Robust Predicate and Function Computation in Continuous Chemical Reaction Networks

Kim Calabrese
University of California–Davis, CA, USA

David Doty
University of California–Davis, CA, USA

Mina Latifi
University of California–Davis, CA, USA

Abstract

We initiate the study of "rate-constant-independent" computation of Boolean predicates (decision problems) and numerical functions in the continuous model of chemical reaction networks (CRNs), which model the amount of a chemical species as a nonnegative, real-valued *concentration*, representing an average count per unit volume. Real-valued numerical functions have previously been studied [20], finding that exactly the continuous, piecewise rational linear (meaning linear with rational slopes) functions $f : \mathbb{R}^k_{>0} \to \mathbb{R}_{>0}$ can be computed *stably* (a.k.a., *rate-independently*), meaning roughly that the CRN gets the answer correct no matter the rate at which reactions occur. For example the reactions $X_1 \to Y$ and $X_2 + Y \to \emptyset$, starting with inputs $X_1 \geq X_2$, converge to output Y having concentration equal to the initial difference of inputs $X_1 - X_2$, no matter the relative rate at which each reaction proceeds.

We first show that, contrary to the case of real-valued functions, continuous CRNs are severely limited in the Boolean predicates they can stably decide, reporting a yes/no answer based only on which inputs are 0 or positive, but not on the exact positive value of any input.

This limitation motivates a slightly relaxed notion of rate-independent computation in CRNs that we call *robust computation*. The standard mass-action rate model is used, in which each reaction (e.g., $A + B \xrightarrow{k} C$) is assigned a *rate* ($A \cdot B \cdot k$ in this example) equal to the product of its reactant concentrations and its *rate constant* k. We say the computation is correct in this model if it converges to the correct output for *any* positive choice of rate constants. This adversary is weaker than the adversary defining stable computation, the latter being able to run reactions at rates that are not those of mass-action for any choice of rate constants (e.g., the stable adversary may deactivate a reaction temporarily, even if all reactants are positive).

We show that CRNs can robustly decide every predicate that is a finite Boolean combination of *threshold predicates*, where a threshold predicate is defined by taking a rational weighted sum of the inputs $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$ and comparing to a constant, answering the question "Is $\sum_{i=1}^k w_i \cdot \mathbf{x}(i) > h$?", for rational weights w_i and real threshold h. Turning to function computation, we show that CRNs can robustly compute any piecewise affine function with rational coefficients, where threshold predicates determine which affine piece to evaluate for a given input $\mathbf{x}$.

2012 ACM Subject Classification Theory of computation → Models of computation

Keywords and phrases chemical reaction networks, analog computation, mean-field limit

Digital Object Identifier 10.4230/LIPIcs.DISC.2025.19

Related Version *Full Version*: `https://arxiv.org/abs/2506.06590` [18]

Funding *Kim Calabrese*: NSF awards 2211793, 1900931.
David Doty: NSF awards 2211793, 1900931 and DoE award DE-SC0024467.
Mina Latifi: NSF awards 2211793, 1900931 and DoE award DE-SC0024467.

39th International Symposium on Distributed Computing (DISC 2025).
Editor: Dariusz R. Kowalski; Article No. 19; pp. 19:1–19:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A *chemical reaction network* (CRN) is a model of reactions among abstract chemical species, such as the reaction $X + Y \to Z$, which indicates that if an X and a Y molecule collide, they can stick together to form a dimer called Z. Since the 19th century [29, 39], this model has been used to describe and predict the behavior of naturally occurring chemicals. It was not until the 21st century, however, that the model was repurposed as a *programming* language [35] for describing the desired behavior of synthetically engineering chemicals such as DNA strand displacement systems, both in theory [36] and practice [22, 38].

CRNs are related to a distributed computing model known as *population protocols* [5], describing anonymous finite-state agents (molecules) that interact (react) asynchronously in pairs (bimolecular reactions), changing state (chemical species) in response. Formally, a population protocol is a CRN in which each reaction has exactly two reactants (inputs) and two products (outputs), with unit *rate constants* (constant multiplier on a reaction rate; see below for definitions.) A population protocol is defined by a transition function $\delta : \Lambda \times \Lambda \to \Lambda \times \Lambda$, where for $A, B, C, D \in \Lambda$, $\delta(A, B) = (C, D)$ is the same as saying the CRN has a reaction $A + B \to C + D$. Although population protocols are a special case of CRNs, nevertheless many computations achievable with more general CRNs can be simulated exactly by population protocols, and conversely most impossibility results on population protocols also apply to the more general CRN model.

Discrete (a.k.a., *stochastic*) CRNs model the amount of each species as a nonnegative integer representing its exact molecular count [28]. In this model, the state of a CRN is given by a continuous-time Markov chain [28], where the probability of a reaction such as $2A + B \xrightarrow{k} C$ occurring is proportional to the counts of A and B, as well as the positive *rate constant* $k \in \mathbb{R}$. Furthermore, reactions can only occur when at least two molecule of A and one molecule of B is present in the configuration. When the number of molecules is very large (a typical DNA nanotechnology experiment may involve over a trillion molecules in a 50 μL test tube), CRNs are well-approximated by what is sometimes called the *mean-field limit*, the *continuous mass-action* model, and very commonly "*the deterministic model*" to contrast with the stochastic nature of the discrete model. In the continuous model, each species has a nonnegative, real-valued *concentration* indicating its average count per unit volume.[1] In this setting a reaction such as $A + B \xrightarrow{k} 2C$ indicates that some real-valued amount of reactants A and B are converted into product species C, e.g., the reaction could turn 1/3 units of A, and the same amount of B, into 2/3 units of C. That is, in the continuous model of CRNs, a reaction encodes the *ratio* of reactants that are turned into products. This is a slightly different interpretation of a reaction than in the discrete model, where the same reaction would be interpreted as "exactly one copy of species A reacts with exactly one copy of species B to produce exactly two copies of species C". While the discrete CRN model has a state that changes at discrete steps, the state of a continuous CRN experiences continuous time evolution. Each reaction has a *rate* equal to the product of the reactant concentrations and rate constant k - $k \cdot A \cdot B$ in this example. The rates define a system of polynomial ordinary differential equations (ODEs) whose unique solution trajectory[2] indicates how species concentrations change over time. The function $S(t)$ which describes

[1] This continuous mean-field limit has been explored within population protocols as well [10, 15, 16].

[2] Polynomials are locally Lipschitz, so by the Picard-Lindelöf Theorem, the ODEs have a *unique* solution; hence the term "the deterministic model".

the concentration of species S at time t is defined by an ODE for $S(t)$, where a reaction in which S appears as a product contributes a positive term to dS/dt and a negative term when S is a reactant; see Section 2.2 for a formal definition.[3]

1.1 Computation with CRNs: Related work

The discrete, integer-valued CRN model has been far more extensively studied for its computational abilities than the continuous model, so we briefly review what is known in this setting. The two most studied types of computation are Boolean-valued predicates $\phi : \mathbb{N}^k \to \{0, 1\}$ [6, 9] and numeric functions $f : \mathbb{N}^k \to \mathbb{N}$ [21, 25]. We designate special *input species* $X_1, \ldots, X_k$ whose initial counts represent the inputs to ϕ or f. These may be the only species present initially ("leaderless" CRNs [5, 9, 25]), or there may be additional species present initially with counts independent of the inputs ("leader-driven" [7, 21]). For a function $f : \mathbb{N}^k \to \mathbb{N}$, the count of a special "output" species Y represents the output of f. For a predicate $\phi : \mathbb{N}^k \to \{0, 1\}$, some species are designated as "yes" voters and some as "no" voters; the CRN's output is undefined if voters for both outputs (or neither) are present, otherwise the output is yes or no depending on which type of voter is present.

Much study has been devoted to *stable* (a.k.a., *rate-independent*) computation, which means intuitively that the CRN generates the correct output no matter in what order reactions proceed. In the discrete stochastic model, rates influence the probability of choosing among several competing reactions to be the next reaction to occur, so rate-independent computation means essentially that, despite the inherent stochasticity of the model, nevertheless the correct output is generated with probability 1.[4] Experimentally, rate constants are often difficult to control precisely, sometimes with a 10-fold difference in rate constants intended to be equal [38], whereas reaction stoichiometry (i.e., how many of each reactant is consumed, and how many of each product is produced) is naturally digital and simpler to engineer exactly. Furthermore there may be violations of the assumptions justifying typical rate laws (for example the solution may not be well-mixed) leading to actual reaction rates deviating from predictions. Whether leaderless or not, exactly the *semilinear* predicates [6] and functions [21, 25] can be stably computed by discrete CRNs. See [6, 9, 21, 25] for formal definitions of these concepts, which are not required to understand this paper.

In this paper, we study the (naturally deterministic) continuous model, we focus on the "rate-independent" characterization of stable computation rather than on probability. Intuitively we want to capture the idea that the CRN generates the correct output "no matter the reaction rates". In contrast, if reaction rates are undisturbed and rate constants can be controlled, then continuous CRNs are Turing universal [27], a consequence of the surprising computational power of polynomial ODEs [17].

Prior work has completely characterized the real-valued functions $f : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ stably computable by continuous CRNs [20]. This definition of stable computation is based on a very general notion of reachability, defined formally in Section B.1. A function f is stably computable by a continuous CRN if and only if it is piecewise rational linear (meaning a finite union of functions that are linear with rational coefficients, e.g., $\frac{4}{3}x_1 - 5x_2$), and *positive-continuous*, meaning intuitively that discontinuities can occur only when some input x_i goes from 0 to positive (Definition B.9); for example the function $f(x_1, x_2) = x_1 + x_2$ if $x_2 > 0$, and $f(x_1, x_2) = 3x_1$ if $x_2 = 0$, is positive-continuous but not continuous.

---

[3] We will often write the time derivative of S, dS/dt as S'.
[4] There are caveats to this, unnecessary to understand this paper; see [23] for a thorough discussion.

1.2 Our results

An open question from [20] concerns stable computation of real-valued decision problems, a.k.a., Boolean predicates $\phi : \mathbb{R}^k_{\geq 0} \to \{0,1\}$. In contrast to functions, we show that the stably decidable predicates are severely limited: exactly the *detection* predicates can be stably decided (Theorem B.2), those that depend only on whether certain inputs are 0 or positive (Definition B.1), but not on their exact positive value. This limitation prompts us to relax the notion of stable computation as suggested by another open question in [20]. We call the concept *robust computation*, which uses mass-action rate laws (see Section 2.2), limited compared to the adversarial rates used in stable computation, but requires the CRN to converge to the correct output for *every* choice of positive rate constants (Definitions 2.6 and 2.7).

Our first main result (Theorem 3.7) shows that every finite Boolean combination of *threshold predicates* (Definition 2.2) is robustly decidable, where a *threshold predicate* $\phi(\mathbf{x}) = 1$ if and only if $\sum_{i=1}^{k} w_i \cdot \mathbf{x}(i) > h$ for rational constants $w_i \in \mathbb{Q}$ and real constant $h \in \mathbb{R}$. Intuitively, there are a finite number of hyperplanes, which cut $\mathbb{R}^k_{\geq 0}$ into a finite number of regions, and $\phi(\mathbf{x})$ depends only on which region $\mathbf{x}$ is in. A famous example of a threshold predicate (which incidentally is not a detection predicate) is *majority* [1–4, 7, 8, 11–14, 24, 26, 31–33, 37] (see Lemma 3.3).

Our second main result (Theorem 3.10) shows that every *threshold-piecewise rational floor-affine function* (Definition 2.3) can be stably computed, which are finite unions of "floor-affine" components $f_i : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$, where a threshold predicate $\phi(\mathbf{x})$ indicates whether f_i is the correct affine function to use for input $\mathbf{x}$ (see Section 3.2). *Floor*-affine means negative outputs are replaced with 0 (Definition 2.1, necessary since concentrations are nonnegative), i.e., $f_i(\mathbf{x}) = \max(0, g(\mathbf{x}))$ for some affine $g : \mathbb{R}^k_{\geq 0} \to \mathbb{R}$. Compared to stable function computation [20], such functions can be discontinuous even within the strictly positive orthant; e.g., $f(x_1, x_2) = x_1 + x_2 + 2$ if $x_1 < x_2$ and $\max(0, x_1/3 - x_2)$ if $x_1 \geq x_2$.

Threshold predicates and affine functions allow a constant "offset", unlike linear functions studied in [20]. This is because unlike [20], we allow a notion similar to "leaders" in discrete distributed computing models, called *initial context*: some non-input species may be present initially, but their initial concentrations do not depend on the input values. Such initial context can be used to implement these constant offsets.[5] We believe our constructions extend naturally to the leaderless setting, and would limit threshold predicates to comparing to $h = 0$ instead of arbitrary constant $h \in \mathbb{R}$, and limit functions to piecewise *linear* (rather than affine), but we have not explored this in detail. Note that some functions are computable with specific rate constants, but are neither stably nor robustly computable; for example $f(x) = 3.5x^2$ is computable by the reactions $2X \xrightarrow{3.5} 2X + Y$ and $Y \xrightarrow{1} \emptyset$.

The high-level goals of stable and robust computation are the same, which is to formalize a notion of computation in an adversarial environment where reaction rates may deviate from standard models. However, robust computation, being based on ODEs in mass-action kinetics, requires vastly different (and typically more sophisticated) techniques to reason about. One goal of this paper is to begin establishing general techniques to prove correctness of such systems. For example, Lemmas 3.1 and 3.2 in Section 3.1 are general lemmas that we use repeatedly to reason "modularly" when the output of an "upstream" CRN U influences a "downstream" CRN D in one direction only, i.e., D does not influence U.

---

[5] As noted in [20, Section 6.2], allowing initial context in stable computation leads to replacing "linear" with "affine" in the characterization.

2 Preliminaries

Let $\mathbb{N}$ denote the nonnegative integers, $\mathbb{Q}$ the rationals, and $\mathbb{R}$ the reals. For any set $A \subseteq \mathbb{R}$, $A_{\geq 0} = A \cap [0, \infty)$ and $A_{>0} = A \cap (0, \infty)$. Given a finite set F and a set S, let S^F denote the set of functions $\mathbf{c} : F \to S$. In the case of $S = \mathbb{R}$ (resp., $\mathbb{N}$), we view $\mathbf{c}$ equivalently as a real-valued (resp., integer-valued) vector indexed by elements of F. Given $a \in F$, we write $\mathbf{c}(a)$, to denote the real number indexed by a. The notation $\mathbb{R}^F_{\geq 0}$ is defined similarly for nonnegative real vectors. For a function of time $A : \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $c \in \mathbb{R} \cup \{\infty\}$, we write $A \to c$ to denote $\lim_{t\to\infty} A(t) = c$; most frequently, A will be a chemical species concentration, or some function thereof.

▶ **Definition 2.1.** *$f : \mathbb{R}^k \to \mathbb{R}$ is* rational affine *if for some rational $w_1, \ldots, w_k \in \mathbb{Q}$ and real $h \in \mathbb{R}$, for all $\mathbf{x} \in \mathbb{R}^k$, $f(\mathbf{x}) = h + \sum_{i=1}^k w_i \cdot \mathbf{x}(i)$.*[6] *If $h = 0$ we say f is* rational linear*. f is* rational floor-affine *if $f(\mathbf{x}) = \max(0, g(\mathbf{x}))$ for some rational affine g.*[7]

For example, the function $f_1(x, y, z) = \sqrt{2} + 2x + \frac{2}{3}y - \frac{3}{7}z$ is rational affine while the function $f_2(x, y, z) = 1 + \sqrt{2}x + \sqrt{3}y + 2z$ is not, although still affine.

The next two definitions capture the class of predicates and functions we show are robustly computable by continuous CRNs in our main results, Theorems 3.7 and 3.10.

▶ **Definition 2.2.** *$\phi : \mathbb{R}^k \to \{0, 1\}$ is a* threshold predicate *with rational* weights *$w_1, \ldots, w_k \in \mathbb{Q}$ and real* threshold *$h \in \mathbb{R}$ if, for all $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$, $\phi(\mathbf{x}) = 1 \iff \sum_{i=1}^k w_i \cdot \mathbf{x}(i) > h$. We say ϕ is a* multi-threshold predicate *if it is a finite Boolean combination of threshold predicates.*

For example, the predicate $\phi(x, y, z) = 1 \iff x + 2y + 3z > \pi$ is a threshold predicate with rational weights $1, 2$ and 3 and threshold π. If ϕ_1, ϕ_2 and ϕ_3 are each threshold predicates, then the predicate ψ defined as $\psi(\mathbf{x}) = (\phi_1(\mathbf{x}) \vee \phi_2(\mathbf{x})) \wedge \overline{\phi_3}(\mathbf{x})$ (where $\vee$ and $\wedge$ denote logical OR and AND, and $\overline{\phi_3}$ is the logical negation of ϕ_3) is a multi-threshold predicate.

▶ **Definition 2.3.** *$f : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ is* threshold-piecewise rational floor-affine *if there is a finite set of rational floor-affine functions $f_1, \ldots, f_l : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$, known as the* affine components *of f, and multi-threshold predicates $\phi_1, \ldots, \phi_l : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$ such that*

1. *The sets $\phi_1^{-1}(1), \ldots, \phi_l^{-1}(1)$ are a partition of $\mathbb{R}^k_{\geq 0}$, i.e., for each $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$, $\phi_i(\mathbf{x}) = 1$ for exactly one $1 \leq i \leq l$.*
2. *For each i and $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$, if $\phi_i(\mathbf{x}) = 1$ then $f(\mathbf{x}) = f_i(\mathbf{x})$, i.e., $\phi_i(\mathbf{x})$ indicates whether f_i is the correct affine component defining $f(\mathbf{x})$.*

For example, if $f_1, f_2 : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ are floor-affine functions and $\phi : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a multi-threshold predicate then the function

$$f(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) & \text{If } \phi(\mathbf{x}) = 1 \\ f_2(\mathbf{x}) & \text{If } \overline{\phi}(\mathbf{x}) = 1 \end{cases}$$

is a threshold-piecewise rational floor-affine function as both ϕ and its negation $\overline{\phi}$ are multi-threshold predicates, and for any input $\mathbf{x}$ exactly one of $\phi(\mathbf{x})$ and $\overline{\phi}(\mathbf{x})$ is 1.

[6] It would seem more natural to require h also to be rational, or conversely to allow each w_i to be real, but this distinction is relevant with CRNs: h will come from the initial concentration of some species, or its negation, whereas the w_i's will come from ratios of integer stoichiometric reaction coefficients. Similar reasoning applies to Definition 2.2.

[7] The significance of rational floor-affine functions is that CRNs can only output nonnegative concentrations; reactions such as $X_1 \to Y$ and $X_2 + Y \to \emptyset$ technically compute, not the affine function $g(x_1, x_2) = x_1 - x_2$, but the floor-affine function $f(x_1, x_2) = \max(0, g(x_1, x_2))$.

2.1 Chemical reaction networks

Throughout this paper, let Λ be a finite set of chemical *species*. Given $S \in \Lambda$ and *state* $\mathbf{c} \in \mathbb{R}_{\geq 0}^{\Lambda}$, $\mathbf{c}(S)$ is the *concentration of* S *in* $\mathbf{c}$. For any $\mathbf{c} \in \mathbb{R}_{\geq 0}^{\Lambda}$, let $[\mathbf{c}] = \{S \in \Lambda \mid \mathbf{c}(S) > 0\}$, the set of species *present* in $\mathbf{c}$ (a.k.a., the *support* of $\mathbf{c}$). We write $\mathbf{c} \leq \mathbf{c}'$ to denote that $\mathbf{c}(S) \leq \mathbf{c}'(S)$ for all $S \in \Lambda$. Given $\mathbf{c}, \mathbf{c}' \in \mathbb{R}_{\geq 0}^{\Lambda}$, we define the vector component-wise operations of addition $\mathbf{c} + \mathbf{c}'$, subtraction $\mathbf{c} - \mathbf{c}'$, and scalar multiplication $x\mathbf{c}$ for $x \in \mathbb{R}$.

A *reaction* over Λ is a triple $\alpha = (\mathbf{r}, \mathbf{p}, k) \in \mathbb{N}^{\Lambda} \times \mathbb{N}^{\Lambda} \times \mathbb{R}_{>0}$, such that $\mathbf{r} \neq \mathbf{p}$, specifying the stoichiometry of the *reactants* $\mathbf{r}$ and *products* $\mathbf{p}$, and the *rate constant* k. For instance, given $\Lambda = \{A, B, C, D\}$, the reaction $A + 2B \xrightarrow{6.7} A + 3C$ is the triple $((1, 2, 0, 0), (1, 0, 3, 0), 6.7)$.

A *chemical reaction network (CRN)* is a pair $\mathcal{C} = (\Lambda, R)$, where Λ is a finite set of chemical *species*, and R is a finite set of reactions over Λ. A *state* of a CRN $\mathcal{C} = (\Lambda, R)$ is a vector $\mathbf{c} \in \mathbb{R}_{\geq 0}^{\Lambda}$. Given a state $\mathbf{c}$ and reaction $\alpha = (\mathbf{r}, \mathbf{p}, k)$, we say that α is *applicable* in $\mathbf{c}$ if $[\mathbf{r}] \subseteq [\mathbf{c}]$ (i.e., $\mathbf{c}$ contains positive concentration of all of the reactants). If no reaction is applicable in state $\mathbf{c}$, we say $\mathbf{c}$ is *static*.

The next two definitions are "syntactic" preparation for stating how a CRN can compute a predicate or function; the "semantic" definitions of stable (Definitions B.5 and B.6) and robust (Definitions 2.6 and 2.7) computation will use these definitions to state under what conditions a CRN "correctly" computes. The first definition is for Boolean predicates.

▶ **Definition 2.4.** *A* chemical reaction decider *(CRD) is a tuple* $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon_{\mathrm{yes}}, \Upsilon_{\mathrm{no}}, \mathbf{i})$ *where* (Λ, R) *is a CRN,* $\Sigma \subseteq \Lambda$ *is the set of* input species*,* $\Upsilon_{\mathrm{yes}} \subseteq \Lambda$ *is the set of* yes voters*,* $\Upsilon_{\mathrm{no}} \subseteq \Lambda \setminus \Upsilon_{\mathrm{yes}}$ *is the set of* no voters*, and* $\mathbf{i} \in \mathbb{R}_{\geq 0}^{\Lambda \setminus \Sigma}$ *is the* initial context*.*

Intuitively initial context $\mathbf{i}$ refers to fixed initial concentrations for non-input species, independent of the input value. A CRD's initial state for predicate input $\mathbf{x} \in \mathbb{R}_{\geq 0}^{k}$ is then $\mathbf{x} + \mathbf{i}$, where we assume some fixed ordering $X_1, \ldots, X_k$ of input species to interpret a vector $\mathbf{x} \in \mathbb{R}_{\geq 0}^{k}$ as a state $\mathbf{x} \in \mathbb{R}_{\geq 0}^{\Sigma}$. The next definition is used for computing numeric functions, identifying a special species Y whose concentration represents output:

▶ **Definition 2.5.** *A* chemical reaction computer *(*CRC*) is a tuple* $\mathcal{C} = (\Lambda, R, \Sigma, Y, \mathbf{i})$ *where* (Λ, R) *is a CRN,* $\Sigma \subseteq \Lambda$ *is the set of* input species*, and* Y *is the* output species*, and* $\mathbf{i} \in \mathbb{R}_{\geq 0}^{\Lambda \setminus \Sigma}$ *is the* initial context*.*

2.2 Robust (rate-*constant*-independent) computation

Stable computation requires a CRN to work against a very powerful adversary who can essentially set the rate of each reaction at each time arbitrarily. This means in particular that the CRN works under a variety of rate laws besides mass-action (defined below). Here we consider a weaker adversary, one that cannot control the rate law – that will be mass-action – but that *can* set the parameters of the rate law, known as *rate constants*. Crucially, these are constant with respect to *time*: the adversary can choose arbitrary positive values for these rate constants, but sets them to those values at time $t = 0$, and the rate constants stay at those values for all future $t > 0$.[8] A CRN robustly computes a function or predicate if it computes the correct output against this adversary: i.e., if the mass-action rate law converges to the correct output, no matter which positive rate constants are chosen.

[8] If we allowed the adversary to change the rate constants over time, then it could mimic the stable computation adversary by adjusting rate constants so as to target particular absolute rates at each time.

A CRN $\mathcal{C} = (\Lambda, R)$ under the *mass-action rate law* is governed by a system of polynomial ordinary differential equations (ODEs) that define functions $S(t)$ representing the concentration of species S at time t. The *rate* $\rho_t(\alpha)$ of a reaction $\alpha = (\mathbf{r}, \mathbf{p}, k)$ at time t is $\rho_t(\alpha) = k \cdot \prod_{S \in \Lambda} S(t)^{\mathbf{r}(S)}$, i.e., the rate constant times each reactant concentration at time t. For example, the rate of $A + 2B \xrightarrow{4.5} C$ is $4.5 \cdot A(t) \cdot B(t)^2$. Each reaction $\alpha = (\mathbf{r}, \mathbf{p}, k)$ contributes a term $\rho_t(\alpha) \cdot (\mathbf{p}(S) - \mathbf{r}(S))$ to the ODE of each species S that is net produced or consumed; the term is α's rate $\rho_t(\alpha)$ times the net stoichiometry of S in α (positive if S is net produced by α, e.g., $S \to 3S$ net produces 2 S's since $\mathbf{p}(S) - \mathbf{r}(S) = 2$ and negative if S is net consumed, e.g., $2S \to S$ net consumes 1 S, i.e., $\mathbf{p}(S) - \mathbf{r}(S) = -1$). For example, the CRN with reactions $A + 2B \xrightarrow{k_1} 3C$ and $2C \xrightarrow{k_2} C$ corresponds to the ODEs

$$A'(t) = -k_1 A(t) B(t)^2$$
$$B'(t) = -2k_1 A(t) B(t)^2$$
$$C'(t) = 3k_1 A(t) B(t)^2 - k_2 C(t)^2$$

This can be formalized as follows. Given a CRN $\mathcal{C} = (\Lambda, R)$ let $\mathbf{A} : \mathbb{R}^{\Lambda}_{\geq 0} \to \mathbb{R}^{R}_{\geq 0}$ map each state $\mathbf{d}$ of $\mathcal{C}$ to the vector $\mathbf{A}(\mathbf{d})$ of instantaneous reaction rates in state $\mathbf{d}$, as given by the mass-action ODEs. With the example reactions above, the state of concentrations $\mathbf{d} = (1, 2, 0)$ would be mapped to the flux vector $\mathbf{A}(\mathbf{d}) = (-4k_1, -8k_1, 12k_1)$. The first coordinate was obtained by setting $A(t) = 1$, $B(t) = 2$ and $C(t) = 0$ in the ODE for A'. The second and third coordinates are obtained similarly by evaluating the ODE for B' and C' respectively. Define the $|\Lambda| \times |R|$ *stoichiometry matrix* $\mathbf{M}$ such that, for species $S \in \Lambda$ and reaction $\alpha = (\mathbf{r}, \mathbf{p}) \in R$, $\mathbf{M}(S, \alpha) = \mathbf{p}(S) - \mathbf{r}(S)$ is the net amount of S produced by α (negative if S is consumed). For example, if we have the reactions $X \to Y$ and $X + A \to 2X + 3Y$, and if the three rows correspond to A, X, and Y, in that order, then $\mathbf{M} = \begin{pmatrix} 0 & -1 \\ -1 & 1 \\ 1 & 3 \end{pmatrix}$. The vector $\mathbf{M} \cdot \mathbf{A}(\mathbf{d})$ gives the rate at which each species concentration is changing in state $\mathbf{d}$. Given an initial state $\mathbf{c}$, the *mass-action trajectory* $\boldsymbol{\tau} : [0, t_{\max}) \to \mathbb{R}^{\Lambda}_{\geq 0}$ *starting at* $\mathbf{c}$, is the solution to the initial value problem

$$\frac{d\boldsymbol{\tau}}{dt} = \mathbf{M} \cdot \mathbf{A}(\boldsymbol{\tau}(t)), \ \boldsymbol{\tau}(0) = \mathbf{c} \tag{1}$$

where $t_{\max} \in \mathbb{R}_{\geq 0} \cup \{\infty\}$.[9] That is, $\boldsymbol{\tau}(t) = (S_1(t), S_2(t), \ldots, S_{|\Lambda|}(t))$ is the vector whose ith component is the concentration of species $S_i \in \Lambda$ at time t. While some CRNs induce ODEs with solutions that do not exist for all time, mass-action ODEs are locally Lipschitz, implying that a unique solution to (1) always exists on some interval. We now define what it means for CRD to decide a predicate in the mass action model with adversarial rate constants.

Intuitively, a CRD robustly decides a predicate ϕ if, for any input $\mathbf{x}$ and any choice of rate constants for the reactions, the induced ODEs (defined by mass action) cause yes voters to remain positive as $t \to \infty$ while no voters converge to zero if $\phi(\mathbf{x}) = 1$ and vice-versa if $\phi(\mathbf{x}) = 0$.

▶ **Definition 2.6** (robustly decide). *Let $\phi : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$ be a predicate. We say a CRD $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon_{\text{yes}}, \Upsilon_{\text{no}}, \mathbf{i})$* robustly decides *(a.k.a.,* rate-constant-independently decides*) ϕ if, for any choice of strictly positive rate constants and every $\mathbf{x} \in \mathbb{R}_{\geq 0}$, the following holds.*

[9] Although $t_{\max} = \infty$ for "typical" CRNs, there are pathological CRNs such as $2X \to 3X$, which diverge to ∞ in finite time; for instance with $X(0) = 1$ and unit rate constant, this CRN has the solution $X(t) = 1/(1 - t)$, which goes to ∞ as $t \to 1$, so $t_{\max} = 1$ for this CRN.

Let $\boldsymbol{\tau}$ be the mass-action trajectory of $\mathcal{C}$, starting at state $\mathbf{x} + \mathbf{i}$. If $\phi(\mathbf{x}) = 1$ (resp. 0), let $\Upsilon_{\mathrm{C}} = \Upsilon_{\mathrm{yes}}$ (resp. Υ_{no}) be the correct voters, and let $\Upsilon_{\mathrm{I}} = \Upsilon_{\mathrm{no}}$ (resp. Υ_{yes}) be the incorrect voters. Define $C(t) = \sum_{V \in \Upsilon_{\mathrm{C}}} V(t)$ be the sum of concentrations of correct voters. Then $\liminf_{t\to\infty} C(t) > 0$ and $\lim_{t\to\infty} I(t) = 0$ for all $I \in \Upsilon_{\mathrm{I}}$.

In other words, the CRD deciding $\phi(\mathbf{x})$ starts with input species concentrations defined by $\mathbf{x}$, and other initially present species indicated by $\mathbf{i}$ (whose initial concentrations are the same for all inputs $\mathbf{x}$). The CRD proceeds by mass-action dynamics as defined above, and converges to a state with only correct voters present. Since concentrations are nonnegative, requiring each individual incorrect voter to converge to 0 is equivalent to requiring their sum to converge to 0, whereas we do not require any individual correct voter to stay bounded above 0, only the sum of correct voters,[10]

We now define what it means for a CRC to robustly compute a real-valued function.

▶ **Definition 2.7** (robustly compute). *Let $f : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$. We say a CRC $\mathcal{C} = (\Lambda, R, \Sigma, Y, \mathbf{i})$* robustly computes *(a.k.a.,* rate-constant-independently computes*) f if, for any choice of strictly positive rate constants and every $\mathbf{x} \in \mathbb{R}_{\geq 0}$, the component $Y(t)$ of $\mathcal{C}$'s mass action trajectory starting from $\mathbf{i} + \mathbf{x}$ satisfies $\lim_{t\to\infty} Y(t) = f(\mathbf{x})$.*

The full definition of stable computation is given in Section B, based on a formal definition of reachability in continuous CRNs. The definition intuitively says a CRN can reach from state $\mathbf{x}$ to $\mathbf{c}$ if one can run some reactions starting at $\mathbf{x}$ and reach to $\mathbf{c}$, without ever running a reaction when one of its reactants is 0. We say that a CRC *stably computes* a function f (or stably decides a predicate ϕ) if, starting from initial state $\mathbf{x}$ encoding the input, for any state $\mathbf{c}$ reachable from $\mathbf{x}$, there is a "correct" state $\mathbf{o}$ reachable from $\mathbf{c}$ (correct meaning the output in $\mathbf{o}$ equals $f(\mathbf{x})$ or $\phi(\mathbf{x})$), that is also *stable*, meaning that every state $\mathbf{o}'$ reachable from $\mathbf{o}$ has the same output as $\mathbf{o}$.

The next definition connects stable and robust computation for some specially structured CRNs. Intuitively, a CRN is *feedforward* if there is an ordering of species so that every reaction producing a species consumes another species earlier in the ordering; formally:

▶ **Definition 2.8.** *A CRN $\mathcal{C} = (\Lambda, R)$ is* feedforward *if Λ can be ordered $\Lambda = \{S_1, S_2, \ldots, S_n\}$ so that, if for each reaction $\alpha = (\mathbf{r}, \mathbf{p}, k) \in R$ and S_j where $\mathbf{p}(j) > \mathbf{r}(j)$, there is $S_i \in \Lambda$ with $i < j$ such that $\mathbf{p}(i) < \mathbf{r}(i)$.*

The following was shown in [20, Corollary 4.11] (in different but equivalent terms).

▶ **Lemma 2.9.** *Each feedforward CRC stably computing function f also robustly computes f.*

Essentially the same proof shows the following.

▶ **Lemma 2.10.** *Each feedforward CRD stably deciding a predicate ϕ also robustly decides ϕ.*

3 Robust computation by continuous CRNs

3.1 Boolean combinations of threshold predicates are robustly decidable

We begin by proving a technical lemma that relates known asymptotic behavior of certain species to the desired asymptotic behavior of others that depend on them. Intuitively, we think of p and g as functions of concentration whose asymptotic behavior has already been

[10] For example two yes voters could oscillate between 0 and 1, so long as they always sum to at least 0.1.

analyzed. In particular, we think of species involved in p and g as belonging to an "upstream" CRC (or CRD) $\mathcal{C}_{\mathrm{U}}$, whose outputs are used in reactions of a "downstream" CRC (or CRD) $\mathcal{C}_{\mathrm{D}}$ that influence concentration of species F. These species influence the concentration of species F, which evolves as $f(t)$. However, f does not affect the concentrations of the species involved in p and g. Assuming that the ODE describing f is of the form $f(t)' = g(t) - p(t)f(t)$, we can use Lemmas 3.1 and 3.2 to reason about asymptotic behavior of f.

▶ **Lemma 3.1.** *Let $p, g : \mathbb{R}_{\geq 0} \to \mathbb{R}$, be differentiable functions, with $p(t) > 0$ and $g(t) \geq 0$ for all $t \in \mathbb{R}_{\geq 0}$. Let $K = \frac{g(0)+p(0)f(0)}{p(0)}$ be a constant. If $f : \mathbb{R} \to \mathbb{R}$ is differentiable and satisfies the first order linear ODE $f'(t) = g(t) - p(t)f(t)$, then for all $t \in \mathbb{R}_{\geq 0}$,*

$$f(t) \leq \frac{g(t)}{p(t)} + K \exp\left(-\int_0^t p(s)ds\right).$$

This lemma allows us to cleanly demonstrate that the concentration of particular voting species V converges to zero. In this context, the function p will represent the concentration of species whose presence causes V to be consumed, and g represents the concentration of species whose presence causes V to be produced. A common pattern in our correctness proofs will be to rearrange the mass-action ODEs for an incorrect voting species into a first order linear ODE, apply Lemma 3.1, and then argue that $g(t)$ converges to zero while $p(t)$ converges to a positive value (or converges to zero slower than g). Further, the non-negativity condition of the hypothesis is trivially satisfied as components of the mass action trajectory are always non-negative, and for some CRNs, strictly positive with suitable initial conditions. Lemma 3.1 is not applicable if $p(t)$ is not strictly positive, but in this case we can still find a bound of a similar form.

▶ **Lemma 3.2.** *Let $p, g : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be differentiable functions, with $p(t) \geq 0$ and $g(t) \geq 0$ for all $t \in \mathbb{R}_{\geq 0}$. Let $K = \frac{g(0)+f(0)(p(0)+1)}{p(0)+1}$ be a constant. If $f : \mathbb{R} \to \mathbb{R}$ is differentiable and satisfies the first order linear ODE $f'(t) = g(t) - p(t)f(t)$, then for all $t \in \mathbb{R}_{\geq 0}$,*

$$f(t) \leq \frac{5g(t)}{p(t) + (t^2+1)^{-1}} + K \exp\left(-\int_0^t p(s)ds\right).$$

Proofs of Lemmas 3.1 and 3.2 are given in [18].

We begin our positive results by showing the *majority predicate* $\phi : \mathbb{R}^2 \to \{0, 1\}$ defined by $\phi(a, b) = 1$ if and only if $a > b$ is robustly decidable. We remark that such a predicate is in general not a detection predicate, showing that the class of robustly decidable predicates is strictly larger than the class of stably decidable one.

We first give an example of a CRD that "almost" decides the majority Let $\hat{\mathcal{D}} = (\hat{\Lambda}, \hat{R}, \Sigma, \Upsilon_{\mathrm{yes}}, \Upsilon_{\mathrm{no}}, \hat{\mathbf{i}})$ with $\Upsilon_{\mathrm{yes}} = \{Y\}$ and $\Upsilon_{\mathrm{no}} = \{N\}$, with initial state $\{aA, bB, 1Y\}$ and $\mathbf{i} = \{1Y\}$, and reactions

$$A + N \overset{k_1}{\to} A + Y \tag{2}$$

$$B + Y \overset{k_2}{\to} B + N \tag{3}$$

$$A + B \overset{k_3}{\to} \emptyset \tag{4}$$

$\hat{\mathcal{D}}$ both robustly (even stably) decides majority when $a \neq b$. Consider the case when $a > b$. Intuitively, only reaction (4) decreases the concentrations of A and B, and thus the concentration of A remains positive (as $a - b > 0$) while the concentration of B converges to zero. As a result, (2) will convert all of species N to species Y. Similarly, when $a < b$ as only concentration of B remains positive (3) converts all of Y to N.

Unfortunately, this CRD is unable to robustly decide majority when $a = b$. In this case, reaction (4) decreases both A and B at the same rate, allowing rate constants to influence the output. The correct output for this case is "no", but the adversary can choose $k_1 = 100, k_2 = 5, k_3 = 1$ so that when A and B approach 0, the rate of reaction (2) will be greater than reaction (3). Thus the CRD will incorrectly converge to $\{1Y\}$ and output "yes". Note that there exist rate constants under which the correct answer is reached. For example, when $k_1 < k_2$, the CRN converges to the state $\{1N\}$.

Since Theorem B.2 implies no CRD can *stably* decide majority, any modification to $\hat{\mathcal{D}}$ so that it *robustly* decides majority must take advantage of mass-action rate laws.

This is why the addition of the auxiliary species C is necessary in Algorithm 3.1. As C converges to zero *asymptotically* slower than A, its presence can speed up the production of N. This use of different asymptotic decays is a powerful tool of mass-action, not available for stable computation, which explains that a CRD can robustly decide majority but cannot stably decide it.

Algorithm 1 The following CRD robustly decides the majority predicate $\phi(a, b) = 1$ if and only if $a > b$. CRD $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon_{\text{yes}}, \Upsilon_{\text{no}}, \mathbf{i})$ is defined as follows. Let $\Upsilon_{\text{yes}} = \{Y\}$ and $\Upsilon_{\text{no}} = \{N\}$ be the yes voter and no voter respectively. Let the initial state be $\{aA, bB, 1Y, 1C\}$ where $\{1Y, 1C\}$ is the initial context $\mathbf{i}$. We add the following reactions:

$$A + N \xrightarrow{k_1} A + Y \tag{5}$$
$$B + Y \xrightarrow{k_2} B + N \tag{6}$$
$$A + B \xrightarrow{k_3} \emptyset \tag{7}$$
$$C + Y \xrightarrow{k_4} C + N \tag{8}$$
$$3C \xrightarrow{k_5} \emptyset \tag{9}$$

▶ **Lemma 3.3.** *The CRD in Algorithm 3.1 robustly decides the majority predicate.*

Proof. Here we show that the CRD given in Algorithm 3.1 robustly decides the majority predicate $\phi(a, b) = 1$ if and only if $a > b$. We consider the following 3 cases separately.

$a > b$**:** To argue that we converge to a correct vote, we must show that $\lim_{t\to\infty} Y(t) = 1$. This is equivalent to showing $\lim_{t\to\infty} N(t) = 0$, as $Y(t) + N(t) = 1$ for all t. We first observe that the only reactions that change the concentration of species A, B, and C strictly decrease the concentration of each species. Therefore, as t approaches infinity, the limits of their concentrations satisfy $\lim_{t\to\infty} A(t) = a - b$ and $\lim_{t\to\infty} B(t) = \lim_{t\to\infty} C(t) = 0$, as the initial concentration of A is strictly greater than the concentration of B. [11]
We observe the CRC induces this mass-action ODE for the concentration of N: $N'(t) = -k_1A(t)N(t) + k_2B(t)Y(t) + k_3C(t)Y(t)$. Using the fact that $Y(t) + N(t) = 1$, we may rewrite the ODE in terms of only $N(t)$ and species whose limit is known: $N'(t) + N(t)(k_1A(t) + k_2B(t) + k_3C(t)) = k_2B(t) + k_3C(t)$. We now want to apply Lemma 3.1, to connect the asymptotic behavior of A, B and C to that of N. To see that this lemma

[11] This can also be seen by application of Lemma 2.9. The sub-CRN with reactions $A + B \to \emptyset$ and $3C \to \emptyset$ is feedforward and stably computes $f(a, b, c) = a - b$ when $a > b$, converging to the static state $\{(a - b)A, 0B, 0C\}$. By Lemma 2.9, this restricted CRN robustly computes $a - b$.

is applicable, we observe that with the initial conditions $A(0) = a > 0$, the function $A(t)$ is always positive as it monotonically decreases and approaches a positive limit. This also implies the lower bound $A(t) \geq a - b$, which is positive in this case. Further, $B(t)$ and $C(t)$ are both non-negative, so we can apply Lemma 3.1 with $f(t) = N(t)$, $p(t) = k_1A(t) + k_2B(t) + k_3C(t)$ and $g(t) = k_2B(t) + k_3C(t)$ to obtain the bound

$$\begin{aligned} N(t) &\leq \frac{k_2B(t) + k_3C(t)}{k_1A(t) + k_2B(t) + k_3C(t)} + K\exp\left(-\int_0^t k_1A(s) + k_2B(s) + k_3C(s)ds\right) \\ &\leq \frac{k_2B(t) + k_3C(t)}{k_1A(t) + k_2B(t) + k_3C(t)} + K\exp\left(-(a-b)k_1t\right) \end{aligned}$$

Since B and C converge to 0 as t approaches infinity, this shows $\lim_{t\to\infty} N(t) = 0$.

$\boldsymbol{a < b}$**:** Symmetric to the previous case.

$\boldsymbol{a = b}$**:** We want to show that in this case CRN converges to the static state $\mathbf{c} = \{1N\}$. With the initial conditions $A(0) = B(0) = a$, it can be found that the functions $A(t)$ and $B(t)$ are equal from their mass action ODEs $A'(t) = B'(t) = -A(t)B(t)$. We can then derive that the closed form for the concentrations $A(t)$ and $B(t)$ is $\frac{a}{k_1t+1}$, where a is the initial concentration of A. Similarly, we can compute that C has closed form $C(t) = \frac{1}{\sqrt{2k_3t+1}}$. We consider the mass action ODE for $Y'(t)$

$$Y'(t) + \left(\frac{2a}{k_1t+1} + \frac{1}{\sqrt{2k_3t+1}}\right)Y(t) = \frac{a}{k_1t+1}$$

This is a first-order linear ordinary differential equation, so we can find an explicit solution for Y by way of integrating factor:

$$Y(t) = \frac{\int_0^t (as+1)^{a/k_1} e^{\sqrt{2k_3s+1}/k_3} ds + 1}{(at+1)^{2a/k_1} e^{\sqrt{2k_3t+1}/k_3}}.$$

Taking the limit as $t \to \infty$, observe that the numerator and the denominator both approach ∞ for any positive choice of k_1 and k_3. Thus,

$$\begin{aligned} \lim_{t\to\infty} Y(t) &= \lim_{t\to\infty} Y'(t) && \text{L'Hôpital's rule} \\ &= \lim_{t\to\infty} \frac{\frac{a}{k_1t+1}}{\frac{2a}{k_1t+1} + \frac{1}{\sqrt{2k_3t+1}}} \\ &= \lim_{t\to\infty} \frac{a}{k_1t+1} \times \frac{(k_1t+1)\sqrt{2k_3t+1}}{2a\sqrt{2k_3t+1} + k_1t+1} = \lim_{t\to\infty} \frac{O\left(t^{3/2}\right)}{O\left(t^2\right)} = 0. \end{aligned}$$

Since $Y(t)$ approaches 0 as $t \to \infty$, $N(t)$ approaches 1 as desired. ◀

Next, we show that threshold predicates are robustly decidable by CRDs.

▶ **Lemma 3.4.** *Every threshold predicate is robustly decidable by a continuous CRC.*

Proof sketch. A full proof is given in Section A. It works by reducing the problem of deciding ϕ to that of deciding majority as in Lemma 3.3. For example, to decide whether $2x_1 - x_2/3 + \frac{5}{4}x_3 > 4$, we start with $4B$, and we convert positive terms to A and negative terms to B with the correct rational multipliers: reactions $X_1 \to 2A$ and $4X_3 \to 5A$ for the positive terms and $3X_2 \to B$ for the negative term. The reactions of Lemma 3.3 then properly decide whether $A > B$, which, since the above reactions converge to $A = 2x_1 + \frac{5}{4}x_3$ and $B = 4 + x_2/3$, is true if and only if the threshold predicate $2x_1 - x_2/3 + \frac{5}{4}x_3 > 4$ holds. ◀

We will now show that Boolean combinations of threshold predicates are also robustly decidable. To show this result, we first prove a lemma that lets us assume CRDs that robustly compute predicates are of a convenient form: without loss of generality, we can assume that a CRD has exactly one yes and no voter. Furthermore, both of their concentrations converge exactly to 1 and 0 as $t \to \infty$.

► **Lemma 3.5.** *Let $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon_{\text{yes}}, \Upsilon_{\text{no}}, \mathbf{i})$ robustly decide the predicate $\phi : \mathbb{R}^k_{\geq 0} \to \{0,1\}$. Then there is a CRD $\mathcal{D}' = (\Lambda', R', \Sigma', \{Y\}, \{N\}, \mathbf{i}')$ that robustly decides ϕ with exactly one yes voter species Y and one no voter species N. Furthermore, the concentration of these voters satisfy $Y(t) + N(t) = 1$ for all $t \in \mathbb{R}_{\geq 0}$.*

Proof sketch. A full proof is given in Section A. For each original yes voter V_{y}, we add a reaction $V_{\text{y}} + N \to V_{\text{y}} + Y$, and similarly $V_{\text{n}} + Y \to V_{\text{n}} + N$, to influence the new voters in the correct direction. ◄

This result shows that Definition 2.6 is equivalent to a model in which we require exactly one yes voter and no voter. Furthermore, this lemma allows us to insist that correct voting species do not just remain above zero, but in fact converge to a particular value.

► **Lemma 3.6.** *Let $\phi_1 : \mathbb{R}^k_{\geq 0} \to \{0,1\}$ and $\phi_2 : \mathbb{R}^k_{\geq 0} \to \{0,1\}$ be robustly decidable predicates. Then the following are also robustly decidable: $\overline{\phi_1}$, $\phi_1 \wedge \phi_2$, and $\phi_1 \vee \phi_2$.*

Proof sketch. A full proof is given in [18]. To decide $\overline{\phi_1}$, it is enough to swap the yes and no voters. To decide $\phi_1 \wedge \phi_2$ or $\phi_1 \vee \phi_2$ we use the following construction. Similarly to Lemma 3.5, we add new voter species that are "influenced" by the voters of the CRDs deciding ϕ_1 and ϕ_2. The new voters are $V_{\text{yy}}, V_{\text{ny}}, V_{\text{yn}}, V_{\text{nn}}$, where the first subscript represents a vote for ϕ_1 and the second subscript a vote for ϕ_2. Yes-voters Y for ϕ_1 influence the new voters via $Y + V_{\text{ny}} \to Y + V_{\text{yy}}$ and $Y + V_{\text{nn}} \to Y + V_{\text{yn}}$ and no-voters N influence via $N + V_{\text{yy}} \to N + V_{\text{ny}}$ and $N + V_{\text{yn}} \to N + V_{\text{nn}}$. Voters for ϕ_2 similarly influence the second subscript. To decide $\phi_1 \vee \phi_2$, let $V_{\text{yy}}, V_{\text{ny}}, V_{\text{yn}}$ be the yes voters. To decide $\phi_1 \wedge \phi_2$, let V_{yy} be the yes voter. ◄

The following is the first main result of this paper.

► **Theorem 3.7.** *Every multi-threshold predicate is robustly decidable by a continuous CRD.*

Proof. Immediate from induction on the number of threshold predicates defining the multi-threshold predicate; Lemma 3.4 is the base case and Lemma 3.6 is the inductive case. ◄

3.2 Piecewise affine functions are robustly computable

The next definition captures the idea that a species S converges because reactions stop changing S, as opposed to reactions producing S at the same rate other reactions consume S.

► **Definition 3.8.** *Consider CRN $\mathcal{C} = (\Lambda, R)$ with initial state $\mathbf{x}$. Recall $\rho_t(\alpha)$ is the rate of $\alpha \in R$ at time t. We say that from $\mathbf{x}$, a species $S \in \Lambda$* approaches static steady state*, if for every reaction $\alpha = (\mathbf{r}, \mathbf{p}, k) \in R$ such that $\mathbf{r}(S) \neq \mathbf{p}(S)$, $\lim_{t\to\infty} \rho_t(\alpha) = 0$.*

Note that if S approaches static steady state then $\lim_{t\to\infty} S(t)$ exists and is finite. In other words, although the CRN may not converge to a single state (perhaps some species oscillate or diverge to ∞), not only does the value of $S(t)$ converge, a stronger condition holds: the CRN converges to rate 0 of every reaction net producing or net consuming S. This contrasts *dynamic steady state*, e.g., for $S \underset{1}{\overset{1}{\rightleftharpoons}} A$, where S approaches a limit, but the reactions producing/consuming S have positive rate even at steady state (where $S = A$).

▶ **Lemma 3.9.** *Every rational floor-affine function $f(\mathbf{x}) = \max(0, h+\sum_{i=1}^{k} w_i \cdot \mathbf{x}(i))$ is stably and robustly computable by a continuous feedforward CRC with output species Y, such that Y approaches static steady state and Y's concentration never exceeds $h + \sum_{i,w_i>0} w_i \cdot \mathbf{x}(i)$.*

Proof. Since all feedforward CRCs that stably compute a function also robustly compute it (Lemma 2.9), it suffices to define a feedforward CRC stably computing f. Define the CRC $\mathcal{C} = (\Lambda, R, \Sigma, Y, \mathbf{i})$, where $\Sigma = \{X_1, \ldots, X_k\}$, $\Lambda = \Sigma \cup \{Y, Y^-\}$, and $\mathbf{i}$ and R are defined below. Let $f(\mathbf{x}) = \max(0, g(\mathbf{x}))$ be rational floor-affine, with $g : \mathbb{R}^k_{\geq 0} \to \mathbb{R}$ a rational affine function $g(\mathbf{x}) = h + \sum_{i=1}^{k} \frac{n_i}{d_i} \cdot \mathbf{x}(i)$, for $h \in \mathbb{R}$, $d_1, \ldots, d_k \in \mathbb{Z}$, and $n_1, \ldots, n_k \in \mathbb{Z}_{>0}$; note we have written each rational w_i as $\frac{n_i}{d_i}$. Start with initial context $\mathbf{i} = \{hY\}$ if $h > 0$ and $\mathbf{i} = \{|h|Y^-\}$ if $h \leq 0$. For every i such that $d_i > 0$, add reaction $n_i X_i \to d_i Y$ to R, and for every i such that $d_i < 0$, add reaction $n_i X_i \to d_i Y^-$ to R. Finally we add reaction $Y + Y^- \to \emptyset$. Except for the last reaction, all reactions are entirely independent – none share a reactant with another reaction. Since the last reaction does not produce any species, $\mathcal{C}$ is feedforward.

From any reachable state, execute reactions of the form $n_i X_i \to d_i Y$ and $n_i X_i \to d_i Y^-$ until all inputs are gone. By inspection of the reactions, this means $\sum_{i,n_i>0} \frac{n_i}{d_i} \mathbf{x}(i)$ amount of Y has been produced in total by such reactions, and $\sum_{i,n_i<0} \frac{n_i}{d_i} \mathbf{x}(i)$ amount of Y^- has been produced in total by such reactions. Then run reaction $Y + Y^- \to \emptyset$ until both reactants are gone. If $g(\mathbf{x}) > 0$ then Y^- will be limiting and this will result in $\sum_{i,n_i>0} \frac{n_i}{d_i} \mathbf{x}(i) - \sum_{i,n_i<0} \frac{n_i}{d_i} \mathbf{x}(i) = \sum_{i=1}^{k} \frac{n_i}{d_i} \mathbf{x}(i) = g(\mathbf{x}) = f(\mathbf{x})$, and if $g(\mathbf{x}) \leq 0$ then Y will go to 0, so $\mathcal{C}$ stably computes $f(\mathbf{x}) = \max(0, g(\mathbf{x}))$.

Since $\mathcal{C}$ is feedforward, its steady state is static [20, Lemma 4.8]. Finally, the fact that Y cannot exceed $h + \sum_{i,w_i>0} w_i \cdot \mathbf{x}(i)$ follows from the fact that we start with hY (if $h > 0$) and the only reactions producing Y are $n_i X_i \to d_i Y$ for i such that $\frac{n_i}{d_i} = w_i > 0$. ◀

The following is the second main result of this paper.

▶ **Theorem 3.10.** *Every threshold-piecewise rational floor-affine function is robustly computable by a continuous CRC.*

Proof. This construction is similar to Lemma 4.4 in [21], though that proof was different (and simpler), applying to stable computation in the discrete CRN model. For a threshold-piecewise rational floor-affine function $f : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$, recall the floor-affine functions $f_1, \ldots, f_l : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ and the multi-threshold predicates $\phi_1, \ldots, \phi_\ell : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$ from Definition 2.3. We equivalently think of each ϕ_j as a set $R_j = \phi_j^{-1}(1)$ that makes ϕ_j true, i.e., f partitions the $\mathbb{R}^k_{\geq 0}$ into ℓ disjoint regions $R_1, \ldots, R_\ell$.

Let $j \in \{1, \ldots, \ell\}$. By Lemma 3.6 there is a CRD $\mathcal{D}_j = (\Lambda_j^{\mathcal{D}}, R_j^{\mathcal{D}}, \Sigma_j^{\mathcal{D}}, \{T_j\}, \{F_j\}, \mathbf{i}_j^{\mathcal{D}})$, robustly deciding ϕ_j, where we may assume a single yes voter T_j and single no voter F_j for each by Lemma 3.5. By Lemma 3.9 there is a CRC $\mathcal{C}_j = (\Lambda_j^{\mathcal{C}}, R_j^{\mathcal{C}}, \Sigma_j^{\mathcal{C}}, Y_j, \mathbf{i}_j^{\mathcal{C}})$ robustly computing f_j. We construct the CRC $\mathcal{C} = (\Lambda, R, \Sigma, Y^P, \mathbf{i})$ to robustly compute f, where $\Sigma = \{X_1, \ldots, X_k\}$. We transform the input species from Σ to those of each $\Sigma_i^{\mathcal{D}}, \Sigma_i^{\mathcal{C}}$ by adding reactions of the form, for each $1 \leq i \leq k$, $X_i \to X_i^{1,\mathcal{D}} + X_i^{1,\mathcal{C}} + \cdots + X_i^{\ell,\mathcal{D}} + X_i^{\ell,\mathcal{C}}$ (similar to the construction in the proof of Lemma 3.6).

Ideally we would convert the correct "local" output Y_j to the "global" output Y, but $\mathcal{C}_j$ may need to consume Y_j; in fact it provably must do so if computing a non-monotone function such as $x_1 - x_2$. To avoid interfering with $\mathcal{C}_j$'s computation, we instead modify its reactions to have additional products that will be used in the conversion. Since we do not change any reactants, and since the new products are new species not appearing in $\mathcal{C}_j$, these new products will preserve the kinetics of the CRC $\mathcal{C}_j$, while enabling the constructed CRC to "read" the output of $\mathcal{C}_j$.[12]

[12] We use a trick known as *dual-rail encoding* [21, 22]; we use it more like [21] as a handy "intermediate" proof technique, not as in [20] in which inputs and outputs of CRCs are encoded in dual-rail.

For each reaction net producing p copies of Y_j in $\mathcal{C}_j$, for example $A + Y_j \to B + 3Y_j$ that net produces 2 Y_j, we modify the reaction to also produce p copies of a new species Y_j^P: $A + Y_j \to B + 3Y_j + 2Y_j^P$. For each reaction net consuming c copies of Y_j in $\mathcal{C}_j$, for example $B + 4Y_j \to Y_j$ that net consumes 3 Y_j, we modify the reaction to also produce c copies of a new species Y_j^C: $B + 4Y_j \to Y_j + 3Y_j^C$. This maintains that $Y_j(t) = Y_j^P(t) - Y_j^C(t)$ for all t. Y_j^P "records" the total amount of Y_j that has ever been produced by any reaction, and Y_j^C the total amount of Y_j that has ever been consumed, so that their difference is the net amount of Y_j produced. If $\mathbf{i}_j^{\mathcal{C}}(Y_j) > 0$, we also modify $\mathbf{i}$ so that $\mathbf{i}(Y_j^P) = \mathbf{i}_j(Y_j)$, i.e., if $\mathcal{C}_j$ starts with some Y_j already "produced", we want to start with the same amount of Y_j^P to maintain $Y_j(t) = Y_j^P(t) - Y_j^C(t)$. (captured by the multiset term $\mathbf{y}$ below)

Let $\Lambda = \{Y^P, Y^C\} \cup \Sigma \cup \bigcup_{j=1}^{\ell} \Lambda_j^{\mathcal{D}} \cup \Lambda_j^{\mathcal{C}} \cup \{Y_j^P, Y_j^C\}$, and $\mathbf{i} = \sum_{j=1}^{\ell} \mathbf{i}_j^{\mathcal{D}} + \mathbf{i}_j^{\mathcal{C}} + \ \mathbf{y}$, where $\mathbf{y}(Y_j^P) = \mathbf{i}_j^{\mathcal{C}}(Y_j)$ and $\mathbf{y}(S) = 0$ for all other $S \in \Lambda$.

R includes all reactions from each $\mathcal{C}_j$ and $\mathcal{D}_j$, after modifying some as explained above, and the following reactions for each $1 \le j \le \ell$:

$$T_j + Y_j^P \overset{k_{j1}}{\to} T_j + \hat{Y}_j^P + Y^P \tag{10}$$

$$T_j + Y_j^C \overset{k_{j2}}{\to} T_j + \hat{Y}_j^C + Y^C \tag{11}$$

$$F_j + \hat{Y}_j^P \overset{k_{j3}}{\to} F_j^P + Y_j^P + Y^C \tag{12}$$

$$F_j + \hat{Y}_j^C \overset{k_{j4}}{\to} F_j^C + Y_j^C + Y^P \tag{13}$$

$$Y^P + Y^C \overset{k}{\to} \emptyset \tag{14}$$

We think of the outputs Y_i^P, Y_i^C produced by $\mathcal{C}_i$ as "inactive", with "active" versions $\hat{Y}_i^P, \hat{Y}_i^C$. Then T_j activates them and F_j deactivates them, while maintaining that as much Y^P, Y^C is produced as the amount of $\hat{Y}_j^P, \hat{Y}_j^C$ that has been net activated by the voters T_j, F_j. Furthermore, to ease analysis of the ODEs, we ensure that Y^P and Y^C do not influence the reaction rates. For, say, reaction (12) to reverse the effect of reaction (10) straightforwardly, it would consume Y^P as a reactant. However, with dual rail it is equivalent to produce Y^C, simplifying the expression for the rate of (12).

By inspection of reactions (10), (11), (12), (13), (14), as well as the modified reactions of $\mathcal{C}_j$, one can verify the conservation law that for all t,

$$\sum_{j=1}^{\ell} \hat{Y}_j^P(t) - \sum_{j=1}^{\ell} \hat{Y}_j^C(t) = Y^P(t) - Y^C(t). \tag{15}$$

since each reaction changes each side of (15) by the same amount. Similarly, the following conservation law holds for each $1 \le j \le \ell$:

$$Y_j^P(t) - Y_j^C(t) + \hat{Y}_j^P(t) - \hat{Y}_j^C(t) = Y_j(t). \tag{16}$$

This completes the description of the construction of $\mathcal{C}$.

We now prove that $\mathcal{C}$ robustly computes f. Let $\mathbf{x} \in \mathbb{R}_{\ge 0}^k$. Assume without loss of generality that $\phi_1(\mathbf{x}) = 1$ (thus $\phi_j(\mathbf{x}) = 0$ for all $j > 1$), i.e., that $f(\mathbf{x}) = f_1(\mathbf{x})$, so that our goal is to converge to activating all $\hat{Y}_1^P, \hat{Y}_1^C$ and deactivating all $\hat{Y}_j^P, \hat{Y}_j^C$ for $j > 1$. Since each $\mathcal{D}_j$ robustly decides ϕ_j, $\lim_{t\to\infty} T_1 = 1$, $\lim_{t\to\infty} F_1 = 0$, and for all $j > 1$, $\lim_{t\to\infty} T_j = 0$, $\lim_{t\to\infty} F_j = 1$.

By the conservation law shown above (15), to demonstrate that $\lim_{t\to\infty} Y^P(t) = f_1(\mathbf{x})$, it suffices to show that $\hat{Y}_1^P(t) - \hat{Y}_1^C(t)$ approaches $f_1(\mathbf{x})$ while $\hat{Y}_j^P(t)$ and $\hat{Y}_j^C(t)$ approach 0 for all $j > 1$. We first will show that for all $j > 1$, $\hat{Y}_j^P(t)$ and $\hat{Y}_j^C(t)$ approach 0. For each j, the ODE for $\hat{Y}_j^P$ is given by $\frac{d}{dt}\hat{Y}_1^P = k_{j_1}T_jY_j^P - k_{j_3}F_j\hat{Y}_j^P$. We apply Lemma 3.2 to obtain that $\hat{Y}_j^P$ is bounded as

$$\hat{Y}_j^P \leq \frac{5k_{j_1}T_j(t)Y_j^P(t)}{k_{j_3}F_j(t) + (t^2+1)^{-1}} + Kk_{j_3}\exp\left(-\int_0^t F_j(s)ds\right).$$

By the correctness of $\mathcal{D}_j$, for all $j > 1$ the concentrations $T_j(t)$ and $F_j(t)$ approach 0 and 1 respectively as t approaches infinity. Further, the function $Y_j^P(t)$ is bounded above by the construction in Lemma 3.9. Hence, the quotient term converges to zero as t approaches infinity. We also observe that $F_j(t)$ approaching 1 implies that the integral $\int_0^t F_j(s)ds$ diverges as $t \to \infty$, which shows that the exponential term also converges to 0. This demonstrates that $\lim_{t\to\infty} \hat{Y}_j^P(t) = 0$, and a similar argument with the ODE for $\hat{Y}_j^C(t)$ shows that $\lim_{t\to\infty} \hat{Y}_j^C(t) = 0$ as well. To show that $\hat{Y}_1^P(t) - \hat{Y}_1^C(t)$ approaches $f_1(\mathbf{x})$, we appeal to the second conservation law derived (16). Since $\mathcal{C}_1$ robustly computes f_1, the limit $\lim_{t\to\infty} Y_1(t) = f_1(\mathbf{x})$. Hence, to show the result we can show simply show that $Y_j^P(t)$ and $Y_j^C(t)$ approach zero. Indeed, the ODE for $Y_j^P(t)$ is given by $\frac{d}{dt}Y_1^P(t) = k_3F_1(t)\hat{Y}_1^P(t) + P_1(\Lambda_1, t) - k_1T_1(t)Y_1^P(t)$ where $P_1(\Lambda_1, t)$ is a function determined by the production of Y_1^P in the CRC $\mathcal{C}_1$. We apply Lemma 3.2 to the ODE with $p(t) = T_1(t)$ and $g(t) = F_1(t)\hat{Y}_1^P(t) + P(\Lambda_1, t)$ to obtain the bound $Y_1^P(t) \leq \frac{k_3F_1(t)\hat{Y}_1^P(t)+P_1(\Lambda_1,t)}{k_1T_1(t)+(t^2+1)^{-1}} + K\exp\left(-\int_0^t k_1T_1(s)ds\right)$. The construction given in Lemma 3.9 has Y_1 approach a static steady state, so the rate of production $P_1(\Lambda_1, t)$ tends to zero as $t \to \infty$. Furthermore, and duel to the previous case, the correctness of $\mathcal{D}_1$ shows that $F_1(t) \to 0$ and $T_1(t) \to 1$, so $Y_1^P(t) \to 0$. Similar arguments also demonstrate that $Y_1^C(t) \to 0$ as well. To complete the result, we note that by inspection of the reactions, at any time t we have $Y^P(t) \geq Y^C(t)$, so reaction (14) causes $Y^C(t) \to 0$. Putting it all together, we have shown the desired limit:

$$\begin{aligned}
\lim_{t\to\infty} Y^P(t) &= \lim_{t\to\infty}\left(\sum_{j=1}^{\ell}\hat{Y}_j^P(t) - \sum_{j=1}^{\ell}\hat{Y}_j^C(t) + Y^C(t)\right) \\
&= \lim_{t\to\infty}\left(\hat{Y}_1^P(t) - \hat{Y}_1^C(t)\right) + \lim_{t\to\infty}\left(\sum_{j=2}^{\ell}\hat{Y}_j^P(t) - \sum_{j=2}^{\ell}\hat{Y}_j^C(t)\right) + \lim_{t\to\infty} Y^C(t) \\
&= f_1(\mathbf{x}) + 0 + 0 = f(\mathbf{x}).
\end{aligned}$$

◀

4 Conclusion

Motivated by the limitations of stable predicate computation, we investigated the robust computation of predicates and numerical functions. While we established positive results on what can be robustly computed, namely multi-threshold predicates (Definition 2.2) and robustly piecewise floor-affine functions (Definition 2.3), the limitations of robust computation remains an open question. We conjecture the positive results are tight, i.e., *exactly* the multi-threshold predicates and robustly piecewise floor-affine functions are robustly computable.

We have assumed the presence of initial context, for example to help include a positive amount of some voter species in all of our constructions, which gave the nice invariant that the sum of voter concentrations is preserved. This choice seems as though it is not strictly necessary, since our construction could instead generate voter species. We conjecture that a leaderless model, without initial context, would only restrict threshold predicates to a constant threshold of 0 and constrain functions to piecewise rational floor-*linear* functions, rather than affine.

Another possible extension of this work is employing the so-called *dual-rail* representation [20] to accommodate negative values as inputs. This means that a (possibly negative) value x is represented as the difference between two nonnegative species concentrations $X^+ - X^-$. This is known to slightly reduce the class of stably computable functions $f : \mathbb{R}^k \to \mathbb{R}$; such a function is stably computable using the dual-rail representation (for both inputs and output) if and only if it is piecewise rational linear and *continuous*.

In the construction given in Theorem 3.10, we employed a technique to modify upstream CRCs, ensuring controlled composition even when certain reactions within these CRCs consumed the output species. This structured approach to dependency management could naturally extend to the broader tricky problem of CRN composition [19, 30, 34].

References

1 Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L Rivest. Time-space trade-offs in population protocols. In *Proceedings of the twenty-eighth annual ACM-SIAM symposium on discrete algorithms*, pages 2560–2579. SIAM, 2017. doi:10.1137/1.9781611974782.169.

2 Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2221–2239. SIAM, 2018. doi:10.1137/1.9781611975031.144.

3 Dan Alistarh, Rati Gelashvili, and Milan Vojnović. Fast and exact majority in population protocols. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, pages 47–56, 2015. doi:10.1145/2767386.2767429.

4 Dana Angluin, James Aspnes, Zoë Diamadi, Michael J Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. In *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 290–299, 2004. doi:10.1145/1011767.1011810.

5 Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, pages 235–253, March 2006. doi:10.1007/S00446-005-0138-3.

6 Dana Angluin, James Aspnes, and David Eisenstat. Stably computable predicates are semilinear. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 292–299, 2006. doi:10.1145/1146381.1146425.

7 Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21:183–199, 2008. doi:10.1007/S00446-008-0067-Z.

8 Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, 2008. doi:10.1007/S00446-008-0059-Z.

9 Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007. doi:10.1007/S00446-007-0040-2.

10 Guillaume Aupy and Olivier Bournez. On the number of binary-minded individuals required to compute sqrt(1/2). *Theoretical Computer Science*, 412(22):2262–2267, 2011. doi:10.1016/j.tcs.2011.01.003.

11 Stav Ben-Nun, Tsvi Kopelowitz, Matan Kraus, and Ely Porat. An o (log3/2 n) parallel time population protocol for majority with o (log n) states. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 191–199, 2020. doi:10.1145/3382734.3405747.

12 Petra Berenbrink, Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Peter Kling, and Tomasz Radzik. A population protocol for exact majority with o (log5/3 n) stabilization time and θ (log n) states. In *32nd International Symposium on Distributed Computing, DISC 2018*, page 10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, 2018. doi:10.4230/LIPIcs.DISC.2018.10.

13 Petra Berenbrink, Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Peter Kling, and Tomasz Radzik. Time-space trade-offs in population protocols for the majority problem. *Distributed Computing*, 34(2):91–111, 2021. doi:10.1007/S00446-020-00385-0.

14 Andreas Bilke, Colin Cooper, Robert Elsässer, and Tomasz Radzik. Brief announcement: Population protocols for leader election and exact majority with o (log2 n) states and o (log2 n) convergence time. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pages 451–453, 2017. doi:10.1145/3087801.3087858.

15 Olivier Bournez, Philippe Chassaing, Johanne Cohen, Lucas Gerin, and Xavier Koegler. On the convergence of population protocols when population goes to infinity. *Applied Mathematics and Computation*, 215(4):1340–1350, 2009. doi:10.1016/j.amc.2009.04.056.

16 Olivier Bournez, Pierre Fraigniaud, and Xavier Koegler. Computing with large populations using interactions. In *Mathematical Foundations of Computer Science 2012 - 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings*, volume 7464, pages 234–246, 2012. doi:10.1007/978-3-642-32589-2_23.

17 Olivier Bournez, Daniel S Graça, and Amaury Pouly. Polynomial time corresponds to solutions of polynomial ordinary differential equations of polynomial length. *Journal of the ACM (JACM)*, 64(6):1–76, 2017. doi:10.1145/3127496.

18 Kim Calabrese, David Doty, and Mina Latifi. Robust predicate and function computation in continuous chemical reaction networks, 2025. doi:10.48550/arXiv.2506.06590.

19 Cameron Chalk, Niels Kornerup, Wyatt Reeves, and David Soloveichik. Composable rate-independent computation in continuous chemical reaction networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(1):250–260, 2019.

20 Ho-Lin Chen, David Doty, Wyatt Reeves, and David Soloveichik. Rate-independent computation in continuous chemical reaction networks. *Journal of the ACM*, 70(3), May 2023. doi:10.1145/3590776.

21 Ho-Lin Chen, David Doty, and David Soloveichik. Deterministic function computation with chemical reaction networks. *Natural Computing*, 13(4):517–534, 2014. Special issue of invited papers from DNA 2012. doi:10.1007/s11047-013-9393-6.

22 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.

23 Rachel Cummings, David Doty, and David Soloveichik. Probability 1 computation with chemical reaction networks. *Natural Computing*, 15(2):245–261, 2016. Special issue of invited papers from DNA 2014. doi:10.1007/s11047-015-9501-x.

24 David Doty, Mahsa Eftekhari, Leszek Gąsieniec, Eric Severson, Przemyslaw Uznański, and Grzegorz Stachowiak. A time and space optimal stable population protocol solving exact majority. In *2021 IEEE 62nd annual symposium on foundations of computer science (FOCS)*, pages 1044–1055. IEEE, 2022.

25 David Doty and Monir Hajiaghayi. Leaderless deterministic chemical reaction networks. *Natural Computing*, 14(2):213–223, 2015. Preliminary version appeared in DNA 2013. doi:10.1007/S11047-014-9435-8.

26 Moez Draief and Milan Vojnović. Convergence speed of binary interval consensus. *SIAM Journal on control and Optimization*, 50(3):1087–1109, 2012. doi:10.1137/110823018.

27 François Fages, Guillaume Le Guludec, Olivier Bournez, and Amaury Pouly. Strong Turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs. In *Computational Methods in Systems Biology: 15th International Conference, CMSB 2017, Darmstadt, Germany, September 27–29, 2017, Proceedings 15*, pages 108–127. Springer, 2017. doi:10.1007/978-3-319-67471-1_7.

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

29 Cato M. Guldberg and Peter Waage. Studies concerning affinity. *Forhandlinger: Videnskabs-Selskabet i Christinia. Norwegian Academy of Science and Letters*, 35, 1864. English translation in [39].

30 Hooman Hashemi, Ben Chugg, and Anne Condon. Composable Computation in Leaderless, Discrete Chemical Reaction Networks. In *DNA 26: 26th International Conference on DNA Computing and Molecular Programming*, volume 174, pages 3:1–3:18, 2020. doi:10.4230/LIPIcs.DNA.2020.3.

31 George B Mertzios, Sotiris E Nikoletseas, Christoforos L Raptopoulos, and Paul G Spirakis. Determining majority in networks with local interactions and very small local memory. *Distributed Computing*, 30(1):1–16, 2017. doi:10.1007/S00446-016-0277-8.

32 Yves Mocquard, Emmanuelle Anceaume, James Aspnes, Yann Busnel, and Bruno Sericola. Counting with population protocols. In *2015 IEEE 14th International Symposium on Network Computing and Applications*, pages 35–42. IEEE, 2015. doi:10.1109/NCA.2015.35.

33 Yves Mocquard, Emmanuelle Anceaume, and Bruno Sericola. Optimal proportion computation with population protocols. In *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*, pages 216–223. IEEE, 2016. doi:10.1109/NCA.2016.7778621.

34 Eric E Severson, David Haley, and David Doty. Composable computation in discrete chemical reaction networks. *Distributed Computing*, 34(6):437–461, 2021. special issue of invited papers from PODC 2019. doi:10.1007/S00446-020-00378-Z.

35 David Soloveichik, Matthew Cook, Erik Winfree, and Jehoshua Bruck. Computation with finite stochastic chemical reaction networks. *natural computing*, 7:615–633, 2008. doi:10.1007/S11047-008-9067-Y.

36 David Soloveichik, Georg Seelig, and Erik Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398, 2010.

37 PG Spirakis, L Gasieniec, Russell Martin, D Hamilton, and G Stachowiac. Deterministic population protocols for exact majority and plurality. *LIPIcs*, 70:14–1, 2017. doi:10.4230/LIPIcs.OPODIS.2016.14.

38 Niranjan Srinivas, James Parkin, Georg Seelig, Erik Winfree, and David Soloveichik. Enzyme-free nucleic acid dynamical systems. *Science*, 358(6369):eaal2052, 2017.

39 Peter Waage and Cato Maximilian Gulberg. Studies concerning affinity. *Journal of chemical education*, 63(12):1044, 1986. English translation; original paper is [29].

A Proofs for Section 3 (Robust computation by continuous CRNs)

▶ **Lemma 3.4.** *Every threshold predicate is robustly decidable by a continuous CRC.*

Proof. Intuitively, this construction translates the threshold problem to a majority problem. We translate the positive and negative contributions in the weighted sum to concentrations of species A and B respectively. We then use the reactions from the majority problem to detect if $A > B$ (with the threshold h added appropriately to either A or B).

Let $\phi : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$ be a threshold predicate. Without loss of generality that we assume all weights are integers by clearing denominators, so $\phi(\mathbf{x}) = 1 \iff \sum_{i=1}^k w_i \mathbf{x}(i) > h$ with each $w_i \in \mathbb{Z}$ and $h \in \mathbb{R}$. We construct the CRD $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon_{\text{yes}} \Upsilon_{\text{no}}, \mathbf{i})$ as follows. $\Sigma = \{X_1, \ldots, X_k\}$, and Λ is implicitly all species described below. Let $\Upsilon_{\text{yes}} = \{Y\}$ and $\Upsilon_{\text{no}} = \{N\}$ be the yes and no voter species respectively. Let the $\mathbf{i} = \{1Y, 1C, |h|AB\}$, where $AB = A$ if $h > 0$ and $AB = B$ otherwise. We then add the following reactions. For any $w_p > 0$ add reaction $X_p \to w_p A$ and for any $w_n < 0$ add reaction $X_n \to |w_n| B$. We also add the reactions from the CRD computing majority in Lemma 3.3.

Without loss of generality, we may insist that the threshold value h is zero as the reactions $L \to hB$ and $L \to |h|A$ influence the concentration of A and B respectively in the "same way" as the reactions $X_i \to w_i(A/B)$ in the sense that the concentration of each X_i species and L have closed form solutions of decaying exponential functions. Let $\mathcal{P}$ be the set of all p for which $w_p > 0$ and let $\mathcal{N}$ be the set of all n for which $w_n < 0$. Let $a = \sum_{p \in \mathcal{P}} w_p \cdot x_p$ and $b = \sum_{n \in \mathcal{N}} |w_n| \cdot x_n$. We consider three cases.

- $\boldsymbol{a > b}$: It suffices to observe that $\lim_{t\to\infty} A(t) = a - b$ and $\lim_{t\to\infty} B(t) = 0$ since the restricted CRN that only contains species that change the concentration of A and B is feed-forward. A similar argument to the one used to prove the $a > b$ case of Lemma 3.3 then shows the result.
- $\boldsymbol{a < b}$: Symmetric to the previous case.
- $\boldsymbol{a = b}$: Our goal is to show that the concentration of species N converges to 1. Equivalently, we will show that the concentration of the yes voter Y converges to zero as the invariant $Y(t)+N(t) = 1$ is held for all t. The following argument will be very similar to the one we used to prove the correctness of our majority CRD. The CRD induces the following ODE for the concentration $Y(t)$: $Y'(t) = k_1A(t)N(t) - k_2B(t)Y(t) - k_4C(t)Y(t)$. Using the fact that $Y(t)+N(t) = 1$ we can rewrite this ODE as $Y'(t)+Y(t)[k_1A(t)+k_2B(t)+k_4C(t)] = k_1A(t)$. From the initial value problem $C'(t) = -C(t)^3$ with $C(0) = 1$, we obtain that a closed form for the concentration of C is $C(t) = (2k_4t + 1)^{-1/2}$. Hence $C(t)$ is strictly positive, so we can apply Lemma 3.1 to find that $Y(t) \le \frac{k_1A(t)}{k_1A(t)+k_2B(t)+k_4C(t)} + K\exp\left(-\int_0^t k_1A(s) + k_2B(s) + k_4C(s)ds\right)$. With the observation that $B(t) \ge 0$ and $A(t) \ge 0$ for all $t \in \mathbb{R}_{\ge 0}$, and substituting in the closed form for $C(t)$, we obtain the bound $Y(t) \le \frac{k_1A(t)}{k_4C(t)} + K\exp\left(-(2k_4^2t + k_4)\right)$. The exponential term clearly converges to zero, so to prove the result it suffices to show the quotient term converges to zero. We observe that the restricted CRN with only reactions that change the concentration of A and B is the feed-forward CRN:

  $$\begin{aligned} \forall p \in \mathcal{P} : X_p &\to w_pA \\ \forall n \in \mathcal{N} : X_n &\to |w_n|B \\ A + B &\to \emptyset \end{aligned}$$

  This CRN stably computes the function $f(a, b) = \min(a-b, 0)$, which by [20] Corollary 4.11 implies that it robustly computes $\min(a-b, 0)$. This shows that $\lim_{t\to\infty} A(t) = a - b = 0$. Taking the limit of the quotient, we obtain $0/0$, a so-called indeterminate form. Hence, we may apply L'Hôpital's rule to obtain the following estimate.

  $$\begin{aligned} \lim_{t\to\infty} \frac{k_1}{k_4}\frac{A(t)}{C(t)} = \frac{k_1}{k_4}\lim_{t\to\infty}\frac{A(t)'}{C(t)'} &= \frac{k_1}{k_4}\lim_{t\to\infty}\frac{\sum_{p\in\mathcal{P}} k_pX_p - k_3A(t)B(t)}{(2k_4t+1)^{-3/2}} \\ &\le \frac{k_1}{k_4}\lim_{t\to\infty}\frac{\sum_{p\in\mathcal{P}} k_pX_p}{(2k_4t+1)^{-3/2}} \end{aligned}$$

  The equation for $A'(t)$ is from its mass action differential equation. Each input species X_i can be found to have a closed form of $w_ix_ik_ie^{-k_pt}$, which shows that the quotient tends to zero as t goes to infinity. Thus, $\lim_{t\to\infty} Y(t) = 0$ as desired. ◀

▶ **Lemma 3.5.** *Let $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon_{\text{yes}}, \Upsilon_{\text{no}}, \mathbf{i})$ robustly decide the predicate $\phi : \mathbb{R}^k_{\ge 0} \to \{0, 1\}$. Then there is a CRD $\mathcal{D}' = (\Lambda', R', \Sigma', \{Y\}, \{N\}, \mathbf{i}')$ that robustly decides ϕ with exactly one yes voter species Y and one no voter species N. Furthermore, the concentration of these voters satisfy $Y(t) + N(t) = 1$ for all $t \in \mathbb{R}_{\ge 0}$.*

Proof. We construct the CRD $\mathcal{D}'$ as follows: Let $\Sigma' = \Sigma$, $\Lambda' = \Lambda \cup \{Y, N\}$. Keep all reactions from the original CRD, and for each yes voter $V_{\text{y}} \in \Upsilon_{\text{yes}}$, add the reaction $V_{\text{y}} + N \overset{k_{V_{\text{y}}}}{\to} V_{\text{y}} + Y$.

Additionally, for each no voter $V_\mathrm{n} \in \Upsilon_\mathrm{no}$ add the reaction $V_\mathrm{n} + Y \overset{k_{V_\mathrm{n}}}{\to} V_\mathrm{n} + N$. Start with initial context $\mathbf{i}' = \mathbf{i} + \{1Y\}$.

To show that $\mathcal{D}'$ robustly decides ϕ, we will show that for all $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$ such that $\phi(\mathbf{x}) = 1$, the concentration of species Y approaches 1 as t approaches infinity when the initial state is $\mathbf{x} + \mathbf{i} + \{1Y\}$. The case when $\phi(\mathbf{x}) = 0$ is a similar argument.

We make the worst-case assumption that the adversary chooses $k_\mathrm{y} = \min_{V_\mathrm{y} \in \Upsilon_\mathrm{yes}} k_{V_\mathrm{y}}$ to be the same (and smallest) rate constant for all reactions of the form $V_\mathrm{y} + N \overset{k_{V_\mathrm{y}}}{\to} V_\mathrm{y} + Y$, and similarly chooses the rate constant $k_\mathrm{n} = \max_{V_\mathrm{n} \in \Upsilon_\mathrm{no}} k_{V_\mathrm{n}}$ for all reactions of the form $V_\mathrm{n} + Y \overset{k_{V_\mathrm{n}}}{\to} V_\mathrm{n} + N$. This is a safe worst-case assumption, since choosing larger rate constants for reaction $V_\mathrm{y} + N \overset{k_{V_\mathrm{y}}}{\to} V_\mathrm{y} + Y$, or smaller rate constants for $V_\mathrm{n} + Y \overset{k_{V_\mathrm{n}}}{\to} V_\mathrm{n} + N$ would only make the network converge faster to the correct values. Furthermore, scaling all rate constants in the system by the same factor modifies only the timescale of the mass-action trajectory of the ODEs, while preserving its shape and eventual convergence. To simplify the system, we scale all rate constants by $1/k_\mathrm{y}$, ensuring that $k_\mathrm{y} = 1$.

The CRD $\mathcal{D}'$ induces the following ODE for the concentration of the species N: $N'(t) + N(t)(k_\mathrm{n} V_\mathrm{n}(t) + k_\mathrm{y} V_\mathrm{y}(t)) = k_\mathrm{n} V_\mathrm{n}(t)$ We may then apply Lemma 3.2 to obtain the bound $N(t) \leq \frac{5k_\mathrm{n} V_\mathrm{n}(t)}{k_\mathrm{n} V_\mathrm{n}(t) + k_\mathrm{y} V_\mathrm{y}(t) + (t^2+1)^{-1}} + K \exp\left(-\int_0^t k_\mathrm{n} V_\mathrm{n}(s) + k_\mathrm{y} V_\mathrm{y}(s) ds\right)$. By the correctness of $\mathcal{C}$, the concentration $V_\mathrm{y}(t)$ remains strictly positive as $t \to \infty$. Hence, the integral $\int_0^t k_\mathrm{n} V_\mathrm{n}(s) + k_\mathrm{y} V_\mathrm{y}(s)\, ds$ diverges to infinity as $t \to \infty$. This demonstrates that the exponential term tends to zero as t grows large. To see the quotient term goes to zero, the correctness of $\mathcal{C}$ also dictates that $V_\mathrm{n}(t) \to 0$ as $t \to \infty$. This shows that $N(t) \to 0$. ◀

B Stable predicate computation by continuous CRNs

In this section, we show that the class of predicates stably decidable by continuous CRNs is exactly the detection predicates. Intuitively, $\phi : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$ is a *detection predicate* if it is a Boolean combination of questions in the form "is initial concentration of species S positive or not?". We formalize this notion as follows.

▶ **Definition B.1.** *A predicate $\phi : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$ is a* simple detection predicate *if there is a $1 \leq i \leq k$ such that the extension of ϕ (the set $\phi^{-1}(1)$ of all input vectors that make ϕ true) is of the form $\{\mathbf{x} \in \mathbb{R}^k_{\geq 0} \mid \mathbf{x}(i) > 0\}$. A* detection predicate *is one expressible as a finite combination of ANDs, ORs, and NOTs of simple detection predicates.*

In other words, a simple detection predicate is defined by a hyperplane (with rational slopes), with output 1 on one side of the hyperplane and output 0 on the other side and on the hyperplane itself. A detection predicate ϕ is defined by a finite number of hyperplanes that partition $\mathbb{R}^k_{\geq 0}$ into a finite number of regions, and ϕ is constant within each region.

The following is main result of this section. Each direction is proven separately in Sections B.2 and B.3 via Lemmas B.8 and B.11.

▶ **Theorem B.2.** *$\phi : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$ is stably decidable by a continuous CRN if and only if ϕ is a detection predicate.*

B.1 Stable (rate-independent) computation

These definitions are taken from [20]; see Section 2.4 of that paper for justification that the notion of segment-reachability in particular (Definition B.4) is reasonable.

▶ **Definition B.3.** *state* $\mathbf{d}$ *is* straight-line reachable (aka 1-segment reachable) *from state* $\mathbf{c}$*, written* $\mathbf{c} \to^1 \mathbf{d}$*, if* $(\exists \mathbf{u} \in \mathbb{R}^R_{\geq 0})$ $\mathbf{c} + \mathbf{M}\mathbf{u} = \mathbf{d}$ *and* $\mathbf{u}(\alpha) > 0$ *only if reaction* α *is applicable at* $\mathbf{c}$*. In this case write* $\mathbf{c} \to^1_{\mathbf{u}} \mathbf{d}$*.*

Intuitively, by a single segment we mean running the reactions applicable at $\mathbf{c}$ at a constant (possibly 0) rate to get from $\mathbf{c}$ to $\mathbf{d}$. In the definition, $\mathbf{u}(\alpha)$ represents the flux, or total amount executed, of reaction $\alpha \in R$.

▶ **Definition B.4.** *Let* $k \in \mathbb{N}$*. state* $\mathbf{d}$ *is* k-segment reachable *from state* $\mathbf{c}$*, written* $\mathbf{c} \rightsquigarrow^k \mathbf{d}$*, if* $(\exists \mathbf{b}_0, \ldots, \mathbf{b}_k)$ $\mathbf{c} = \mathbf{b}_0 \to^1 \mathbf{b}_1 \to^1 \mathbf{b}_2 \to^1 \ldots \to^1 \mathbf{b}_k$*, with* $\mathbf{b}_k = \mathbf{d}$*. State* $\mathbf{d}$ *is* segment-reachable *(or simply* reachable*) from state* $\mathbf{c}$*, written* $\mathbf{c} \rightsquigarrow \mathbf{d}$*, if* $(\exists k \in \mathbb{N})$ $\mathbf{c} \rightsquigarrow^k \mathbf{d}$*.*

Often Definition B.4 is used implicitly, when we make statements such as, "Run reaction 1 until X is gone, then run reaction 2 until Y is gone", which implicitly defines two straight lines in concentration space.

We now formalize what it means for a CRN to "rate-independently" compute (stably decide) a predicate ϕ. We define the *global output partial function* $\Phi : \mathbb{N}^\Lambda \dashrightarrow \{0,1\}$ as $\Phi(\mathbf{x}) = 1$. If the only voter species with positive concentration in the state $\mathbf{x}$ are yes voters, then $\Phi(\mathbf{x}) = 1$. Conversely, if the only voter species with positive concentration are no voters, $\Phi(\mathbf{x}) = 0$. Lastly, if neither of these conditions is met, the output function $\Phi(\mathbf{x})$ is undefined. We say a state $\mathbf{o}$ is *stable* if, for all $\mathbf{c}$ such that $\mathbf{o} \rightsquigarrow \mathbf{c}$: $\Phi(\mathbf{o}) = \Phi(\mathbf{c})$.

▶ **Definition B.5** (stably decide)**.** *Let* $\phi : \mathbb{R}^k_{\geq 0} \to \{0,1\}$ *be a predicate. We say a CRD* $\mathcal{D}$ stably decides ϕ *if, for all* $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$*, and all* $\mathbf{c}$ *such that* $\mathbf{x} \rightsquigarrow \mathbf{c}$*, there exists a stable state* $\mathbf{o}$ *such that* $\mathbf{c} \rightsquigarrow \mathbf{o}$ *and* $\Phi(\mathbf{o}) = \phi(\mathbf{x})$*. We say a set* A *is* stably decidable *if its indicator function* χ_A *is stably decidable.*

▶ **Definition B.6** (stably compute)**.** *Let* $f : \mathbb{R}^k_{\geq 0} \to \mathbb{R}$ *be a function. We say a CRC* $\mathcal{C} = (\Lambda, R, \Sigma, \{Y\}, \sigma)$ stably computes f *if for all* $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$ *and all* $\mathbf{c}$ *such that* $\mathbf{x} \rightsquigarrow \mathbf{c}$*, there exists a stable state* $\mathbf{o}$ *such that* $\mathbf{c} \rightsquigarrow \mathbf{o}$ *and* $\mathbf{o}(Y) = f(\mathbf{x})$*.*

▶ **Theorem B.7.** *A function* $f : \mathbb{R}^k_{\geq 0} \to \mathbb{R}$ *is stably computable by a continuous CRC if and only if it is positive-continuous and piecewise rational linear.*

A linear function's graph defines a k-dimensional hyperplane in $\mathbb{R}^{k+1}_{\geq 0}$ that passes through the origin (i.e., linear but not affine). If a piecewise rational linear function is positive-continuous, we switch from one linear component f_i to another f_j only where their hyperplanes intersect. Thus the question "*is linear component* f_i *the correct linear component to apply to compute* f *on input* $\mathbf{x}$*?*" is itself a *multi-threshold predicates* as defined in Definition 2.2, but with constant $h = 0$.

B.2 Positive result: All detection predicates are stably decidable by continuous CRNs

We first show that continuous CRNs can stability decide all detection predicates.

▶ **Lemma B.8.** *Every detection predicate is stably decidable by a continuous CRN.*

Proof. To decide the detection predicate $\phi : \mathbb{R}^k_{\geq 0} \to \{0,1\}$, we let $\Sigma = \{X_{\{1\}}, X_{\{2\}}, \ldots, X_{\{k\}}\}$ and let $\Lambda = \{X_U \mid U \subseteq \{1, \ldots, k\}, U \neq \emptyset\}$ be the set of species. The set of yes voters Υ_{yes} is the set of all X_U such that $\phi(\mathbf{x}) = 1$ for any $\mathbf{x}$ with $\text{supp}(\mathbf{x}) = U$ (note this is well-defined since ϕ is a detection predicate; it must have the same output on all $\mathbf{x}$ with identical support) and the set of no voters be each other species. The reactions in the CRN are

$X_U + X_T \to X_{U \cup T}$. This stably decides ϕ, because eventually all species become X_{U^*} where U^* represents the set of species actually present initially. To handle the case of $\mathbf{x} = \mathbf{0}$, start with initial context of some positive amount of a voter V that votes yes or not based on whether $\phi(\mathbf{0}) = 1$, and all other species X_U consume V via $X_U + V \to X_U$. ◀

B.3 Negative result: All stably decidable predicates are detection predicates

In this section we show that, unlike the case of computing functions $f : \mathbb{R}^k \to \mathbb{R}$ with numerical output (the focus of [20]), CRNs stably computing *predicates* are much more severely limited in computational power. We prove this using the fact, proven in [20], that such functions f must be *positive-continuous*, meaning that discontinuities can only occur when some input coordinate $\mathbf{x}(i)$ goes from 0 to positive. (Note the conceptual similarity to detection predicates, which can change their output only when some input coordinate $\mathbf{x}(i)$ goes from 0 to positive.) We connect this to predicates by showing that any CRN $\mathcal{C}$ stably computing a predicate ϕ can be augmented to stably compute a function σ_ϕ that is continuous only if the predicate stably computed by $\mathcal{C}$ is a detection predicate, proving that since such functions σ_ϕ must be continuous, then ϕ must be a detection predicate.

We recall the definition of a positive-continuous function from [20]. Intuitively, a positive-continuous function is only allowed to have discontinuities whenever some input goes from 0 to positive. For example, the function $f(x_1, x_2) = x_1$ if $x_2 > 0$ and $f(x_1, x_2) = 0$ otherwise is positive continuous though not continuous.

▶ **Definition B.9.** *A function* $f : \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is* positive-continuous*, for all* $U \subseteq \{1, \ldots, k\}$*,* f *is continuous on the domain* $D_U = \left\{\mathbf{x} \in \mathbb{R}^k_{\geq 0} \mid \mathbf{x}(i) > 0 \iff i \in U\right\}$*.*

To understand the definition, it helps to understand first what the domains D_U look like. In the case $k = 2$. The domains D_U are the origin, the positive x and y axes, and the set $\{(x, y) \mid x, y > 0\}$. If $U = \{1\}$, then D_U is the positive x axis as the vectors $\mathbf{x} \in D_U$ must satisfy $\mathbf{x}(1) > 0$ and $\mathbf{x}(2) = 0$. A positive continuous function is one which must be continuous inside each domain, but is allowed discontinuities as it moves across boundaries. If useful, a positive continuous function can be understood as a collection of 2^k continuous functions, each defined on a different piece of the positive orthant.

Given a predicate $\phi : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$, define the *sum characteristic function* of ϕ, σ_ϕ : $\mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ for all $\mathbf{x} \in \mathbb{R}^k_{\geq 0}$ by

$$\sigma_\phi(\mathbf{x}) = \begin{cases} \sum_{i=1}^k \mathbf{x}(i) = \|\mathbf{x}\|_1 & \text{if } \phi(\mathbf{x}) = 1 \\ 0 & \text{if } \phi(\mathbf{x}) = 0 \end{cases}$$

▶ **Lemma B.10.** *Let* $\phi : \mathbb{R}^k_{\geq 0} \to \{0, 1\}$*. If* ϕ*'s sum characteristic function* σ_ϕ *is positive-continuous, then* ϕ *is a detection predicate.*

Proof. We show the contrapositive, that if ϕ is not a detection predicate, then the induced σ_ϕ is not positive-continuous. Suppose ϕ is not a detection predicate. Then, for some $U \subseteq \{1, \ldots, k\}$ the region D_U contains points $\mathbf{x}, \mathbf{y} \in D_U$ such that $\phi(\mathbf{x}) \neq \phi(\mathbf{y})$, as all detection predicates cannot change their output within a domain D_U. Note that this implies $D_U \neq \{\mathbf{0}\}$, i.e., $U \neq \emptyset$. Let ℓ denote the straight line connecting $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^k$. Such an ℓ is completely contained in D_U, as D_U is a convex set. Consider the image of ℓ under σ_ϕ, denoted $\sigma_\phi(\ell) \subseteq \mathbb{R}_{\geq 0}$. Since at least one point along ℓ does not satisfy ϕ, we have $0 \in \sigma_\phi(\ell)$. Furthermore, since $\mathbf{x}, \mathbf{y} \neq \mathbf{0}$, there is $\varepsilon > 0$ such that $\|\mathbf{x}\|_1 > \varepsilon$ and $\|\mathbf{y}\|_1 > \varepsilon$. Since $\|\cdot\|_1$

is a linear function, this implies that for all points $\mathbf{z} \in \ell$, $\|\mathbf{z}\|_1 > \varepsilon$. Thus, all non-zero elements of $\sigma_\phi(\ell)$ are greater than ε. This implies 0 is an isolated point of $\sigma_\phi(\ell)$. Since $\sigma_\phi(\ell)$ contains an isolated point, it is not a connected set. As a continuous function must preserve connectedness, this implies that σ_ϕ is not continuous on D_U, i.e., not positive continuous. ◀

▶ **Lemma B.11.** *If $\phi : \mathbb{R}^k_{\geq 0} \to \{0,1\}$ is stably decidable by a CRD, then ϕ is a detection predicate.*

Proof. Let $\mathcal{D}$ be a CRD stably deciding ϕ. We show how to convert $\mathcal{D}$ into a CRC $\mathcal{C}$ that stably computes the sum characteristic function σ_ϕ. Since all functions stably computable by a CRC are positive-continuous [20], Lemma B.10 implies ϕ is a detection predicate.

For each input species X_i, add the reaction $X_i \to X'_i + Y$, where X'_i is the i'th input species to $\mathcal{D}$. For each yes-voter T in $\mathcal{D}$, add the reaction $T + \hat{Y} \to T + Y$, and for each no-voter F, add the reaction $F + Y \to F + \hat{Y}$.

Let $\mathbf{c}$ be a state reached from the initial state $\mathbf{x}$. If any of the input species X_i are present in $\mathbf{c}$, we apply the reaction $X_i \to X'_i + Y$ until all these species are gone from the state. Once this is done, the concentrations of Y and $\hat{Y}$ satisfy $[Y] + [\hat{Y}] = \sum_{i=1}^{k} \mathbf{x}_i$. Denote this reached state by $\mathbf{c}'$. Since $\mathcal{D}$ stably decides ϕ, there is a stable state $\mathbf{o}$ reachable from $\mathbf{c}'$ in which all species present are yes-voters if $\phi(\mathbf{x}) = 1$ and no-voters otherwise. In the former case, we may apply the reaction $T + \hat{Y} \to T + Y$ to convert all $\hat{Y}$ to Y. As no voters are present in the state, it is stable. Furthermore, the concentration of $\hat{Y}$ is exactly $\|\mathbf{x}\|_1$. If $\phi(\mathbf{x}) = 0$, then only no voters are present so running the reaction $F + Y \to F + \hat{Y}$ eventually will remove all Y, stabilizing on the correct output of 0. ◀