf{u}'_{u'} = \mathbf{u}_u + \mathbf{p}$. For all $S \in \Gamma$, $\mathbf{u}'_{u'}(S) \geq \mathbf{p}(S) \geq u'$ by (2), and for all $S \in \Pi$, $\mathbf{u}'_{u'}(S) \geq \mathbf{u}_u \geq u \geq u'$ by (1); i.e., $\mathbf{u}'_{u'}(S) \geq u'$ for all $S \in \Gamma \cup \Pi$. Let $\bar{U} = (\bar{\mathbf{u}}_{\bar{u}})$ be an infinite nondecreasing subsequence of U' (by Lemma 3.2). (Note that reindexing due to taking a subsequence can only decrease the index of an element, and thus we still have that for all $S \in \Gamma \cup \Pi$, $\bar{\mathbf{u}}_{\bar{u}}(S) \geq \bar{u}$.) Then $\Gamma \cup \Pi$ is N-pumpable with pumping sequence $\bar{U}$. Since Π is a maximal N-pumpable set, this implies $\Gamma \subseteq \Pi$. □

The previous proposition showed that species already large *before* pumping a maximal pumpable set are necessarily contained in it; the next proposition, in a sense, goes in the other direction, showing this for species that can be made large *after* pumping. One way to think of this is that pumpable sets can be naturally "chained." Suppose Π is maximal N-pumpable with pumping sequence U, and Ω is U-pumpable. Then $\Omega \subseteq \Pi$ by the maximality of Π. Intuitively, this is because by "withholding" sufficiently many copies of species in Π, we can reach a state in which all species in $\Pi \cup \Omega$ are "large," hence N-pumpable. The following lemma captures chaining of pumpable sets more generally, allowing for intervening Π-fast reactions (i.e., the paths $U \Longrightarrow^{\Pi} Y$ below).

Proposition 4.6. *Let N, U, Y be infinite nondecreasing sequences of states. Let Π be maximal N-pumpable with pumping sequence U, and suppose $U \Longrightarrow^{\Pi} Y$. Let Ω be Y-pumpable. Then $\Omega \subseteq \Pi$.*

Proof. We argue that if $\Omega \not\subseteq \Pi$, Π is not maximal N-pumpable. Let W be a corresponding pumping sequence for Ω, and let $U = (\mathbf{u}_u)$ and $W = (\mathbf{w}_w)$. By assumption, $N \Longrightarrow^{\Pi} U \Longrightarrow^{\Pi} Y \Longrightarrow^{\Omega} W$. This implies that for all $w \in \mathbb{N}$, there is $u_w \in \mathbb{N}$ (such that $u_w \to \infty$ as $w \to \infty$) and path p_w such that $\mathbf{u}_{u_w} \Longrightarrow_{p_w}^{\Pi \cup \Omega} \mathbf{w}_w$. Since U is a pumping sequence for Π, for any u, for all $S \in \Pi$, $\mathbf{u}_u(S) \geq u$. Also recall U is nondecreasing. Thus by starting with a sufficiently larger $\hat{u}_w \geq u_w$ and then taking the same path p_w, we can get $\mathbf{u}_{\hat{u}_w} \Longrightarrow_{p_w}^{\Pi \cup \Omega} \hat{\mathbf{w}}_w$, where $\hat{\mathbf{w}}_w \geq \mathbf{w}_w$, and for all $S \in \Pi$, $\hat{\mathbf{w}}_w(S) \geq w$. In other words, by the fact that W is a pumping sequence for Ω, we had that for all $S \in \Omega$, $\mathbf{w}_w(S) \geq w$, but now we ensure that for all $S \in \Pi \cup \Omega$, $\hat{\mathbf{w}}_w(S) \geq w$. Let $\hat{W}$ be the sequence of $\hat{\mathbf{w}}_w$ constructed in this manner for each w, and let $\bar{W} = (\bar{\mathbf{w}}_w)$ be an infinite nondecreasing subsequence of $\hat{W}$ (by Lemma 3.2). (Note that reindexing due to taking a subsequence can only decrease the index of an element, and thus we still have that for all $S \in \Pi \cup \Omega$, $\bar{\mathbf{w}}_w(S) \geq w$.) Finally, observe that since we chose $\hat{u}_w \geq u_w$ and $u_w \to \infty$ as $w \to \infty$, we can conclude that $Y \Longrightarrow^{\Pi \cup \Omega} \bar{W}$. This implies that overall $N \Longrightarrow^{\Pi \cup \Omega} \bar{W}$. Thus, $\Pi \cup \Omega$ is N-pumpable with pumping sequence $\bar{W}$. If $\Omega \not\subseteq \Pi$, then this shows that Π is not maximal N-pumpable. □

The consequence of pumping a maximal set Π is that we know that the counts of all other species (not in Π) are bounded no matter which path we take, as long as it is Π-friendly or c-fast (for large enough c). This is captured in following two propositions, first for Π-friendly, and second for c-fast paths.

Proposition 4.7. *Let N be an infinite nondecreasing sequence of states and let Π be maximal N-pumpable, with pumping sequence $U = (\mathbf{u}_u)$. Then there is a bound c_U (depending only on U and the CRN) such that, for all $S \in \Lambda \setminus \Pi$, $\mathbf{c} \in \mathbb{N}^{\Lambda}$, and $u \in \mathbb{N}$ such that $\mathbf{u}_u \Longrightarrow^{\Pi} \mathbf{c}$, $\mathbf{c}(S) < c_U$.*

Proof. Fix N as in the statement of the Proposition. It suffices to consider only one maximal pumpable set since there are a finite number of them, so fix a maximal pumpable Π. Let $U = (\mathbf{u}_u)$ be a pumping sequence for Π, so that $N \Longrightarrow^{\Pi} U$. Let $S \in \Lambda$ be any species that can grow arbitrarily large from U, i.e., $(\forall c \in \mathbb{N})(\exists \mathbf{c}_c \in \mathbb{N}^{\Lambda})(\exists u \in \mathbb{N})$ $\mathbf{u}_u \Longrightarrow^{\Pi} \mathbf{c}_c$ and $\mathbf{c}_c(S) \geq c$. Let Y be an infinite

nondecreasing subsequence of $(\mathbf{c}_c)$, and let $\Omega = \{S\}$. Then Ω is Y-pumpable (trivially, with Y as the pumping sequence). Since $U \Longrightarrow^{\Pi} Y$, Proposition 4.6 then implies $\Omega \subseteq \Pi$, so $S \in \Pi$. □

Proposition 4.8. *Let N be an infinite nondecreasing sequence of states, and let Π be maximal N-pumpable, with pumping sequence $U = (\mathbf{u}_u)$. Let c_U be the bound from Proposition 4.7. For all $\mathbf{c} \in \mathbb{N}^\Lambda$, if $\mathbf{u}_u \Longrightarrow \mathbf{c}$ by a c_U-fast path p then: (1) p is Π-friendly, and (2) $\forall S \in \Lambda$, $\mathbf{c}(S) \geq c_U$ implies $S \in \Pi$.*

Proof. Conclusion (1) follows by contradiction as follows. Let α be the first reaction along p that is not Π-friendly. Since the state immediately preceding this reaction is reachable by a Π-friendly path, Proposition 4.7 tells us that all species $S \in \Lambda \setminus \Pi$ have count less than c_U. Therefore α occurs when the count of all its reactants is less than c_U, hence it is not c_U-fast, a contradiction. Finally, conclusion (2) follows from (1) by Proposition 4.7 applied on the entire path p. □

Note that the above proposition means that the only way to get a species outside of Π "large" is by executing a "slow" reaction (between two reactants not in Π).

Finally, products of reactions whose reactants are both maximal pumpable must also be pumpable, since we can use these reactions to produce a "large" amount of the product species.

Proposition 4.9. *Let N be an infinite nondecreasing sequence of states, and let Π be maximal N-pumpable. If the CRN contains a reaction with all reactants in Π, then all products are in Π.*

Proof. Let α be a reaction with all reactants in Π. For each $u \in \mathbb{N}$, let $\mathbf{u}_u$ be reachable from some $\mathbf{n}_n$ such that $\mathbf{u}_u(S) \geq u$ for all $S \in \Pi$. From state $\mathbf{u}_u$ execute α $u/2$ times (that may be the maximum number of times the reaction can execute if it is of the form $X + X \to \dots$). This results in a state in which all products of the reaction α have count at least $u/2$. Since $u/2$ grows unboundedly, and α is Π-friendly, Proposition 4.7 establishes that the products are in Π. □

We observe that, due to Theorem 3.1, if N is specially structured to have a constant difference between adjacent states, there is pumping sequence U for N that is similarly structured to have constant difference between adjacent states. This additional structure on the pumping sequence will prove essential for proving the Π perturbation claim (Claim 1) in the main argument.

Lemma 4.10. *Let $\mathbf{h} \in \mathbb{N}^\Lambda$ and let $N = (\mathbf{n}_n)$ be such that $\mathbf{n}_{n+1} = \mathbf{n}_n + \mathbf{h}$ for all $n \in \mathbb{N}$. Let Π be maximal N-pumpable. Then there is a pumping sequence $U = (\mathbf{u}_u)$ for Π, and $\mathbf{d} \in \mathbb{N}^\Pi$ where for all $S \in \Pi$, $\mathbf{d}(S) > 0$, such that for all $u \in \mathbb{N}$, $\mathbf{u}_u + \mathbf{d} = \mathbf{u}_{u+1}$.*

Proof. Let $X = \{\ \mathbf{n}_n \mid n \in \mathbb{N}\ \}$. Note that X is semilinear (in fact linear). Since Π is N-pumpable, let $U' = (\mathbf{u}'_0 \leq \mathbf{u}'_1 \leq \dots)$ be any pumping sequence for Π, so that for each $u \in \mathbb{N}$, there is an n such that $\mathbf{n}_n \Longrightarrow^{\Pi} \mathbf{u}'_u$ and $\mathbf{u}'_u(S) \geq u$ for all $S \in \Pi$. Remove all reactions from our original CRN that are not Π-friendly to obtain a new CRN C_Π. We then apply Theorem 3.1 to the new CRN C_Π to obtain the following. There exist $\mathbf{b}_1, \dots, \mathbf{b}_l \in \mathbb{N}^\Lambda$ and monoids $M_1, \dots, M_l \subseteq \mathbb{N}^\Lambda$ such that $\text{post}^{C_\Pi}(X) = \bigcup_{j=1}^{l} (\mathbf{b}_j + M_j)$. Let $u_0 > \max\limits_{1 \leq j \leq l} |\mathbf{b}_j|$. Since $\mathbf{u}'_{u_0} \in \text{post}^{C_\Pi}(X)$, there is a j such that $\mathbf{u}'_{u_0} \in \mathbf{b}_j + M_j$.

Let $\mathbf{d} = \mathbf{u}'_{u_0} - \mathbf{b}_j$; then $\mathbf{d} \in M_j$. Note $\mathbf{d} > 0$ by our choice of u_0. Define $\mathbf{u}_0 = \mathbf{u}'_{u_0}$, and for all $u \in \mathbb{N}$, define $\mathbf{u}_{u+1} = \mathbf{u}_u + \mathbf{d}$. For all $u \in \mathbb{N}$, $\mathbf{u}_u \in \mathbf{b}_j + M_j$ because M_j is closed under addition. Thus by the definition of $\text{post}^{C_\Pi}(X)$, for all $u \in \mathbb{N}$, there is an $n \in \mathbb{N}$ such that $\mathbf{n}_n \Longrightarrow^{\Pi} \mathbf{u}_u$. Since $\mathbf{u}'_{u_0}(S) > \mathbf{b}_j(S)$ for all $S \in \Pi$, $\mathbf{d}(S) = \mathbf{u}'_{u_0}(S) - \mathbf{b}_j(S) > 0$.

For $(\mathbf{u}_u)$ to be a pumping sequence for Π, we additionally need $n \to \infty$ as $u \to \infty$. We can ensure this without loss of generality as follows: For all u, let $n(u)$ be an index of N such

that $\mathbf{n}_{n(u)} \Longrightarrow^{\Pi} \mathbf{u}_u$. Note that for all $u \in \mathbb{N}$, $\mathbf{n}_{n(u)+u} = \mathbf{n}_{n(u)} + u\mathbf{h} \Longrightarrow^{\Pi} \mathbf{u}_u + u\mathbf{h}$. Thus if $n(u)$ is bounded, we can define $\hat{\mathbf{d}} = \mathbf{d} + \mathbf{h}$, and define a new sequence $(\hat{\mathbf{u}}_u)$ by $\hat{\mathbf{u}}_0 = \mathbf{u}_0$ and, for all $u \in \mathbb{N}$, $\hat{\mathbf{u}}_{u+1} = \hat{\mathbf{u}}_u + \hat{\mathbf{d}}$, so $\hat{\mathbf{u}}_u = \mathbf{u}_0 + u\hat{\mathbf{d}}$. For all $u \in \mathbb{N}$, defining $\hat{n}(u) = n(u) + u$, we have $\mathbf{n}_{\hat{n}(u)} = \mathbf{n}_{n(u)+u} \Longrightarrow^{\Pi} \mathbf{u}_u + u\mathbf{h} = \mathbf{u}_0 + u\mathbf{d} + u\mathbf{h} = \hat{\mathbf{u}}_u$; note that $\hat{n}(u) \to \infty$ as $u \to \infty$.

The requirement that $\hat{\mathbf{d}} \in \mathbb{N}^{\Pi}$ (i.e., that $\mathbf{d}$ does not contain species outside of Π) follows by the maximality of Π. We then let $\mathbf{d}$ and $(\mathbf{u}_u)$ in the statement of the lemma be $\hat{\mathbf{d}}$ and $(\hat{\mathbf{u}}_u)$, respectively. □

4.2.2 Parallel decomposition

Intuitively, the following lemma shows that CRNs reacting by Π-friendly reactions can be effectively decomposed into separate non-interacting "test tubes" (in the context of a large excess of species in Π). 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 $\mathbf{z}' + \mathbf{z}'' = \mathbf{z}$. Using this parallel decomposition lemma we will repeatedly argue that if something can happen from the whole test tube then it could happen from one of the halves — often arriving at a contradiction.

Lemma 4.11. *Suppose* $\mathbf{x}_1 + \mathbf{x}_2 \Longrightarrow^{\Pi} \mathbf{y}$. *Then there are* $\mathbf{p}, \mathbf{p}', \mathbf{p}'' \in \mathbb{N}^{\Pi}$, *and* $\mathbf{y}', \mathbf{y}'' \in \mathbb{N}^{\Lambda}$ *such that* $\mathbf{p} + \mathbf{x}_1 \Longrightarrow^{\Pi} \mathbf{p}' + \mathbf{y}'$ *and* $\mathbf{p} + \mathbf{x}_2 \Longrightarrow^{\Pi} \mathbf{p}'' + \mathbf{y}''$, *where* $\mathbf{y}' + \mathbf{y}'' = \mathbf{y}$ *and* $\mathbf{p}' + \mathbf{p}'' = 2\mathbf{p}$.

Proof. First we define two concepts that will be used in the proof: *parallel decomposition* and *longest parallel prefix*. For a reaction sequence q applied to a state $\mathbf{x}$ to give $\mathbf{x} \Longrightarrow_q \mathbf{y}$, where $\mathbf{x}$ is written as a sum of two states $\mathbf{x}_1 + \mathbf{x}_2 = \mathbf{x}$, we say that q has a *parallel decomposition from* $(\mathbf{x}_1, \mathbf{x}_2)$ if there exists a partition of q into two disjoint subsequences of reactions (l, r) such that $\mathbf{x}_1 \Longrightarrow_l \mathbf{y}_1$, $\mathbf{x}_2 \Longrightarrow_r \mathbf{y}_2$, and $\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2$. In other words, if we imagine splitting $\mathbf{x}$ into two "tubes" $\mathbf{x}_1$ and $\mathbf{x}_2$, then the evolution determined by the reaction sequence q can be interpreted as happening entirely within the tubes.

Suppose a reaction sequence p is applicable to $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, but p does not have a parallel decomposition from $(\mathbf{x}_1, \mathbf{x}_2)$. Then there is a longest prefix q of p (possibly q is empty) such that q has a parallel decomposition (l, r) from $(\mathbf{x}_1, \mathbf{x}_2)$. We call q the *longest parallel prefix* of p from $(\mathbf{x}_1, \mathbf{x}_2)$. Let $(\mathbf{l}, \mathbf{r})$ be such that $\mathbf{x}_1 \Longrightarrow_l \mathbf{l}$ and $\mathbf{x}_2 \Longrightarrow_r \mathbf{r}$. In other words, q is the furthest that $\mathbf{x}_1$ and $\mathbf{x}_2$ can evolve on their own before the next reaction in p requires a molecule from $\mathbf{l}$ and a molecule from the other $\mathbf{r}$. Therefore the next reaction must be bimolecular $L + R \to \ldots$, and it must be the case that, without loss of generality, $\mathbf{l}(R) = 0$, $\mathbf{l}(L) > 0$ and $\mathbf{r}(L) = 0$, $\mathbf{r}(R) > 0$; otherwise one of the reaction sequences l or r could be extended by that reaction while remaining a parallel decomposition, and q would not be the longest prefix of p with a parallel decomposition.

Now we proceed with the proof of the lemma. Let p be the Π-friendly reaction sequence such that $\mathbf{x}_1 + \mathbf{x}_2 \Longrightarrow_p \mathbf{y}$. Let $\mathbf{p}_n \in \mathbb{N}^{\Pi}$ consist of exactly n molecules of every species in Π. For any $\mathbf{p}_n$ we can apply the path p in the context of $\mathbf{p}_n$: $2\mathbf{p}_n + \mathbf{x}_1 + \mathbf{x}_2 \Longrightarrow_p 2\mathbf{p}_n + \mathbf{y}$. Let q_n be the longest parallel prefix of p from $(\mathbf{p}_n + \mathbf{x}_1, \mathbf{p}_n + \mathbf{x}_2)$ and let (l_n, r_n) be the parallel decomposition of q_n from $(\mathbf{p}_n + \mathbf{x}_1, \mathbf{p}_n + \mathbf{x}_2)$. Let $\mathbf{l}_n$ and $\mathbf{r}_n$ be such that $\mathbf{p}_n + \mathbf{x}_1 \Longrightarrow_{r_n} \mathbf{l}_n$ and $\mathbf{p}_n + \mathbf{x}_2 \Longrightarrow_{l_n} \mathbf{r}_n$.

In the remainder of the proof we will show that for large enough n, $q_n = p$. Showing $q_n = p$ completes the proof: $\mathbf{p}_n + \mathbf{x}_1 \Longrightarrow \mathbf{l}_n$ by a Π-friendly reaction sequence and $\mathbf{p}_n + \mathbf{x}_2 \Longrightarrow \mathbf{r}_n$ by a Π-friendly reaction sequence where $\mathbf{l}_n + \mathbf{r}_n = \mathbf{y} + 2\mathbf{p}$.

If q_n is not all of p then the next reaction in p after q_n must be of the form $L + R \to \ldots$ where, without loss of generality, $\mathbf{l}_n(R) = 0$, $\mathbf{l}_n(L) > 0$ and $\mathbf{r}_n(L) = 0$, $\mathbf{r}_n(R) > 0$. Since p is Π-friendly, at least one reactant L or R is in Π. Now, for the same p, consider q_{n+1}, the longest parallel prefix of p from $(\mathbf{p}_{n+1} + \mathbf{x}_1, \mathbf{p}_{n+1} + \mathbf{x}_2)$. The following argument shows that q_{n+1} must be longer than q_n by

at least 1 reaction. Suppose $q_{n+1} = q_n$. If $L \in \Pi$ then $\mathbf{r}_{n+1}(L) = 1$, and if $R \in \Pi$ then $\mathbf{l}_{n+1}(R) = 1$. Since we still have $\mathbf{r}_{n+1}(R) > 0$ and $\mathbf{l}_{n+1}(L) > 0$, the above reaction from p can occur in either l_{n+1} or r_{n+1}, contradicting that q_{n+1} is the longest parallel prefix. Using the base case $|q_0| \geq 0$, we conclude that for all $n \geq |p|$, $q_n = p$. □

The following lemma is a key consequence of the parallel decomposition lemma above. In general, a YES voter cannot be produced from an output stable NO state (by definition), while combining *two* output stable NO states *could* lead to the production of a YES voter.[18] The following, however, establishes that this path to producing a YES voter cannot be fast if the output stable NO states come from a pumping sequence for a maximal pumpable set.

Lemma 4.12. *Let Y be an infinite nondecreasing sequence of states, and let Ω be maximal Y-pumpable, with pumping sequence $W = (\mathbf{w}_w)$, where all $\mathbf{w}_w$ are output stable NO states. Let c_W be the bound from Proposition 4.7 applied to pumping sequence W. Then for all $\mathbf{w}_w$, $\mathbf{w}_{w'}$, states $\mathbf{z}$ and paths p such that $\mathbf{w}_w + \mathbf{w}_{w'} \Longrightarrow_p \mathbf{z}$ and $\mathbf{z}$ contains a YES voter, path p is not $2c_W$-fast.*

Proof. We consider two possibilities — that p is Ω-friendly, and that p is not Ω-friendly — and argue that the first is impossible and that the second implies that p is not $2c_W$-fast.

If p is Ω-friendly, then by the parallel decomposition lemma (Lemma 4.11) applied to $\mathbf{w}_w + \mathbf{w}_{w'} \Longrightarrow_p^{\Omega} \mathbf{z}$ there are $\mathbf{p}, \mathbf{p}', \mathbf{p}'' \in \mathbb{N}^{\Omega}$, $\mathbf{z}', \mathbf{z}'' \in \mathbb{N}^{\Lambda}$ such that $\mathbf{p} + \mathbf{w}_w \Longrightarrow \mathbf{p}' + \mathbf{z}'$ and $\mathbf{p} + \mathbf{w}_{w'} \Longrightarrow \mathbf{p}'' + \mathbf{z}''$ and $\mathbf{z}' + \mathbf{z}'' = \mathbf{z}$. Since $\mathbf{z}$ contains a YES voter, $\mathbf{z}'$ or $\mathbf{z}''$ must contain a YES voter. Note that for large enough $\hat{w}$, $\mathbf{w}_{\hat{w}} \geq \mathbf{p} + \mathbf{w}_w$ and $\mathbf{w}_{\hat{w}} \geq \mathbf{p} + \mathbf{w}_{w'}$ since species in Ω have count at least $\hat{w}$ in $\mathbf{w}_{\hat{w}}$. Thus for large enough $\hat{w}$, we can produce a YES voter from $\mathbf{w}_{\hat{w}}$. Since $\mathbf{w}_{\hat{w}}$ is output stable NO by assumption, this is a contradiction.

Thus we conclude that p is not Ω-friendly. Assume p is $2c_W$-fast. Path p must begin with a (possibly empty) Ω-friendly portion, followed by a reaction α that is not Ω-friendly. Let $\mathbf{x}$ be the state immediately before this reaction occurs in p. Since p is $2c_W$-fast, it must be that $\mathbf{x}$ contains count $2c_W$ of some species S that is not in Ω (otherwise α would be Ω-friendly). Since the initial portion of p that leads to $\mathbf{x}$ is Ω-friendly, we have $\mathbf{w}_w + \mathbf{w}_{w'} \Longrightarrow^{\Omega} \mathbf{x}$ and the parallel decomposition lemma (Lemma 4.11) applies: there are $\mathbf{p}, \mathbf{p}', \mathbf{p}'' \in \mathbb{N}^{\Omega}$, $\mathbf{x}', \mathbf{x}'' \in \mathbb{N}^{\Lambda}$ such that $\mathbf{p} + \mathbf{w}_w \Longrightarrow^{\Omega} \mathbf{p}' + \mathbf{x}'$ and $\mathbf{p} + \mathbf{w}_{w'} \Longrightarrow^{\Omega} \mathbf{p}'' + \mathbf{x}''$ and $\mathbf{x}' + \mathbf{x}'' = \mathbf{x}$. Thus either $\mathbf{x}'$ or $\mathbf{x}''$ must contain at least $2c_W/2 = c_W$ of S. Since for large enough $\hat{w}$, $\mathbf{w}_{\hat{w}} \geq \mathbf{p} + \mathbf{w}_w$ and $\mathbf{w}_{\hat{w}} \geq \mathbf{p} + \mathbf{w}_{w'}$, starting from $\mathbf{w}_{\hat{w}}$ we can reach a state containing at least c_W of $S \in \Lambda \setminus \Omega$ by an Ω-friendly path. In this we obtain a contradiction of Proposition 4.7. □

4.2.3 Simplified proof for leaderless CRDs

In this section, we show a simplified version of our main negative result that speed fault free CRDs stably decide only detection predicates. Its proof illustrates how the tools developed in previous subsections can be used to show that certain CRDs are not speed fault free, without requiring the full technical detail required for our main negative result. The lemma below shows that *leaderless* CRDs (those with initial context $\mathbf{0}$) that are speed fault free decide only predicates $\psi : \mathbb{N}^k \to \{0, 1\}$ that are closed under doubling, i.e., for all $\mathbf{x} \in \mathbb{N}^k$, $\psi(\mathbf{x}) = \psi(2\mathbf{x})$ (note all detection predicates are closed under doubling). The lemma immediately implies that any leaderless CRD stably deciding the "$2A$ predicate" of Figure 1 ($\psi(x) = 1 \iff x \geq 2$) is not speed fault free.

Intuitively, the proof of the lemma below involves splitting the initial state representing $2\mathbf{x}$ into two equal tubes each representing $\mathbf{x}$ such that each tube reaches an output stable state with no

[18] For example, consider the reaction $A + A \to Y$, with A voting NO and Y voting YES, and consider the states $\{1A\}$ and $\{2A\}$.

$\psi(2\mathbf{x})$ voters. Then if the CRD were speed fault free, once the two tubes are allowed to interact, there should be an f-fast path to producing a $\psi(2\mathbf{x})$ voter. By initially choosing more fuel molecules, we should be able to find such a path for arbitrarily large f. Then we use Lemma 4.12 which gives an upper bound on how large f can be, a contradiction.

Lemma 4.13. *Let $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon, F, \phi, \mathbf{s})$ be a fueled CRD with $\mathbf{s} = \mathbf{0}$ stably deciding a predicate $\psi : \mathbb{N}^k \to \{0, 1\}$ such that for some $\mathbf{x} \in \mathbb{N}^k$, $\psi(\mathbf{x}) \neq \psi(2\mathbf{x})$. Then $\mathcal{D}$ is not speed fault free.*

Proof. Suppose $\mathcal{D}$ stably decides ψ. Let $\mathbf{x} = (x_1, \ldots, x_k)$ be as in the statement of the lemma. Define $N = (\mathbf{n}_n)$ for all $n \in \mathbb{N}$ by $\mathbf{n}_n = \mathbf{n}_n(\mathbf{x})$ as in Definition 4.1; i.e., counts $x_1, \ldots, x_k$ of $A_1, \ldots, A_k$, count n of F, and count 0 of all other species. Since $\mathcal{D}$ stably decides ψ, for each $n \in \mathbb{N}$, there is an output stable state $\mathbf{y}_n$ such that $\mathbf{n}_n \Longrightarrow \mathbf{y}_n$ and $\Phi(\mathbf{y}_n) = \psi(\mathbf{x})$; assume without loss of generality that $\psi(\mathbf{x}) = 0$. Define sequence $Y = (\mathbf{y}_n)$. Let $\Omega \subseteq \Lambda$ be a maximal Y-pumpable set of species with pumping sequence $W = (\mathbf{w}_w)$.

For each $n \in \mathbb{N}$, consider the initial state $2\mathbf{n}_n = \mathbf{n}_{2n}(2\mathbf{x})$; this is a valid initial state representing input $2\mathbf{x}$, for which $\psi(2\mathbf{x}) \neq \psi(\mathbf{x})$. By the above argument, for each $w \in \mathbb{N}$, there is an $n \in \mathbb{N}$ such that $2\mathbf{n}_n \Longrightarrow 2\mathbf{w}_w$. By the fact that $\psi(\mathbf{x}) \neq \psi(2\mathbf{x})$ and the fact that $\mathcal{D}$ stably decides ψ, for each w there is an output stable state $\mathbf{z}_w$ such that $2\mathbf{w}_w \Longrightarrow_{q_w} \mathbf{z}_w$ and $\mathbf{z}_w$ contains a YES voter (since each $\mathbf{w}_w$ is output stable NO, the state $2\mathbf{w}_w$ has no YES voters, and the CRD must produce one before it can reach an output stable YES state). By Lemma 4.12, no such path q_w can be $2c_W$-fast. Recall that $N \Longrightarrow Y \Longrightarrow W$ implies that $n \to \infty$ (index of N and Y) as $w \to \infty$ (index of W). Thus, there are infinitely many n such that from initial state $2\mathbf{n}_n$ the CRD can reach to a state $2\mathbf{w}_w$, from which every path to a correct output stable state is not $2c_W$-fast, implying $\mathcal{D}$ is not speed fault free. □

Remark 4.2. Recall that we have defined CRDs to use the symmetric output convention with both NO and YES voters, while the examples in the introduction (Fig. 1) adhere to an asymmetric output convention where YES output is represented by the presence of Y and NO output is represented by its absence. In the asymmetric output convention, the proof above shows that if $\psi(\mathbf{x}) = 0$ while $\psi(2\mathbf{x}) = 1$, then the CRD is not speed fault free. Thus the above lemma is sufficient to prove that no speed fault free CRD, without an initial context, can compute the "2A predicate" as described in the introduction.

Remark 4.3. Although, Lemma 4.13 states merely that the CRD is not speed fault free, the proof actually shows a stronger conclusion. That the CRD is not speed fault free means that for some input there are reachable states such that every path to an output stable state is "slow" — i.e., that stabilization is "slow". However, the proof shows that there are reachable states such that every path to the correct output, stable or not, is "slow" — i.e., convergence must be slow as well.

In the following sections we work to remove the restriction to leaderless CRDs and to extend the above lemma to a larger variety of predicates. Note that the proof of Lemma 4.13 appeals to Lemma 4.12, which requires the two states $\mathbf{u}_u, \mathbf{u}_{u'}$ to be from the same pumping sequence. This highlights a central difficulty of extending the above proof to cover a non-zero initial context. If the initial context contains, for example, a leader (a single copy of a certain species L), then it must go into one or the other tube but not both, and the set of pumpable species might be different depending on the presence or absence of the leader.

Even without a leader, consider the non-detection predicate $\psi(x_1, x_2) = 1 \iff x_1 = x_2$, which is closed under doubling since $x_1 = x_2 \iff 2x_1 = 2x_2$; we cannot apply the above proof on ψ since the it requires that ψ not to be closed under doubling.

The remainder of the paper develops the technical tools needed to handle these difficulties. Though we still make heavy use of the parallel decomposition lemma, we apply it to more carefully chosen states.

4.2.4 Reaction ordering lemma

Intuitively, the next lemma states that a "fast" reaction sequence that decreases certain species from large counts to small counts must contain reactions of a certain restricted form. In particular the form is as follows: if Δ is the set of species whose counts decrease from large to small, then we can write the species in Δ in some order $D_1, D_2, \ldots, D_l$, such that for each $1 \leq i \leq l$, there is a reaction α_i that consumes D_i, and every other species involved in α_i is either not in Δ, or comes later in the ordering. These reactions will be used in the proof of the main result to do controlled "surgery" on fast reaction sequences, because they give a way to alter the count of D_i by inserting or removing the reactions α_i, knowing that this will not affect the counts of $D_1, \ldots, D_{i-1}$. Specifically, the reaction ordering lemma is key to proving the Π perturbation claim (Claim 1) in the main argument.

Lemma 4.14. *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. Define $\Delta = \{\ D \in \Lambda \mid \mathbf{x}(D) \geq c_2$ and $\mathbf{y}(D) \leq c_1\ \}$. 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 $D_i + S \to P_1 + \ldots + P_k$, 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 $\mathbf{c}(S) \geq c_2$.*

Proof. We define the ordering based on increasing sets $\varnothing = \Delta_0 \subset \Delta_1 \subset \Delta_2 \subset \ldots \Delta_{l-1} \subset \Delta_l = \Delta$, where for each $1 \leq i \leq l$, $\Delta_i \setminus \Delta_{i-1} = \{D_i\}$.

We define the ordering inductively "in reverse," by first defining D_l, then D_{l-1}, etc. For all $1 \leq i \leq l$, define $\Theta_i : \mathbb{N}^\Lambda \to \mathbb{N}$ for all states $\mathbf{c}$ by $\Theta_i(\mathbf{c}) = \sum_{D \in \Delta_i} \mathbf{c}(D)$. Θ_l is well-defined since $\Delta_l = \Delta$, and Θ_i is well-defined once we have defined $D_{i+1}, \ldots, D_l$, because $\Delta_i = \Delta \setminus \{D_{i+1}, \ldots, D_l\}$.

Because $\mathbf{y}(D) \leq c_1$ for all $D \in \Delta$, it follows that $\Theta_i(\mathbf{y}) \leq i \cdot c_1 \leq |\Lambda| \cdot c_1$. Recall that $\mathbf{x}(D) \geq c_2$ for all $D \in \Delta$. Let r be the suffix of q after the last state $\mathbf{c}'$ along q such that $\Theta_i(\mathbf{c}') \geq c_2$. Then in all states $\mathbf{c}$ in r (not including $\mathbf{c}'$ itself), $\mathbf{c}(D) < c_2$ for all $D \in \Delta_i$. Because $\Theta_i(\mathbf{c}') \geq c_2$, while $\Theta_i(\mathbf{y}) \leq |\Lambda| \cdot c_1$, and $c_2 > |\Lambda| \cdot c_1$, r must contain a subsequence s of reactions, each of which strictly decreases Θ_i, and the total decrease in Θ_i over all of s is at least $(c_2 - |\Lambda| \cdot c_1)$ between states $\mathbf{c}'$ and $\mathbf{y}$.

We now examine the form of any reaction in s. Since every reaction in s strictly decreases Θ_i, the reaction must have a reactant in Δ_i. Since s is c_2-fast, and all states $\mathbf{c}$ along s have $\mathbf{c}(D) < c_2$ for $D \in \Delta_i$, the reaction cannot be unimolecular since the count of D is too low for the reaction to be c_2-fast. So the reaction must be bimolecular with the other reactant S having count at least c_2. This implies $S \notin \Delta_i$ (since all $D \in \Delta_i$ have count $< c_2$ between $\mathbf{c}'$ and $\mathbf{y}$). For the reaction to strictly decrease Θ_i, all products $P \notin \Delta_i$ (otherwise Θ_i would either stay equal or increase after applying the reaction). In fact, this implies every reaction in s decreases Θ_i by exactly 1. Since there are at least $c_2 - |\Lambda| \cdot c_1$ instances of such reactions in s, and there are at most $|R|$ total types of reactions, by the pigeonhole principle at least one reaction type must repeat in s at least $\frac{c_2 - |\Lambda| \cdot c_1}{|R|}$ times. We call D_i the reactant of this reaction that is in Δ_i, and continue in the same manner to define Δ_{i-1}, D_{i-1}, etc. □

4.2.5 Proof of the full negative result

Throughout this subsection, let $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon, F, \phi, \mathbf{s})$ be an arbitrary speed fault free fueled CRD with $\Sigma = \{A_1, \ldots, A_k\}$. Supposing for the sake of contradiction that $\mathcal{D}$ decides some non-detection

predicate $\psi : \mathbb{N}^k \to \{0,1\}$, then there must exist some index i (assume without loss of generality that $i = 1$), and an input value $(x_1, x_2, \ldots, x_k) \in \mathbb{N}^k$, where $x_1 \geq 1$, such that $\psi(x_1, x_2, \ldots, x_k) \neq \psi(x_1 + 1, x_2, \ldots, x_k)$. Assume without loss of generality that $\psi(x_1, x_2, \ldots, x_k) = 0$.

Definition of sequence N. For each $n \in \mathbb{N}$, write the initial state of $\mathcal{D}$ with answer NO and n copies of F as $\mathbf{n}_n$ $(= \{x_1 A_1,\ x_2 A_2,\ \ldots,\ x_k A_k,\ nF\} + \mathbf{s}$ (where $\mathbf{s}$ is a fixed initial context)). Let $N = (\mathbf{n}_0, \mathbf{n}_1, \ldots)$ be the (increasing) sequence of these initial states.

The high level intuition behind our proof is as follows. First we start with x_1 copies of A_1 (i.e., infinite sequence $N = (\mathbf{n}_n)$ for all n) and pump a maximal set of species Π (pumping sequence U). Recall that the input defining all of these has answer NO, but that a single extra copy of A_1 implies that the answer should be YES. By the speed fault free assumption there is an f-fast path, and thus a Π-friendly path, to output stable NO states Y. Thus the A_1 "communicates" with the rest of the CRD only via perturbations to these high count species Π. The Π-perturbation claim (Claim 1, proven in Subsection 4.2.6) relies on the reaction ordering lemma on fast paths to show that a perturbation on Π can be effectively "neutralized". We force an additional copy of A_1 to undergo the same path as a previous copy, and then neutralize its effect on Π. Then we force the CRD to go to a state that is contained in the sum of two output stable NO states from the same pumping sequence W (for a maximal Y-pumpable set Ω). Then by Lemma 4.12 we conclude that the path to producing a YES voter from this state cannot be fast, a contradiction.

We now formally define these and other infinite sequences of states and sets of species. For clarity of exposition we first define U and Y in a more unconstrained way, but then refine the construction to satisfy additional constraints in the subsection to follow.

Definition of sequences U, Y, W, and sets Π, Γ, Ω. Let Π be maximal N-pumpable with pumping sequence $U = (\mathbf{u}_u)$. Recall that by definition $\mathbf{n}_n$ contains n molecules of the fuel species F. Because every $\mathbf{u}_u$ is reachable from some $\mathbf{n}_n$ where $n \to \infty$ as $u \to \infty$, if the CRD is speed fault free then each $\mathbf{u}_u$ must be able to reach some output stable NO state by an f-fast path, where $f \to \infty$ as $u \to \infty$. Consider only the output stable NO states with $f \geq c_U$, the bound from Proposition 4.7, and let $Y = (\mathbf{y}_y)$ be an infinite non-decreasing subsequence of these states (using Lemma 3.2). Let Γ be the species unbounded in Y (i.e., $S \in \Gamma$ iff $(\forall c \exists y)\ \mathbf{y}_y(S) > c$). Note by Proposition 4.8, $U \Longrightarrow^{\Pi} Y$. Let Ω be maximal Y-pumpable with pumping sequence $W = (\mathbf{w}_w)$. By Proposition 4.6, $\Omega \subseteq \Pi$.

By the above definition for all w, there are y, u, and n such that $\mathbf{n}_n \Longrightarrow^{\Pi} \mathbf{u}_u \Longrightarrow^{\Pi} \mathbf{y}_y \Longrightarrow^{\Omega} \mathbf{w}_w$, where y depends on w, u depends on y, and n depends on u, where $n \to \infty$ as u, y, or $w \to \infty$. Letting n_0 be the index of N corresponding to $\mathbf{w}_0$, and recalling that $\Omega \subseteq \Pi$, we can write the complete path as $\mathbf{n}_{n_0} \Longrightarrow^{\Pi} \mathbf{w}_0$. We can equivalently rewrite this path as: $(\mathbf{n}_{n_0} \setminus \{A_1\}) + \{A_1\} \Longrightarrow^{\Pi} \mathbf{w}_0$. Now by the parallel decomposition lemma (Lemma 4.11), there are $\mathbf{p}, \mathbf{p}' \in \mathbb{N}^{\Pi}$ and a reaction sequence p such that $\mathbf{p} + \{A_1\} \Longrightarrow_p \mathbf{p}' + \mathbf{w}$ where $\mathbf{w} \leq \mathbf{w}_0$ (letting $\mathbf{x}_1 = \{A_1\}$ and $\mathbf{x}_2 = \mathbf{n}_{n_0} \setminus \{A_1\}$). Since U is a pumping sequence for Π, for all large enough u, $\mathbf{u}_u \geq \mathbf{p}$, and so $\mathbf{u}_u + \{A_1\} \geq \mathbf{p} + \{A_1\}$. Thus for all large enough u, $\mathbf{u}_u + \{A_1\} \Longrightarrow_p \mathbf{u}_u + \mathbf{e} + \mathbf{w}$, where $\mathbf{e} = \mathbf{p}' - \mathbf{p} \in \mathbb{Z}^{\Pi}$ (because $\mathbf{u}_u + \{A_1\} = (\mathbf{u}_u - \mathbf{p}) + \mathbf{p} + \{A_1\} \Longrightarrow_p (\mathbf{u}_u - \mathbf{p}) + \mathbf{p}' + \mathbf{w} = \mathbf{u}_u + \mathbf{e} + \mathbf{w}$).

In the subsection to follow we will show that U and Y can be chosen to satisfy the following "Π-perturbation" claim. The additional constraints that must be imposed upon U and Y, and how to satisfy them, are postponed until the next subsection for clarity of exposition. The Π-perturbation claim intuitively says that we can absorb a perturbation over Π, into a perturbation over Γ. Then, the perturbed $\mathbf{y}_y$ is still bounded above by some other output stable NO state $\mathbf{y}_{y'}$, and thus a YES voter cannot be reached. We will use this claim to drive the CRD from $\mathbf{u}_u + \mathbf{e} + \mathbf{w}$ to a state from which a YES voter cannot be reached, a contradiction.

Claim 1 (Π **Perturbation Claim**). *For all* $\mathbf{e} \in \mathbb{Z}^\Pi$, *there is a* $y_0 \in \mathbb{N}$, *such that for all* $y \geq y_0$, *there are infinitely many* $u \in \mathbb{N}$ *and* $\mathbf{g}_u \in \mathbb{N}^\Gamma$ *such that* $\mathbf{u}_u + \mathbf{e} \Longrightarrow \mathbf{y}_y + \mathbf{g}_u$.

The claim implies that for all large enough y, there are infinitely many $u \in \mathbb{N}$ and corresponding $\mathbf{g}_u \in \mathbb{N}^\Gamma$, such that $\mathbf{u}_u + \{A_1\} \Longrightarrow \mathbf{y}_y + \mathbf{g}_u + \mathbf{w}$. Choose $y \geq y_0$ (constant from Claim 1) and $w_0 \in \mathbb{N}$ such that $\mathbf{y}_y \Longrightarrow \mathbf{w}_{w_0}$ (i.e., find a large enough y that has a corresponding $\mathbf{w}_w$, whose index we will call w_0). Then, for infinitely many $u \in \mathbb{N}$, $\mathbf{u}_u + \{A_1\} \Longrightarrow \mathbf{w}_{w_0} + \mathbf{g}_u + \mathbf{w}$.

Recall that for all u, $\mathbf{u}_u$ is reachable from some $\mathbf{n}_n$ such that $n \to \infty$ as $u \to \infty$. Recall $\mathbf{n}_n$ contains n fuel molecules F, and that initial state $\mathbf{n}_n + \{A_1\}$ should result in answer YES. Thus, assuming the CRD is speed fault free, there must be an f-fast path $\mathbf{w}_{w_0} + \mathbf{g}_u + \mathbf{w} \Longrightarrow_{q_u} \mathbf{z}_u$ where $\mathbf{z}_u$ contains a YES voter, and $f \to \infty$ as $u \to \infty$. Choose u large enough that $f \geq 2c_W$, where c_W is the bound from Proposition 4.7 applied on the maximal Y-pumpable set Ω with pumping sequence W. Next, observe that $\Gamma \subseteq \Omega$; this follows by Proposition 4.5 taking (left side variables referring to the statement of the proposition and right side variables referring to definitions in the present context) $\Pi = \Omega$, $N = Y$, and $W = U$. Thus $\mathbf{g}_u \in \mathbb{N}^\Omega$. To summarize the above, $\mathbf{w}_{w_0} + \mathbf{g}_u + \mathbf{w} \Longrightarrow_{q_u} \mathbf{z}_u$ where $\mathbf{z}_u$ contains a YES voter and q_u is $2c_W$-fast, and $\mathbf{g}_u \in \mathbb{N}^\Omega$.

Recall that $W = (\mathbf{w}_w)$ is a pumping sequence for Ω and thus it is nondecreasing, and $\mathbf{w}_w$ contains at least count w of all species in Ω. Then, since $\mathbf{g}_u \in \mathbb{N}^\Omega$, we can find a large enough $\hat{w}$ such that $\mathbf{w}_{\hat{w}} \geq \mathbf{w}_{w_0} + \mathbf{g}_u$. Further recall that $\mathbf{w}_0 \geq \mathbf{w}$. Thus applying path q_u to $\mathbf{w}_{\hat{w}} + \mathbf{w}_0$ leads to a superset of $\mathbf{z}_u$. In other words, we can produce a YES voter from $\mathbf{w}_{\hat{w}} + \mathbf{w}_0$ via a $2c_W$-fast path. Recall that all states in W are output stable NO (since they are reachable from Y). Thus we obtain a contradiction by Lemma 4.12 with the maximal Y-pumpable set Ω and pumping sequence W.

Since we assumed at the start of the subsection that $\mathcal{D}$ was an arbitrary speed fault free CRD deciding a non-detection predicate, this contradiction implies the main technical result of this paper, which together with Lemma 4.4 implies our main theorem, Theorem 4.3.

Lemma 4.15. *Speed-fault free CRDs decide only detection predicates.*

4.2.6 Proving the Π perturbation claim (Claim 1)

First we prove an intermediate lemma that will be useful in proving Claim 1. Intuitively, it shows that the path $\mathbf{u}_u \Longrightarrow \mathbf{y}_y$ can be modified to convert a positive or negative "perturbation" of the large count species in U ($\mathbf{e} \in \mathbb{Z}^\Pi$) to a perturbation of the large count species in Y ($\mathbf{e}^\Gamma \in \mathbb{Z}^\Gamma$). However, the modified path is only valid in the "context" of an excess amount of species in Π (i.e., $\mathbf{p} \in \mathbb{N}^\Pi$).

Lemma 4.16. *Let* N *be an infinite nondecreasing sequence of states, and let* Π *be maximal* N*-pumpable, with pumping sequence* $U = (\mathbf{u}_u)$. *Let* Y *be an infinite sequence of states such that for all* $y \in \mathbb{N}$, $\mathbf{u}_u \Longrightarrow \mathbf{y}_y$ *by an* f*-fast path, and* $u, f \to \infty$ *as* $y \to \infty$. *Let* Γ *be the set of species unbounded in* Y. *Then for any* $\mathbf{e} \in \mathbb{Z}^\Pi$, *there is* $\mathbf{p} \in \mathbb{N}^\Pi, \mathbf{e}^\Gamma \in \mathbb{Z}^\Gamma$ *and there is a* $y_0 \in \mathbb{N}$, *such that for all* $y \geq y_0$, $\mathbf{p} + \mathbf{u}_u + \mathbf{e} \Longrightarrow \mathbf{p} + \mathbf{y}_y + \mathbf{e}^\Gamma$.

Proof. For any y, there is u and an f-fast path p such that $\mathbf{u}_u \Longrightarrow_p \mathbf{y}_y$ and $u, f \to \infty$ as $y \to \infty$. Let c_1 be the largest count of any species not in Γ in the sequence Y. Apply the reaction ordering lemma (Lemma 4.14) on path p for large enough y such that $c_2 = \min\{u, f\}$ satisfies $c_2 > |\Lambda| \cdot c_1$ and $c_2 \geq c_U$, where c_U is the bound from Proposition 4.7.

Recall $(\mathbf{u}_u)$ is a pumping sequence for maximal Π and thus for all $S \in \Pi$, $\mathbf{u}_u(S) \geq u$, while all other species are bounded (by Proposition 4.7). So for large enough c_2 (obtained from a large

enough y), since $c_2 \leq u$, $\forall S \in \Pi$, $\mathbf{u}_u(S) \geq c_2$ while $\forall S \in \Lambda \setminus \Pi$, $\mathbf{u}_u(S) < c_2$. This implies that $\Pi \setminus \Gamma = \{\ D \in \Lambda \mid \mathbf{u}_u(D) \geq c_2 \text{ and } \mathbf{y}_y(D) \leq c_1\ \}$. Define $\Delta = \Pi \setminus \Gamma$.

Using the reaction ordering lemma, we can write $\Delta = \{D_1, \ldots, D_l\}$, such that for each $1 \leq i \leq l$, there is a bimolecular reaction α_i with the following properties. (That properties (2) and (3) follow from the reaction ordering lemma is shown below.) (1) D_i is a reactant. (2) All products are either elements of Γ, or are D_j for $j > i$. (3) The other reactant (S) is either an element of Γ, or is D_j for $j > i$. (4) α_i occurs at least $(c_2 - |\Lambda| \cdot c_1)/|R|$ times in p.

Property (3) can be shown as follows. By Proposition 4.8, the other reactant S must be in Π because Π is maximal pumpable and the count of S is at least c_2 (by the reaction ordering lemma) which is at least c_U (the bound of Proposition 4.7, by construction of c_2). Since S is not one of $D_1, \ldots, D_i$ by the reaction ordering lemma, it must be either in Γ or D_j for $j > i$. Property (2) follows using Proposition 4.9 since both reactants are in Π as just established.

We now describe how to modify the reaction sequence p by inserting or removing the above reactions to get rid of the Δ component of $\mathbf{e}$. However, as a result of the modification, if we simply start in $\mathbf{u}_n + \mathbf{e}$, we may not be able to complete the modified path p because some species in Π might go negative. However, for large enough $\mathbf{p} \in \mathbb{N}^{\Pi}$, $\mathbf{p} + \mathbf{u}_u + \mathbf{e} \Longrightarrow \mathbf{p} + \mathbf{y}_y + \mathbf{e}^{\Gamma}$ where $\mathbf{e}^{\Gamma} \in \mathbb{Z}^{\Gamma}$.

We iteratively fix the counts of species in Δ one by one, in the ordering given, i.e., we first adjust p to fix D_1, then we fix D_2 (while showing that the fixing of D_2 cannot affect the count of D_1 in any state, so it remains fixed), etc. We start with $\mathbf{e}_0 = \mathbf{e}$. Having fixed $D_1, \ldots, D_{i-1}$, and obtaining new $\mathbf{e}_{i-1} \in \mathbb{Z}^{\Pi}$ such that $\mathbf{e}_{i-1}$ is zero on $D_1, \ldots, D_{i-1}$, we process D_i as follows. If $\delta_i = \mathbf{e}_{i-1}(D_i) > 0$: add δ_i instances of reaction α_i at the end of the reaction sequence. If $\delta_i < 0$: remove δ_i instances of α_i where they occur in the reaction sequence; property (4) ensures that p contains enough instances of α_i (see below). In this way we obtain $\mathbf{e}_i$. By property (2) and (3), adding or removing instances of α_i affects only the counts of species in Γ and $D_{i+1}, \ldots, D_l$. Since we fix these counts in the prescribed order, when we are done, the counts of each D_i is zero in $\mathbf{e}^{\Gamma} = \mathbf{e}_k$, while the counts of elements of Γ may have been altered (upward or downward). Note that as we fix D_i by adding or removing α_i, we are affecting the counts of D_j for $j > i$ and Γ. Although the counts of D_j for $j > i$ are compensated later, they may temporarily dip below zero had we not added a large enough $\mathbf{p} \in \mathbb{N}^{\Pi}$. Further, counts of Γ are never fixed, and thus $\mathbf{p}$ must be large enough that $\mathbf{p} + \mathbf{y}_y + \mathbf{e}^{\Gamma}$ is non-negative on Γ.

Finally, we derive a bound on the number of reaction instances that we may need to remove, which places a bound on c_2 to ensure that there are enough instances by property (4) above.

Bound on the amount of fixing: Let $c_{\mathrm{b}} = |\mathbf{e}|$, and let c_{s} be the maximum stoichiometric coefficient (which bounds the amount that species can change each time the reaction is added or removed). We add or remove at most $|\delta_1| \leq c_{\mathrm{b}}$ instances of α_1, which affects the count of $D_2, \ldots, D_l$ and species in Γ by at most $c_{\mathrm{b}}c_{\mathrm{s}}$. Thus, $|\delta_2| \leq c_{\mathrm{b}} + |\delta_1|c_{\mathrm{s}} \leq c_{\mathrm{b}}(1 + c_{\mathrm{s}})$ (the original c_{b} error plus the additional error from altering the number of α_1 reactions). In general, $|\delta_i| \leq c_{\mathrm{b}} + (|\delta_1| + \cdots + |\delta_{i-1}|)c_{\mathrm{s}} \leq (1 + c_{\mathrm{s}})^{i-1}c_{\mathrm{b}}$. Thus if we let $c_2 \geq |\Lambda| \cdot c_1 + (1 + c_{\mathrm{s}})^{l-1}c_{\mathrm{b}}|R|$ (where $l = |\Delta|$ is the upper bound on i), we will have enough reaction instances by property (4) to remove (since $(c_2 - |\Lambda| \cdot c_1)/|R| \geq (1 + c_{\mathrm{s}})^{l-1}c_{\mathrm{b}}$). Note that this bound, and thus the values of $\mathbf{p}$ and $\mathbf{e}^{\Gamma}$, depend only on $\mathbf{e}$. Since $c_2 = \min\{u, f\}$ and u, f are arbitrarily large, we can achieve a sufficiently large value of c_2. □

In order to prove Claim 1, we need to be more deliberate in constructing sequences $U = (\mathbf{u}_u)$ and $Y = (\mathbf{y}_y)$, such that two additional constraints are satisfied. Intuitively, these constraints give a way to repeatedly convert some amount of Π species into some amount of Γ species without changing anything else. Eventually these constraints will allow us to replace $\mathbf{p} + \mathbf{u}_u$ in Lemma 4.16 with $\mathbf{u}_{u'}$ for a larger index u', and ensure that the excess of species in Π can be consumed after

Lemma 4.16 is applied.

Recall Γ is the set of species that grow unbounded in sequence Y.

Constraint 1. There are $\mathbf{u} \in \mathbb{N}^{\Lambda}$ and $\mathbf{d} \in \mathbb{N}^{\Pi}$ where $\forall S \in \Pi$, $\mathbf{d}(S) > 0$, such that $\mathbf{u}_u = \mathbf{u} + u\mathbf{d}$.
Constraint 2. There are y_1, k, and $\mathbf{g} \in \mathbb{N}^{\Gamma}$ where $\forall S \in \Gamma, \mathbf{g}(S) > 0$, such that $\mathbf{y}_{y_1} + k\mathbf{d} \Longrightarrow \mathbf{y}_{y_1} + \mathbf{g}$.

Refined construction of sequences U and Y. U can be constructed to satisfy constraint (1) by following Lemma 4.10 with $\mathbf{h} = \{1F\}$ (since $\mathbf{n}_{n+1} = \mathbf{n}_n + \{1F\}$ by construction). To satisfy constraint (2) we construct sequence Y as follows. First infinite sequence Y' is constructed inductively: Let $\mathbf{y}'_0$ be some output stable NO state reached from $\mathbf{u}_0$. Let $\mathbf{y}'_y$ be an output stable NO state reached from $\mathbf{y}'_{y-1} + \mathbf{d}$ (call this path p_y). Since $\mathbf{y}'_{y-1} + \mathbf{d}$ is reachable from $\mathbf{u}_y$, which in turn is reachable from some $\mathbf{n}_n$ such that $n \to \infty$ as $y \to \infty$, we can ensure that paths p_y are f-fast such that $f \to \infty$ as $y \to \infty$ (assuming the CRD is speed fault free). Finally, let $Y = (\mathbf{y}_y)$ be an infinite non-decreasing subsequence of Y' (invoking Lemma 3.2) restricted to the states $\mathbf{y}'_y$ for which p_y is at least c_U-fast for bound c_U from Proposition 4.7 (for the U constructed above).

We now show that Y constructed in this manner satisfies constraint (2). There must be $\mathbf{y}_{y_0}$ and $\mathbf{y}_{y_1}$ in Y such that for all species $S \in \Lambda \setminus \Gamma$, $\mathbf{y}_{y_0}(S) = \mathbf{y}_{y_1}(S)$, and for all species $S \in \Gamma$, $\mathbf{y}_{y_0}(S) < \mathbf{y}_{y_1}(S)$. Then observe that for some k, $\mathbf{y}_{y_0} + k\mathbf{d} \Longrightarrow \mathbf{y}_{y_1} = \mathbf{y}_{y_0} + \mathbf{g}$ where $\mathbf{g} = \mathbf{y}_{y_1} - \mathbf{y}_{y_0} \in \mathbb{N}^{\Gamma}$, which is positive over all of Γ.

With the two additional constraints on $U = (\mathbf{u}_u)$ and $Y = (\mathbf{y}_y)$ described above we are able to prove the Π-perturbation claim used in the previous subsection.

Claim 1. (Π **Perturbation Claim**) *For all $\mathbf{e} \in \mathbb{Z}^{\Pi}$, there is a $y_0 \in \mathbb{N}$, such that for all $y \geq y_0$, there are infinitely many $u \in \mathbb{N}$ and $\mathbf{g}_u \in \mathbb{N}^{\Gamma}$ such that $\mathbf{u}_u + \mathbf{e} \Longrightarrow \mathbf{y}_y + \mathbf{g}_u$.*

Proof. Given, any $\mathbf{e} \in \mathbb{Z}^{\Pi}$, let y be at least as large as y_0 from Lemma 4.16 and y_1 from constraint (2). By Lemma 4.16, there is u' and $\mathbf{p} \in \mathbb{N}^{\Pi}, \mathbf{e}^{\Gamma} \in \mathbb{Z}^{\Gamma}$ such that $\mathbf{p} + \mathbf{u}_{u'} + \mathbf{e} \Longrightarrow_{p'} \mathbf{p} + \mathbf{y}_y + \mathbf{e}^{\Gamma}$. Suppose that instead of $\mathbf{p} + \mathbf{u}_{u'} + \mathbf{e}$, using the constant k from constraint (2), we let $u = u' + ku''$ for some u'' to be chosen later, and start with $\mathbf{u}_u + \mathbf{e} = ku''\mathbf{d} + \mathbf{u}_{u'} + \mathbf{e}$ (by constraint (1)) such that $ku''\mathbf{d} \geq \mathbf{p}$ (which is true for large enough u''). Then by the same path p' we have: $ku''\mathbf{d} + \mathbf{u}_{u'} + \mathbf{e} \Longrightarrow_{p'} ku''\mathbf{d} + \mathbf{y}_y + \mathbf{e}^{\Gamma}$.

Let r be the path described in constraint (2). We want to use this path u'' times to convert $ku''\mathbf{d} \in \mathbb{N}^{\Pi}$ to $u''\mathbf{g} \in \mathbb{N}^{\Gamma}$ starting from $ku''\mathbf{d} + \mathbf{y}_y + \mathbf{e}^{\Gamma}$. But $\mathbf{e}^{\Gamma} \in \mathbb{Z}^{\Gamma}$ can be negative — how can we be sure that we don't go below zero when taking r multiple times? We use the following argument, which depends on making u'' large enough as a function of $\mathbf{e}^{\Gamma}$: In the first $u''/2$ applications of path r we rely on the remaining $(ku''/2)\mathbf{d}$ to remain non-negative. Then, in the second $u''/2$ applications of r, we rely on the $(u''/2)\mathbf{g}$ produced in the first half to remain non-negative. More precisely, for large enough u'', $(ku''/2)\mathbf{d} + \mathbf{e}^{\Gamma} \geq 0$, and so we can take the path $ku''\mathbf{d} + \mathbf{y}_y + \mathbf{e}^{\Gamma} \Longrightarrow (ku''/2)\mathbf{d} + \mathbf{y}_y + (u''/2)\mathbf{g} + \mathbf{e}^{\Gamma}$ by repeating path r $u''/2$ times. Here, $\mathbf{g} \in \mathbb{N}^{\Gamma}$ such that $\forall S \in \Gamma, \mathbf{g}(S) > 0$. Thus, $(u''/2)\mathbf{g}$ can be arbitrarily large on Γ if we use a large enough u''. Therefore, for sufficiently large u'', $(u''/2)\mathbf{g} + \mathbf{e}^{\Gamma} \geq 0$. Then, by taking path r another $u''/2$ times, $(ku''/2)\mathbf{d} + \mathbf{y}_y + (u''/2)\mathbf{g} + \mathbf{e}^{\Gamma} \Longrightarrow (u''/2)\mathbf{g} + \mathbf{y}_y + (u''/2)\mathbf{g} + \mathbf{e}^{\Gamma} = \mathbf{y}_y + u''\mathbf{g} + \mathbf{e}^{\Gamma}$. We obtain the statement of the claim with $u = u' + ku''$ for all large enough u'', and $\mathbf{g}_u = u''\mathbf{g} + \mathbf{e}^{\Gamma} \in \mathbb{N}^{\Gamma}$. □

Acknowledgements. We thank Damien Woods, Anne Condon, Chris Thachuk, Bonnie Kirkpatrick, Monir Hajiaghayi, and Ján Maňuch for useful discussions. We are also grateful to anonymous reviewers for pointing out a number of issues in the original version of this manuscript.

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] Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, July 2008.

[5] Dana Angluin, James Aspnes, Michael J Fischer, and Hong Jiang. Self-stabilizing population protocols. In *Principles of Distributed Systems*, pages 103–117. Springer, 2006.

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

[7] Luca Cardelli. Morphisms of reaction networks that couple structure to function. *BMC systems biology*, 8(1):84, 2014.

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

[9] 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.

[10] Ho-Lin Chen, Rachel Cummings, David Doty, and David Soloveichik. Speed faults in computation by chemical reaction networks. In Fabian Kuhn, editor, *DISC 2014: Proceedings of the 28th International Symposium on Distributed Computing, Austin, TX, USA, October 12-15, 2014*, volume 8784 of *Lecture Notes in Computer Science*, pages 16–30. Springer Berlin Heidelberg, 2014.

[11] Ho-Lin Chen, David Doty, and David Soloveichik. Deterministic function computation with chemical reaction networks. *Natural computing*, 13(4):517–534, 2014.

[12] 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.

[13] 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.

[14] Matthew Cook, David Soloveichik, Erik Winfree, and Jehoshua Bruck. Programmability of chemical reaction networks. In Anne Condon, David Harel, Joost N. Kok, Arto Salomaa, and Erik Winfree, editors, *Algorithmic Bioprocesses*, pages 543–584. Springer Berlin Heidelberg, 2009.

[15] Rachel Cummings, David Doty, and David Soloveichik. Probability 1 computation with chemical reaction networks. In *DNA Computing and Molecular Programming*, pages 37–52. Springer, 2014.

[16] 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.

[17] 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.

[18] David Doty and Monir Hajiaghayi. Leaderless deterministic chemical reaction networks. *Natural Computing*, 14(2):213–223, 2015. Preliminary version appeared in DNA 2013.

[19] David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. In *DISC 2015: Proceedings of the 29th International Symposium on Distributed Computing, Tokyo, Japan*, October 2015.

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

[21] John E. Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theor. Comput. Sci.*, 8:135–159, 1979.

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

[23] Jérôme Leroux. Vector addition systems reachability problem (a simpler solution). In *The Alan Turing Centenary Conference*, pages 214–228, June 2012.

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

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

[26] 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.