Theoretical Computer Science 816 (2020) 114–143

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Robust biomolecular finite automata[☆]

Titus H. Klinge[a,*], James I. Lathrop[b], Jack H. Lutz[b]

[a] Drake University, Des Moines, IA 50311 USA
[b] Iowa State University, Ames, IA 50011 USA

ARTICLE INFO

Article history:
Received 21 December 2018
Received in revised form 30 August 2019
Accepted 8 January 2020
Available online 16 January 2020
Communicated by N. Jonoska

Keywords:
Biomolecular automata
Input/output chemical reaction networks
Concentration signals
Molecular programming
Robustness

ABSTRACT

We present a uniform method for translating an arbitrary nondeterministic finite automaton (NFA) into a deterministic mass action *input/output chemical reaction network* (*I/O CRN*) that simulates it. The I/O CRN receives its input as a continuous time signal consisting of concentrations of chemical species that vary to represent the NFA's input string in a natural way. The I/O CRN exploits the inherent parallelism of chemical kinetics to simulate the NFA in *real time* with a number of chemical species that is *linear* in the size of the NFA. We prove that the simulation is correct and that it is robust with respect to perturbations of the input signal, the initial concentrations of species, the output (decision), and the rate constants of the reactions of the I/O CRN.

1. Introduction

Molecular programming combines computer science principles with the information processing capabilities of DNA and other biomolecules in order to control the structure and behavior of matter at the nanoscale. Molecular programming has its origin in Seeman's development of DNA nanotechnology in the 1980s [39] (indeed, "molecular programming" and "DNA nanotechnology" are still nearly synonymous), but the field has made progress in the present century at a rate whose increase is reminiscent of Moore's law. The achievements of molecular programming are far too numerous to survey here, but they include the self-assembly of virtually any two- or three-dimensional nanoscale structure that one wants to prescribe [19,26,28,38,45], DNA strand displacement networks that simulate logic circuits and neural networks [34–36], and molecular robots that perform various functions while either walking on nanoscale tracks or floating free in solution [13,15,18,40,44,48,49]. All this has been achieved in real laboratory experiments, and applications to synthetic biology, medicine, and computer electronics are envisioned. Theoretical progress includes demonstrations that various molecular programming paradigms are, in principle, Turing universal [3,14,17,23,32,33,41,46,47], thereby indicating that the full generality and creativity of algorithmic computation may be deployed in molecular and biological arenas.

Our objective in this paper is to begin mitigating the "in principle" of the preceding sentence. This is important for two reasons. First, although such theoretical results are steps in the right direction, processes that require unrealistically precise control of unrealistically large numbers of molecules simply cannot be implemented. Second, processes that can be implemented, but only with inordinately precise control of parameters are inherently unreliable and hence inherently unsafe in many envisioned applications. Our objective here is thus to identify a class of computations that can be implemented *robustly* in the molecular world, i.e., implemented in such a way that they will *provably* perform correctly, even when crucial

[☆] This research was supported in part by National Science Foundation Grants 1247051, 1545028, and 1900716.

[*] Corresponding author.
E-mail addresses: titus.klinge@drake.edu (T.H. Klinge), jil@iastate.edu (J.I. Lathrop), lutz@iastate.edu (J.H. Lutz).

https://doi.org/10.1016/j.tcs.2020.01.008
0304-3975/© 2020 Elsevier B.V. All rights reserved.

parameters are perturbed by small amounts. Future research can then strive to enhance this robustness and to extend the class of computations that enjoy it.

In this paper we give a uniform method for translating nondeterministic finite automata to chemical reaction networks that implement them robustly. Nondeterministic finite automata (NFAs) are over half a century old [37] and far from Turing universal, but they have many applications and remain an active research topic [8,9,27]. Applications of NFAs that are likely to extend to molecular programming include their uses in monitoring and parsing large data streams and in implementing and verifying secure and/or safe communication protocols. Chemical reaction networks (CRNs) are also over half a century old [5]. Their role in molecular programming did not become fully apparent until recently, when Soloveichik, Seelig, and Winfree [42] showed that there is a systematic method for translating an arbitrary CRN, which is an abstract mathematical object, into a set of DNA strands and complexes that simulates the CRN via toehold-mediated strand displacement. This method has been refined and incorporated into various compilers [7,12], and CRNs are now the programming language of choice for many molecular programming investigations. The two most widely used semantics (operational meanings) for CRNs are deterministic mass action semantics and stochastic mass action semantics. In this paper we use deterministic mass action, which implies that the state of a CRN at any time is determined by the *real-valued concentrations* of its molecular species at that time.

An NFA is a real-time device that reads its input string *sequentially*, left to right, changing states appropriately in response to each symbol prior to reading the next symbol. Accordingly, we translate each NFA to an *input/output CRN* (*I/O CRN*), which is a CRN that receives the NFA's input string formatted as a continuous time *concentration signal* consisting of concentrations of input species that vary to represent the input string in a natural way. (Concentration signals are likely to be useful in other molecular programming contexts, e.g., in modularizing CRN constructions.) Using the inherent parallelism of chemical kinetics, our I/O CRN implements the NFA in *real time*, processing each input symbol before the next one arrives, and it does so with a number of molecular species that is *linear* in the size of the NFA that it implements. Specifically, if the NFA has q states, s symbols, and d transitions, then our I/O CRN consists of two modular components. The first module is a preprocessor that transforms the input concentration signals into approximate square waves and consists of $(s+2)(n+4)$ species and $2(s+2)(n+2)$ reactions, where n is logarithmic in q. The second module, which actually simulates the NFA, has $4q+s+2$ species and $5q+d$ reactions. As one example, the compiler of [12] would then translate these modules into DNA gates and strands for a strand displacement network consisting of $4(s+2)(n+2)$ gates and $7(s+2)(n+1)$ strands for the first module, and $10q+2d$ gates and $24q+5d$ strands for the second module. Other CRN-to-DNA compilers could be used here with some variation of these numbers.

Our translation would thus appear to make small NFAs implementable in laboratories now and NFAs of modest size implementable in the near future. However, a significant caveat here concerns an imperfection in current strand displacement implementations of CRNs. A *catalyst* of a chemical reaction is a molecule that enables, but does not actually participate in, the reaction. That is, the amount of the catalyst in solution is not affected by the reaction. Our model here requires an I/O CRN to only use its inputs as catalysts, i.e., to read its input concentration signal *nondestructively*. We also use catalysis in this manner to modularize our construction and its analysis. In contrast, current strand displacement implementations of catalysis may sequester significant amounts of a catalyst pending subsequent arrival of reactants and fuel molecules, thereby suppressing the amount of the catalyst for an indeterminate amount of time. It is to be hoped that strand displacement implementations of CRNs will soon handle catalysis more faithfully. In the meantime, however, we emphasize that our results here are entirely about the relationship between NFAs and CRNs.

Most importantly, our I/O CRN's correct implementation of the NFA is robust with respect to small perturbations of four things, namely, its input signal, the initial concentrations of its species, its output signal (acceptance or rejection of its input string), and the rate constants of its reactions. One key to achieving this robustness is a signal restoration technique akin to the approximate majority algorithm of [2,10,11].

The rest of this paper is organized as follows. Section 2 defines the I/O CRN model, an extension of the CRN model. Section 3 introduces a specific notion of a requirement and then uses such requirements to specify robustness properties of I/O CRNs. Section 4 gives a construction and theorem for a CRN module that enhances input signals by removing noise. Section 5 presents the main result of the paper, a construction for the robust simulation of an NFA using I/O CRNs and the proof that the construction is correct. Some concluding remarks are given in Section 6. Finally, detailed technical proofs of certain lemmas are provided in the appendices.

2. Input/output reaction networks

The chemical reaction network model used here must, like the sequential automata that it simulates, have a provision for reading its input over a period of time, processing early parts of the input before later parts become available. This section describes a chemical reaction network model with such a provision. Inputs are read as *concentration signals*, which consist of concentrations of designated input species that vary over time under external control. This model takes its name from the fact that its deterministic mass action semantics, developed below, is a special case of the "input/output systems" of control theory.

Formally, we fix a countably infinite set $\mathbf{S} = \{X_0, X_1, \ldots\}$, whose elements we call *species*. Informally, we regard each species as an abstract name of a type of molecule, and we avoid excessive subscripts by writing elements of $\mathbf{S}$ in other ways, e.g., X, Y, Z, $\widehat{X}$, $\widetilde{X}$, etc.

A *reaction* over a finite set $S \subseteq \mathbf{S}$ is formally a triple

$$\rho = (\mathbf{r}, \mathbf{p}, k) \in \mathbb{N}^S \times \mathbb{N}^S \times (0, \infty),$$

where $\mathbb{N}^S$ is the set of functions from S into $\mathbb{N} = \{0, 1, 2, \ldots\}$, and $\mathbf{r} \neq \mathbf{p}$. Since S is finite, it is natural to also regard elements of $\mathbb{N}^S$ as vectors. Given such a reaction ρ, we write $\mathbf{r}(\rho) = \mathbf{r}$, $\mathbf{p}(\rho) = \mathbf{p}$, and $k(\rho) = k$, and we call these three things the *reactant vector*, the *product vector*, and the *rate constant*, respectively, of the reaction ρ. The species in the *support set* $\operatorname{supp}(\mathbf{r}) = \{X \in S \mid \mathbf{r}(X) > 0\}$ are the *reactants* of ρ, and the species in $\operatorname{supp}(\mathbf{p})$ are the *products* of ρ.

We usually write reactions in a more intuitive, chemistry-like notation. For example, if $S = \{X, Y, Z\}$, then we write

$$X + Y \xrightarrow{k} X + 2Z$$

for the reaction $(\mathbf{r}, \mathbf{p}, k)$, where $\mathbf{r}, \mathbf{p} : S \to \mathbb{N}$ are defined by $\mathbf{r}(X) = \mathbf{r}(Y) = 1$, $\mathbf{r}(Z) = 0$, $\mathbf{p}(X) = 1$, $\mathbf{p}(Y) = 0$, and $\mathbf{p}(Z) = 2$.

The *net effect* of a reaction ρ is the (nonzero) vector $\Delta\rho = \mathbf{p}(\rho) - \mathbf{r}(\rho) \in \mathbb{Z}^S$. A species X satisfying $\mathbf{r}(\rho)(X) = \mathbf{p}(\rho)(X) > 0$, as in the example above, is called a *catalyst* of the reaction ρ.

An *input/output chemical reaction network* (*I/O CRN*) is an ordered triple $N = (U, R, S)$, where $U, S \subseteq \mathbf{S}$ are finite; $U \cap S = \emptyset$; R is a finite set of reactions over $U \cup S$; and species in U only appear as catalysts in R. Elements of S are called *state species*, or *operating species*, of N. Elements of U are called *input species* of N.

Given a finite set $W \subseteq \mathbf{S}$ of species, we define the *W-signal space* to be the set $C[W] = C^\omega([0, \infty), [0, \infty)^W)$, where $C^\omega(\mathcal{X}, \mathcal{Y})$ is the set of real analytic functions from $\mathcal{X}$ to $\mathcal{Y}$. A function $\mathbf{w} \in C[W]$ is a *concentration signal* that specifies the *concentration* $\mathbf{w}(t)(Y) \in [0, \infty)$ of each species $Y \in W$ at each time $t \in [0, \infty)$.

For sets $W, W' \subseteq \mathbf{S}$, we also use the set $C[W, W'] = C([0, \infty)^W, [0, \infty)^{W'})$.

Intuitively, an I/O CRN $N = (U, R, S)$ is a system that transforms an input signal $\mathbf{u} \in C[U]$ to an output signal $\mathbf{v}$. We now make this intuition precise.

A *context* of an I/O CRN $N = (U, R, S)$ is an ordered triple $\mathbf{c} = (\mathbf{u}, V, h)$, where $\mathbf{u} \in C[U]$ is an *input signal*, $V \subseteq S$ is a set of *output species*, and $h \in C[S \cup U, V]$ is an *output function*. We write $\mathcal{C}_N$ for the set of contexts of N.

The *deterministic mass action semantics* (or *deterministic mass action kinetics*) of an I/O CRN N specifies how N behaves in a context $(\mathbf{u}, V, h)$.

Let $N = (U, R, S)$ be an I/O CRN. A *state* of N is a vector $\mathbf{x} \in [0, \infty)^S$; an *input state* of N is a vector $\mathbf{u} \in [0, \infty)^U$; and a *global state* of N is a vector $(\mathbf{x}, \mathbf{u}) \in [0, \infty)^{S \cup U}$, where $\mathbf{x}$ is a state of N and $\mathbf{u}$ is an input state of N. (Our double usage of the notation $\mathbf{u}$ for a single input state and also for a function specifying a time-varying input state $\mathbf{u}(t)$ is deliberate and minimizes obfuscation. The same holds for $\mathbf{x}$ and $\mathbf{x}(t)$ below.) For each reaction $\rho \in R$ and each $(\mathbf{x}, \mathbf{u}) \in [0, \infty)^{S \cup U}$, the (*deterministic mass action*) *rate of* ρ *in the global state* $(\mathbf{x}, \mathbf{u})$ is

$$\operatorname{rate}_{\mathbf{x},\mathbf{u}}(\rho) = k(\rho)(\mathbf{x}, \mathbf{u})^{\mathbf{r}(\rho)}, \tag{2.1}$$

where $(\mathbf{x}, \mathbf{u})^{\mathbf{r}(\rho)}$ is the product, for all $Y \in S \cup U$, of c $(\mathbf{x}, \mathbf{u})(Y)^{\mathbf{r}(\rho)(Y)}$. For example, if ρ is the reaction $X + Y \xrightarrow{k} X + 2Z$, where $X \in U$ and $Y, Z \in S$, then $\operatorname{rate}_{\mathbf{x},\mathbf{u}}(\rho) = k\mathbf{u}(X)\mathbf{x}(Y)$. Intuitively, the frequency with which an X and a Y react with one another is proportional to $\mathbf{u}(X)\mathbf{x}(Y)$, and the constant of proportionality k summarizes other factors, not depending on $(\mathbf{x}, \mathbf{u})$ (e.g. temperature, salinity of solution, properties of X and Y), that also govern the rate at which ρ occurs.

For each state species $Y \in S$, we define the *deterministic mass action function* $F_Y : [0, \infty)^{S \cup U} \to \mathbb{R}$ by

$$F_Y(\mathbf{x}, \mathbf{u}) = \sum_{\rho \in R} \operatorname{rate}_{\mathbf{x},\mathbf{u}}(\rho)\Delta\rho(Y) \tag{2.2}$$

for all $\mathbf{x} \in [0, \infty)^S$ and $\mathbf{u} \in [0, \infty)^U$. Then $F_Y(\mathbf{x}, \mathbf{u})$ is the total rate at which the concentration of Y is changing in the global state $(\mathbf{x}, \mathbf{u})$. Now let $(\mathbf{u}, V, h)$ be a context of the I/O CRN N. Then $\mathbf{u}(t)(X)$ is the concentration of each input species $X \in U$ at each time $t \in [0, \infty)$. Hence, if the state of N is $\mathbf{x}(t) \in [0, \infty)^S$ at time t, then the concentration of each state species Y must obey the ordinary differential equation (ODE)

$$y'(t) = F_Y(\mathbf{x}(t), \mathbf{u}(t)). \tag{2.3}$$

If we let $\mathcal{E}_Y$ be the ODE (2.3) for each $Y \in S$, then the *deterministic mass action system* of the I/O CRN N is the coupled system

$$(\mathcal{E}_Y \mid Y \in S) \tag{2.4}$$

of ODEs. If we define the vector-valued function $F : [0, \infty)^{S \cup U} \to \mathbb{R}^S$ by

$$F(\mathbf{x}, \mathbf{u}) = (F_Y(\mathbf{x}, \mathbf{u}) \mid Y \in S) \tag{2.5}$$

for all $\mathbf{x} \in [0, \infty)^S$ and $\mathbf{u} \in [0, \infty)^U$, then the mass action system (2.4) can also be written in the vector form

$$\mathbf{x}'(t) = F(\mathbf{x}(t), \mathbf{u}(t)). \tag{2.6}$$

The I/O CRN N is initialized to a state $\mathbf{x}_0 \in [0,\infty)^S$ at time 0 in the context $(\mathbf{u}, V, h)$, and this state then evolves according to the mass action system (2.6). The *deterministic mass action initial value problem* (*IVP*) of N in the context $(\mathbf{u}, V, h)$ with the initial state $\mathbf{x}_0$ is thus the initial value problem consisting of the mass action system (2.6) together with the initial value condition

$$y(0) = \mathbf{x}_0(Y) \text{ for each } Y \in S. \tag{2.7}$$

By the standard existence-uniqueness theory for ODEs [4,43], this mass action IVP has a solution $\mathbf{x}(t)$ that is defined for all $t \in [0,b)$ for some $b \in (0,\infty]$, and this solution is unique. It is not difficult to show, then, that $\mathbf{x}(t) \in [0,\infty)^S$ holds for all $t \in [0,b)$, i.e., that concentrations remain nonnegative. The I/O CRNs defined in this paper are all very well behaved, so that $b = \infty$, i.e., $\mathbf{x}(t)$ is well defined for all $t \in [0,\infty)$ and all input signals and initial values considered here.

In the context $(\mathbf{u}, V, h)$ of N, the observed output of N is given by the output function $h : [0,\infty)^{S\cup U} \to [0,\infty)^V$. In most cases, this function h is some approximation, due to experimental error, of the zero-error projection function $h_0 : [0,\infty)^{S\cup U} \to [0,\infty)^V$ defined by

$$h_0(\mathbf{x}, \mathbf{u})(Y) = \mathbf{x}(Y) \tag{2.8}$$

for all $Y \in V$. If $\mathbf{x}(t)$ is defined as in the preceding paragraph, then the *output signal* of the I/O CRN N in the context $\mathbf{c} = (\mathbf{u}, h)$ with the initial state $\mathbf{x}_0$ is the (continuous) function $N_{\mathbf{c},\mathbf{x}_0} : [0,\infty) \to [0,\infty)^V$ defined by $N_{\mathbf{c},\mathbf{x}_0}(t) = h(\mathbf{x}(t), \mathbf{u}(t))$ for all $t \in [0,\infty)$.

In the language of control theory [6,16], an *input/output system* is a system of the form (2.6), where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ range over more general state spaces $\mathcal{X}$ and $\mathcal{U}$, together with a function $h : \mathcal{X} \times \mathcal{U} \to \mathcal{V}$ for some space $\mathcal{V}$ of values. The input signal $\mathbf{u}$ is often called a *control signal*, and the output function h is often called a *measurement function*.

In most papers, a *chemical reaction network* (*CRN*) is an ordered pair $N = (S, R)$ such that $(\emptyset, R, S)$ is an I/O CRN as defined here. Such CRNs are *autonomous* in the two equivalent senses that (i) the system (2.6) has the simpler form

$$\mathbf{x}'(t) = F(\mathbf{x}(t)), \tag{2.9}$$

the right-hand side of which only depends on the time t indirectly, via the state $\mathbf{x}(t)$; and (ii) once the initial state $\mathbf{x}(0)$ is determined, the CRN's state evolves according to (2.9), without further outside influence. It is clear by inspection of (2.1)-(2.6) that the deterministic mass action system (2.9) of an autonomous CRN is *polynomial*, meaning that the components of the vector $F(\mathbf{x}(t))$ are polynomial in the components $y(t)$ of $\mathbf{x}(t)$. In contrast, the I/O CRNs considered in the present paper have mass action systems (2.6) that are neither autonomous nor polynomial.

Further discussions of chemical reaction networks with deterministic mass action semantics appear in [21,22,25,31].

We conclude this section by noting that I/O CRNs offer a natural means for modularizing constructions. It is often convenient to write the components of an I/O CRN $N = (U, R, S)$ as $U[N] = U$, $R[N] = R$, and $S[N] = S$. The *join* of a finite family $\mathcal{N}$ of I/O CRNs is the I/O CRN

$$\bigsqcup \mathcal{N} = (U^* \setminus S^*, R^*, S^*),$$

where $U^* = \bigcup_{N\in\mathcal{N}} U[N]$, $R^* = \bigcup_{N\in\mathcal{N}} R[N]$, and $S^* = \bigcup_{N\in\mathcal{N}} S[N]$. If $S[N] \cap S[N'] = \emptyset$ for distinct $N, N' \in \mathcal{N}$, then the reactions of N and N' do not interfere with each other, and $\bigsqcup \mathcal{N}$ is the *modular composition* of the I/O CRNs in $\mathcal{N}$.

3. Requirements and robustness

This section specifies what a requirement for an input/output chemical reaction network is and what it means for a reaction network to satisfy a requirement robustly.

Intuitively, a requirement for an I/O CRN N with an initial state $\mathbf{x}_0$ says that, in any context $\mathbf{c} = (\mathbf{u}, V, h)$ satisfying a context assumption $\alpha(\mathbf{c})$, a desired relationship $\phi(\mathbf{u}, N_{\mathbf{c},\mathbf{x}_0})$ should hold between the input signal $\mathbf{u}$ and the output signal $N_{\mathbf{c},\mathbf{x}_0}$. More formally, a *requirement* for N is an ordered pair $\Phi = (\alpha, \phi)$, where the predicates $\alpha : \mathcal{C}_N \to \{\mathbf{false}, \mathbf{true}\}$ and $\phi : C[U] \times C[V] \to \{\mathbf{false}, \mathbf{true}\}$ are called the *context assumption* and the *input/output requirement* (*I/O requirement*), respectively, of Φ. The I/O CRN N *exactly satisfies* a requirement $\Phi = (\alpha, \phi)$ with the initial state $\mathbf{x}_0 \in [0,\infty)^S$, and we write $N, \mathbf{x}_0 \models \Phi$, if the implication

$$\alpha(\mathbf{c}) \implies \phi(\mathbf{u}, N_{\mathbf{c},\mathbf{x}_0}) \tag{3.1}$$

holds for every context $\mathbf{c} = (\mathbf{u}, V, h) \in \mathcal{C}_N$. The I/O CRN N *exactly satisfies* Φ, and we write $N \models \Phi$, if there exists $\mathbf{x}_0 \in [0,\infty)^S$ such that $N, \mathbf{x}_0 \models \Phi$.

Two things should be noted about the above definition. First, a requirement only concerns input and outputs. Two different I/O CRNs with different sets of state species may satisfy the same requirement. Second, in order for $N \models \Phi$ to hold, a *single* initial state $\mathbf{x}_0$ must cause (3.1) to hold for *every* context $\mathbf{c}$.

It is often sufficient to satisfy a requirement approximately, rather than exactly. To quantify the approximation here, we use the *supremum norm* defined by $\|f\| = \sup_{t\in[0,\infty)} |f(t)|$ for all $f \in C([0,\infty), \mathbb{R}^W)$, where

$$|\mathbf{x}| = \left(\sum_{Y\in W} \mathbf{x}(Y)^2\right)^{1/2}$$

is the Euclidean norm on $\mathbb{R}^W$. It is well known that $\|f-g\|$ is then a well behaved distance between functions $f,g \in C([0,\infty),\mathbb{R}^W)$, hence also between functions $f,g\in C[W]$. For $f\in C[W]$ and $\epsilon\in[0,\infty)$ we thus define the *closed ball of radius ϵ about f in $C[W]$* to be the set

$$B_\epsilon(f) = \{g\in C[W] \mid \|g-f\|\le\epsilon\}.$$

For $\epsilon\in[0,\infty)$ we say that the I/O CRN N *ϵ-satisfies* a requirement $\Phi=(\alpha,\phi)$ with the initial state $\mathbf{x}_0\in[0,\infty)^S$, and we write $N,\mathbf{x}_0 \models_\epsilon \Phi$, if the implication

$$\alpha(\mathbf{c}) \implies (\exists\mathbf{v}\in B_\epsilon(N_{\mathbf{c},\mathbf{x}_0}))\phi(\mathbf{u},\mathbf{v}) \tag{3.2}$$

holds for every context $\mathbf{c}=(\mathbf{u},V,h)\in\mathcal{C}_N$. The I/O CRN N *ϵ-satisfies* Φ, and we write $N\models_\epsilon\Phi$, if there exists $\mathbf{x}_0\in[0,\infty)^S$ such that $N,\mathbf{x}_0\models_\epsilon\Phi$.

It is clear by inspection of (3.1) and (3.2) that $\models$ is equivalent to $\models_0$.

We now come to robustness. Intuitively, an I/O CRN N with an initial state $\mathbf{x}_0$ robustly ϵ-satisfies a requirement $\Phi=(\alpha,\phi)$ if, for every context $\mathbf{c}$ satisfying $\alpha(\mathbf{c})$, the following holds: For every "$\hat{\mathbf{c}}$ close to $\mathbf{c}$," every "$\hat{\mathbf{x}}_0$ close to $\mathbf{x}_0$," and every "$\hat{N}$ close to N," the right-hand side of (3.2) holds with $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ in place of $N_{\mathbf{c},\mathbf{x}_0}$. To make this intuition precise, we define the three phrases in quotation marks.

We have already used the supremum norm to define the distance $\|f-g\|$ between two signals $f,g\in C[W]$. We use the same idea and notation to define the distance between two functions $f,g\in C[W,W']$ and the closed ball of radius ϵ about f in $C[W,W']$. Given contexts $\mathbf{c}=(\mathbf{u},V,h)$ and $\hat{\mathbf{c}}=(\hat{\mathbf{u}},\hat{V},\hat{h})$, and given $\delta_1,\delta_2\in[0,\infty)$, we say that $\hat{\mathbf{c}}$ is *(δ_1,δ_2)-close* to $\mathbf{c}$ if $V=\hat{V}$ and $(\hat{\mathbf{u}},\hat{h})\in B_{\delta_1}(\mathbf{u})\times B_{\delta_2}(h)$.

Given $\mathbf{x},\hat{\mathbf{x}}\in[0,\infty)^S$ and $\delta\in[0,\infty)$, we say that $\hat{\mathbf{x}}$ is *δ-close* to $\mathbf{x}$ if $\hat{\mathbf{x}}\in B_\delta(\mathbf{x})$, where the closed ball $B_\delta(\mathbf{x})$ in $[0,\infty)^S$ is defined in the obvious way using the Euclidean norm.

The definition of "$\hat{N}$ close to N" takes a bit more work, because it allows for the fact that $\hat{N}$ may be an implementation of N in which the "rate constants" are only approximately constant. Nevertheless, the intuition is simple: A *δ-perturbation* of N is a variant $\hat{N}$ of an I/O CRN in which an *adversary* is allowed to *vary* each rate constant k, subject to the constraint that $|\hat{k}(t)-k|\le\delta$ for all $t\in[0,\infty)$.

Formally, a *time-dependent reaction* over a finite set $S\subseteq\mathbf{S}$ is a triple

$$\rho=(\mathbf{r},\mathbf{p},\hat{k})\in\mathbb{N}^S\times\mathbb{N}^S\times C([0,\infty),(0,\infty)).$$

As before, we write $\mathbf{r}(\rho)=\mathbf{r}$, $\mathbf{p}(\rho)=\mathbf{p}$, and $\hat{k}(\rho)=\hat{k}$, and we use more intuitive notions like

$$X+Z\xrightarrow{\hat{k}}2Y+Z,$$

remembering that $\hat{k}$ is now a function of time, rather than a constant. An *input/output time-dependent CRN* (*I/O tdCRN*) is then an ordered triple $\hat{N}=(U,\hat{R},S)$, where U and S are as in the I/O CRN definition and $\hat{R}$ is a finite set of time-dependent reactions over S. The deterministic mass action semantics of an I/O tdCRN $\hat{N}=(U,\hat{R},S)$ is defined in the obvious way, rewriting (2.1)–(2.6) as

$$\mathrm{rate}_{\mathbf{x},\mathbf{u}}(\rho)(t)=\hat{k}(\rho)(t)(\mathbf{x},\mathbf{u})^{\mathbf{r}(\rho)}, \tag{3.3}$$

$$F_Y(\mathbf{x},\mathbf{u},t)=\sum_{\rho\in\hat{R}}\mathrm{rate}_{\mathbf{x},\mathbf{u}}(\rho)(t)\Delta_\rho(Y), \tag{3.4}$$

$$y'(t)=F_Y(\mathbf{x}(t),\mathbf{u}(t),t), \tag{3.5}$$

$$(\mathcal{E}_Y \mid Y\in S), \tag{3.6}$$

$$F(\mathbf{x},\mathbf{u},t)=(F_Y(\mathbf{x},\mathbf{u},t)\mid Y\in S), \tag{3.7}$$

$$\mathbf{x}'(t)=F(\mathbf{x}(t),\mathbf{u}(t),t). \tag{3.8}$$

The *output signal* $\hat{N}_{\mathbf{c},\mathbf{x}_0}$ of an I/O tdCRN $\hat{N}$ in the context $\mathbf{c}$ with initial state $\mathbf{x}_0$ is defined in the now-obvious manner.

Let $N=(U,R,S)$ be an I/O CRN, and let $\delta\in[0,\infty)$. A *δ-perturbation* of N is an I/O tdCRN $\hat{N}=(U,\hat{R},S)$ in which $\hat{R}$ is exactly like R, except that each reaction $(\mathbf{r},\mathbf{p},k)$ is replaced by a time-dependent reaction $(\mathbf{r},\mathbf{p},\hat{k})$ satisfying

$$|\hat{k}(t)-k|\le\delta \tag{3.9}$$

for all $t\in[0,\infty)$.

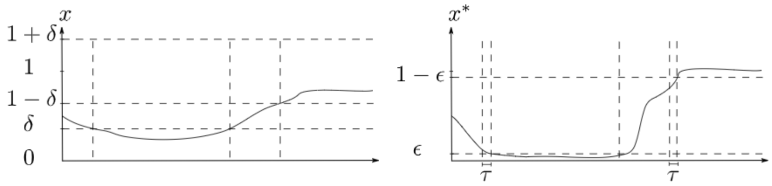

Fig. 1. The two graphs demonstrate the relationship the I/O requirement ϕ imposes on the input and output signals. The output signal $x^*(t)$ is essentially an "enhanced" version of the input signal $x(t)$.

Putting this all together, let $N = (U, R, S)$ be an I/O CRN, let $\mathbf{x}_0 \in [0,\infty)^S$ be an initial state of N, let $\Phi = (\alpha, \phi)$ be a requirement for N, let $\epsilon \in [0,\infty)$, and let $\boldsymbol{\delta} = (\delta_u, \delta_h, \delta_0, \delta_k) \in (0,\infty)^4$ be a vector of *strictly positive* real numbers. We say that N and $\mathbf{x}_0$ *$\boldsymbol{\delta}$-robustly ϵ-satisfy* Φ, and we write $N, \mathbf{x}_0 \models^{\boldsymbol{\delta}}_{\epsilon} \Phi$, if, for every $\mathbf{c} = (\mathbf{u}, V, h) \in \mathcal{C}_N$ satisfying $\alpha(\mathbf{c})$, every $\hat{\mathbf{c}}$ that is (δ_u, δ_h)-close to $\mathbf{c}$, every $\hat{\mathbf{x}}_0$ that is δ_0-close to $\mathbf{x}_0$, and every $\hat{N}$ that is δ_k-close to N, there exists $\mathbf{v} \in B_\epsilon(\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0})$ such that $\phi(\mathbf{u}, \mathbf{v})$ holds. Finally, we say that N *$\boldsymbol{\delta}$-robustly ϵ-satisfies* Φ, and we write $N \models^{\boldsymbol{\delta}}_{\epsilon} \Phi$, if there exists $\mathbf{x}_0 \in [0,\infty)^S$ such that $N, \mathbf{x}_0 \models^{\boldsymbol{\delta}}_{\epsilon} \Phi$.

We extend the notations $\models$, etc., to the satisfaction of finite sets $\boldsymbol{\Phi}$ of requirements Φ in the obvious way.

4. Input enhancement

An essential part of our NFA construction is a device that reduces noise in the input signal. This part of our I/O CRN is a separate module that does not depend on any aspect of the NFA being simulated other than its number of states. In fact, this preprocessing module consists of several identical submodules, one for each input species of the NFA logic module. The goal of the preprocessor is to transform the concentration $x(t)$ of each input species $X \in U$ into a concentration $x^*(t)$ that approximates a square-wave. In particular, when the concentration $x(t)$ is high, then $x^*(t)$ is close to 1, and when $x(t)$ is low, then $x^*(t)$ is close to 0.

We now formally state the requirement of the input enhancer. Let $\tau > 0$, and let $X \in \mathbf{S}$ be a species. Define $\Phi^{(X)} = \Phi^{(X)}(\tau) = (\alpha, \phi)$ to be the requirement where the context assumption $\alpha : C[\{X\}] \to \{\mathbf{false}, \mathbf{true}\}$ is defined by

$$\alpha(\mathbf{u}, V, h) \equiv \Big[V = \{X^*, \overline{X}^*\} \text{ and } h = h_0 \Big], \tag{4.1}$$

where h_0 is the zero-error projection function from equation (2.8). Notice that α requires that the I/O CRN has one input species X and two output species X^* and $\overline{X}^*$. The two output species are a "dual rail" encoding of the input species. Thus, $\overline{X}^*$ represents the Boolean complement of X^* and should be close to 0 if $x(t)$ is high and close to 1 if $x(t)$ is low.

The I/O requirement ϕ of $\Phi^{(X)}$ requires more work to specify, so we begin by defining some helpful terminology and notation. If $I = [a, b] \subseteq [0, \infty)$ is a closed interval, we write $\text{len}(I) = b - a$ to denote the length of the interval. If $I = [a, b]$ and $\text{len}(I) \geq \tau$, we define the *τ-left truncation of I* to be the subinterval $I_\tau = [a + \tau, b]$.

Let $\mathbf{u} \in C[\{X\}]$ be an input signal, and let $\mathbf{v} \in C[\{X^*, \overline{X}^*\}]$ be an output signal. An *input event* is an ordered pair (b, I) where $b \in \{0, 1\}$ is a bit, $I \subseteq [0, \infty)$ is a closed interval with $\text{len}(I) \geq \tau$, and $\mathbf{u}(t)(X) = b$ for all $t \in I$. The set of all input events over $\mathbf{u}$ is denoted **IEV**. Intuitively, an input event is a segment of the input signal which has length at least τ in which the input is held at b. An *output event* is an ordered pair (b, I) where $b \in \{0, 1\}$, $I \subseteq [0, \infty)$ is a closed interval, and the following two conditions hold for all $t \in I$:

1. If $b = 1$, then $\mathbf{v}(t)(X^*) \geq 1$ and $\mathbf{v}(t)(\overline{X}^*) = 0$.
2. If $b = 0$, then $\mathbf{v}(t)(X^*) = 0$ and $\mathbf{v}(t)(\overline{X}^*) \geq 1$.

The set of all output events over $\mathbf{v}$ is denoted **OEV**.

We now define the I/O requirement ϕ of $\Phi^{(X)}$ to be

$$\phi(\mathbf{u}, \mathbf{v}) \equiv \big[(b, I) \in \mathbf{IEV} \implies (b, I_\tau) \in \mathbf{OEV} \big]. \tag{4.2}$$

Intuitively, Φ requires that whenever the input signal has exactly concentration $b \in \{0, 1\}$, the output signal converges to b and $1 - b$ in τ time. Therefore the output species encode both the original bit b and its compliment $1 - b$. As an example, suppose an input signal $x(t)$ contains input events which only get within $\delta \in (0, \frac{1}{3})$ of the bits it is encoding. Then, what we desire is an I/O CRN which is capable of *improving* this signal so that it gets within $\epsilon < \delta$ of the bits it is encoding and only introducing a delay of at most τ. Fig. 1 depicts this relationship in more detail.

We now specify the I/O CRN that is capable of robustly satisfying the requirement $\Phi^{(X)}$. We first state the construction formally and then give an intuitive overview of its operation.

Construction 4.1. *Given strictly positive real numbers τ, ϵ, and $\boldsymbol{\delta} = (\delta_u, \delta_h, \delta_0, \delta_k)$ where $\delta_u \in (0, \frac{1}{3})$, $\delta_h \in [0, \epsilon)$, and $\delta_0 \in (0, \frac{1}{2})$, let $b = \frac{1-\delta_u}{2\delta_u}$ and $n = \lceil 2 \log_b(\frac{8}{\epsilon - \delta_h}) \rceil$. Define the I/O CRN $N^{(X)} = N^{(X)}(\tau, \epsilon, \boldsymbol{\delta}) = (U, R, S)$ where*

$$U = \{X\}, \qquad S = \{X_i \mid 0 \le i \le n\} \cup \{X^*, \overline{X}^*\},$$

and where R consists of the reactions

$$\begin{aligned} X + X_i &\xrightarrow{k_1} X + X_{i+1} \quad (\forall\, 0 \le i < n) \\ X_i &\xrightarrow{k_1} X_0 \qquad\qquad (\forall\, 0 < i \le n) \\ X_n + \overline{X}^* &\xrightarrow{k_2} X_n + X^* \\ X^* &\xrightarrow{k_2} \overline{X}^*, \end{aligned}$$

and the rate constants k_1 and k_2 are defined by

$$\begin{aligned} k_1 &= 2\delta_k + \frac{2n\log(2n)}{\tau(1-\delta_u)} + \frac{2}{\tau}\log\left(10\left(\frac{8}{\epsilon-\delta_h}\right)^2\left(\frac{2}{1-\delta_u}\right)^n\right) + \frac{\delta_k(2+\delta_u)}{\delta_u}, \\ k_2 &= \frac{2}{\tau}\log\left(\frac{3}{\epsilon-\delta_h}\right) + 4\delta_k. \end{aligned}$$

We also define the initial state $\mathbf{x}_0^{(X)}$ of $N^{(X)}$ by

$$\begin{aligned} &\mathbf{x}_0^{(X)}(X^*) = 0, \\ &\mathbf{x}_0^{(X)}(\overline{X}^*) = 1 + \delta_0, \\ &\mathbf{x}_0^{(X)}(X_0) = \frac{10}{\epsilon-\delta_h}\left(\frac{2}{1-\delta_u}\right)^n + \delta_0, \\ &\mathbf{x}_0^{(X)}(X_i) = 0 \quad (\forall\, 0 < i \le n). \end{aligned}$$

The species of $N(X)$ are naturally separated into two parts. The first part is the cascade of species $X_0, \dots, X_n$. This cascade is designed so that every species X_i “falls down” to X_0 at a constant rate, and each species X_i “climbs up” to the next species X_{i+1} at a rate proportional to the input X. As a result, whenever the concentration of X is low, the top of the cascade X_n is *extremely* low. Similarly, whenever the concentration of X is relatively high, the concentration of X_n becomes relatively high.

The second part of the construction consists of the species X^* and $\overline{X}^*$ which are the output species. The sum of the concentrations of these species is always constant, and the presence of the species X_n causes X^* to dominate, and the absence of X_n causes $\overline{X}^*$ to dominate. The cascade and the two species X^* and $\overline{X}^*$ collaborate to enhance the input signal.

The length of the cascade, the rate constants, and the initial concentrations are carefully set and depend on the parameters of $N^{(X)}$. For example, the length of the cascade increases as ϵ decreases since the output must be enhanced by a larger amount. The constants k_1, k_2, $\mathbf{x}_0^{(X)}(X_0)$ are chosen to make the bounding arguments especially easy in the proof of the following theorem. However, the theorem shows that the construction is robust to perturbations to all of these parameters, so they need not be precise.

Theorem 4.2 *(Input enhancement theorem). If $\tau > 0$, $\epsilon \in (0, \frac{1}{2})$, $\boldsymbol{\delta} = (\delta_u, \delta_h, \delta_0, \delta_k)$ with $\delta_u \in (0, \frac{1}{3})$, $\delta_h \in (0, \epsilon)$, $\delta_0 \in (0, \frac{1}{2})$, $\delta_k > 0$, and $N^{(X)} = N^{(X)}(\tau, \epsilon, \boldsymbol{\delta})$ and $\mathbf{x}_0^{(X)}$ are constructed according to Construction 4.1, then*

$$N^{(X)}, \mathbf{x}_0^{(X)} \models_{\epsilon}^{\delta} \Phi^{(X)}(\tau). \tag{4.3}$$

A detailed proof of Theorem 4.2 is provided in Appendix A.

5. Robust I/O CRN simulation of NFAs

In this section we give the main result of this paper: a uniform translation of an NFA to an I/O CRN that simulates it robustly. Finite automata are ubiquitous in computer science, but details and notation vary, so we briefly review the specific model used in this paper. (See, e.g., [30].)

A *nondeterministic finite automaton (NFA)* is an ordered 5-tuple $M = (Q, \Sigma, \Delta, I, F)$, where Q is a finite set of *states*; Σ is a finite *input alphabet*; $I \subseteq Q$ is the set of *initial states*; $F \subseteq Q$ is the set of *accepting states*; and $\Delta : Q \times \Sigma \to \mathcal{P}(Q)$ is the *transition function*. Here we are using the notation $\mathcal{P}(Q)$ for the *power set* of Q, i.e., the set of all subsets of Q. When convenient we identify the transition function Δ with the set of all *transitions* of M, which are triples $(q, a, r) \in Q \times \Sigma \times Q$ satisfying $r \in \Delta(q, a)$. Informally, the *size* of M is determined by the three cardinalities $|Q|$, $|\Sigma|$, and $|\Delta|$.

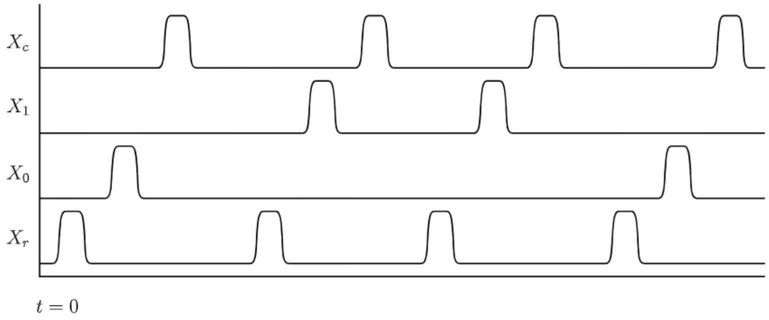

Fig. 2. Example input signal for 0110.

The *extended transition function* of the above NFA M is the function $\widehat{\Delta} : \mathcal{P}(Q) \times \Sigma^* \to \mathcal{P}(Q)$ defined by the recursion

$$\widehat{\Delta}(A, \lambda) = A, \text{ and}$$
$$\widehat{\Delta}(A, wa) = \bigcup_{q \in \widehat{\Delta}(A,w)} \Delta(q, a)$$

for all $A \subseteq Q$, $w \in \Sigma^*$, and $a \in \Sigma$, where λ is the *empty string*. The NFA M *accepts* an input string $w \in \Sigma^*$ if $\widehat{\Delta}(I, w) \cap F \neq \emptyset$, i.e., if there is a chain of transitions leading from some state in I to some state in F. Otherwise, M *rejects* w.

Given an NFA $M = (Q, \Sigma, \Delta, I, F)$, our first objective is to specify a requirement $\Phi = (\alpha, \phi)$ for an I/O CRN $N = (U, R, S)$ to simulate M. The details of R and S can be specified later, but U is an implicit parameter of Φ, so we at this juncture define the set of input species of N to be

$$U = \{X_a \mid a \in \Sigma\} \cup \{X_r, X_c\}, \tag{5.1}$$

where r ("reset") and c ("copy") are special symbols not occurring in Σ.

We now explain how an input $w \in \Sigma^*$ for M is provided to N as a concentration signal. The intuition is that the input w is presented as a sequence of pulses in the concentrations of $|\Sigma| + 2$ species, namely X_r, X_c, and X_a for each $a \in \Sigma$. Each character a in the string w is represented by a sequence of three pulses starting with a pulse in the concentration of X_r, followed by a pulse in the concentration of X_a, and finally ending with a pulse in the concentration of X_c. An example sequence of pulses for the binary string 0110 is shown in Fig. 2. To formally specify this intuition as a context assumption, a bit more terminology is needed. If $I = [a, b]$ and $J = [c, d]$ are closed intervals in $\mathbb{R}$, then I *lies to the left of* J, and we write $I < J$, if $b < c$.

Given an input signal $\mathbf{u} \in C[U]$ for N, we define the following.

1. For $X \in U$, an X-*pulse* in $\mathbf{u}$ is an interval $[b, b+4]$, where $b \in [0, \infty)$, with the following four properties.
   (a) For all $\widehat{X} \in U \setminus \{X\}$ and $t \in [b, b+4]$, $\hat{x}(t) = 0$.
   (b) For all $t \in \{b\} \cup [b+3, b+4]$, $x(t) = 0$.
   (c) For all $t \in [b, b+1] \cup [b+2, b+3]$, $x(t) \in [0, 1]$.
   (d) For all $t \in [b+1, b+2]$, $x(t) = 1$.
2. For $a \in \Sigma$, an a-*event* in $\mathbf{u}$ is an interval $[b, b+12]$ such that $[b, b+4]$ is an X_r-pulse in $\mathbf{u}$, $[b+4, b+8]$ is an X_a-pulse in $\mathbf{u}$, and $[b+8, b+12]$ is an X_c-pulse in $\mathbf{u}$.
3. A *symbol event* in $\mathbf{u}$ is an interval $I \subseteq [0, \infty)$ that is an a-event in $\mathbf{u}$ for some $a \in \Sigma$.
4. The input signal $\mathbf{u}$ is *proper* if there is a sequence $(I_i \mid 0 \leq i < k)$ of symbol events in $\mathbf{u}$ such that $0 \leq k \leq \infty$, $1 < I_i < I_{i+1}$ holds for all $0 \leq i < k-1$, and $\mathbf{u}(t)(X) = 0$ holds for all $X \in U$ and $t \in [0, \infty) \setminus \bigcup_{i=0}^{k-1} I_i$.
5. If $\mathbf{u}$ is proper, the sequence $(I_i \mid 0 \leq i < k)$ testifies to this fact, I_i is an a_i-event for each $0 \leq i < k$, and $t \in [0, \infty)$, then the *string presented by* $\mathbf{u}$ *at time* t is the string

$$w(\mathbf{u})(t) = a_0 a_1 \cdots a_{j-1},$$

   where j is the greatest integer such that $0 \leq j < k$ and $I_j \leq t$.
6. The input signal $\mathbf{u}$ is *terminal* if it is proper and the sequence $(I_i \mid 0 \leq i < k)$ testifying to this fact is finite, i.e., $k \in \mathbb{N}$. In this case, the *terminus* of $\mathbf{u}$ is the time $\tau(\mathbf{u}) =$ *if* $k = 0$ *then* 1 *else* the right endpoint of the interval I_{k-1}, and the *string presented by* $\mathbf{u}$ is the string

$$w(\mathbf{u}) = w(\mathbf{u})(\tau(\mathbf{u})).$$

We now have enough terminology to formally state what it means for an I/O CRN to simulate an NFA. Given an NFA $M = (Q, \Sigma, \Delta, I, F)$, we define the requirement $\Phi = \Phi(M) = (\alpha, \phi)$ as follows. The context assumption α of Φ is defined by

$$\alpha(\mathbf{u}, V, h) \equiv \left[\mathbf{u} \text{ is terminal and } V = \{Y_q \mid q \in F\} \text{ and } h = h_0\right], \tag{5.2}$$

where h_0 is the zero-error projection function (2.8).

The I/O requirement ϕ of Φ is defined by

$$\phi(\mathbf{u}, \mathbf{v}) \equiv \psi_1 \text{ and } \psi_2, \tag{5.3}$$

where ψ_1 and ψ_2 are the formulas

$$\psi_1 \equiv \left[M \text{ accepts } w(\mathbf{u}) \implies (\forall t > \tau(\mathbf{u}))(\exists Y \in V)\mathbf{v}(t)(Y) = 1\right], \tag{5.4}$$

$$\psi_2 \equiv \left[M \text{ rejects } w(\mathbf{u}) \implies (\forall t > \tau(\mathbf{u}))(\forall Y \in V)\mathbf{v}(t)(Y) = 0\right]. \tag{5.5}$$

The two parts ψ_1 and ψ_2 of the I/O requirement correspond to how the I/O CRN should output “accept” and “reject,” respectively. If the input string presents a string that should be accepted, ψ_1 requires that the output signal have at least one species $Y \in V$ that is held at a value of 1 indefinitely. Similarly, if the input string should be rejected, ψ_2 requires that the output signal hold all species in V at a value of 0 indefinitely.

We now specify our translation of an arbitrary NFA into an I/O CRN that simulates it. The I/O CRN consists of two separate modules: the input enhancement module from Section 4, and a module responsible for the NFA logic. We begin by defining the I/O CRN that computes the logic of the NFA, and later we join this I/O CRN with the input enhancement module.

Construction 5.1. *Given an NFA $M = (Q, \Sigma, \Delta, I, F)$ and strictly positive real numbers ϵ and $\boldsymbol{\delta} = (\delta_u, \delta_h, \delta_0, \delta_k)$ satisfying $\epsilon > \delta_h + \delta_0$, we define the I/O CRN $N^* = N^*(M, \epsilon, \boldsymbol{\delta}) = (U^*, R^*, S^*)$ as follows.*

The set U^* is the preprocessed equivalent to the species (5.1) specified earlier, i.e.,

$$U^* = \{X_a^* \mid a \in \Sigma\} \cup \{X_r^*, X_c^*\}.$$

The set S^* contains the following three types of species.

1. State species. *For each state $q \in Q$ there is a species Y_q. Intuitively, the concentration of Y_q is close to 1 in N when M could (as permitted by its nondeterminism) be in state q.*
2. Portal species. *For each state $q \in Q$ there is a species Z_q that is used as a buffer to facilitate transitions into the state q.*
3. Dual species. *For each state species Y_q and portal species Z_q, there are species $\overline{Y}_q$ and $\overline{Z}_q$. We refer to the species Y_q, Z_q as* basic species *in order to further distinguish them from their duals $\overline{Y}_q$, $\overline{Z}_q$. Intuitively, a dual of a basic species is one that has exactly the opposite operational meaning, i.e., when Y_q has high concentration, $\overline{Y}_q$ has low concentration and vice versa.*

We define S^* to be the collection of species of these three types, noting that $|S^*| = 4|Q|$.

The reactions of N^* are of four types, designated as follows.

1. Reset reactions. *For each state $q \in Q$ we have the reaction*

$$X_r^* + Z_q \xrightarrow{k_1} X_r^* + \overline{Z}_q. \tag{5.6}$$

2. Transition reactions. *For each transition $(q, a, r) \in \Delta$ of M we have the reaction*

$$X_a^* + Y_q + \overline{Z}_r \xrightarrow{k_1} X_a^* + Y_q + Z_r. \tag{5.7}$$

3. Copy back reactions. *For each state $q \in Q$ we have the reactions*

$$X_c^* + Z_q + \overline{Y}_q \xrightarrow{k_2} X_c^* + Z_q + Y_q, \tag{5.8}$$

$$X_c^* + \overline{Z}_q + Y_q \xrightarrow{k_2} X_c^* + \overline{Z}_q + \overline{Y}_q. \tag{5.9}$$

4. State restoration reactions. *For each state $q \in Q$ we have the reactions*

$$2Y_q + \overline{Y}_q \xrightarrow{k_2} 3Y_q \tag{5.10}$$

$$2\overline{Y}_q + Y_q \xrightarrow{k_2} 3\overline{Y}_q. \tag{5.11}$$

Note these reactions are an implementation of the termolecular signal restoration algorithm in [29].

The rate constants k_1 and k_2 are defined by

$$k_1 = \frac{30|Q|}{\epsilon - \delta_h - \delta_0}, \tag{5.12}$$

$$k_2 = 18 \log\left(\frac{20|Q|}{\epsilon - \delta_h - \delta_0}\right). \tag{5.13}$$

We define R^* to be the collections of reactions of these four types, noting that $|R^*| = |\Delta| + 5|Q|$. We also note that $U^* \cap S^* = \emptyset$ and species in U^* only appear as catalysts in R^*, so N^* is indeed an I/O CRN.

Intuitively, N^* simulates the NFA M in the following way. The state species Y_q and $\overline{Y}_q$ for $q \in Q$ are used to store the states that M could be in at any time. More specifically, these species encode the set $\widehat{\Delta}(I, w)$ where w is the string processed so far. Whenever the input signal provides another symbol event to N^*, it processes the event in three stages, each corresponding to the three pulses of the symbol event. The first pulse of a symbol event is the "reset" pulse via the species X^*_r. When N^* receives this pulse, it forces all of the concentration of the portal species Z_q into the species $\overline{Z}_q$ using the reactions of equation (5.6). After the X^*_r pulse is completed, every Z_q species has concentration close to 0 and every $\overline{Z}_q$ species has concentration close to 1. This reset process prepares these portal species to compute the transition function.

The second pulse of the symbol event is an X^*_a pulse for some symbol $a \in \Sigma$. When this pulse arrives, N^* computes the transition function of the NFA M using the reactions from equation (5.7). Therefore, after this pulse is processed, the portal species Z_q will be close to 1 if and only if $q \in \widehat{\Delta}(I, wa)$.

The last pulse of the symbol event is the "copy" pulse via the species X^*_c. During this pulse, N^* copies the values of the portal species $Z_q, \overline{Z}_q$ back into the state species $Y_q, \overline{Y}_q$ using reactions (5.8) and (5.9). Therefore, after the X^*_c pulse has been processed, the set $\widehat{\Delta}(I, wa)$ is encoded into the state species $Y_q, \overline{Y}_q$ which completes the computation.

Finally, the reactions from equations (5.10) and (5.11) ensure that the state will remain valid in the absence of a symbol event indefinitely.

Two observations concerning Theorem 5.1 are useful. First, N^* is designed to simulate the nondeterminism of M in real time by computing all transitions in parallel. Second, this parallelism causes *leak* from one state to the next proportional to the number of states $|Q|$. This leak causes the simulation to fail if the input signal is too noisy. To mitigate this, the input enhancer from Section 4 preprocesses the noisy input signal in order to present a signal to N^* that guarantees correct simulation. We also note that although $|Q|$ is encoded into the rate constants k_1 and k_2, these constants need not be precise since we prove that our construction is robust with respect to adversarial manipulation of the rate constants.

We now specify the complete I/O CRN that simulates the NFA which includes the input enhancement module. Recall that our set of input species U consists of $|\Sigma| + 2$ elements, one for each symbol in the input alphabet Σ and two for the special symbols r and c. Our preprocessing module consists of one input enhancing I/O CRN for each input species.

Construction 5.2. *Given NFA $M = (Q, \Sigma, \Delta, I, F)$ and strictly positive real numbers ϵ and $\boldsymbol{\delta} = (\delta_u, \delta_h, \delta_0, \delta_k)$, we define the family of I/O CRNs $\mathcal{N} = \mathcal{N}(M, \epsilon, \boldsymbol{\delta})$ by*

$$\mathcal{N} = \{N^*\} \cup \left\{N^{(X_a)} \mid a \in \Sigma \cup \{r, c\}\right\}, \tag{5.14}$$

where $N^* = N^*(M, \epsilon, \boldsymbol{\delta})$ is constructed according to Construction 5.1 and $N^{(X_a)} = N^{(X_a)}(\frac{1}{2}, \gamma, \boldsymbol{\delta}^*)$ for each $a \in \Sigma \cup \{r, c\}$ is constructed according to Construction 4.1 where

$$\gamma = \frac{\epsilon - \delta_h - \delta_0}{(34|Q|)^4} \tag{5.15}$$

and $\boldsymbol{\delta}^* = (\delta_u, 0, \delta_0, \delta_k)$.

We also define the I/O CRN $N = N(M, \epsilon, \boldsymbol{\delta}) = (U, R, S)$ to be the join of this family of I/O CRNs

$$N = \bigsqcup \mathcal{N}. \tag{5.16}$$

Note that N from Construction 5.2 is indeed an I/O CRN because $\mathcal{N}$ is *modular*, and the set of input species U matches that of equation (5.1) defined earlier.

We now state the main theorem of this paper.

Theorem 5.3. *If $M = (Q, \Sigma, \Delta, I, F)$ is an NFA and ϵ, $\boldsymbol{\delta} = (\delta_u, \delta_h, \delta_0, \delta_k)$ are strictly positive real numbers satisfying*

$$\delta_u, \delta_h, \delta_0, \delta_k < \frac{1}{20}, \tag{5.17}$$

$$\delta_h + \delta_0 < \epsilon, \tag{5.18}$$

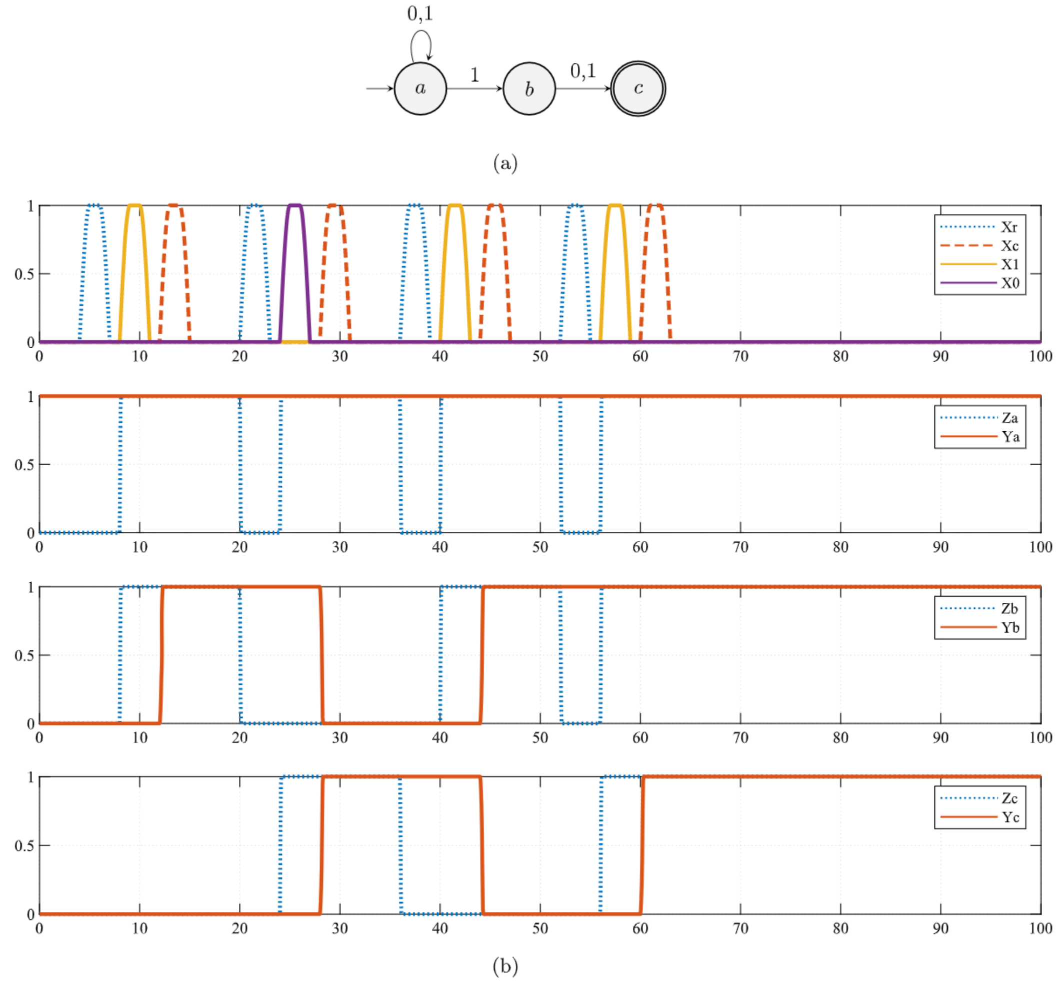

Fig. 3. (a) An NFA that recognizes the language of all strings whose second-to-last bit is 1; (b) A MATLAB simulation of the I/O CRN from Construction 5.2 of this NFA on input 1011. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

and $N = N(M, \epsilon, \boldsymbol{\delta})$ *is constructed according to Construction 5.2, then*

$$N \models_{\epsilon}^{\boldsymbol{\delta}} \Phi(M). \tag{5.19}$$

As an example, Fig. 3 uses MATLAB to show the I/O simulation of the indicated NFA. This NFA being simulated accepts a binary string if and only if its second-to-last bit is a 1.

The rest of this section is devoted to proving Theorem 5.3. We begin by assuming the hypothesis and construct an initial state for N that we use to simulate the NFA. Since N contains many input enhancement I/O CRNs, we initialize N so that each of them are initialized properly according to Construction 4.1. Therefore, let $N^{(X)} = (U^{(X)}, R^{(X)}, S^{(X)})$ be the input enhancement module of N for $X \in U$, and let $\mathbf{x}_0^{(X)}$ be the initial state of $N^{(X)}$ constructed according to Construction 4.1. Now let $\mathbf{x}_0$ be a state of N defined by

$$\begin{aligned}
&(\forall q \in I) && \mathbf{x}_0(Y_q) = 1 = 1 - \mathbf{x}_0(\overline{Y}_q),\\
&(\forall q \in Q \setminus I) && \mathbf{x}_0(Y_q) = 0 = 1 - \mathbf{x}_0(\overline{Y}_q),\\
&(\forall q \in Q) && \mathbf{x}_0(Z_q) = 0 = 1 - \mathbf{x}_0(\overline{Z}_q),
\end{aligned}$$

and

$$(\forall X \in U)(\forall \widehat{X} \in S^{(X)}) \quad \mathbf{x}_0(\widehat{X}) = \mathbf{x}_0^{(X)}(\widehat{X}).$$

The initial state $\mathbf{x}_0$ ensures that every input enhancement module $N^{(X)}$ is initialized properly. The initial state $\mathbf{x}_0$ is also defined so that the state species encode the set of start states I and the portal species encode the empty set. We also note that $\mathbf{x}_0(Y_q) + \mathbf{x}_0(\overline{Y}_q) = 1$ and $\mathbf{x}_0(Z_q) + \mathbf{x}_0(\overline{Z}_q) = 1$ for all $q \in Q$.

Each input enhancement module $N^{(X)} = N^{(X)}(\frac{1}{2}, \gamma, \delta^*)$ is constructed with delay $\tau = \frac{1}{2}$ and robustness parameters $\delta^* = (\delta_u, 0, \delta_0, \delta_k)$. A measurement perturbation of $\delta_h = 0$ is used since the enhanced signals are internally used by N^* and never measured. The number γ throttles the leak introduced by the state species of Q. Furthermore, we know that $N, \mathbf{x}_0 \models_{\gamma}^{\delta^*} \Phi^{(X)}$ for each $X \in U$ where $\Phi^{(X)} = \Phi^{(X)}(\frac{1}{2})$ is the input enhancement requirement from Section 4. This follows from Theorem 4.2 which says that $N^{(X)}, \mathbf{x}_0^{(X)} \models_{\gamma}^{\delta^*} \Phi^{(X)}$ along with the fact that N is a modular composition I/O CRNs which includes $N^{(X)}$ for each $X \in U$.

Since each input enhancer satisfies its requirement, each input event will be enhanced to have at most γ error. This is an important step for N^* to simulate the NFA because of the leak introduced by the state species.

We now enumerate the ODEs generated by N. Using the mass action function (3.5), for each $q \in Q$, the ODEs of the species $Y_q, Z_q, \overline{Y}_q, \overline{Z}_q$ of N are

$$\frac{dy_q}{dt} = k_2 x_c^* z_q \overline{y}_q - k_2 x_c^* \overline{z}_q y_q + k_2 y_q^2 \overline{y}_q - k_2 y_q \overline{y}_q^2, \tag{5.20}$$

$$\frac{dz_q}{dt} = -k_1 x_r^* z_q + \sum_{(s,a,q)\in\Delta} k_1 x_a^* y_s \overline{z}_q, \tag{5.21}$$

$$\frac{d\overline{y}_q}{dt} = -\frac{dy_q}{dt}, \tag{5.22}$$

$$\frac{d\overline{z}_q}{dt} = -\frac{dz_q}{dt}, \tag{5.23}$$

respectively.

Notice that $\frac{dy_q}{dt} + \frac{d\overline{y}_q}{dt} = 0$ and $\frac{dz_q}{dt} + \frac{d\overline{z}_q}{dt} = 0$. This implies that the sum of the concentrations of Y_q and $\overline{Y}_q$ is constant and the sum of the concentrations of Z_q and $\overline{Z}_q$ is constant. Unfortunately, we cannot assume these sums are 1 because of the initial state perturbation. Therefore, for each $q \in Q$ we define the constants

$$p(Y_q) = \mathbf{x}_0(Y_q) + \mathbf{x}_0(\overline{Y}_q) \tag{5.24}$$

$$p(Z_q) = \mathbf{x}_0(Z_q) + \mathbf{x}_0(\overline{Z}_q), \tag{5.25}$$

noting that $1 - \delta_0 < p(Y_q), p(Z_q) < 1 + \delta_0$, since the initial state $\mathbf{x}_0$ can be perturbed by at most δ_0.

We prove that $N, \mathbf{x}_0 \models_{\epsilon}^{\delta} \Phi$ by showing that N and $\mathbf{x}_0$ robustly satisfy a family of weaker requirements. To formally state these requirements, more notation and terminology is needed.

For $A \subseteq Q$ we use $Y_A = \{Y_q \mid q \in A\}$ and $Z_A = \{Z_q \mid q \in A\}$ to denote the set of all state species of A and portal species of A, respectively. For $B \subseteq Q$ and vector $\mathbf{x} \in [0,\infty)^{Y_Q}$, we say that Y_Q *encodes* B *in* $\mathbf{x}$ if $(\forall q \in B)\ \mathbf{x}(Y_q) = p(Y_q)$ and $(\forall q \in Q \setminus B)\ \mathbf{x}(Y_q) = 0$.

We also have terminology for approximately encoding a set. For $\eta \geq 0$, we say that Y_Q *η-encodes* B *in* $\mathbf{x}$ if $(\forall q \in B)$ $|p(Y_q) - \mathbf{x}(Y_q)| < \eta$ and $(\forall q \in Q \setminus B)\ \mathbf{x}(Y_q) < \eta$. We extend this terminology to the set of portal species Z_Q in the obvious way. Furthermore, because an I/O CRN produces a solution of states $\mathbf{x}(t)$ that are indexed by time, we occasionally refer to encoding sets *at time* t when the state $\mathbf{x}(t)$ is clear from context.

We now specify the family of requirements. For $w \in \Sigma^*$, let $\Phi_w = (\alpha_w, \phi_w)$ be a requirement where α_w is defined by

$$\alpha_w(\mathbf{u}, V, h) \equiv \left[\alpha(\mathbf{u}, Y_F, h) \text{ and } w(\mathbf{u}) = w \text{ and } V = Y_Q\right], \tag{5.26}$$

and where ϕ_w is defined by

$$\phi_w(\mathbf{u}, \mathbf{v}) \equiv (\forall t \geq \tau(\mathbf{u}))\left[Y_Q \text{ encodes } \widehat{\Delta}(I, w) \text{ in } \mathbf{v}(t)\right]. \tag{5.27}$$

Therefore the requirement Φ_w requires that if the I/O CRN receives an input that presents the string $w \in \Sigma^*$, then after processing w it must output an encoding of $\widehat{\Delta}(I, w)$.

To show that Theorem 5.3 holds, we prove that $N, \mathbf{x}_0 \models_{\eta}^{\delta^*} \Phi_w$ holds for all $w \in \Sigma^*$ where η is the constant

$$\eta = \frac{\epsilon - \delta_h - \delta_0}{(80|Q|)^2}. \tag{5.28}$$

We prove this via induction over the strings $w \in \Sigma^*$ via the following two lemmas, and then show these lemmas suffice to prove Theorem 5.3.

Lemma 5.4 *(Base case).* $N, \mathbf{x}_0 \models_{\eta}^{\delta^*} \Phi_\lambda$.

Lemma 5.5 *(Induction step). For all $w \in \Sigma^*$ and $a \in \Sigma$*

$$N, \mathbf{x}_0 \models_{\eta}^{\delta^*} \Phi_w \implies N, \mathbf{x}_0 \models_{\eta}^{\delta^*} \Phi_{wa}. \tag{5.29}$$

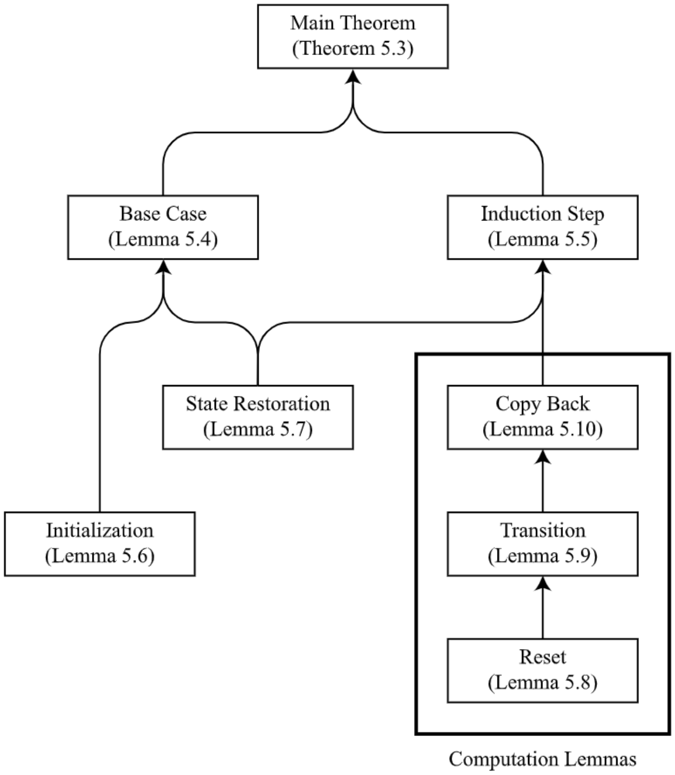

Fig. 4. Refinements of the main theorem into lemmas.

Proof of Theorem 5.3. Assume the hypothesis. Let $\mathbf{c} = (\mathbf{u}, V, h)$ be a context that satisfies $\alpha(\mathbf{c})$, let $\hat{\mathbf{c}} = (\hat{\mathbf{u}}, V, \hat{h})$ be (δ_u, δ_h)-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_0-close to $\mathbf{x}_0$, let $\hat{N}$ be δ_k-close to N, let $w = w(\mathbf{u})$, and let $\hat{\mathbf{c}}_w = (\hat{\mathbf{u}}, Y_Q, h_0)$. It suffices to show that $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to a signal $\mathbf{v} \in C[V]$ such that $\Phi(\mathbf{u}, \mathbf{v})$ is satisfied.

By the induction of Lemmas 5.4 and 5.5, we know that $N, \mathbf{x}_0 \models_{\eta}^{\delta^*} \Phi_w$. It follows that Y_Q η-encodes $\widehat{\Delta}(I, w)$ in $\hat{N}_{\hat{\mathbf{c}}_w,\hat{\mathbf{x}}_0}(t)$ for all $t \geq \tau(\mathbf{u})$. If the NFA M accepts the string w, then $F \cap \widehat{\Delta}(I, w) \neq \emptyset$, so there exists a $q \in F$ such that $\hat{N}_{\hat{\mathbf{c}}_w,\hat{\mathbf{x}}_0}(t)(Y_q) > p(Y_q) - \eta$ for all $t \geq \tau(\mathbf{u})$. Since the perturbed initial state $\hat{\mathbf{x}}_0$ is δ_0-close to $\mathbf{x}_0$, it follows that $p(Y_q) > 1 - \delta_0$. Thus,

$$\hat{N}_{\hat{\mathbf{c}}_w,\hat{\mathbf{x}}_0}(t)(Y_q) > 1 - \delta_0 - \eta$$

for all $t \geq \tau(\mathbf{u})$. We also know that the only difference between $\hat{N}_{\hat{\mathbf{c}}_w,\hat{\mathbf{x}}_0}$ and $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is the effect of the measurement perturbation by δ_h. Thus, $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}(t)(Y_q) > 1 - \delta_0 - \eta - \delta_h$. Finally, since $\epsilon > \delta_h + \delta_0 + \eta$, it follows that $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}(t)(Y_q) > 1 - \epsilon$, and since $Y_q \in V$, the function $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to satisfying ψ_1 of ϕ.

Similarly, if M rejects w, then $F \cap \widehat{\Delta}(I, w) = \emptyset$, therefore for all $Y \in V$ and $t \geq \tau(\mathbf{u})$, $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}(t)(Y) < \eta + \delta_h < \epsilon$. Therefore $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to satisfying ψ_2 of ϕ. It follows that $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to a function $\mathbf{v} \in C[V]$ such that the I/O requirement $\phi(\mathbf{u}, \mathbf{v})$ holds. Therefore $N, \mathbf{x}_0 \models_{\epsilon}^{\delta} \Phi$. □

It remains to be shown that Lemmas 5.4 and 5.5 hold. The proofs of these are extensive and are broken down into several supporting lemmas which are visualized in Fig. 4.

The first two of these supporting lemmas are:

Lemma 5.6 *(Initialization lemma). Y_Q δ_0-encodes I at time $t = \frac{1}{2}$.*

Lemma 5.7 *(State restoration lemma). Let $A \subseteq Q$ be a set of states of the NFA, and let $t_1, t_2 \in [0, \infty)$ be times such that $t_1 + \frac{1}{2} \leq t_2$. If the following two conditions hold:*

1. *Y_Q $\frac{1}{20}$-encodes A at time t_1 and*
2. *$x^*(t) \leq \gamma$ for all $t \in [t_1, t_2]$ and for all $X^* \in U^*$,*

then Y_Q η-encodes A for all $t \in [t_1 + \frac{1}{2}, t_2]$.

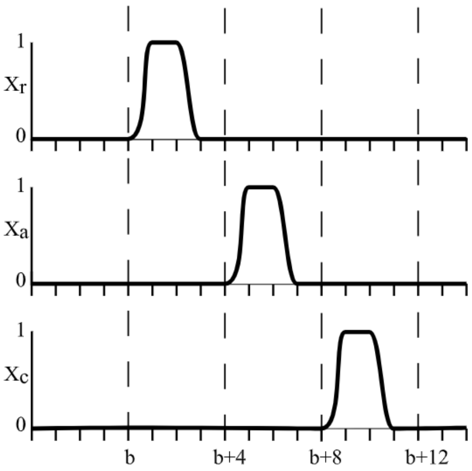

Fig. 5. The X_r-, X_a,- and X_c-pulses of the final a-event.

Lemma 5.6 requires that during the first half-second, the encoding of the initial states is not negatively affected. This ensures that the input enhancement modules will activate without any noise accumulating in the state species. The proof of Lemma 5.6 is included in Appendix B.

Lemma 5.7 states that if the state species of N are approximately encoding a set of states A after the last symbol event of its input, then not only will N continue to encode A, it will *improve* the accuracy of its encoding to η. This lemma serves two purposes: to *restore* the accuracy of an encoding after processing a symbol event, and to *maintain* that accuracy as long as no more symbol events arrive. The proof of Lemma 5.7 is included in Appendix C.

Using the initialization and state restoration lemmas, we can now prove the base case of the induction.

Proof of Lemma 5.4. Let $\mathbf{c} = (\mathbf{u}, V, h)$ be a context satisfying $\alpha_\lambda(\mathbf{c})$, let $\hat{\mathbf{c}} = (\hat{\mathbf{u}}, V, h)$ be $(\delta_u, 0)$-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_0-close to $\mathbf{x}_0$, and let $\hat{N}$ be δ_k-close to N. To show that $N, \mathbf{x}_0 \models_\eta^{\delta^*} \Phi_\lambda$, we now only need to show that $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is η-close to a signal $\mathbf{v} \in C[V]$ that satisfies $\phi_\lambda(\mathbf{u}, \mathbf{v})$. Let $\hat{\mathbf{x}}(t)$ be the unique solution of the IVP defined by $\hat{N}$ and the initial condition $\hat{\mathbf{x}}_0$. It now suffices to show that Y_Q η-encodes I for all $t \geq \tau(\mathbf{u}) = 1$.

By Lemma 5.6, we know that Y_Q δ_0-encodes I in state $\hat{\mathbf{x}}(\frac{1}{2})$. Then the hypothesis of Lemma 5.7 is satisfied with $A = I$ and $t_1 = \frac{1}{2}$. Since $w(\mathbf{u}) = \lambda$, no symbol event will ever occur in the input. It follows that every species $X \in U$ will have a concentration less than δ_u in $\hat{\mathbf{u}}$ for all $t \in [0, \infty)$. By Theorem 4.2, all the input enhancers will activate by time $t = \frac{1}{2}$, and so each enhanced signal $X^* \in U^*$ will be held at a concentration less than γ for all $t \geq \frac{1}{2}$. Since this will remain true indefinitely, any choice of $t_2 \geq 1$ will satisfy the hypothesis of Lemma 5.7. Thus, Lemma 5.7 tells us Y_Q will η-encode I for all $t \geq 1$. □

We now turn our attention to proving Lemma 5.5. Let $w \in \Sigma^*$ and $a \in \Sigma$ and assume the inductive hypothesis $N, \mathbf{x}_0 \models_\eta^{\delta^*} \Phi_w$ holds. Let $\mathbf{c} = (\mathbf{u}, V, h)$ be a context satisfying $\alpha_{wa}(\mathbf{c})$, let $\hat{\mathbf{c}} = (\hat{\mathbf{u}}, V, h)$ be $(\delta_u, 0)$-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_0-close to $\mathbf{x}_0$, and let $\hat{N}$ be δ_k-close to N. It suffices to show that $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is η-close to a function $\mathbf{v} \in C[V]$ which satisfies $\phi_{wa}(\mathbf{u}, \mathbf{v})$. We must show that for all $t \geq \tau(\mathbf{u})$ the set Y_Q η-encodes $\widehat{\Delta}(I, wa)$ at time t.

Let $I = [b, b+12]$ be the final symbol event of the input $\mathbf{u}$. Then we know that I is an a-event and that $\tau(\mathbf{u}) = b + 12$. All the remaining work of proving Lemma 5.5 involves closely examining the behavior of $\hat{N}$ during the a-event of $[b, b+12]$. Recall that an a-event consists of three pulses as shown in Fig. 5:

1. an X_r-pulse during $[b, b+4]$ that resets portal species Z_Q to encode $\emptyset$,
2. an X_a-pulse during $[b+4, b+8]$ that computes the transition function of the NFA and stores the result in the portal species, and
3. an X_c-pulse that copies the values of the portal species back into the state species Y_Q.

Since the a-event is partitioned into three separate pulses, it is natural to break the proof into three parts, each corresponding to one of the pulses.

Lemma 5.8 *(Reset lemma). Z_Q η-encodes $\emptyset$ at time $b+4$.*

Lemma 5.9 *(Transition lemma). Z_Q $\frac{1}{20}$-encodes $\widehat{\Delta}(I, wa)$ during the interval $[b+8, b+12]$.*

Lemma 5.10 *(Copy back lemma). Y_Q $\frac{1}{20}$-encodes $\widehat{\Delta}(I, wa)$ at time $b + 11.5$.*

These three lemmas are the *computation lemmas* and their proofs are provided in Appendix D. Intuitively, the Reset Lemma simply says that the portal species are "reset" to encode the empty set during the X_r-pulse so that the transition function of the NFA can be properly computed during the X_a-pulse. Similarly, the Transition Lemma says that the transition function is successfully computed during the X_a-pulse and maintained through the X_c-pulse. Finally, the Copy Back Lemma says that near the end of the X_c-pulse, Y_Q is properly encoding the correct set of states $\widehat{\Delta}(I, wa)$.

Using the above computation lemmas, we now finish the proof of the induction step.

Proof of Lemma 5.5. By Lemmas 5.8, 5.9, and 5.10, we know that Y_Q $\frac{1}{20}$-encodes the set $\widehat{\Delta}(I, wa)$ at time $b + 11.5$. Since this a-event $[b, b+12]$ is the last event in the input signal $\mathbf{u}$, the terminus of $\mathbf{u}$ is $\tau(\mathbf{u}) = b + 12$. This means that $x(t) < \delta_0$ for each $X \in U$ and for all time $t \geq b + 11$. By Theorem 4.2, the input enhancers will clean up the input signals so that $x^*(t) < \gamma$ for all $t \geq b + 11.5$. Finally, by Lemma 5.7, we know that Y_Q will η-encode $\widehat{\Delta}(I, wa)$ for all $t \geq b + 12 = \tau(\mathbf{u})$. Thus, $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{u}}}$ is η-close to a signal $\mathbf{v} \in C[V]$ that satisfies the I/O requirement ϕ_{wa}. □

This concludes the proofs of the main supporting lemmas of Theorem 5.3. Proofs of Lemmas 5.6 and 5.7 are provided in Appendix B and C, respectfully, and the proofs of the computation lemmas are given in Appendix D.

6. Conclusion

Unlike traditional CRNs where input is constrained to an initial state, I/O CRNs have designated input species for providing input signals over time. These input species may only be used catalytically which requires the I/O CRN to access its input nondestructively. We also introduced a notion of satisfying a requirement *robustly*. In particular, robust I/O CRNs must satisfy their requirement even in the presence of adversarial perturbations to their input signals, output measurement (decision), initial concentrations, and reaction rate constants.

Using these definitions, we showed that any nondeterministic finite automaton can be translated into an I/O CRN that robustly simulates it. Our translation also *efficiently* simulates an NFA by exploiting the inherent parallelism in the I/O CRN model. Specifically, the nondeterminism (existential quantification) is achieved directly by simulating all possible paths through the finite-state machine in parallel.

A key contribution of this paper is our *proof* that the translation robustly simulates the NFA in an adversarial environment. The proof was refined into two main parts corresponding to the two modules of our construction. The first module diminishes noise in the input signals to an acceptable level and reshapes it to be closer to a square wave. We intentionally specified this module separately so that it can be used in other molecular computation devices. The second module is responsible for computing the NFA transition function and maintains its state until the next symbol event occurs.

It should be noted that the nondestructive (catalytic) access that I/O CRNs have to their inputs does not correspond to current practice in DNA strand displacement CRNs. In fact, it is not immediately clear how to implement such access, especially for long input signals. However, the advantages of such catalytic interfaces for modular design and verification demonstrated here and elsewhere [20] indicate that future research should aim at achieving nondestructive access to inputs.

Finally, we hope that future research will improve our robustness bounds, extend robustness to stochastic CRNs, and shed new light on possible tradeoffs among the number of species, the computation time, and robustness.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank students and faculty auditors in our molecular programming course for helpful and challenging discussions. We thank Neil Lutz for a discussion that helped us finalize the I/O CRN model. We thank two anonymous referees for very useful suggestions.

Appendix A. Proof of input enhancement theorem

This section is dedicated to proving Theorem 4.2. The proof is naturally partitioned into two parts: Appendix A.1 is dedicated to the analysis of the ODEs generated by the cascade of species in Construction 4.1, and Appendix A.2 presents a complete proof of the theorem.

A.1. Cascade analysis

In this section, we only concern ourselves with analyzing systems of ODEs. The construction below is a simplified specification of the ODEs generated by the cascade from Construction 4.1. We use f ("forward") and b ("backward") for the rate constants of climbing up the cascade and falling to the bottom of the cascade, respectively. Notice that we also fold in the concentration of the input species X into the constant f. This simplification allows us to thoroughly analyze the behavior of the cascade whenever X is held constant and is enough to prove the theorem in the following section.

Construction A.1. *Given $f > 0$, $b > 0$, and $n \in \mathbb{N}$, let $x_0, \ldots, x_n : [0, \infty) \to [0, \infty)$ be functions that satisfy the ODEs*

$$\frac{dx_0}{dt} = \sum_{i=1}^{n} bx_i - fx_0, \tag{A.1}$$

$$\frac{dx_i}{dt} = fx_{i-1} - (f+b)x_i \quad \textit{for } 0 < i < n, \tag{A.2}$$

$$\frac{dx_n}{dt} = fx_{n-1} - bx_n. \tag{A.3}$$

We will now solve for explicit solutions to an IVP generated by the ODEs above using induction. These solutions have similar structure, so we define the following family of functions to describe their solution.

Construction A.2. *Given $f > 0$, $b > 0$, $p > 0$ and $i \in \mathbb{N}$, let $F_i : [0, \infty) \to [0, \infty)$ be the function*

$$F_i(t) = p\left(\frac{f}{f+b}\right)^i e^{-(f+b)t} \sum_{k=i}^{\infty} \frac{t^k (f+b)^k}{k!}. \tag{A.4}$$

Observation A.3. *If $f > 0$, $b > 0$, $p > 0$, $i \in \mathbb{N}$ and F_i is constructed according to Construction A.2, then*

$$\int e^{(f+b)t} \cdot F_i(t)dt = \frac{1}{f} e^{(f+b)t} F_{i+1}(t) + C \tag{A.5}$$

for some $C \in \mathbb{R}$.

Proof. Assume the hypothesis. Then by the definition of F_i from equation (A.4),

$$\int e^{(f+b)t} \cdot F_i(t)dt = p\left(\frac{f}{f+b}\right)^i \int \sum_{k=i}^{\infty} \frac{t^k (f+b)^k}{k!} dt. \tag{A.6}$$

The integral can be evaluated to obtain

$$\int \sum_{k=i}^{\infty} \frac{t^k (f+b)^k}{k!} dt = \frac{1}{f+b} \sum_{k=i+1}^{\infty} \frac{t^k (f+b)^k}{k!} + C_1,$$

for some $C_1 \in \mathbb{R}$. Inserting this into (A.6), we obtain

$$\begin{aligned}\int e^{(f+b)t} \cdot F_i(t)dt &= p\left(\frac{f}{f+b}\right)^i \left(\frac{1}{f+b} \sum_{k=i+1}^{\infty} \frac{t^k (f+b)^k}{k!} + C_1\right) \\ &= \frac{1}{f} e^{(f+b)t} F_{i+1}(t) + C\end{aligned}$$

for some $C \in \mathbb{R}$, which is the right side of (A.5). □

Lemma A.4. *If $f > 0$, $b > 0$, $p > 0$, $n \in \mathbb{N}$, and for $0 \le i \le n$ the functions x_i and F_i are constructed according to Construction A.1 and A.2 such that $x_0(0) = p$ and $x_i(0) = 0$ for all $0 < i \le n$, then for all $t \in [0, \infty)$,*

$$x_i(t) = F_i(t) - F_{i+1}(t) \quad \textit{for } 0 \le i < n, \tag{A.7}$$

$$x_n(t) = F_n(t). \tag{A.8}$$

Proof. Assume the hypothesis. We begin by proving (A.7) by induction on i.

Since $\sum_{i=0}^{n} \frac{dx_i}{dt} = 0$, it follows that $\sum_{i=0}^{n} x_i(t) = p$ for all $t \in [0, \infty)$. Therefore (A.1) can be simplified to

$$\frac{dx_0}{dt} = b(p - x_0) - fx_0,$$

and has solution

$$x_0(t) = p - p\left(\frac{f}{f+b}\right)\left(1 - e^{-(f+b)t}\right) = F_0(t) - F_1(t).$$

For the induction step, assume that $x_i(t) = F_i(t) - F_{i+1}(t)$ for some $0 \le i < n-1$. By (A.2), the derivative of x_{i+1} is

$$\frac{dx_{i+1}}{dt} = f x_i - (f+b)x_{i+1},$$

which can be solved using the integrating factor method. Thus, we obtain the family of solutions

$$\begin{aligned} x_{i+1}(t) &= e^{-(f+b)t}\int e^{(f+b)t} f x_i(t)dt \\ &= e^{-(f+b)t}\int e^{(f+b)t} f\left(F_i(t) - F_{i+1}(t)\right)dt. \end{aligned}$$

It immediately follows from Observation A.3 that

$$x_{i+1}(t) = F_{i+1}(t) - F_{i+2}(t) + C \cdot e^{-(f+b)t}$$

for some $C \in \mathbb{R}$. By the initial condition, $x_{i+1}(0) = 0$, and therefore $C = 0$. This completes the induction and shows that (A.7) holds.

It remains to be shown that (A.8) holds. Since $\sum_{i=0}^{n} x_i(t) = p$ for all $t \in [0, \infty)$, we know that

$$x_n(t) = p - \sum_{i=0}^{n-1} x_i(t)$$

for all $t \in [0, \infty)$ which can be written in terms of equation (A.4) in the following way:

$$x_n(t) = F_0(t) + \sum_{i=0}^{n-1} F_{i+1}(t) - \sum_{i=0}^{n-1} F_i(t).$$

Finally, we obtain (A.8) after canceling terms in the above equation. □

Lemma A.5. *Under the assumptions of Lemma A.4,*

$$x_n(t) = F_n(t) > p\left(\frac{f}{f+b}\right)^n\left(1 - ne^{-\frac{1}{n}(f+b)t}\right) \tag{A.9}$$

for all $t \in [0, \infty)$.

Proof. It suffices to show that

$$e^{-(f+b)t}\sum_{k=n}^{\infty}\frac{t^k(f+b)^k}{k!} > 1 - ne^{-\frac{1}{n}(f+b)t}. \tag{A.10}$$

The left-hand side of the (A.10) is related to the incomplete gamma function γ by

$$e^{-(f+b)t}\sum_{k=n}^{\infty}\frac{t^k(f+b)^k}{k!} = \frac{\gamma(n,(f+b)t)}{(n-1)!}.$$

The incomplete gamma function is well understood and many useful bounds exist. One particularly useful bound by Alzer [1, 24] is

$$\frac{\gamma(a,x)}{\Gamma(a)} > \left(1 - e^{-s_a x}\right)^a,$$

for $a \ge 1$ where $s_a = |\Gamma(1+a)|^{-\frac{1}{a}}$. It follows that

$$\frac{\gamma(n,(f+b)t)}{(n-1)!} > \left(1 - e^{-s_n(f+b)t}\right)^n,$$

where $s_n = (n!)^{-\frac{1}{n}}$. Since $n^n \ge n!$, we know $s_n > \frac{1}{n}$, whence

$$\frac{\gamma(n,(f+b)t)}{(n-1)!} > \left(1 - e^{-\frac{1}{n}(f+b)t}\right)^n > 1 - ne^{-\frac{1}{n}(f+b)t}. \quad \square$$

Corollary A.6. *Under the assumptions of Lemma A.4,*

$$x_n(t) = F_n(t) > \frac{p}{2}\left(\frac{f}{f+b}\right)^n \tag{A.11}$$

for all $t \geq \frac{n\log(2n)}{f+b}$.

Lemma A.7. *If* $f > 0$, $b > 0$, $p > 0$, $n \in \mathbb{N}$, *and for* $0 \leq i \leq n$ *the functions* x_i *and* F_i *are constructed according to Construction A.1 and A.2 such that* $x_i(0) = 0$ *for all* $0 \leq i < n$ *and* $x_n(0) = p$, *then for all* $t \in [0, \infty)$,

$$x_i(t) = \frac{b}{f}F_{i+1}(t) \quad \text{for } 0 \leq i < n. \tag{A.12}$$

Proof. Assume the hypothesis. We prove (A.12) by induction on i.

Since $\sum_{i=0}^{n}\frac{dx_i}{dt} = 0$, it follows that $\sum_{i=0}^{n} x_i(t) = p$ for all $t \in [0, \infty)$. Therefore (A.1) can be simplified to

$$\frac{dx_0}{dt} = b(p - x_0) - f x_0,$$

and has solution

$$x_0(t) = p\left(\frac{b}{f+b}\right)\left(1 - e^{-(f+b)t}\right) = \frac{b}{f}\cdot F_1(t).$$

For the induction step, assume that $x_i(t) = \frac{b}{f}F_{i+1}(t)$ for some $0 \leq i < n-1$. By (A.2), the derivative of x_{i+1} is

$$\frac{dx_{i+1}}{dt} = f x_i - (f+b)x_{i+1}.$$

By the integrating factor method, we obtain the solution

$$x_{i+1}(t) = e^{-(f+b)t}\int e^{(f+b)t} f x_i(t)dt = e^{-(f+b)t}\int e^{(f+b)t} bF_{i+1}(t)dt.$$

It follows from Observation A.3 that

$$x_{i+1}(t) = \frac{b}{f}F_{i+2}(t) + Ce^{-(f+b)t}$$

for some $C \in \mathbb{R}$. By the initial condition, $x_{i+1}(0) = 0$, so $C = 0$. □

Lemma A.8. *Under the assumptions of Lemma A.7,*

$$x_n(t) < pe^{-bt} + p\left(\frac{f}{f+b}\right)^n\left(1 - e^{-bt}\right) \tag{A.13}$$

for all $t \in [0, \infty)$.

Proof. Assume the hypothesis. By equation (A.3), the derivative of x_n is

$$\frac{dx_n}{dt} = f x_{n-1}(t) - bx_n(t).$$

Therefore x_n has a solution of the form

$$x_n(t) = e^{-bt}\int e^{bt} f x_{n-1}(t)dt.$$

By Lemma A.7, we know that $x_{n-1}(t) = \frac{b}{f}F_n(t)$, therefore

$$\begin{aligned} x_n(t) &= e^{-bt}\int e^{bt} bF_n(t)dt \\ &= e^{-bt}\int e^{bt} bp\left(\frac{f}{f+b}\right)^n e^{-(f+b)t}\sum_{k=n}^{\infty}\frac{t^k(f+b)^k}{k!}dt \\ &= bp\left(\frac{f}{f+b}\right)^n e^{-bt}\int e^{-ft}\sum_{k=n}^{\infty}\frac{t^k(f+b)^k}{k!}dt. \end{aligned}$$

Using the Taylor series of the exponential function, we can rearrange the integral to obtain

$$\int e^{-ft}\sum_{k=n}^{\infty}\frac{t^k(f+b)^k}{k!}dt=\int e^{-ft}\left(e^{(f+b)t}-\sum_{k=0}^{n-1}\frac{t^k(f+b)^k}{k!}\right)dt$$
$$=\int e^{bt}dt-\sum_{k=0}^{n-1}\frac{(f+b)^k}{k!}\int t^ke^{-ft}dt.$$

Since

$$\int t^ke^{-ft}dt=-\frac{k!}{f^{k+1}}e^{-ft}\sum_{i=0}^{k}\frac{t^if^i}{i!}+C_1,$$

for some $C_1\in\mathbb{R}$, and

$$\int e^{bt}dt=\frac{1}{b}e^{bt}+C_2,$$

for some $C_2\in\mathbb{R}$, we see that

$$x_n(t)=bp\left(\frac{f}{f+b}\right)^n e^{-bt}\left[\frac{1}{b}e^{bt}-\sum_{k=0}^{n-1}\frac{(f+b)^k}{k!}\left(-\frac{k!}{f^{k+1}}e^{-ft}\sum_{i=0}^{k}\frac{t^if^i}{i!}\right)+C_3\right]$$
$$=p\left(\frac{f}{f+b}\right)^n+p\frac{b}{f}e^{-bt}\sum_{k=0}^{n-1}\left(\frac{f}{f+b}\right)^{n-k-1}e^{-ft}\sum_{i=0}^{k}\frac{t^if^i}{i!}+C_4e^{-bt}$$

for some $C_3, C_4\in\mathbb{R}$. By the initial condition, $x_n(0)=p$. Therefore we can solve for C_4 in the equation

$$p=p\left(\frac{f}{f+b}\right)^n+p\frac{b}{f}\sum_{k=0}^{n-1}\left(\frac{f}{f+b}\right)^{n-k-1}+C_4,$$

and we see that

$$C_4=p-p\left(\frac{f}{f+b}\right)^n-p\frac{b}{f}\sum_{k=0}^{n-1}\left(\frac{f}{f+b}\right)^{n-k-1}.$$

After substituting this value for C_4 into our equation for x_n, we obtain

$$x_n(t)=pe^{-bt}+p\left(\frac{f}{f+b}\right)^n(1-e^{-bt})+A,$$

where

$$A=p\frac{b}{f}e^{-bt}\sum_{k=0}^{n-1}\left(\frac{f}{f+b}\right)^{n-k-1}\left(e^{-ft}\sum_{i=0}^{k}\frac{t^if^i}{i!}-1\right).$$

The lemma immediately follows from the fact that $A<0$. □

At this point, we have derived the solutions and bounds necessary for the cascade of species $X_0,\dots,X_n$ from Construction 4.1. However, we must prove a few lemmas concerning the other two species X^* and $\overline{X}^*$ that interact with the top of the cascade.

Construction A.9. *Given $f>0$ and $b>0$, let $x,\overline{x}:[0,\infty)\to[0,\infty)$ be functions that satisfy the ODEs*

$$\frac{dx}{dt}=f\overline{x}-bx, \tag{A.14}$$
$$\frac{d\overline{x}}{dt}=bx-f\overline{x}. \tag{A.15}$$

Lemma A.10. *If x and $\overline{x}$ are functions constructed according to Construction A.9 with $f > 0$ and $b > 0$, then for all $t \in [0, \infty)$,*

$$x(t) = p\left(\frac{f}{f+b}\right)\left(1 - e^{-(f+b)t}\right) + x(0) \cdot e^{-(f+b)t} \tag{A.16}$$

$$\overline{x}(t) = p - x(t), \tag{A.17}$$

where $p = x(0) + \overline{x}(0)$.

Proof. Assume the hypothesis. Since $\frac{dx}{dt} + \frac{d\overline{x}}{dt} = 0$, it follows that for all $t \in [0, \infty)$

$$x(t) + \overline{x}(t) = x(0) + \overline{x}(0),$$

and therefore (A.17) holds.

To show (A.16) holds, we solve the ODE (A.14) which can be written as

$$\frac{dx}{dt} = f(p - x) - bx,$$

which has solution (A.16). □

Lemma A.11. *If $\epsilon \in (0, \frac{1}{2})$, $\tau > 0$, and $x, \overline{x}$ are constructed according to Construction A.9 with $p = x(0) + \overline{x}(0)$, and*

$$f \geq \frac{1}{\tau}\log\left(\frac{2p}{\epsilon}\right), \qquad b \leq f\left(\frac{\epsilon}{2p}\right),$$

then $x(t) > p - \epsilon$ and $\overline{x}(t) < \epsilon$ for all $t \geq \tau$.

Proof. Assume the hypothesis. Then by Lemma A.10, for all $t \geq \tau$,

$$\begin{aligned} x(t) &\geq p\left(\frac{f}{f+b}\right)\left(1 - e^{-(f+b)\tau}\right) + x(0)e^{-(f+b)\tau} \\ &\geq p\left(\frac{f}{f+b}\right)\left(1 - e^{-f\tau}\right). \end{aligned}$$

Since $f \geq \frac{1}{\tau}\log\left(\frac{2p}{\epsilon}\right)$, $b \leq f\left(\frac{\epsilon}{2p}\right)$, and $\frac{\epsilon}{2p} < \frac{\epsilon}{2p-\epsilon}$,

$$x(t) \geq p\left(\frac{1}{1 + \frac{\epsilon}{2p}}\right)\left(1 - \frac{\epsilon}{2p}\right) = p\left(1 - \frac{\epsilon}{2p}\right)^2 > p - \epsilon. \quad \square$$

A.2. Proof of input enhancement theorem

We now have the machinery that we need to prove the Input Enhancement Theorem.

Proof of Theorem 4.2. Assume the hypothesis. Then $\tau > 0$, $\epsilon \in (0, \frac{1}{2})$, $\boldsymbol{\delta} = (\delta_u, \delta_h, \delta_0, \delta_k)$ with $\delta_u \in (0, \frac{1}{3})$, $\delta_h \in (0, \epsilon)$, $\delta_0 \in (0, \frac{1}{2})$, $\delta_k > 0$, and $N(X) = N^{(X)}(\tau, \epsilon, \boldsymbol{\delta})$ and $\mathbf{x}_0^{(X)}$ are constructed according to Construction 4.1. We now must show that $N^{(X)}, \mathbf{x}_0^{(X)} \models_{\epsilon}^{\boldsymbol{\delta}} \Phi^{(X)}(\tau)$.

Now let $n = |S| - 2$, let $\mathbf{c} = (\mathbf{u}, V, h)$ be a context satisfying $\alpha(\mathbf{c})$, let $\hat{\mathbf{c}} = (\hat{\mathbf{u}}, V, \hat{h})$ be (δ_u, δ_h)-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_0-close to $\mathbf{x}_0^{(X)}$, let $\hat{N}$ be δ_k-close to $N^{(X)}$, and let p and p^* be the constants

$$p = \sum_{i=0}^{n} \hat{\mathbf{x}}_0(X_i), \qquad p^* = \hat{\mathbf{x}}_0(X^*) + \hat{\mathbf{x}}_0(\overline{X}^*).$$

It now suffices to show that $\hat{N}_{\hat{\mathbf{c}}, \hat{\mathbf{x}}_0}$ is ϵ-close to a function $\mathbf{v} \in \mathcal{C}[V]$ that satisfies $\phi(\mathbf{u}, \mathbf{v})$, i.e., if (b, I) is an input event for $\mathbf{u}$, then (b, I_τ) must be an output event for $\mathbf{v}$. We prove this in two cases corresponding to $b = 1$ and $b = 0$ by invoking many of the lemmas from the previous section.

The state species of $\hat{N}$ are naturally split up into two parts. The first part is the cascade of species $X_0, \ldots, X_n$, and the second part are the species $X^*, \overline{X}^*$ which are affected by the top of the cascade. The ODEs for species $X_0, \ldots, X_n$ of $\hat{N}$ can be derived from the reactions in Construction 4.1 along with the perturbed mass action function from equation (3.5) and are

$$\frac{dx_0}{dt} = \sum_{i=1}^{n} \hat{k}_1 x_i - (\hat{k}_1 x)x_0, \tag{A.18}$$

$$\frac{dx_i}{dt} = (\hat{k}_1 x)x_{i-1} - (\hat{k}_1 x + \hat{k}_1)x_i \quad \text{for } 0 < i < n, \tag{A.19}$$

$$\frac{dx_n}{dt} = (\hat{k}_1 x)x_{n-1} - \hat{k}_1 x_n. \tag{A.20}$$

Similarly, the ODEs for X^* and $\overline{X}^*$ are

$$\frac{dx^*}{dt} = (\hat{k}_2 x_n)\overline{x}^* - \hat{k}_2 x^*, \tag{A.21}$$

$$\frac{d\overline{x}^*}{dt} = \hat{k}_2 x^* - (\hat{k}_2 x_n)\overline{x}^*. \tag{A.22}$$

Since $\frac{dx^*}{dt} + \frac{d\overline{x}^*}{dt} = 0$, it is easy to show that $x^*(t) + \overline{x}^*(t) = p^*$ for all $t \in [0, \infty)$. Similarly, $\sum_{i=0}^{n} x_i(t) = p$ for all $t \in [0, \infty)$.

Let $(1, I)$ be an input event for $\mathbf{u}$, where $I = [t_1, t_2]$. Since the input signal can be perturbed by δ_u, it follows that $x(t) > 1 - \delta_u$ for all $t \in I$. We also know that the rate constants can be perturbed by δ_k. To minimize the concentration of X_n in the interval I, we assume that all the concentration of $X_0, \ldots, X_n$ is in X_0 at time t_1. We also maximize the rate of falling down the cascade and minimize the rate of climbing the cascade.

Therefore by Lemma A.4, for all $t \in I$,

$$x_n(t) > p\left(\frac{f}{f+b}\right)^n \sum_{i=n}^{\infty} \frac{t^i (f+b)^i}{i!} e^{-(f+b)(t-t_1)},$$

where $f = (k_1 - \delta_k)(1 - \delta_u)$ and $b = k_1 + \delta_k$. Since x_n is monotonically increasing, for all $t \in [t_1 + \frac{\tau}{2}, t_2]$, $x_n(t) \geq x_n(\frac{\tau}{2})$, and therefore

$$x_n(t) > p\left(\frac{f}{f+b}\right)^n \sum_{i=n}^{\infty} \frac{t^i (f+b)^i}{i!} e^{-(f+b)\frac{\tau}{2}}.$$

By Lemma A.5, for all $t \in [t_1 + \frac{\tau}{2}, t_2]$,

$$x_n(t) > p\left(\frac{f}{f+b}\right)^n \left(1 - ne^{-\frac{1}{n}(f+b)\frac{\tau}{2}}\right).$$

Since $k_1 > \delta_k + \frac{2n}{\tau(1-\delta_u)} \log(2n)$, Corollary A.6 tells us

$$\begin{aligned} x_n(t) &> p\left(\frac{f}{f+b}\right)^n \left(\frac{1}{2}\right) \\ &= \frac{p}{2}\left(\frac{(k_1 - \delta_k)(1 - \delta_u)}{(k_1 - \delta_k)(1 - \delta_u) + k_1 + \delta_k}\right)^n \\ &= \frac{p}{2}\left(\frac{1 - \delta_u}{1 - \delta_u + u}\right)^n, \end{aligned}$$

where $u = \frac{k_1 + \delta_k}{k_1 - \delta_k}$. Since $k_1 > 2\frac{\delta_k(2+\delta_u)}{\delta_u}$, we know that $u < 1 + \delta_u$ and therefore

$$x_n(t) > \frac{p}{2}\left(\frac{1 - \delta_u}{2}\right)^n.$$

Since the initial condition can be perturbed by at most δ_0, $p > \frac{10}{\epsilon - \delta_h}\left(\frac{2}{1-\delta_u}\right)^n$, therefore

$$x_n(t) > \frac{5}{\epsilon - \delta_h},$$

for all $t \in [t_1 + \frac{\tau}{2}, t_2]$.

Recall that the ODEs for X^* and $\overline{X}^*$ are (A.21) and (A.22). To minimize the concentration of X^* in the interval $[t_1 + \frac{\tau}{2}, t_2]$, we minimize the production of X^* and maximize the production of $\overline{X}^*$. By Lemma A.10, for all $t \in [t_1 + \tau, t_2]$,

$$x^*(t) > p^*\left(\frac{f}{f+b}\right)\left(1 - e^{-(f+b)\frac{\tau}{2}}\right),$$

where $f = (k_2 - \delta_k)\frac{5}{\epsilon - \delta_h}$ and $b = k_2 + \delta_k$. Since $k_2 > 4\delta_k$, then $\frac{k_2+\delta_k}{k_2-\delta_k} < \frac{5}{3}$ and therefore

$$b = k_2 + \delta_k < \frac{\epsilon - \delta_h}{3}\left[(k_2 - \delta_k)\frac{5}{\epsilon - \delta_h}\right] = \frac{\epsilon - \delta_h}{3} f,$$

whence

$$b < \frac{\epsilon - \delta_h}{2p^*} f.$$

Since $\frac{5}{\epsilon - \delta_h} > 1$ and $k_2 = \frac{2}{\tau}\log\left(\frac{3}{\epsilon - \delta_h}\right) + 4\delta_k$,

$$f = (k_2 - \delta_k)\frac{5}{\epsilon - \delta_h} > \frac{2}{\tau}\log\left(\frac{3}{\epsilon - \delta_h}\right) > \frac{2}{\tau}\log\left(\frac{2p^*}{\epsilon - \delta_h}\right).$$

By Lemma A.11, for all $t \in [t_1 + \tau, t_2]$,

$$x^*(t) > p^* - \epsilon + \delta_h.$$

Since the initial state can only be perturbed by at most δ_0 and the output function can only introduce δ_h error. It immediately follows that

$$N_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}(t) > 1 - \epsilon.$$

Therefore $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}(t)$ is ϵ-close to satisfying the requirement that $(1, I_\tau)$ is an output event.

It remains to be shown that $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to handling input events of the form $(0, I)$. To show this, let $(0, I)$ be an input event, and let $I = [t_1, t_2]$. Therefore $x(t) < \delta_u$ for all $t \in I$. Similar to the above argument, by Lemma A.8, for all $t \in I$,

$$x_n(t) < pe^{-b(t-t_1)} + p\left(\frac{f}{f+b}\right)^n\left(1 - e^{-b(t-t_1)}\right)$$

where $f = (k_1 + \delta_k)\delta_u$ and $b = k_1 - \delta_k$. Since this function is monotonically decreasing, for all $t \in [t_1 + \frac{\tau}{2}, t_2]$,

$$\begin{aligned}
x_n(t) &< p\left(\frac{f}{f+b}\right)^n + pe^{-b\frac{\tau}{2}} \\
&= p\left(\frac{(k_1+\delta_k)\delta_u}{(k_1+\delta_k)\delta_u + k_1 - \delta_k}\right)^n + pe^{-b\frac{\tau}{2}} \\
&= p\left(\frac{\delta_u}{\delta_u + u}\right)^n + pe^{-b\frac{\tau}{2}},
\end{aligned}$$

where $u = \frac{k_1 - \delta_k}{k_1 + \delta_k}$. Since $k_1 > \frac{\delta_k(2-\delta_u)}{\delta_u}$, we know that $u > 1 - \delta_u$, whence for all $t \in [t_1 + \frac{\tau}{2}, t_2]$,

$$x_n(t) < p\delta_u^n + pe^{-b\frac{\tau}{2}}.$$

Since $p < \frac{10}{\epsilon - \delta_h}\left(\frac{2}{1-\delta_u}\right)^n + 2\delta_0$,

$$\begin{aligned}
x_n(t) &< \frac{10}{\epsilon - \delta_h}\left(\frac{2\delta_u}{1-\delta_u}\right)^n + \delta_u^n + pe^{-b\frac{\tau}{2}} \\
&< \frac{10 + \epsilon - \delta_h}{\epsilon - \delta_h}\left(\frac{2\delta_u}{1-\delta_u}\right)^n + pe^{-b\frac{\tau}{2}} \\
&< \frac{32}{3(\epsilon - \delta_h)}\left(\frac{1-\delta_u}{2\delta_u}\right)^{-n} + pe^{-b\frac{\tau}{2}}.
\end{aligned}$$

Since $n \geq \log_{\left(\frac{1-\delta_u}{2\delta_u}\right)}\left(\frac{64}{(\epsilon-\delta_h)^2}\right)$,

$$x_n(t) < \frac{32}{3(\epsilon - \delta_h)}\left(\frac{(\epsilon - \delta_h)^2}{64}\right) + pe^{-b\frac{\tau}{2}} = \frac{\epsilon - \delta_h}{6} + pe^{-b\frac{\tau}{2}}.$$

As we showed before, $p < \frac{32}{3(\epsilon-\delta_h)}\left(\frac{2}{1-\delta_u}\right)^n + 2\delta_0$, whence

$$x_n(t) < \frac{\epsilon-\delta_h}{6} + \frac{32}{3(\epsilon-\delta_h)}\left(\frac{4}{1-\delta_u}\right)^n e^{-b\frac{\tau}{2}} + 2\delta_0 e^{-b\frac{\tau}{2}}.$$

Since $b = k_1 - \delta_k > \frac{2}{\tau}\log\left(\frac{640}{(\epsilon-\delta_h)^2}\left(\frac{2}{1-\delta_u}\right)^n\right)$,

$$x_k(t) < \frac{\epsilon-\delta_h}{6} + \frac{\epsilon-\delta_h}{60} + 2\delta_0 e^{-b\frac{\tau}{2}} < \frac{\epsilon-\delta_h}{6} + \frac{\epsilon-\delta_h}{30},$$

whence for all $t \in [t_1 + \frac{\tau}{2}, t_2]$

$$x_n(t) < \frac{\epsilon-\delta_h}{5}.$$

We now bound the concentration of X^* and $\overline{X}^*$. If $f = (k_2+\delta_k)\frac{\epsilon-\delta_h}{5}$ and $b = k_2 - \delta_k$, then by Lemma A.10

$$\overline{x}^*(t) > p\left(\frac{f}{f+b}\right)\left(1 - e^{-(f+b)\frac{\tau}{2}}\right),$$

for all $t \in [t_1+\tau, t_2]$. Since $\frac{k_2-\delta_k}{k_2+\delta_k} > \frac{3}{5}$,

$$f = (k_2+\delta_k)\frac{\epsilon-\delta_h}{5} < \frac{\epsilon-\delta_h}{3}(k_2-\delta_k) < \frac{\epsilon-\delta_h}{2p^*}b.$$

Since $b = k_2 - \delta_k > \frac{2}{\tau}\log\left(\frac{3}{\epsilon-\delta_h}\right) > \frac{2}{\tau}\log\left(\frac{2p^*}{\epsilon-\delta_h}\right)$, by Lemma A.11, for all $t \in [t_1+\tau, t_2]$,

$$\overline{x}^*(t) > p^* - \epsilon + \delta_h.$$

Since $x^*(t) + \overline{x}^*(t) = p^*$ for all $t \in [0,\infty)$, it follows that $x^*(t) < \epsilon - \delta_h$ for all $t \in I_\tau$. Since $p^* > 1 + \delta_0$, it follows that $\overline{x}^*(t) > 1 - \epsilon + \delta_h$ for all $t \in I_\tau$.

Finally, since the output function can at most deviate by δ_h from the solutions of $x^*(t)$ and $\overline{x}^*(t)$, it is clear that $\hat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to having $(0, I_\tau)$ as a valid output event. □

Appendix B. Proof of initialization lemma

Proof of Lemma 5.6. In this proof, we must show that the I/O CRN N, when initialized with its initial state $\mathbf{x}_0$, will still be δ_0-encoding its set of initial states I at time $t = \frac{1}{2}$. Before we begin the argument, we must fix the arbitrary perturbations of the I/O CRN. Since this lemma applies to both the base case and the induction step of our overall argument, we fix an arbitrary input string $w \in \Sigma^*$. Now let $\mathbf{c} = (\mathbf{u}, V, h)$ be a context satisfying $\alpha_w(\mathbf{c})$, let $\hat{\mathbf{c}} = (\hat{\mathbf{u}}, V, h)$ be $(\delta_u, 0)$-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_0-close to $\mathbf{x}_0$, let $\hat{N}$ be δ_k-close to N, and let $\hat{\mathbf{x}}(t)$ be the unique solution of $\hat{N}$ when initialized to $\hat{\mathbf{x}}_0$. To complete the proof we must show that Y_Q η-encodes I in the state $\hat{\mathbf{x}}(\frac{1}{2})$. In other words, for each $q \in I$ we must show that $y_q(\frac{1}{2}) > p(Y_q) - \eta$ and for each $q \notin I$ that $y_q(\frac{1}{2}) < \eta$.

For the first part, let $q \in I$. Recall that the ODE for Y_q is

$$\begin{aligned}\frac{dy_q}{dt} &= \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q + \hat{k}_2 y_q^2 \overline{y}_q - \hat{k}_2 y_q \overline{y}_q^2 \\ &> \hat{k}_2 y_q^2 \overline{y}_q - \hat{k}_2 y_q \overline{y}_q^2 - \hat{k}_2 \delta_0 \overline{z}_q y_q.\end{aligned}$$

Since $\hat{k}_2$ is δ_k-close to the constant k_2, we have

$$\frac{dy_q}{dt} > (k_2-\delta_k) y_q^2 \overline{y}_q - (k_2+\delta_k) y_q \overline{y}_q^2 - (k_2+\delta_k)\delta_0 \overline{z}_q y_q.$$

Since the sum of the concentrations of Z_q and $\overline{Z}_q$ is the constant $p(Z_q)$, we know that $\overline{z}_q(t)$ is bounded by $p(Z_q)$. Thus,

$$\frac{dy_q}{dt} > (k_2-\delta_k) y_q^2 \overline{y}_q - (k_2+\delta_k) y_q \overline{y}_q^2 - (k_2-\delta_k)\delta_0 p(Z_q) y_q.$$

Since $\overline{y}_q = p(Y_q) - y_q$, we can simplify the ODE to

$$\begin{aligned}\frac{dy_q}{dt} >& (k_2-\delta_k) y_q^2 (p(Y_q) - y_q) - (k_2+\delta_k) y_q (p(Y_q) - y_q)^2 \\ &- (k_2-\delta_k)\delta_0 p(Z_q) y_q.\end{aligned}$$

At this point, everything in the ODE is a constant except for the function y_q. If we let a, b, c, and p be the constants

$$a = k_2 - \delta_k, \qquad b = k_2 + \delta_k, \qquad c = (k_2 - \delta_k)p(Z_q)\delta_0, \qquad p = p(Y_q),$$

then we can rewrite the ODE in a simpler form

$$\frac{dy_q}{dt} > ay_q^2(p - y_q) - by_q(p - y_q)^2 - cy_q.$$

The above ODE has identical structure to that of the termolecular signal restoration algorithm from [29]. This means that if the inequality $c < \frac{p^2a^2}{4(a+b)}$ holds, we can make use of Theorem 3.2 from [29] to bound the concentration of Y_q during the interval $[0, \frac{1}{2}]$. It is routine to verify this and can easily be shown using the facts that $k_2 > 25$, $1 - \delta_0 < p(Z_q) < 1 + \delta_0$, and $\delta_0, \delta_k < \frac{1}{20}$.

At this point, we know that during the interval $[0, \frac{1}{2}]$ the behavior of Y_q is bounded by the termolecular signal restoration algorithm and that the constant c is small enough to introduce bistability to the system. We now show that the concentration of Y_q is attracted to the stable fixed point close to 1 and therefore remains unaffected.

Let E_1 and E_2 be the constants

$$E_1 = p\left(\frac{b}{a+b}\right) + A, \tag{B.1}$$

$$E_2 = p - A, \tag{B.2}$$

where $A = \frac{p}{2}\left(\frac{a}{a+b}\right)\left(1 - \sqrt{1 - c^*}\right)$ and $c^* = c \cdot \frac{4(a+b)}{p^2a^2}$. These constants are two of the equilibrium points of the signal restoration algorithm mentioned previously. Because of the stability of these points shown in [29], if $y_q(0) > E_1$, then Y_q will converge to E_2, whereas if $y_q(0) < E_1$ then it would converge to 0. It is routine to verify that $y_q(0) > E_1$ but can be easily shown using the bounds mentioned previously along with $y_q(0) > 1 - \delta_0$.

Since E_1 is the decision point of the signal restoration algorithm, Theorem 3.2 from [29] tells us that the concentration of Y_q will converge away from the constant E_1 to the constant E_2. It immediately follows from the fact that $E_1 < y_q(0) < E_2$ that $y_q(t) \geq y_q(0) > 1 - \delta_0$ for all $t \in [0, \frac{1}{2}]$.

It remains to be shown that if $q \notin I$, then $y_q(\frac{1}{2}) < \delta_0$. This immediately follows by the symmetry imposed by the dual relationship of Y_q and $\overline{Y}_q$. □

Appendix C. Proof of state restoration lemma

Proof of Lemma 5.7. In this proof, we must show that the I/O CRN N, when initialized with $\mathbf{x}_0$, is capable of maintaining and improving its encoding of a set $A \subseteq Q$. Before we begin the argument, we must fix the arbitrary perturbations of the I/O CRN. Since this lemma also applies to both the base case and the induction step of our overall argument, we fix an arbitrary input string $w \in \Sigma^*$. Now let $\mathbf{c} = (\mathbf{u}, V, h)$ be a context satisfying $\alpha_w(\mathbf{c})$, let $\hat{\mathbf{c}} = (\hat{\mathbf{u}}, V, h)$ be $(\delta_u, 0)$-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_0-close to $\mathbf{x}_0$, let $\hat{N}$ be δ_k-close to N, and let $\hat{\mathbf{x}}(t)$ be the unique solution of $\hat{N}$ when initialized to $\hat{\mathbf{x}}_0$.

We also assume the hypothesis of Lemma 5.7, i.e., that Y_Q $\frac{1}{20}$-encodes the set $A \subseteq Q$ in state $\hat{\mathbf{x}}(t_1)$ and $x^*(t) \leq \gamma$ for all $t \in [t_1, t_2]$ and for each $X^* \in U^*$. To complete the proof, we must show that Y_Q η-encodes A for all $t \in [t_1 + \frac{1}{2}, t_2]$.

Let $q \in A$. Recall that the ODE for Y_q is

$$\frac{dy_q}{dt} = \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q + \hat{k}_2 y_q^2 \overline{y}_q - \hat{k}_2 y_q \overline{y}_q^2.$$

We will be examining the behavior of Y_q during the interval $[t_1, t_2]$. During this interval, we know by the hypothesis that $x^*(t) \leq \gamma$ for each $X^* \in U^*$. Thus we know that during the interval $[t_1, t_2]$

$$\begin{aligned}\frac{dy_q}{dt} &> \hat{k}_2 y_q^2 \overline{y}_q - \hat{k}_2 y_q \overline{y}_q^2 - \hat{k}_2 \gamma \overline{z}_q y_q, \\ &> (k_2 - \delta_k) y_q^2 (p(Y_q) - y_q) - (k_2 + \delta_k) y_q (p(Y_q) - y_q)^2 \\ &\quad - (k_2 + \delta_k)\gamma p(Z_q) y_q.\end{aligned}$$

Now if we define the constants a, b, c, and p to be

$$a = k_2 - \delta_k, \qquad b = k_2 + \delta_k, \qquad c = (k_2 + \delta_k)p(Z_q)\gamma, \qquad p = p(Y_q),$$

then we can rewrite the above ODE as

$$\frac{dy_q}{dt} > ay_q^2(p - y_q) - by_q(p - y_q)^2 - cy_q.$$

We also define the constants E_1 and E_2

$$E_1 = p\left(\frac{b}{a+b}\right) + A, \tag{C.1}$$

$$E_2 = p - A, \tag{C.2}$$

where $A = \frac{p}{2}\left(\frac{a}{a+b}\right)\left(1-\sqrt{1-c^*}\right)$ and $c^* = c \cdot \frac{4(a+b)}{p^2a^2}$.

It is easy to verify that $c < \frac{p^2a^2}{4(a+b)}$ and $E_1 < y_q(t_1)$, so Theorem 3.2 from [29] tells us that $y_q(t)$ is converging toward the value E_2.

We now show that the constant E_2 is sufficiently high to restore the concentration of Y_q to at least $p(Y_q) - \eta$. Using the definition of γ in (5.15), k_2 in (5.13), and the fact $\delta_k, \delta_0 < \frac{1}{20}$, it is not difficult to show that:

$$A < \frac{2c}{pa} = \frac{2\gamma(k_2+\delta_k)p(Z_q)}{p(Y_q)(k_2-\delta_k)} < 2\gamma\left(\frac{1+\frac{\delta_k}{k_2}}{1-\frac{\delta_k}{k_2}}\right)\left(\frac{1+\delta_0}{1-\delta_0}\right) < \frac{\eta}{8}. \tag{C.3}$$

It follows that

$$E_2 = p(Y_q) - A > p(Y_q) - \frac{\eta}{8}. \tag{C.4}$$

By Theorem 3.3 in [29], the amount of time Δt it takes the signal restoration reactions to restore the concentration of Y_q to $p(Y_q) - \eta$ from $p(Y_q) - \delta_0$ is bounded by

$$\Delta t < \frac{a+b}{abp^2\left(1 - c\frac{4(a+b)}{p^2a^2}\right)}\log u, \tag{C.5}$$

where

$$u = \frac{(p-\eta-E_1)(E_2-p+\delta_0)}{(p-\delta_0-E_1)(E_2-p+\eta)}.$$

Using previous definitions and bounds, it is routine to verify that the right-hand side of (C.5) is bounded by $\frac{1}{2}$.

It follows that within $\frac{1}{2}$ time, the concentration of Y_q reaches $p(Y_q) - \eta$, and therefore for all $t \in [t_1 + \frac{1}{2}, t_2]$, $y_q(t) > p(Y_q) - \eta$.

This finishes one half of the proof, namely, that if $q \in A$ that the I/O CRN N robustly keeps the value of Y_q η-close to $p(Y_q)$ during the interval $[t_1 + \frac{1}{2}, t_2]$. It remains to be shown that for $q \notin A$, $y_q(t) < \eta$ for all $t \in [t_1 + \frac{1}{2}, t_2]$. This follows by the symmetry of Y_q and its dual $\overline{Y}_q$. □

Appendix D. Proofs of computation lemmas

In this appendix, we prove Lemmas 5.8, 5.9, 5.10. Recall that there are variables defined in Section 5 that are relevant to these lemmas, and we repeat them here for readability. Let $w \in \Sigma^*$ and $a \in \Sigma$ and assume the inductive hypothesis $N, \mathbf{x}_0 \models^{\delta^*}_{\eta} \Phi_w$ of Lemma 5.5 holds. Let $\mathbf{c} = (\mathbf{u}, V, h)$ be a context satisfying $\alpha_{wa}(\mathbf{c})$, let $\hat{\mathbf{c}} = (\hat{\mathbf{u}}, V, h)$ be $(\delta_u, 0)$-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_0-close to $\mathbf{x}_0$, and let $\hat{N}$ be δ_k-close to N.

Let $I = [b, b+12]$ be the final symbol event of the input $\mathbf{u}$. Then I is an a-event and $\tau(\mathbf{u}) = b + 12$. The proofs of computation lemmas involves closely examining the behavior of the I/O CRN N during the three pulses of this a-event s shown in Fig. 6:

Before we start proving the computation lemmas, we first state and prove some helpful observations.

Observation D.1. *Y_Q η-encodes $\widehat{\Delta}(I, w)$ during the interval* $[b, b+8]$.

Proof. Let $\mathbf{u}^*$ be a terminal input such that $\mathbf{u}^*(t) = \mathbf{u}(t)$ for all $t \in [0, b]$ and $\mathbf{u}^*(t) = \mathbf{0}$ for all $t > b$. Then $w(\mathbf{u}^*) = w$ and $\tau(\mathbf{u}^*) \le b$. Since $N, \mathbf{x}_0 \models^{\delta^*}_{\eta} \Phi_w$ by our inductive hypothesis, we know that Y_Q η-encodes $\widehat{\Delta}(I, w)$ starting at time $\tau(\mathbf{u}^*)$. Since $\mathbf{u}$ and $\mathbf{u}^*$ agree at every $t \in [0, b]$, the concentrations of Y_q for each $q \in Q$ must also agree at every time $t \in [0, b]$. Therefore Y_Q η-encodes $\widehat{\Delta}(I, w)$ at time b. Finally, the input species X_c is below δ_u during the interval $[b, b+8]$, so the values of the species in Y_Q will be maintained by the state restoration reactions as shown in the proof of Lemma 5.7 in Appendix C. □

Observation D.2. $x_r^*(b), x_c^*(b) < \gamma$.

Proof. This follows from the fact that the concentrations of inputs X_r and X_c are less than δ_u in the interval $[b - \frac{1}{2}, b]$. Therefore the preprocessed input species X_r^* and X_c^* have time to drop below γ before time b. □

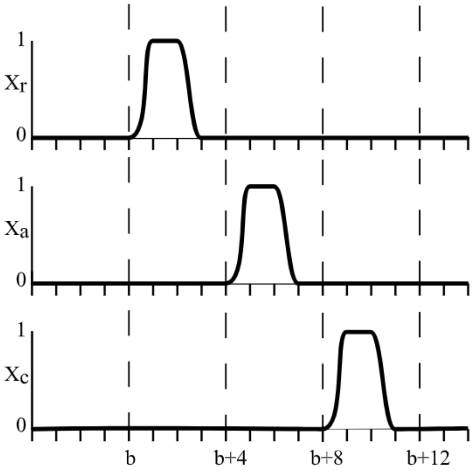

Fig. 6. The X_r-, X_a,- and X_c-pulses of the final a-event.

D.1. *Proof of reset lemma*

Proof of Lemma 5.8. For $q \in Q$, the ODE for Z_q is

$$\begin{aligned}\frac{dz_q}{dt} &= -\hat{k}_1 x_r^* z_q + \sum_{(s,a,q)\in\Delta} \hat{k}_1 x_a^* y_s \overline{z}_q \\ &< -(k_1 - \delta_k) x_r^* z_q + \sum_{(s,a,q)\in\Delta} (k_1 + \delta_k) x_a^* p(Y_r) \overline{z}_q .\end{aligned}$$

The interval $[b, b+4]$ is an X_r-pulse and therefore every species X_a^* for $a \in \Sigma$ must have concentration γ-close to zero. Therefore

$$\begin{aligned}\frac{dz_q}{dt} &< -(k_1 - \delta_k) x_r^* z_q + \sum_{(s,a,q)\in\Delta} p(Y_r)(k_1 + \delta_k)\gamma \overline{z}_q \\ &< -(k_1 - \delta_k) x_r^* z_q + |Q|(1+\delta_0)(k_1+\delta_k)\gamma \overline{z}_q .\end{aligned}$$

During the interval $[b+1, b+2]$, the species X_r is δ_u-close to one. Therefore during the interval $[b+1.5, b+2]$ the species X_r^* is γ-close to 1. Thus

$$\frac{dz_q}{dt} < -(k_1 - \delta_k)(1-\gamma) z_q + |Q|(1+\delta_0)(k_1+\delta_k)\gamma \overline{z}_q .$$

Let $\hat{f}$, $\hat{b}$, and $\hat{p}$ be constants defined by

$$\hat{f} = |Q|(1+\delta_0)(k_1+\delta_k)\gamma \qquad \hat{b} = (k_1 - \delta_k)(1-\gamma), \qquad \hat{p} = p(Z_q),$$

then we can rewrite the above ODE as

$$\frac{dz_q}{dt} < \hat{f}\overline{z}_q - \hat{b} z_q ,$$

which has identical structure to the ODE from Construction A.9. By Lemma A.10, we have the bound

$$\begin{aligned}z_q(b+2) &< \hat{p}\left(\frac{\hat{f}}{\hat{f}+\hat{b}}\right)\left(1 - e^{-(\hat{f}+\hat{b})\frac{1}{2}}\right) + z_q(b+1.5)\cdot e^{-(\hat{f}+\hat{b})\frac{1}{2}} \\ &< \hat{p}\left(\frac{\hat{f}}{\hat{f}+\hat{b}}\right)\left(1 - e^{-(\hat{f}+\hat{b})\frac{1}{2}}\right) + \hat{p}e^{-(\hat{f}+\hat{b})\frac{1}{2}} \\ &= \hat{p}\left(\frac{\hat{f}}{\hat{f}+\hat{b}}\right) + \hat{p}\left(\frac{\hat{b}}{\hat{f}+\hat{b}}\right)e^{-(\hat{f}+\hat{b})\frac{1}{2}} < \hat{p}\hat{f} + \hat{p}e^{-\frac{\hat{b}}{2}} .\end{aligned}$$

Using the definition of k_1 from equation (5.12), it is easy to show that $k_1 > \delta_k + \frac{2}{1-\gamma}\log\left(\frac{4(1+\delta_0)}{\eta}\right)$, therefore

$$z_q(b+2) < \hat{p}\hat{f} + \frac{\eta}{4}.$$

It is routine but easy to verify that $\hat{p}\hat{f} > \frac{\eta}{4}$, and so $z_q(b+2) < \frac{\eta}{2}$.

During the interval $[b+2, b+4]$, the derivative of Z_q is bounded by

$$\frac{dz_q}{dt} < \hat{f}(\hat{p} - z_q), < \hat{p}\hat{f}, < \frac{\eta}{4},$$

which means less than $\frac{\eta}{2}$ of Z_q is produced over the interval $[b+2, b+4]$. Therefore $z_q(b+4) < \eta$. Since q was arbitrary, this means that Z_Q η-encodes the set $\emptyset$ at time $b+4$. □

D.2. Proof of transition lemma

Proof of Lemma 5.9. We prove this in two steps. First we prove that if $q \in \widehat{\Delta}(I, wa)$, then for all $t \in [b+8, b+12]$ $z_q(t) > p(Z_q) - \frac{1}{20}$, and second we prove that if $q \notin \widehat{\Delta}(I, wa)$, then for all $t \in [b+8, b+12]$ $z_q(t) < \frac{1}{20}$. We also depend on the fact that Lemma 5.8 states that all the portal species have been reset to a concentration less than η at time $b+4$.

For the first part, let $q \in \widehat{\Delta}(I, wa)$. Then there exists a state $s \in \widehat{\Delta}(I, w)$ such that $(s, a, q) \in \Delta$. This means that there is at least one reaction from equation (5.7) and Construction 5.1 that computes the transition $(s, a, q) \in \Delta$. Therefore we can bound the ODE corresponding to Z_q by

$$\begin{aligned}\frac{dz_q}{dt} &= -\hat{k}_1 x_r^* z_q + \sum_{(s,a,q)\in\Delta} \hat{k}_1 x_a^* y_s \bar{z}_q,\\ &> -\hat{k}_1 x_r^* z_q + \hat{k}_1 x_a^* y_s \bar{z}_q,\\ &> -(k_1+\delta_k)x_r^* z_q + (k_1-\delta_k)x_a^* y_s \bar{z}_q.\end{aligned}$$

During the interval $[b+5, b+6]$, the input signal is at the peak of the X_a-pulse, and therefore during the interval $[b+5.5, b+6]$ we know that

$$\frac{dz_q}{dt} > -(k_1+\delta_k)\gamma z_q + (k_1-\delta_k)(1-\gamma)y_s\bar{z}_q.$$

By Observation D.1, the set Y_Q η-encodes $\widehat{\Delta}(I, w)$ during $[b, b+8]$, and since $s \in \widehat{\Delta}(I, w)$ we know that

$$\frac{dz_q}{dt} > -(k_1+\delta_k)\gamma z_q + (k_1-\delta_k)(1-\gamma)(1-\eta)\bar{z}_q.$$

Now let $\hat{f}$, $\hat{b}$, and $\hat{p}$ be the constants

$$\hat{f} = (k_1-\delta_k)(1-\gamma)(1-\eta) \qquad \hat{b} = (k_1+\delta_k)\gamma \qquad \hat{p} = p(Z_q),$$

so that

$$\frac{dz_q}{dt} > \hat{f}\bar{z}_q - \hat{b}z_q.$$

Then by Lemma A.10, we have the bound

$$\begin{aligned}z_q(b+6) &> z_q(b+5.5)e^{-(\hat{f}+\hat{b})\frac{1}{2}} + \hat{p}\left(\frac{\hat{f}}{\hat{f}+\hat{b}}\right)\left(1-e^{-(\hat{f}+\hat{b})\frac{1}{2}}\right)\\ &> \hat{p}\frac{\hat{f}}{\hat{f}+\hat{b}} - \hat{p}e^{-\frac{\hat{f}}{2}} > \hat{p}\left(1-\frac{\hat{b}}{\hat{f}}\right) - \hat{p}e^{-\frac{\hat{f}}{2}} = \hat{p} - \hat{p}\left(\frac{\hat{b}}{\hat{f}} + e^{-\frac{\hat{f}}{2}}\right).\end{aligned}$$

Using the definition of k_1 from equation (5.12), it is easy to show that $k_1 > \delta_k + \frac{2}{(1-\gamma)(1-\eta)}\log\left(\frac{4(1+\delta_0)}{\eta}\right)$. Therefore we know that

$$z_q(b+6) > \hat{p} - \hat{p}\frac{\hat{b}}{\hat{f}} + \frac{\eta}{4}.$$

It is routine but easy to show that $\hat{p}\frac{\hat{b}}{\hat{f}} < \frac{\eta}{4}$, therefore we have the bound

$$z_q(b+6) > \hat{p} - \frac{\eta}{2}.$$

Finally, we know that Z_q is bounded during the interval $[b+6, b+12]$, Z_q by

$$\frac{dz_q}{dt} > -\hat{b}\hat{p} = -(k_1+\delta_k)\gamma p(Z_q) > -(k_1+\delta_k)\gamma(1+\delta_0).$$

It is also routine but easy to verify that $(k_1+\delta_k)\gamma(1+\delta_0) < \frac{\eta}{12}$, and therefore $\frac{dz_q}{dt} > -\frac{\eta}{12}$ during this interval. This means at most $\frac{\eta}{2}$ of Z_q can be destroyed by time $b+12$, and thus, for all $t \in [b+8, b+12]$, $z_q(t) > p(Z_q) - \eta > p(Z_q) - \frac{1}{20}$.

It remains to be shown that if $q \notin \widehat{\Delta}(I, wa)$, then for all $t \in [b+8, b+12]$ $z_q(t) < \frac{1}{20}$. Let $q \notin \widehat{\Delta}(I, wa)$. Then for all $(s, a, q) \in \widehat{\Delta}(I, wa)$, $s \notin \widehat{\Delta}(I, w)$. Therefore, we have the following bound for Z_q in the interval $[b+4, b+12]$.

$$\begin{aligned}\frac{dz_q}{dt} &= -\hat{k}_1 x_r^* z_q + \sum_{(s,a,q)\in\Delta} \hat{k}_1 x_a^* y_s \bar{z}_q < \sum_{(s,a,q)\in\Delta} \hat{k}_1 x_a^* y_s \bar{z}_q \\ &< |Q|(k_1+\delta_k)(1+\delta_0)\eta(1+\delta_0) = |Q|(k_1+\delta_k)(1+\delta_0)^2\eta.\end{aligned}$$

Therefore

$$\begin{aligned}z_q(b+12) &< z_q(b+4) + 8|Q|(k_1+\delta_k)(1+\delta_0)^2\eta \\ &< \eta + 8|Q|(k_1+\delta_k)(1+\delta_0)^2\eta < \frac{1}{20}. \quad \square\end{aligned}$$

D.3. Proof of copy back lemma

Proof of Lemma 5.10. In this proof, we assume the result of Lemma 5.9, i.e., Z_Q $\frac{1}{20}$-encodes $\widehat{\Delta}(I, wa)$ during the interval $[b+8, b+12]$. We now focus on the behavior of Y_q during the interval $[b+8, b+12]$. We begin by examining the ODE for Y_q which is

$$\frac{dy_q}{dt} = A + B,$$

where $A = \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q$ and $B = \hat{k}_2 y_q^2 \overline{y}_q - \hat{k}_2 y_q \overline{y}_q^2$. We begin by bounding the signal restoration part of the ODE with

$$\begin{aligned}B &> y_q\overline{y}_q\left((k_2-\delta_k)y_q - (k_2+\delta_k)\overline{y}_q\right) \\ &= -k_2 y_q \overline{y}_q \left(2\overline{y}_q - p(Y_q)\left(1 - \frac{\delta_k}{k_2}\right)\right).\end{aligned}$$

Since $y_q + \overline{y}_q = p(Y_q)$, it is not difficult to show that minimizing B under these constraints yields the inequality

$$B > -\frac{k_2}{6} p(Y_q)^3 \left(1 + \frac{\delta_k}{k_2}\right)^3.$$

Now let $q \in \widehat{\Delta}(I, wa)$. Then by Lemma 5.9, $z_q(t) > p(Z_q) - \frac{1}{20}$ for all $t \in [b+8, b+12]$. During the interval $[b+9, b+10]$, the input species X_c is at a peak which means that X_c^* is above $1-\gamma$ during the interval $[b+9.5, 10]$. Therefore the derivative for Y_q during this interval is bounded by

$$\begin{aligned}\frac{dy_q}{dt} &> \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q - \frac{k_2}{6} p(Y_q)^3 \left(1 + \frac{\delta_k}{k_2}\right)^3 \\ &= \hat{a}(\hat{p} - y_q) - \hat{b} y_q - \hat{c}\end{aligned}$$

where

$$\begin{aligned}\hat{a} &= k_2\left(1 - \frac{\delta_k}{k_2}\right)(1-\gamma)\left(1 - \delta_0 - \frac{1}{20}\right) \\ \hat{b} &= k_2\left(1 + \frac{\delta_k}{k_2}\right)(1+\delta_0)\frac{1}{20}\end{aligned}$$

$$\hat{c} = \frac{k_2}{6} p(Y_q)^3 \left(1 + \frac{\delta_k}{k_2}\right)^3$$
$$\hat{p} = p(Y_q).$$

This ODE is easily solvable, and therefore we obtain the bound

$$y_q(b+9.75) > \frac{\hat{p}\hat{a} - \hat{c}}{\hat{a}+\hat{b}} \left(1 - e^{-(\hat{a}+\hat{b})\frac{1}{4}}\right). \tag{D.1}$$

It is routine but easy to verify that

$$\frac{\hat{p}\hat{a} - \hat{c}}{\hat{a}+\hat{b}} > \frac{2}{3} p(Y_q) \left(1 + \frac{\delta_k}{k_2}\right). \tag{D.2}$$

Using the specification of k_2 from equation (5.13), it is clear that $k_2 > \delta_k + \frac{4\log 4}{(1-\gamma)(1-\delta_0-\frac{1}{20})}$. Therefore we know that

$$1 - e^{-(\hat{a}+\hat{b})\frac{1}{4}} > 1 - e^{-\hat{a}\frac{1}{4}} = \frac{3}{4}.$$

Plugging this inequality along with equation (D.2) into equation (D.1), we obtain

$$y_q(b+9.75) > \frac{2}{3} p(Y_q) \left(1 + \frac{\delta_k}{k_2}\right) \left(\frac{3}{4}\right) = \frac{p(Y_q)}{2} \left(1 + \frac{\delta_k}{k_2}\right).$$

During the interval $[b+9.75, 10]$, the derivative of Y_q is still bounded by

$$\begin{aligned} \frac{dy_q}{dt} &> \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q - k_2 y_q \overline{y}_q \left(2\overline{y}_q - p(Y_q)\left(1 - \frac{\delta_k}{k_2}\right)\right) \\ &= \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q + k_2 y_q \overline{y}_q \left(2y_q - p(Y_q)\left(1 + \frac{\delta_k}{k_2}\right)\right). \end{aligned}$$

Since the concentration of Y_q is greater than $\frac{p(Y_q)}{2}\left(1 + \frac{\delta_k}{k_2}\right)$ at time $t = 9.75$, we know that the ODE of Y_q during the interval $[b+9.75, b+10]$ is bounded by

$$\frac{dy_q}{dt} > \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q > \hat{a}(\hat{p} - y_q) - \hat{b} y_q.$$

By Lemma A.10 we obtain the bound

$$\begin{aligned} y_q(b+10) &> \hat{p}\left(\frac{\hat{a}}{\hat{a}+\hat{b}}\right)\left(1 - e^{-(\hat{a}+\hat{b})\frac{1}{4}}\right) + y_q(b+9.75) \cdot e^{-(\hat{a}+\hat{b})\frac{1}{4}} \\ &> \hat{p}\left(\frac{\hat{a}}{\hat{a}+\hat{b}}\right)\left(1 - e^{-\hat{a}\frac{1}{4}}\right) > \hat{p}\left(1 - \frac{\hat{b}}{\hat{a}}\right)\left(1 - e^{-\hat{a}\frac{1}{4}}\right) \\ &> \hat{p} - \hat{p}\frac{\hat{b}}{\hat{a}} - \hat{p}e^{-\hat{a}\frac{1}{4}}. \end{aligned}$$

It is routine but easy to verify that $\hat{p}\frac{\hat{b}}{\hat{a}} < \frac{1}{40}$ and $\hat{p}e^{-\hat{a}\frac{1}{4}} < \frac{1}{40}$, so we have the bound $y_q(b+10) > p(Y_q) - \frac{1}{20}$.

It remains to be shown that if $q \notin \widehat{\Delta}(I, wa)$ that $y_q(b+10) < \frac{1}{20}$. This holds by symmetry of Y_q and $\overline{Y}_q$. □

References

[1] Horst Alzer, On some inequalities for the incomplete gamma function, Math. Comput. 66 (218) (1997) 771–778.

[2] Dana Angluin, James Aspnes, David Eisenstat, A simple population protocol for fast robust approximate majority, Distrib. Comput. 21 (2) (2008) 87–102.

[3] Dana Angluin, James Aspnes, David Eisenstat, Eric Ruppert, The computational power of population protocols, Distrib. Comput. 20 (4) (2007) 279–304.

[4] Tom M. Apostol, Calculus, Volume 2: Multi-Variable Calculus and Linear Algebra with Applications to Differential Equations and Probability, John Wiley & Sons, 1969.

[5] Rutherford Aris, Prolegomena to the rational analysis of systems of chemical reactions, Arch. Ration. Mech. Anal. 19 (2) (1965) 81–99.

[6] Karl Johan Aström, Richard M. Murray, Feedback Systems: An Introduction for Scientists and Engineers, Princeton University Press, 2008.

[7] Stefan Badelt, Seung Woo Shin, Robert F. Johnson, Qing Dong, Chris Thachuk, Erik Winfree, A general-purpose CRN-to-DSD compiler with formal verification, optimization, and simulation capabilities, in: Proceedings of the 23rd International Conference on DNA Computing and Molecular Programming, in: Lecture Notes in Computer Science, 2017, pp. 232–248.

[8] Filippo Bonchi, Damien Pous, Checking NFA equivalence with bisimulations up to congruence, in: Proceedings of the 40th Symposium on Principles of Programming Languages, ACM, 2013, pp. 457–468.

[9] Filippo Bonchi, Damien Pous, Hacking nondeterminism with induction and coinduction, Commun. ACM 58 (2) (2015) 87–95.

[10] Luca Cardelli, Morphisms of reaction networks that couple structure to function, BMC Syst. Biol. 8 (2014) 84.
[11] Luca Cardelli, Attila Csikász-Nagy, The cell cycle switch computes approximate majority, Sci. Rep. 2 (2012).
[12] Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik, Georg Seelig, Programmable chemical controllers made from DNA, Nat. Nanotechnol. 8 (10) (2013) 755–762.
[13] Juan Cheng, Sarangapani Sreelatha, Ruizheng Hou, Artem Efremov, Ruchuan Liu, Johan R.C. van der Maarel, Zhisong Wang, Bipedal nanowalker by pure physical mechanisms, Phys. Rev. Lett. 109 (2012) 238104.
[14] Matthew Cook, David Soloveichik, Erik Winfree, Jehoshua Bruck, Programmability of chemical reaction networks, in: Anne Condon, David Harel, Joost N. Kok, Arto Salomaa, Erik Winfree (Eds.), Algorithmic Bioprocesses, in: Natural Computing Series, Springer, 2009, pp. 543–584.
[15] Frits Dannenberg, Marta Kwiatkowska, Chris Thachuk, Andrew J. Turberfield, DNA walker circuits: computational potential, design, and verification, in: Proceedings of the 19th International Conference on DNA Computing and Molecular Programming, in: Lecture Notes in Computer Science, vol. 8141, Springer, 2013, pp. 31–45.
[16] Domitilla Del Vecchio, Richard M. Murray, Biomolecular Feedback Systems, Princeton University Press, 2014.
[17] David Doty, Jack H. Lutz, Matthew J. Patitz, Robert T. Schweller, Scott M. Summers, Damien Woods, The tile assembly model is intrinsically universal, in: Proceedings of the 53rd Symposium on Foundations of Computer Science, IEEE, 2012, pp. 302–310.
[18] Shawn M. Douglas, Ido Bachelet, George M. Church, A logic-gated nanorobot for targeted transport of molecular payloads, Science 335 (6070) (2012) 831–834.
[19] Shawn M. Douglas, Adam H. Marblestone, Surat Teerapittayanon, Alejandro Vazquez, George M. Church, William M. Shih, Rapid prototyping of 3D DNA-origami shapes with caDNAno, Nucleic Acids Res. (2009) 1–6.
[20] Samuel J. Ellis, Titus H. Klinge, James I. Lathrop, Robust chemical circuits, in: Selected papers from the International Conference on the Theory and Practice of Natural Computing 2017, Biosystems 186 (2019) 103983, https://doi.org/10.1016/j.biosystems.2019.103983.
[21] Irving Robert Epstein, John Anthony Pojman, An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves, Patterns, and Chaos, Oxford University Press, 1998.
[22] Péter Érdi, János Tóth, Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models, Manchester University Press, 1989.
[23] François Fages, Guillaume Le Guludec, Olivier Bournez, Amaury Pouly, Strong Turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs, in: Proceedings of the 15th International Conference on Computational Methods in Systems Biology, Springer International Publishing, 2017, pp. 108–127.
[24] Walter Gautschi, The incomplete gamma functions since Tricomi, in: Tricomi's Ideas and Contemporary Applied Mathematics, vol. 147, Francesco Giacomo Tricomi (Ed.), Accademia Nazionale dei Lincei, 1998, pp. 203–237.
[25] Jeremy Gunawardena, Chemical reaction network theory for in-silico biologists, http://www.jeremy-gunawardena.com/papers/crnt.pdf, 2003.
[26] Dongran Han, Suchetan Pal, Jeanette Nangreave, Zhengtao Deng, Yan Liu, Hao Yan, DNA origami with complex curvatures in three-dimensional space, Science 332 (6027) (2011) 342–346.
[27] Thomas A. Henzinger, Jean-François Raskin, The equivalence problem for finite automata: technical perspective, Commun. ACM 58 (2) (2015) 86.
[28] Yonggang Ke, Luvena L. Ong, William M. Shih, Peng Yin, Three-dimensional structures self-assembled from DNA bricks, Science 338 (6111) (2012) 1177–1183.
[29] Titus H. Klinge, Robust signal restoration in chemical reaction networks, in: Proceedings of the 3rd International Conference on Nanoscale Computing and Communication, ACM, 2016, 6.
[30] Dexter Kozen, Automata and Computability, Springer, 1997.
[31] Gábor Lente, Deterministic Kinetics in Chemistry and Systems Biology: The Dynamics of Complex Reaction Networks, Springer, 2015.
[32] Fumiya Okubo, Takashi Yokomori, The computational capability of chemical reaction automata, Nat. Comput. 15 (2) (2016) 215–224.
[33] Fumiya Okubo, Takashi Yokomori, The computing power of determinism and reversibility in chemical reaction automata, in: Andrew Adamatzky (Ed.), Reversibility and Universality: Essays Presented to Kenichi Morita on the Occasion of his 70th Birthday, Springer International Publishing, 2018, pp. 279–298.
[34] Lulu Qian, Erik Winfree, Scaling up digital circuit computation with DNA strand displacement cascades, Science 332 (6034) (2011) 1196–1201.
[35] Lulu Qian, Erik Winfree, A simple DNA gate motif for synthesizing large-scale circuits, J. R. Soc. Interface 8 (62) (2011) 1281–1297.
[36] Lulu Qian, Erik Winfree, Jehoshua Bruck, Neural network computation with DNA strand displacement cascades, Nature 475 (7356) (2011) 368–372.
[37] Michael O. Rabin, Dana Scott, Finite automata and their decision problems, IBM J. Res. Dev. 3 (2) (1959) 114–125.
[38] Paul W.K. Rothemund, Folding DNA to create nanoscale shapes and patterns, Nature 440 (7082) (2006) 297–302.
[39] Nadrian C. Seeman, Nucleic acid junctions and lattices, J. Theor. Biol. 99 (2) (1982) 237–247.
[40] Lloyd M. Smith, Nanotechnology: molecular robots on the move, Nature 465 (7295) (2010) 167–168.
[41] David Soloveichik, Matthew Cook, Erik Winfree, Jehoshua Bruck, Computation with finite stochastic chemical reaction networks, Nat. Comput. 7 (4) (2008) 615–633.
[42] David Soloveichik, Georg Seelig, Erik Winfree, DNA as a universal substrate for chemical kinetics, Proc. Natl. Acad. Sci. USA 107 (12) (2010) 5393–5398.
[43] Gerald Teschl, Ordinary Differential Equations and Dynamical Systems, American Mathematical Society, 2012.
[44] Anthony S. Walsh, HaiFang Yin, Christoph M. Erben, Matthew J.A. Wood, Andrew J. Turberfield, DNA cage delivery to mammalian cells, ACS Nano 5 (7) (2011) 5427–5432.
[45] Bryan Wei, Mingjie Dai, Peng Yin, Complex shapes self-assembled from single-stranded DNA tiles, Nature 485 (7400) (2012) 623–626.
[46] Erik Winfree, Algorithmic self-assembly of DNA, PhD thesis, California Institute of Technology, 1998.
[47] Damien Woods, Intrinsic universality and the computational power of self-assembly, Philos. Trans. R. Soc. A, Math. Phys. Eng. Sci. 373 (2046) (2015).
[48] Bernard Yurke, Andrew J. Turberfield, Allen P. Mills, Friedrich C. Simmel, Jennifer L. Neumann, A DNA-fuelled molecular machine made of DNA, Nature 406 (6796) (2000) 605–608.
[49] David Yu Zhang, Georg Seelig, Dynamic DNA nanotechnology using strand-displacement reactions, Nat. Chem. 3 (2) (2011) 103–113.