2016

Modular and Robust Computation with Deterministic Chemical Reaction Networks

Titus Klinge
Iowa State University

Modular and robust computation with deterministic chemical reaction networks

by

Titus H. Klinge

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Computer Science

Program of Study Committee:
James I. Lathrop, Co-Major Professor
Jack H. Lutz, Co-Major Professor
Eric R. Henderson
Andrew S. Miner
Giora Slutzki

Iowa State University

Ames, Iowa

2016

TABLE OF CONTENTS

ACKNOWLEDGMENTS*

I thank my co-advisers, Jim Lathrop and Jack Lutz, whose role in my development as a scientist cannot be overstated. Both of them were a continuous source of encouragement and inspiration to me—not only by their words, but also by how they lived and worked. I am glad that Jim and Jack are not only my academic mentors, but also my good friends.

I also thank the members of the Iowa State Laboratory for Molecular Programming for their friendship and countless scientific discussions. In no particular order, I thank Adam Case, Gianfranco Ciardo, Sam Ellis, Eric Henderson, Xiang Huang, Xiaoyuan Li, Robyn Lutz, Divita Mathur, Andy Miner, Brian Nakayama, Don Nye, and Don Stull.

Many other friends were a source of encouragement during my studies. I especially thank Rob Barnes, Bronson Mayse, Brian Peck, Alden Peterson, Ian Willard, and Ben Wyatt for their support and friendship these five years.

Finally I thank my family. Everyone in my family has been very supportive and encouraging while I was at Iowa State. In particular, I thank my parents, Paul and Kathy, who have shaped the course of my life and my studies more than any others. I love you both dearly. I also thank my brothers, Nathan and Tim, for challenging me to move into more difficult studies early in my education. I wouldn't be where I am without those occasional nudges.

* This research was supported in part by National Science Foundation Grants 1247051 and 1545028.

ABSTRACT

In this thesis, we present four results concerning the computing capabilities of chemical reaction networks (CRNs) under deterministic mass action semantics[2]: (1) we introduce a modular method for computing concentration signals using CRN extension operators; (2) we present a thorough analysis of two CRN signal restoration algorithms that prevent certain concentration signals from degrading over time; (3) we introduce a new model called *input/output chemical reaction networks* (*I/O CRNs*) which generalizes the CRN model to allow receiving input signals over time; and (4) we investigate what I/O CRNs can compute robustly and prove that I/O CRNs are capable of robustly simulating any nondeterministic finite automaton (NFA).

CRN extension operators are operations that can be applied to a CRN to add extra functionality without affecting its original behavior. We show that common operators such as addition, multiplication, integration, and many others can be characterized as CRN extension operators. By iteratively applying these extensions, complex concentration signals can be modularly constructed from simple CRNs. To explore the full generality of these extensions, we introduce a notion of *weakly* CRN-computable signals and show that any CRN that can weakly compute a signal can be extended to exactly compute it.

The two signal restoration algorithms that we investigate are related to the approximate majority algorithm for population protocols originally developed by Angluin, Aspnes, and Eisenstat. Under deterministic semantics, these algorithms are commonly used to prevent discrete memory from deteriorating over time. We investigate the behavior of these algorithms in the presence of adversarial reactions and show that under modest

[2] CRNs under deterministic mass action semantics are also referred to as "mass action CRNs" or "CRNs with mass action kinetics."

conditions they are capable of maintaining discrete memory indefinitely. We also give tight analytical bounds on how these algorithms evolve over time.

The I/O CRN model that we introduce has two important impacts. (1) Concentration signals are a more natural way to provide arbitrarily long inputs to a CRN. Classical CRNs are restricted to encoding inputs into the initial state of the system which makes providing an arbitrarily long input (e.g. a binary string) quite difficult. (2) It promotes modular design of CRNs. Designing chemical systems from modular components that communicate via concentration signals is now possible, and we demonstrate its effectiveness in multiple constructions including our I/O CRN implementation of NFAs.

Our notion of robustness requires that an I/O CRN tolerate perturbations with respect to four things, namely, its initial state, rate constants, input signal, and the measurement of its output signal. We investigate what I/O CRNs are capable of computing under this notion of robustness, and we prove that I/O CRNs can robustly compute the regular languages by simulating an NFA in real time.

CHAPTER 1. INTRODUCTION

Molecular programming, also called DNA nanotechnology, is an emerging field of research aiming to algorithmically control the structure and function of matter at the molecular level. The origin of the field is attributed to Seeman in 1982 for his work on self-assembling DNA crystals [58] along with Winfree's 1998 work proving that this DNA self-assembly technique is, in principle, Turing universal [68]. Since then, the advancements of molecular programming have proceeded at a rate comparable to Moore's law and have led to methods for precise delivery of nanoscale cargo [25, 66, 19, 59], self-assembling arbitrary two- and three-dimensional nano-structures [56, 38, 11], and implementing DNA-based circuits and neural nets [52, 53, 54]. Many useful models for chemical systems have also been proven to be Turing universal [6, 20, 51, 57], including certain types of chemical reaction networks [63].

Chemical reaction networks (CRNs) are an abstract mathematical model used to study the dynamics of well-mixed chemical systems, and they date back to Aris' work in 1965 [8]. In the 20th century, CRNs were used to study various chemical phenomena and to explore interesting chemical behavior that might be possible. They were not considered a practical design tool, because finding actual chemicals to implement a CRN was often a long and difficult process. Today, CRNs are regarded as a programming language (in the literal sense of computer science) due to some results in the last decade. In 2009, Soloveichik, Seelig, and Winfree showed that any CRN can, in principle, be implemented with DNA strands [24], and their work has since been refined into a CRN-to-DNA compiler [18]. These DNA constructions simulate the chemical reactions of a CRN with a process called

DNA strand displacement introduced by Yurke et al. in 2000 [69, 46, 70, 65]. Due to this CRN compiler, CRNs became a more practical development tool, and understanding their computing ability became an important research topic.

CRNs are used to model several types of chemical systems and as a result have multiple semantics (i.e. operational meanings). The two most prevalent CRN semantics are called stochastic mass action semantics and deterministic mass action semantics, and we often refer to CRNs under such semantics as stochastic CRNs and deterministic CRNs, respectively. Stochastic CRNs (SCRNs) discretely model the state of the system (i.e. amount of each molecule present) as a vector of integers and are used to model chemical systems contained in a small volume (e.g. inside a cell). The SCRN model is essentially equivalent to other well-known models of computation such as Petri nets [49, 35], vector addition systems [50, 43], and population protocols [6, 23]. SCRNs have also been shown to be Turing universal if some small probability of error is allowed [63], but without error they are limited to computing the semi-linear predicates [4]. Other important SCRN computability results can be found in [20, 23, 21, 15, 16] and the references therein.

Deterministic CRNs (DCRNs) model the state of the system as a vector of real-valued *concentrations* and are used to model chemical systems in relatively large volumes which can be modeled continuously[1]. The DCRN model is the deterministic limit of the SCRN model [45, 31], and a state of a DCRN evolves according to autonomous polynomial ordinary differential equations (ODEs). DCRNs only form a restricted class of polynomial ODEs but are capable of approximating any polynomial system with arbitrarily small error, over any fixed time interval, and with only bimolecular reactions [42, 41, 67]. They can also be used to implement various models of computation including finite logic circuits and register machines with bounded register capacity [37]. Today it remains an open question whether DCRNs are Turing universal (i.e. simulate a Turing machine with unbounded memory). Several claims that DCRNs are universal have surfaced including

[1] A single milliliter of water at room temperature contains over 10^{22} molecules, so "relatively large" here includes some fairly small volumes.

Magnasco [48] and Stansifer [20]. However, Magnasco only demonstrates that DCRNs can simulate finite circuits, and Stansifer's proof has not appeared in print. Arbitrary polynomial ODEs are legitimately Turing universal [32], but this construction uses ODEs that are not implementable by DCRNs [34].

These computability results make CRNs an attractive model for molecular programming, but there are numerous obstacles to overcome before they truly become practical. One obstacle is that CRNs are rather primitive, and even simple algorithms such as detecting the absence of a signal are challenging to design [27, 26]. Software tools such as Microsoft Visual GEC[2] and MATLAB SimBiology[3] provide molecular programmers with CRN simulation utilities, but the burden of design is still left to the developer.

Another obstacle is robustness. Many applications of molecular programming are safety critical, therefore ensuring that a CRN can tolerate worst case conditions is often necessary. Furthermore, the CRN model is only an approximation of a physical system, so CRNs that depend on unrealistically precise parameters have little hope of functioning once physically implemented. This has led to research of the CRN model when parameters are not well defined, such as initial conditions [60, 2] and rate constants [17, 16, 62], but for CRNs to truly be practical, they must be robust with respect to both of these aspects.

In this thesis, we present results regarding the deterministic CRN model that help toward overcoming these two obstacles. To reduce the burden of CRN design, we introduce two separate notions for modular CRN development. The first notion is called a *closed sub-CRN* which is a CRN that is "contained within another CRN." Closed sub-CRNs are defined using a Kuratowski closure operator [10, 61] that guarantees that the sub-CRN is "closed" in the sense that it does not depend on the enclosing CRN in any way. However, the enclosing CRN may depend on the sub-CRN to facilitate a more complex computation. Thus closed sub-CRNs can be regarded as independent *modules* of the enclosing CRN and can be studied and tested in isolation.

[2] Visual GEC by Microsoft: http://lepton.research.microsoft.com/webgec.
[3] MATLAB SimBiology by MathWorks, Inc.: http://www.mathworks.com/products/simbiology.

We formally introduce closed sub-CRNs in section 2.1, and in chapter 3 we explore their use for computing complex concentration signals. Specifically, we use closed sub-CRNs to define *CRN extension operators* which are operations that can be used to extend CRNs in certain ways. When an extension operator is applied to a CRN, it produces a new CRN that contains new functionality while preserving the behavior of the original CRN with a closed sub-CRN. CRN extension operators are related to Cardelli's notion of CRN morphisms [14], and every CRN produced by a CRN extension operator can emulate the original CRN via one of these morphisms. We also prove that many common operators can be formulated as CRN extension operators including addition, multiplication, integration, and many others. By iteratively applying extension operators, it is possible to modularly construct complex concentration signals from relatively simple CRNs. To fully explore the generality of this extension method, we define a notion for CRNs *weakly* computing a concentration signal. We say that a CRN weakly computes a concentration signal if that concentration signal can be produced by a polynomial combination of the output signals of the CRN. We also prove that if a CRN can weakly compute a signal, then it can be extended via an operator to explicitly compute that signal.

Our second notion of modularity is introduced in chapter 5 and is a new model called *input/output chemical reaction networks* (*I/O CRNs*). An I/O CRN is a generalization of a deterministic CRN that has a provision for receiving input over time via a concentration signal and naturally promotes modular development. For instance, a complex I/O CRN could be composed of two or more simpler I/O CRNs that pass signals to and from one another. These smaller I/O CRNs might be dependent on one another via these signals, so neither of the I/O CRNs could be considered *closed* in the same sense of a closed sub-CRN. Rather, these I/O CRNs are dependent on one another through an *interface* of the concentration signals passed between them. In the classic CRN model, two processes that depend on one another cannot be easily separated and tested in isolation. Whereas,

the I/O CRN model naturally supports this separation, making it easier to develop complex systems by joining multiple modules together.

Another major aspect of this thesis concerns robustness. In chapter 5 we also define what it means for an I/O CRN to robustly satisfy a requirement. Specifically, our notion of robustness requires that an I/O CRN be robust with respect to four things: (i) input signal, (ii) initial state, (iii) rate constants, and (iv) output measurement. We chose these four aspects because if an I/O CRN fails to be robust in any of these ways, there often are significant consequences. In practice, we do not have perfectly precise control over these areas. For instance, even with a CRN-to-DNA compiler, rate constants can only be approximated. Therefore if a CRN depends on these rate constants being exact, it certainly will not behave as designed. Even the assumption that rate constants are actually "constant" is not appropriate because fluctuations in temperature and other environmental factors cause these rate constants to vary over time. Similarly, an I/O CRN that requires a perfectly precise input signal or initial condition cannot always be relied upon, so it is important to be able to tolerate perturbations in these as well.

Using this notion of robustness, in chapter 6 we present an I/O CRN that is capable of robustly cleaning up an input signal that encodes a binary sequence. More precisely, these concentration signals encode bits by the contrast of "high" and "low" concentrations. The signal conditioning I/O CRN specifically takes input signals of this form and "cleans them up" so that the output signal better approximates an ideal square wave. One of the most interesting challenges overcome by this I/O CRN is that it is capable of getting the output signal to be arbitrarily close to zero when it needs to be low. Therefore this system can be easily used to prepare signals to be used as a *switch* to cleanly turn on and off certain modules of an I/O CRN. In fact, our I/O CRN construction for computing NFAs heavily relies on the signal conditioning I/O CRNs presented in this chapter.

In chapter 7, we show that I/O CRNs are capable of robustly simulating nondeterministic finite automata (NFAs). NFAs are over half a century old [55] and far from

Turing universal, but they have many applications and remain an active research topic [12, 36, 13]. We translate an arbitrary NFA to an I/O CRN that simulates it 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. The I/O CRN also consists of two modules. The first module is the signal conditioning module constructed in chapter 6.1 that transforms the input signals into approximate square waves. The second module actually simulates the NFA by computing the transition function and encoding the current states by contrasting high and low concentrations of state species.

Our NFA translation also depends on a *signal restoration* algorithm which is a memory refresh technique that utilizes bistability to prevent these concentration signals from deteriorating over time. In chapter 4, we give a thorough analysis of two signal restoration algorithms in deterministic CRNs and give sufficient conditions for their correctness even in the presence of adversarial reactions. The first algorithm consists of two termolecular reactions (reactions with three reactants, i.e., input molecules) and is the simpler of the two algorithms [40, 37]. The simplicity of this algorithm makes it easy to use and relatively easy to analyze. We also give an exact analytical solution for the runtime of this algorithm that demonstrates that the algorithm converges in logarithmic time.

The second signal restoration algorithm consists of four bimolecular reactions and was developed in 2008 by Angluin, Aspnes, and Eisenstat for population protocols [23, 3]. This second algorithm is significantly harder to analyze, but it has the advantage that bimolecular reactions are easier to physically implement. By utilizing a precise relationship with the termolecular algorithm, we give precise asymptotic analysis of the algorithm and sufficient conditions for it to work in the presence of adversarial reactions. We also give tight analytical bounds on its runtime and show it also converges in logarithmic time.

CHAPTER 2. CHEMICAL REACTION NETWORKS*

In this chapter we review the chemical reaction network model under deterministic mass action semantics. Formally, we fix a countable set $\mathbf{S}$ whose elements we call *species*. Species are abstract molecule types and are typically denoted by capital roman characters (e.g. X, Y, Z), but to differentiate species that are related we often use various decorations to differentiate between them (e.g. $X, X_0, \overline{X}, \widehat{X}$).

A *reaction* over a finite set $S \subseteq \mathbf{S}$ is a triple $\rho = (\mathbf{r}, \mathbf{p}, k)$ where $\mathbf{r} \in \mathbb{N}^S$ is the *reactant vector*, $\mathbf{p} \in \mathbb{N}^S$ is the *product vector*, and $k \in (0, \infty)$ is the *rate constant*, respectively. Here we use $\mathcal{X}^{\mathcal{Y}}$ to denote the set of all functions mapping the set $\mathcal{Y}$ into $\mathcal{X}$. Therefore elements of $\mathbb{N}^S$ can naturally be regarded as vectors of natural numbers that are indexed by the elements of S. Given a reaction $\rho = (\mathbf{r}, \mathbf{p}, k)$, we use the notation $\mathbf{r}(\rho) = \mathbf{r}$, $\mathbf{p}(\rho) = \mathbf{p}$, and $k(\rho) = k$ to access the individual components of ρ.

For a reaction $\rho = (\mathbf{r}, \mathbf{p}, k)$, a species $X \in S$ is called a *reactant* of ρ if $\mathbf{r}(X) > 0$, called a *product* of ρ if $\mathbf{p}(X) > 0$, and called a *catalyst* of ρ if $\mathbf{r}(X) = \mathbf{p}(X) > 0$. Intuitively, the reactants and products of a reaction are the molecules consumed and produced by the reaction, respectively. The *net effect* of a reaction ρ is the vector $\Delta\rho = \mathbf{p}(\rho) - \mathbf{r}(\rho) \in \mathbb{Z}^S$.

We also borrow the intuitive notation of chemistry to specify reactions. For example, if $S = \{A, B, C\} \subseteq \mathbf{S}$, we write

$$A + 2B \xrightarrow{k} A + C \tag{2.1}$$

* The exposition in this chapter is joint work with Jim Lathrop and Jack Lutz and appeared in earlier form in [40].

to specify the reaction ρ where $k(\rho) = k$ and the vectors $\mathbf{r}(\rho)$ and $\mathbf{p}(\rho)$ are defined by

$$\mathbf{r}(\rho)(A) = 1, \qquad \mathbf{r}(\rho)(B) = 2, \qquad \mathbf{r}(\rho)(C) = 0,$$
$$\mathbf{p}(\rho)(A) = 1, \qquad \mathbf{p}(\rho)(B) = 0, \qquad \mathbf{p}(\rho)(C) = 1.$$

Here the species A and B are the reactants of ρ, A and C are the products of ρ, and the net effect of the reaction is the vector defined by $\Delta\rho(A) = 0$, $\Delta\rho(B) = -2$, and $\Delta\rho(C) = 1$. The species A is also a catalyst of ρ because it participates in the reaction but is unaffected by it.

A *chemical reaction network* (*CRN*) is an ordered pair $N = (S, R)$ where $S \subseteq \mathbf{S}$ is a finite set of species and R is a finite set of reactions over S. At this point, we have fully specified the *syntax* of a CRN. The rest of the chapter is dedicated to specifying the *deterministic mass action semantics* of CRNs.

Under deterministic mass action semantics, a *state* of a CRN $N = (S, R)$ is a vector $\mathbf{x} \in [0, \infty)^S$, and for each $Y \in S$, we call $\mathbf{x}(Y)$ the *concentration* of Y in $\mathbf{x}$.

Given a finite set $S \subseteq \mathbf{S}$ of species, we define the *S-signal space* to be the set $C[S] = C([0, \infty), [0, \infty)^S)$, where $C(\mathcal{X}, \mathcal{Y})$ is the set of continuous functions from $\mathcal{X}$ to $\mathcal{Y}$. A function $\mathbf{x} \in C[S]$ is called a *concentration signal* and assigns a state $\mathbf{x}(t) \in [0, \infty)^S$ to each time $t \in [0, \infty)$. (Our use of $\mathbf{x}$ to denote both a single state and a concentration signal is deliberate to reduce obfuscation.) If a concentration signal $\mathbf{x} \in C[S]$ is obvious from context, for each species $Y \in S$ we use the corresponding lowercase character y to denote the concentration of Y in the concentration signal $\mathbf{x}$. For example, if $S = \{X, Y, Z\}$, then we write $x(t)$, $y(t)$, and $z(t)$ to denote the concentrations $\mathbf{x}(t)(X)$, $\mathbf{x}(t)(Y)$, and $\mathbf{x}(t)(Z)$ at time t, respectively.

Given a CRN $N = (S, R)$, a reaction $\rho \in R$, and a state $\mathbf{x} \in [0, \infty)^S$, we define the *rate of* $\rho \in R$ *in* $\mathbf{x}$ to be the real-valued quantity

$$\text{rate}_{\mathbf{x}}(\rho) = k(\rho) \prod_{Y \in S} \mathbf{x}(Y)^{\mathbf{r}(\rho)(Y)}. \tag{2.2}$$

Therefore, if ρ is the reaction $A + 2B \xrightarrow{k} A + C$, then $\text{rate}_{\mathbf{x}}(\rho) = k \cdot \mathbf{x}(A) \cdot \mathbf{x}(B)^2$. Intuitively, the rate of a reaction is proportional to the frequency the reactants collide with one another, and the rate constant encapsulates factors that do not depend on the state (e.g. temperature).

For a CRN $N = (S, R)$ and species $Y \in S$, the *deterministic mass action rate function of Y in N* is the function $F_Y : [0, \infty)^S \to \mathbb{R}$ defined by

$$F_Y(\mathbf{x}) = \sum_{\rho \in R} \Delta\rho(X) \cdot \text{rate}_{\mathbf{x}}(\rho) \tag{2.3}$$

for all $\mathbf{x} \in [0, \infty)^S$. The function $F_Y(\mathbf{x})$ specifies the total rate at which the concentration of Y is changing in the state $\mathbf{x}$. Hence, if the CRN $N = (S, R)$ is in the state $\mathbf{x}(t) \in [0, \infty)^S$ at time t, then each species $Y \in S$ must satisfy the ordinary differential equation (ODE)

$$y'(t) = F_Y(\mathbf{x}(t)). \tag{2.4}$$

We say that a state $\mathbf{x} \in [0, \infty)^S$ is an *equilibrium state* (or *equilibrium point*) if $F_Y(\mathbf{x}) = 0$ for all $Y \in S$.

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

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

for all $\mathbf{x} \in [0, \infty)^S$, then the *deterministic mass action system* of the CRN is the vector

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

An *initialized chemical reaction network* (*ICRN*) is an ordered pair $(N, \mathbf{x}_0)$ where $N = (S, R)$ is a CRN and $\mathbf{x}_0 \in [0, \infty)^S$ is an *initial state* of N. The ICRN $(N, \mathbf{x}_0)$ specifies an initial value problem consisting of the mass action system (2.6) along with

the initial condition $y(0) = \mathbf{x}_0(Y)$ for all $Y \in S$. By the standard existence-uniqueness theory for ODEs [7, 64], this initial value problem has a unique solution $\mathbf{x}_{N,\mathbf{x}_0} \in C[S]$ defined for some interval $[0, b)$ where $b \in (0, \infty]$.

Note that CRNs are *autonomous* in the two equivalent senses that (i) the right-hand side of the system (2.6) 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 (2.6) without further outside influence. It is also clear by inspection of (2.2)-(2.6) that the deterministic mass action system (2.6) of a CRN is *polynomial*, meaning that the components of the vector $F(\mathbf{x}(t))$ are polynomial in the components $y(t)$ of $\mathbf{x}(t)$.

We define the *solution space* of a CRN $N = (S, R)$ to be the set

$$\mathbf{sol}(N) = \{\mathbf{x}_{N,\mathbf{x}_0} \mid \mathbf{x}_0 \in [0, \infty)^S\} \tag{2.7}$$

of solutions ranging over all possible initial states. For each solution $\mathbf{x} \in \mathbf{sol}(N)$, we define the *domain* of $\mathbf{x}$, and we write $\mathrm{dom}(\mathbf{x})$, to be the maximal interval $[0, b)$ for which $\mathbf{x}$ is defined.

Under deterministic mass action semantics, the set $\mathbf{sol}(N)$ completely specifies the possible behaviors of the CRN $N = (S, R)$. Further discussions of chemical reaction networks with deterministic mass action semantics appear in [29, 28, 33, 47].

2.1 Closed Sub-CRNs

In this section, we formally define our notion of a closed sub-CRN, but first we define some necessary notation and terminology needed to specify the notion properly.

Let $N = (S, R)$ be a CRN. For species $X \in S$ and $Y \in S$ and reaction $\rho \in R$, we say that X *affects* Y *via* ρ, and we write $X \underset{\rho}{\longrightarrow} Y$, if $\mathbf{r}(\rho)(X) > 0$ and $\Delta\rho(Y) \neq 0$. In other words, $X \underset{\rho}{\longrightarrow} Y$ if and only if the rate of the reaction ρ depends on the concentration of X in some way *and* ρ has a non-zero net effect on Y.

Let $f^n : \mathcal{P}(S) \to \mathcal{P}(S)$ for all $n \in \mathbb{Z}^+$ be a family of functions defined by the recursion

$$f^1(T) = \{Y \mid (\exists X \in T)(\exists \rho \in R)\ X \underset{\rho}{\longrightarrow} Y\}, \tag{2.8}$$

$$f^{n+1}(T) = f^1(f^n(T)), \tag{2.9}$$

for all $T \subseteq S$. Here we are using $\mathcal{P}(S)$ to denote the power set of the set S (i.e. the set of all possible subsets of S). For $T \subseteq S$, let the sets $S_T \subseteq S$ and $R_T \subseteq R$ be defined by

$$S_T = \bigcup_{i=1}^{\infty} f^n(T), \tag{2.10}$$

$$R_T = \{\rho \in R \mid (\exists X \in S_T)\ \Delta\rho(X) \neq 0\}, \tag{2.11}$$

noting that S_T and R_T are both finite. We call the CRN $N_T = (S_T, R_T)$ the *closure of* T *in* N.

We say $N = (S, R)$ is a *closed sub-CRN* of $\widehat{N} = (\widehat{S}, \widehat{R})$, and we write $N \trianglelefteq \widehat{N}$, if $N = \widehat{N}_S$. Intuitively, $N \trianglelefteq \widehat{N}$ holds when N is closed in the sense that it is unaffected by the parts of $\widehat{N}$ outside of N, and therefore N is completely independent and self-sustaining. However, the CRN $\widehat{N}$ may depend on the closed sub-CRN N contained within it.

If $N = (S, R)$ is a CRN, $T \subseteq S$ and $\mathbf{x} \in [0, \infty)^S$ is a state of N, then the *restriction of* $\mathbf{x}$ to T is the projection $\mathbf{x}_T \in [0, \infty)^T$ defined by

$$\mathbf{x}_T(Y) = \mathbf{x}(Y) \tag{2.12}$$

for all $Y \in T$. Similarly, if $\mathbf{x} \in C[S]$ is a concentration signal, the *restriction of* $\mathbf{x}$ to T is the vector-valued function $\mathbf{x}_T$ defined by

$$\mathbf{x}_T(t)(Y) = \mathbf{x}(t)(Y) \tag{2.13}$$

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

CHAPTER 3. CRN EXTENSION OPERATORS*

In this chapter, we examine the possible concentration signals that CRNs can produce and introduce a modular means of constructing these signals via *CRN extension operators*. We begin by formally defining what it means for a concentration signal to be *CRN-computable* along with other definitions necessary to develop this notion. We then prove various closure properties of the set of CRN-computable concentration signals. These properties are intentionally proven in a way that makes designing modular concentration signals natural.

Definition 3.1. A concentration signal $\mathbf{u} \in C[U]$ is *CRN-computable* if there exists a CRN $N = (S, R)$ with $U \subseteq S$ that has a solution $\mathbf{x} \in \mathbf{sol}(N)$ such that $\mathbf{x}_U = \mathbf{u}$. The set of all such CRN-computable concentration signals is denoted **CCRN**.

It is important to note that a CRN may use auxiliary species that do not appear in the computed signal (i.e. the set $S \setminus U$ may be non-empty). This greatly increases the total number of CRN-computable functions because extra dimensions may be utilized to compute the signal.

Example 3.2. The concentration signal $x(t) = e^t$ is CRN-computable.

Proof. Let $N = (S, R)$ be a CRN with $S = \{X\}$ and $R = \{\rho\}$ where ρ is the reaction $X \xrightarrow{1} 2X$. Let $\mathbf{x}_0 \in [0, \infty)^S$ be an initial state of N such that $\mathbf{x}_0(X) = 1$. Then the ICRN $(N, \mathbf{x}_0)$ has a unique solution $\mathbf{x} \in \mathbf{sol}(N)$. It suffices to show that this solution satisfies $x(t) = e^t$ for all $t \in [0, \infty)$.

* The material in this chapter is joint work with Jim Lathrop and Jack Lutz and will appear in a forthcoming paper.

According to equation (2.4), the concentration $x(t)$ must obey the ODE

$$x'(t) = x(t). \tag{3.1}$$

This ODE is easily solvable by separation of variables and has solution $x(t) = x(0) \cdot e^t$. It follows by the initial condition $x(0) = 1$ that $x(t) = e^t$ for all $t \in [0, \infty)$. □

Although in the above example we built the concentration signal for X from scratch, this proves to be difficult for complicated signals. Therefore we now begin developing a natural means of extending existing CRN-computable concentration signals to produce more complex signals.

Definition 3.3. A *concentration signal operator* is a function $H : C[U] \to C[V]$ where $U, V \subseteq \mathbf{S}$ are finite and $U \cap V = \emptyset$. We say that the set **CCRN** of CRN-computable functions is *closed under operator* H if for all $\mathbf{u} \in C[U]$

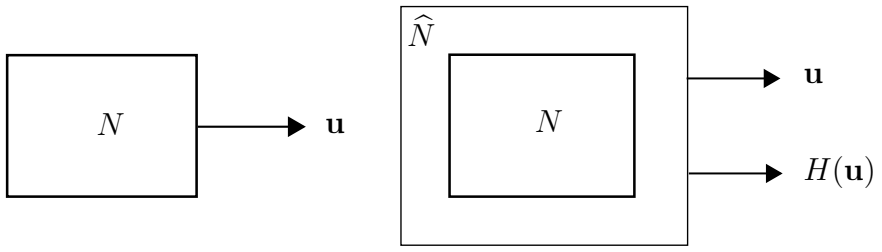
$$\mathbf{u} \in \mathbf{CCRN} \implies H(\mathbf{u}) \in \mathbf{CCRN}.$$

This notion of a concentration signal operator is not only useful for proving closure properties on the set **CCRN** but also for modular development of concentration signals.

Definition 3.4. A concentration signal operator $H : C[U] \to C[V]$ is an *extension operator* if for every CRN $N = (S, R)$ with $U \subseteq S$ and $S \cap V = \emptyset$, there exists a CRN $\widehat{N} = (\widehat{S}, \widehat{R})$ such that $N \trianglelefteq \widehat{N}$, $V \subseteq \widehat{S}$, and for every solution $\mathbf{x} \in \mathbf{sol}(N)$, there exists a solution $\hat{\mathbf{x}} \in \mathbf{sol}(\widehat{N})$ where $\hat{\mathbf{x}}_S = \mathbf{x}$ and $\hat{\mathbf{x}}_V = H(\mathbf{x}_U)$.

Observation 3.5. The set **CCRN** of CRN-computable signals is closed under all extension operators.

Intuitively, an extension operator is an operator that can be "added on" to an existing CRN while preserving the CRNs original behavior. This is demonstrated in figure 3.1.



(a) Original CRN (b) Extended CRN

Figure 3.1: A visual depiction of an extension operator $H : C[U] \to C[V]$ applied to a CRN N to produce the CRN $\widehat{N}$.

Lemma 3.6. The concentration signal operator $H : C[\{X\}] \to C[\{Y\}]$ defined by

$$H(\mathbf{u})(t)(Y) = \int_0^t \mathbf{u}(s)(X)ds \tag{3.2}$$

for all $\mathbf{u} \in C[\{X\}]$ and $t \in [0, \infty)$, is an extension operator.

Proof. Let H be as given, and let $N = (S, R)$ be a CRN such that $X \in S$ and $Y \notin S$. Let $\widehat{N} = (\widehat{S}, \widehat{R})$ be the CRN defined by $\widehat{S} = S \cup \{Y\}$ and $\widehat{R} = R \cup \{\rho\}$ where ρ is the reaction $X \xrightarrow{1} X + Y$. Since ρ only uses species in S as catalysts, it has no effect on the original behavior of N. It follows that N is a closed sub-CRN of $\widehat{N}$.

Now let $\mathbf{x} \in \mathbf{sol}(N)$, and let $\hat{\mathbf{x}}_0$ be an initial state of $\widehat{N}$ such that $\hat{\mathbf{x}}_0(Y) = 0$ and $\hat{\mathbf{x}}_0(Z) = \mathbf{x}(0)(Z)$ for each $Z \in S$. Then the ICRN $(\widehat{N}, \hat{\mathbf{x}}_0)$ specifies a unique solution $\hat{\mathbf{x}} \in \mathbf{sol}(\widehat{N})$. Since $N \trianglelefteq \widehat{N}$ and $\mathbf{x}(0)(Z) = \hat{\mathbf{x}}(0)(Z)$ for all species $Z \in S$, it follows that $\hat{\mathbf{x}}_S = \mathbf{x}$.

It remains to be shown that $\hat{\mathbf{x}}(t)(Y) = \int_0^t \mathbf{x}(s)(X)ds$ for all $t \in [0, \infty)$. To show this, we examine the ODE for Y in $\widehat{N}$. Since the only reaction affecting Y is ρ, the ODE corresponding to Y is

$$y'(t) = x(t).$$

By integrating both sides of this equation, we obtain

$$y(t) - y(0) = \int_0^t x(s)ds.$$

The lemma immediately follows by the initial condition, $y(0) = 0$. □

This lemma demonstrates that it is possible to nondestructively compute the definite integral of any concentration signal from any CRN by simply adding one species and one reaction. By iteratively applying different extension operators to a CRN, it is possible to modularly construct complex concentration signals without affecting any intermediate functionality.

There are also some powerful concentration signal constructions that are *not* CRN extension operators. For example, the following theorem shows that it is possible to dynamically speed up or slow down the behavior of a CRN by modifying every reaction of the CRN to include one extra catalyst. This catalyst is then used as a mechanism to uniformly affect the speed of the CRN computation.

Theorem 3.7 (Time Dilation)**.** If $X \notin U$, then **CCRN** is closed under the operator $H : C[U \cup \{X\}] \to C[U]$ defined by

$$H(\mathbf{u}, x)(t) = \mathbf{u}\left(\int_0^t x(s)ds\right) \tag{3.3}$$

for all $\mathbf{u} \in C[U]$ and $x \in C[\{X\}]$.

Proof. Assume the hypothesis, and let $\mathbf{u} \in C[U]$ and $x \in C[\{X\}]$ be two CRN-computable functions. It suffices to show that $H(\mathbf{u}, x)$ is also CRN-computable.

Let $N^{(1)} = (S^{(1)}, R^{(1)})$ and $N^{(2)} = (S^{(2)}, R^{(2)})$ be CRNs that compute $\mathbf{u}$ and x, respectively, such that $S^{(1)} \cap S^{(2)} = \emptyset$. Let $N = (S, R)$ be a CRN with $S = S^{(1)} \cup S^{(2)}$ and $R = \widehat{R}^{(1)} \cup R^{(2)}$ where $\widehat{R}^{(1)} = \{\hat{\rho} \mid \rho \in R^{(1)}\}$ and where each reaction $\hat{\rho}$ is a copy of ρ except with X added as a catalyst, i.e.,

$$\begin{aligned}
&\mathbf{r}(\hat{\rho})(X) = \mathbf{p}(\hat{\rho})(X) = 1,\\
&\mathbf{r}(\hat{\rho})(Y) = \mathbf{p}(\hat{\rho})(Y) = 0 \text{ for all } Y \in S^{(2)} \setminus \{X\},\\
&\mathbf{r}(\hat{\rho})(Y) = \mathbf{r}(\rho)(Y) \text{ for all } Y \in S^{(1)}, \text{ and}\\
&\mathbf{p}(\hat{\rho})(Y) = \mathbf{p}(\rho)(Y) \text{ for all } Y \in S^{(1)}.
\end{aligned}$$

Since the species in $S^{(2)}$ are unaffected by these reactions, it is clear that $N^{(2)} \trianglelefteq N$.

Let $\mathbf{x}^{(1)} \in \mathbf{sol}(N^{(1)})$ and $\mathbf{x}^{(2)} \in \mathbf{sol}(N^{(2)})$ be solutions that compute $\mathbf{u}$ and x, respectively, and let $\mathbf{x} \in \mathbf{sol}(N)$ be the solution with $\mathbf{x}(0)(Y) = \mathbf{x}^{(1)}(0)(Y)$ for all $Y \in S^{(1)}$ and $\mathbf{x}(0)(Z) = \mathbf{x}^{(2)}(0)(Z)$ for each $Z \in S^{(2)}$.

It now suffices to show that $\mathbf{x}_U = H(\mathbf{u}, x)$.

Let $f : [0, \infty) \to [0, \infty)$ be a function defined by $f(t) = \int_0^t x(s)ds$. For each species $Y \in S^{(1)}$, let $\hat{y} : [0, \infty) \to [0, \infty)$ be the function defined by $\hat{y}(t) = \mathbf{x}^{(1)}(f(t))(Y)$. Equation (2.4) and the chain rule tell us that the derivative of each $\hat{y}$ is

$$\frac{d\hat{y}}{dt} = \frac{d}{dt}\left(\mathbf{x}^{(1)}(f(t))(Y)\right) = f'(t) \cdot F_Y^{(1)}(\mathbf{x}^{(1)}(f(t)))$$

where $F_Y^{(1)}$ is the deterministic mass action rate function of species Y in $N^{(1)}$. By equations (2.2) and (2.3) along with the fact that $f'(t) = x(t)$,

$$\begin{aligned}\frac{d\hat{y}}{dt} &= x(t) \sum_{\rho \in R^{(1)}} \Delta\rho(Y) \cdot \mathrm{rate}_{\mathbf{x}^{(1)}(f(t))}(\rho) \\ &= x(t) \sum_{\rho \in R^{(1)}} \Delta\rho(Y) \cdot k(\rho) \prod_{Z \in S^{(1)}} \mathbf{x}^{(1)}(f(t))(Z)^{\mathbf{r}(\rho)(Z)}.\end{aligned}$$

Recall that for each $Z \in S^{(1)}$, we defined the function $\hat{z}(t) = \mathbf{x}^{(1)}(f(t))(Z)$. Therefore

$$\frac{d\hat{y}}{dt} = x(t) \sum_{\rho \in R^{(1)}} \Delta\rho(Y) \cdot k(\rho) \prod_{Z \in S^{(1)}} \hat{z}(t)^{\mathbf{r}(\rho)(Z)}. \tag{3.4}$$

Now, we will compare these ODEs to those of each species $Y \in S^{(1)}$ in the CRN N. Equation (2.4) tells us that for each $Y \in S^{(1)}$,

$$\mathbf{x}'(t)(Y) = F_Y(\mathbf{x}(t)),$$

where F_Y is the deterministic mass action function of Y in N, and therefore

$$\mathbf{x}'(t)(Y) = \sum_{\rho \in R} \Delta\rho(Y) \cdot \mathrm{rate}_{\mathbf{x}(t)}(\rho).$$

Since each species $Y \in S^{(1)}$ is only affected by the reactions in $\widehat{R}^{(1)}$ in N, we can rewrite this ODE as

$$\mathbf{x}'(t)(Y) = \sum_{\hat{\rho} \in \widehat{R}^{(1)}} \Delta\hat{\rho}(Y) \cdot \text{rate}_{\mathbf{x}(t)}(\hat{\rho}).$$

Using equation (2.2), we can expand the reaction rate function to obtain

$$\mathbf{x}'(t)(Y) = \sum_{\hat{\rho} \in \widehat{R}^{(1)}} \Delta\hat{\rho}(Y) \cdot k(\hat{\rho}) \prod_{Z \in S} \mathbf{x}(t)(Z)^{\mathbf{r}(\hat{\rho})(Z)}.$$

Since $\hat{\rho}$ is identical to ρ with an additional catalyst X, we can rewrite this ODE as

$$\mathbf{x}'(t)(Y) = \mathbf{x}(t)(X) \sum_{\rho \in R^{(1)}} \Delta\rho(Y) \cdot k(\rho) \prod_{Z \in S^{(1)}} \mathbf{x}(t)(Z)^{\mathbf{r}(\rho)(Z)}. \tag{3.5}$$

Now equation (3.5) has identical structure to that of equation (3.4) except the functions $\hat{y}(t)$ are replaced with $\mathbf{x}(t)(Y)$ for each $Y \in S^{(1)}$. Therefore $\hat{y}(t) = \mathbf{x}(t)(Y)$ if the initial conditions are identical.

Since $f(0) = 0$, it is clear that $\hat{y}(0) = \mathbf{x}^{(1)}(0)(Y) = \mathbf{x}(0)(Y)$ for each $Y \in S^{(1)}$. Therefore $\mathbf{x}(t)(Y) = \hat{y}(t) = \mathbf{x}^{(1)}(f(t))(Y)$ for all $Y \in S^{(1)}$ and $t \in [0, \infty)$. This shows that $\mathbf{x}_U = H(\mathbf{u}, x)$, and therefore **CCRN** is closed under operator H. □

The operation defined in Theorem 3.7 is not an extension operator (because it requires modifying the original CRN). Nevertheless, it is one of the most useful and robust operations on concentration signals.

At this point, the signal operations we have shown in Lemma 3.6 and Theorem 3.7 are well behaved. By "well behaved" we mean that these operators do not have any unintended side effects and operate as expected on all initial states. We will now start proving closure properties that depend on reactions that are not well behaved on all possible inputs. An example of such a reaction is demonstrated in the following example.

Example 3.8. If $N = (S, R)$ is a CRN with $S = \{X\}$ and R consists of the single reaction $2X \xrightarrow{1/2} 3X$, then every solution $\mathbf{x} \in \mathbf{sol}(N)$ has the form

$$x(t) = \frac{x(0)}{1 - x(0) \cdot t}. \tag{3.6}$$

Moreover, $\mathrm{dom}(\mathbf{x}) = [0, \frac{1}{x(0)})$ if $x(0) > 0$ and $\mathrm{dom}(\mathbf{x}) = [0, \infty)$ if $x(0) = 0$.

Proof. Assume the hypothesis. Then the ODE for X is $x'(t) = \frac{x(t)^2}{2}$. By separation of variables and integration, it is clear that the solution is $x(t) = \frac{x(0)}{1-x(0)\cdot t}$. This also implies that $x(t) = 0$ for all $t \in [0, \infty)$ if $x(0) = 0$. Similarly, if $x(0) > 0$, it is clear that $x(t)$ is undefined at time $t = \frac{1}{x(0)}$. □

Even though CRNs such as these are unquestionably unrealistic, we will continue exploring which concentration signals the CRN model can compute using these extensions. In chapters 6 and 7, we will explore in more detail what CRNs can compute *robustly*.

We will now prove a theorem that is paramount to proving the remaining closure properties of CRN-computable signals. The theorem simply shows that it is possible compute the *reciprocal* of a one-dimensional concentration signal and is intentionally stated in a way to be easily referenced in the forthcoming proofs.

Theorem 3.9 (reciprocal theorem)**.** If $N = (S, R)$ is a CRN with $X, \overline{X} \in S$ such that

$$F_{\overline{X}}(\mathbf{x}) = -\mathbf{x}(\overline{X})^2 \cdot F_X(\mathbf{x}) \tag{3.7}$$

for all $\mathbf{x} \in [0, \infty)^S$ where $F_{\overline{X}}$ and F_X are the deterministic mass action rate functions of $\overline{X}$ and X in N, respectively, and if $\mathbf{x} \in \mathbf{sol}(N)$ such that $\overline{x}(0) = \frac{1}{x(0)}$, then $\overline{x}(t) = \frac{1}{x(t)}$ for all $t \in [0, \infty)$.

Proof. Assume the hypothesis. By equation (3.7), the ODEs of $\overline{X}$ and X are related by

$$\frac{d\overline{x}}{dt} = -\overline{x}^2 \frac{dx}{dt}.$$

By separation of variables and integration, we obtain

$$-\int \frac{1}{\overline{x}^2} d\overline{x} = \int dx + C$$

for some $C \in \mathbb{R}$. By evaluating the integrals, we see that

$$\frac{1}{\overline{x}} = x + C.$$

Since $\overline{x}(0) = \frac{1}{x(0)}$, it follows that $C = 0$, and therefore $\overline{x}(t) = \frac{1}{x(t)}$ for all $t \in [0, \infty)$. □

Corollary 3.10. The concentration signal operator $H : C[\{X\}] \to C[\{\overline{X}\}]$ defined by

$$H(\mathbf{u})(t)(\overline{X}) = \frac{1}{\mathbf{u}(t)(X)} \tag{3.8}$$

for all $\mathbf{u} \in C[\{X\}]$ such that $\mathbf{u}(0)(X) > 0$, is an extension operator.

Two things are worth noting about the above theorem and corollary. First, if $x(0) = 0$, it is not possible to compute the reciprocal of X because $\frac{1}{x(0)}$ is not defined. Second, the extension described in Corollary 3.10 is not always well behaved on initial states that do not satisfy $\overline{x}(0) = \frac{1}{x(0)}$. We demonstrate this with the following example.

Example 3.11. Let $N = (S, R)$ be a CRN where $S = \{X, \overline{X}\}$ and R consists of the reactions $X \xrightarrow{k} \emptyset$ and $2\overline{X} + X \xrightarrow{k} 3\overline{X} + X$ where $k > 0$. Then the deterministic mass action functions of X and $\overline{X}$ are related by

$$F_{\overline{X}}(\mathbf{x}) = k \cdot \mathbf{x}(\overline{X})^2 \cdot \mathbf{x}(X) = -\mathbf{x}(\overline{X})^2 \cdot F_X(\mathbf{x})$$

for all $\mathbf{x} \in [0, \infty)^S$.

By Theorem 3.9, for all $\mathbf{x} \in \mathbf{sol}(N)$ with $\overline{x}(0) = \frac{1}{x(0)}$, the concentration of $\overline{X}$ will be the reciprocal of X for all $t \in [0, \infty)$. However, if $x(0) > 1$ and $\overline{x}(0) = \frac{1}{x(0)-1}$, then $\overline{x}(t) = \frac{1}{x(t)-1}$ for all $t \in [0, 1)$, and $\overline{x}(t)$ will be undefined at time $t = 1$.

The above example shows that the reciprocal of a perfectly harmless reaction $X \xrightarrow{k} \emptyset$ will blow up in constant time if initialized improperly. In fact, in this example, it will blow up whenever $\overline{x}(0) > \frac{1}{x(0)}$.

The rest of the extension operators we show in this chapter depend on the reciprocal theorem in some way. Therefore, these extensions depend on precise initial conditions and are not well-behaved if initialized incorrectly. These new extensions are more general than those of Lemma 3.6 and Theorem 3.7, and to explore their full generality, we define a notion of a *weakly* CRN-computable signal. A weakly CRN-computable signal is simply a function that can be written as a polynomial of CRN-computable concentration signals.

An important feature of weakly CRN-computable signals is that they can be negative. Our previous definitions do not support negative concentration signals, therefore we extend those definitions in the following way.

Definition 3.12. A *signal* is a continuous function $\mathbf{w} : [0, \infty) \to \mathbb{R}^W$ where $W \subseteq \mathbf{S}$ is a finite set of species. The set of all continuous signals over the set W is denoted $C_{\mathbb{R}}[W]$.

We have been calling functions $\mathbf{u} \in C[U]$ concentration signals. Our reuse of the terminology *signal* to refer to functions $\mathbf{w} \in C_{\mathbb{R}}[W]$ is a natural and intuitive generalization of the original term.

Definition 3.13. A function $P : [0, \infty)^U \to \mathbb{R}$ where $U \subseteq \mathbf{S}$ is a finite set a species is called a *polynomial map* if there are constants $k \in \mathbb{N}$, $c_0, \ldots, c_{k-1} \in \mathbb{R}$, and $n_{i,Y} \in \mathbb{N}$ for each $0 \leq i < k$ and $Y \in U$ such that

$$P(\mathbf{x}) = \sum_{i=0}^{k-1} \left(c_i \prod_{Y \in U} \mathbf{x}(Y)^{n_{i,Y}} \right) \tag{3.9}$$

for all $\mathbf{x} \in [0, \infty)^U$.

Definition 3.14. A signal $\mathbf{w} : C_{\mathbb{R}}[W]$ is called *weakly CRN-computable* if there exists a CRN $N = (S, R)$, a polynomial map $P_Y : [0, \infty)^S \to \mathbb{R}$ for each $Y \in W$, and a solution

$\mathbf{x} \in \mathbf{sol}(N)$ such that $\mathbf{w}(t)(Y) = P_Y(\mathbf{x}(t))$ for all $t \in [0, \infty)$ and for all $Y \in W$. The set of all weakly CRN-computable signals is denoted **WCRN**.

Observation 3.15. Every CRN-computable signal is weakly CRN-computable.

Since weakly CRN-computable signals are allowed to be negative, it is easier to prove closure properties over the set **WCRN**. However, our primary interest is what CRNs are capable of computing in the stronger sense. We will now prove that if a weakly CRN-computable signal is strictly positive at time zero, then it is also CRN-computable. In order to prove this result, we first need to prove a few closure properties of **WCRN** and introduce a new method of extending CRNs.

Recall that an extension operator makes it possible to extend a CRN to compute a concentration signal in terms of another concentration signal. We now show that it is possible construct specific deterministic mass action rate functions with simple extensions.

Definition 3.16. A function of the form $G_X : [0, \infty)^{U \cup \{X\}} \to \mathbb{R}$ where $X \in \mathbf{S}$ and $U \subseteq \mathbf{S}$ is finite is called a *rate function*.

A rate function G_X is called *valid* if for all CRNs $N = (S, R)$ with $U \subseteq S$ and $X \notin S$, there exists a CRN $\widehat{N} = (\widehat{S}, \widehat{R})$ such that $N \trianglelefteq \widehat{N}$, $\widehat{S} = S \cup \{X\}$, and $\widehat{F}_X(\hat{\mathbf{x}}) = G_X(\hat{\mathbf{x}}_{U \cup \{X\}})$ for all $\hat{\mathbf{x}} \in [0, \infty)^{\widehat{S}}$ where $\widehat{F}_X$ is the deterministic mass action function of X in $\widehat{N}$.

A rate function $P_X : [0, \infty)^{U \cup \{X\}} \to \mathbb{R}$ is *polynomial* if it is a polynomial map.

Lemma 3.17. If $P_X : [0, \infty)^{U \cup \{X\}} \to \mathbb{R}$ and $Q_X : [0, \infty)^{U \cup \{X\}} \to \mathbb{R}$ are polynomial rate functions with strictly positive coefficients and $G_X : [0, \infty)^{U \cup \{X\}} \to \mathbb{R}$ is a rate function defined by

$$G_X(\mathbf{x}) = P_X(\mathbf{x}) - \mathbf{x}(X) \cdot Q_X(\mathbf{x}) \tag{3.10}$$

for all $\mathbf{x} \in [0, \infty)^{U \cup \{X\}}$, then G_X is valid.

Proof. Assume the hypothesis, and let $N = (S, R)$ be a CRN such that $U \subseteq S$ and $X \notin S$. Since P_X and Q_X are polynomial rate functions, G_X can be written as

$$G_X(\mathbf{x}) = \sum_{i=0}^{k-1} \left(c_i \prod_{Y \in U \cup \{X\}} \mathbf{x}(Y)^{n_{i,Y}} \right) - \mathbf{x}(X) \sum_{i=0}^{l-1} \left(d_i \prod_{Y \in U \cup \{X\}} \mathbf{x}(Y)^{m_{i,Y}} \right)$$

for all $\mathbf{x} \in [0, \infty)^{U \cup \{X\}}$where $k, l \in \mathbb{N}$, $c_0, \ldots, c_{k-1} \in (0, 1)$, $d_0, \ldots, d_{l-1} \in (0, 1)$, $n_{i,Y} \in \mathbb{N}$ for all $0 \leq i < k$ and for all $Y \in U \cup \{X\}$, and $m_{i,Y} \in \mathbb{N}$ for all $0 \leq i < l$ and for all $Y \in U \cup \{X\}$.

Now let $\widehat{N} = (\widehat{S}, \widehat{R})$ be a CRN such that $\widehat{S} = S$ and $\widehat{R} = R \cup \widehat{R}^+ \cup \widehat{R}^-$ where $\widehat{R}^+ = \{\alpha_0, \ldots, \alpha_{k-1}\}$ and $\widehat{R}^- = \{\beta_0, \ldots, \beta_{l-1}\}$, and for each $0 \leq i < k$, the reaction α_i is defined by $k(\alpha_i) = c_i$ and

$$\begin{aligned}
&\mathbf{r}(\alpha_i)(X) = \mathbf{p}(\alpha_i)(X) - 1 = n_{i,X}, \\
&\mathbf{r}(\alpha_i)(Y) = \mathbf{p}(\alpha_i)(Y) = n_{i,Y} \text{ for all } Y \in U, \\
&\mathbf{r}(\alpha_i)(Z) = \mathbf{p}(\alpha_i)(Z) = 0 \text{ for all } Z \in \widehat{S} \setminus (U \cup \{X\}),
\end{aligned}$$

and for each $0 \leq j < l$, the reaction β_j is defined by $k(\beta_j) = d_j$ and

$$\begin{aligned}
&\mathbf{r}(\beta_j)(X) = \mathbf{p}(\beta_j)(X) + 1 = m_{j,X} + 1, \\
&\mathbf{r}(\beta_j)(Y) = \mathbf{p}(\beta_j)(Y) = m_{j,Y} \text{ for all } Y \in U, \\
&\mathbf{r}(\beta_j)(Z) = \mathbf{p}(\beta_j)(Z) = 0 \text{ for all } Z \in \widehat{S} \setminus (U \cup \{X\}).
\end{aligned}$$

Intuitively, each reaction α_i corresponds exactly to the ith term of P_X, and each reaction β_j corresponds to the jth term of Q_X (and also takes into account the extra $\mathbf{x}(X)$). Since each of the reactions of $\widehat{R}^+$ and $\widehat{R}^-$ only use species of S as catalysts, it is clear that $N \trianglelefteq \widehat{N}$. It is also easy to see that the deterministic mass action function for X in $\widehat{N}$ is

$$\widehat{F}_X(\hat{\mathbf{x}}) = P_X(\hat{\mathbf{x}}_{U \cup \{X\}}) - \hat{\mathbf{x}}(X) \cdot Q_X(\hat{\mathbf{x}}_{U \cup \{X\}}) = G_X(\hat{\mathbf{x}}_{U \cup \{X\}})$$

for all $\hat{\mathbf{x}} \in [0, \infty)^{\widehat{S}}$. □

Lemma 3.17 shows it is possible to build a CRN extension such that a new species X has a polynomial ODE like that of equation (3.10).

We now prove a few simple closure properties of **WCRN** that will also help us in the proof of the main theorem.

Definition 3.18. A *signal operator* is a function of the form $H : C_{\mathbb{R}}[U] \to C_{\mathbb{R}}[V]$ where $U, V \subseteq \mathbf{S}$ are finite and $U \cap V = \emptyset$. We say that the set **WCRN** of weakly CRN-computable signals is *closed under operator* H if for all $\mathbf{u} \in C_{\mathbb{R}}[U]$)

$$\mathbf{u} \in \mathbf{WCRN} \implies H(\mathbf{u}) \in \mathbf{WCRN}.$$

Observation 3.19. The set **WCRN** is closed under addition and multiplication. More formally, it is closed under the operators $H_1, H_2 : C_{\mathbb{R}}[\{X, Y\}] \to C_{\mathbb{R}}[\{Z\}]$ defined by

$$H_1(\mathbf{u})(t)(Z) = \mathbf{u}(t)(X) + \mathbf{u}(t)(Y)$$
$$H_2(\mathbf{u})(t)(Z) = \mathbf{u}(t)(X) \cdot \mathbf{u}(t)(Y)$$

for all $\mathbf{u} \in C_{\mathbb{R}}[\{X, Y\}]$ and $t \in [0, \infty)$.

Proof. This follows from the fact that adding or multiplying two polynomials produces another polynomial. □

Lemma 3.20. The set **WCRN** is closed under differentiation. More formally, it is closed under the operator $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{\widehat{X}\}]$ defined by

$$H(t)(\mathbf{u})(\widehat{X}) = \mathbf{u}'(t)(X)$$

for all $\mathbf{u} \in C_{\mathbb{R}}[\{X\}]$ and $t \in [0, \infty)$.

Proof. Since deterministic mass action rate functions are polynomials, the derivative of the concentration of each species is a polynomial. This implies that the derivative of a polynomial map is also a polynomial. □

The closure properties from Observation 3.19 and Lemma 3.20 are rather trivial because weakly CRN-computable signals are polynomials. In fact, these closure properties do not even require an extension because a single CRN can weakly compute any polynomial combination of its species concentrations.

We now prove the theorem that weakly CRN-computable signals can also be computable if they are strictly positive initially.

Theorem 3.21. If $\mathbf{w} \in C_{\mathbb{R}}[W]$ is weakly CRN-computable, $b \in (0, \infty]$, and $\mathbf{w}(t)(Y) > 0$ for all $t \in [0, b)$ and $Y \in W$, then $\mathbf{w}$ is CRN-computable and is defined for all $t \in [0, b)$.

Proof. Assume the hypothesis. Let $N = (S, R)$ be a CRN that weakly computes $\mathbf{w}$ where $S \cap W = \emptyset$, and let the polynomial maps $P_Y : [0, \infty)^S \to \mathbb{R}$ for each $Y \in W$ be the witnesses that N weakly computes $\mathbf{w}$. Since the derivative of a polynomial map is also a polynomial map, let $\widehat{P}_Y : [0, \infty)^S \to \mathbb{R}$ be the derivative of P_Y for each $Y \in W$.

By Lemma 3.17, it is possible to extend the CRN $N = (S, R)$ to a CRN $\widehat{N} = (\widehat{S}, \widehat{R})$ such that $N \trianglelefteq \widehat{N}$, $\widehat{S} = S \cup \{Y, \overline{Y} \mid Y \in W\}$ and for each $Y \in W$ the deterministic mass action functions $\widehat{F}_Y$ and $\widehat{F}_{\overline{Y}}$ in $\widehat{N}$ are defined by

$$\widehat{F}_Y(\hat{\mathbf{x}}) = \hat{\mathbf{x}}(Y)\hat{\mathbf{x}}(\overline{Y})\widehat{P}_Y(\hat{\mathbf{x}}_S) \tag{3.11}$$

$$\widehat{F}_{\overline{Y}}(\hat{\mathbf{x}}) = -\hat{\mathbf{x}}(\overline{Y})^2\widehat{F}_Y(\hat{\mathbf{x}}). \tag{3.12}$$

This is possible because the rate functions from equations (3.11) and (3.12) are of the same form of equation (3.10) and therefore valid.

Equation (3.12) along with Theorem 3.9 tell us that for each $Y \in W$, it is possible for $\overline{Y}$ to be exactly the reciprocal of Y. By the hypothesis, we know that $\mathbf{w}(t)(Y) > 0$ for all $t \in [0, b)$ and $Y \in W$, therefore the reciprocal of $\mathbf{w}(0)(Y)$ is well-defined.

Let $\mathbf{x} \in \mathbf{sol}(N)$ be a solution that weakly computes $\mathbf{w}$, i.e., $\mathbf{w}(t)(Y) = P_Y(\mathbf{x}(t))$ for all $t \in [0, \infty)$ and $Y \in W$. Now let $\hat{\mathbf{x}} \in \mathbf{sol}(\widehat{N})$ be the solution defined by $\hat{\mathbf{x}}_S = \mathbf{x}$ and $\hat{\mathbf{x}}(0)(Y) = \frac{1}{\hat{\mathbf{x}}(0)(\overline{Y})} = \mathbf{w}(0)(Y)$ for each $Y \in W$. Then by the reciprocal theorem,

$\hat{\mathbf{x}}(t)(Y) = \frac{1}{\hat{\mathbf{x}}(t)(\overline{Y})}$ for all $t \in [0, \infty)$ and $Y \in W$. This means that every species $Y \in W$ in $\widehat{N}$ obeys the ODE

$$\hat{\mathbf{x}}'(t)(Y) = \widehat{P}_Y(\hat{\mathbf{x}}_S(t)).$$

Since $\widehat{P}_Y$ is the derivative of P_Y and $\hat{\mathbf{x}}(0)(Y) = \mathbf{w}(0)(Y)$ for each $Y \in W$, it is clear that $\hat{\mathbf{x}}(t)(Y) = \mathbf{w}(t)(Y)$ for all $t \in [0, b)$ because $\mathbf{w}(t)(Y) > 0$ for all $t \in [0, b)$. Thus $\hat{\mathbf{x}}_W(t) = \mathbf{w}(t)$ for all $t \in [0, b)$, and thus $\widehat{N}$ computes $\mathbf{w}$ over the domain $[0, b)$. □

The above theorem shows that all the closure properties we prove on the set **WCRN** immediately applies to the set **CCRN**. This is a surprising result since any weakly CRN-computable signal that is initially strictly positive but eventually becomes negative is still CRN-computable. It is important to notice that if the weakly CRN-computable signal has a component Y that becomes zero at time b, then the CRN constructed in Theorem 3.21 that computes it has a species $\overline{Y}$ that is undefined at time b. Therefore, although the signal is computable, it is only defined over the interval $[0, b)$.

The rest of this section is devoted to proving more sophisticated closure properties on the set **WCRN**. We first prove that **WCRN** is closed under integration. However, in order to prove this, we need a generalized version of Lemma 3.6 which is shown below.

Observation 3.22. The concentration signal operator $H : C[U] \to C[\{X\}]$ defined by

$$H(\mathbf{u})(t)(X) = k \int_0^t \prod_{Y \in U} \mathbf{u}(s)(Y)^{n_Y} ds \tag{3.13}$$

where $k \in (0, \infty)$ and $n_Y \in \mathbb{N}$ for each $Y \in U$, is an extension operator.

Proof. This proof is identical to that of Lemma 3.6 except the new reaction ρ changes. In particular, ρ is defined by $k(\rho) = k$, $\mathbf{r}(\rho)(Y) = \mathbf{p}(\rho)(Y) = n_Y$ for each $Y \in U$, and $\mathbf{p}(\rho)(X) = \mathbf{r}(\rho)(X) + 1 = 1$. □

Lemma 3.23. The set **WCRN** is closed under integration, i.e., for all $C \in \mathbb{R}$, it is closed under the operator $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{\widehat{X}\}]$ defined by

$$H(t)(\mathbf{u})(\widehat{X}) = \int \mathbf{u}(t)(X)dt + C$$

for all $\mathbf{u} \in C_{\mathbb{R}}[\{X\}]$ and $t \in [0, \infty)$.

Proof. Let H be as given for some $C \in \mathbb{R}$, and let $\mathbf{u} \in C_{\mathbb{R}}[\{X\}]$ be a weakly CRN-computable signal. Then there is a CRN $N = (S, R)$, a polynomial map $P_X : [0, \infty)^S \to \mathbb{R}$, and a solution $\mathbf{x} \in \mathbf{sol}(N)$ such that $\mathbf{u}(t)(X) = P_X(\mathbf{x}(t))$ for all $t \in [0, \infty)$. Since P_X is a polynomial map, there exist constants $k \in \mathbb{N}$, $c_0, \ldots, c_{k-1} \in \mathbb{R}$, and $n_{i,Y} \in \mathbb{N}$ for each $0 \leq i < k$ and $Y \in S$ such that

$$P_X(\mathbf{x}(t)) = \sum_{i=0}^{k-1} \left(c_i \prod_{Y \in S} \mathbf{x}(t)(Y)^{n_{i,Y}} \right)$$

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

Observation 3.22 tells us that the integral of every term of P_X is CRN-computable via an extension. Let $\widehat{N} = (\widehat{S}, \widehat{R})$ be a CRN such that $N \trianglelefteq \widehat{N}$ with additional species $T_0, \ldots, T_{k-1} \in \widehat{S}$ where each T_i is added to $\widehat{N}$ according to Observation 3.22 to compute the integral of $\prod_{Y \in S} \mathbf{x}(t)(Y)^{n_{i,Y}}$. Then there exists a solution $\hat{\mathbf{x}} \in \mathbf{sol}(\widehat{N})$ such that $\hat{\mathbf{x}}_S = \mathbf{x}$ and

$$\hat{\mathbf{x}}(t)(T_i) = \int_0^t \prod_{Y \in S} \mathbf{x}(s)(Y)^{n_{i,Y}} ds$$

for all $0 \leq i < k$ and $t \in [0, \infty)$. We note that for each $0 \leq i < k$ we can rewrite this integral as

$$\hat{\mathbf{x}}(t)(T_i) = \int \prod_{Y \in S} \mathbf{x}(t)(Y)^{n_{i,Y}} dt + C_i$$

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

We now define the polynomial map $P_{\widehat{X}} : [0,\infty)^{\widehat{S}} \to \mathbb{R}$ by

$$P_{\widehat{X}}(\tilde{\mathbf{x}}) = \sum_{i=0}^{k-1} c_i \tilde{\mathbf{x}}(T_i) + C^*$$

for all $\tilde{\mathbf{x}} \in [0,\infty)^{\widehat{S}}$ where the constant $C^* \in \mathbb{R}$ is defined by $C^* = C - \sum_{i=0}^{k-1} c_i \cdot C_i$. It follows that for all $t \in [0,\infty)$,

$$P_{\widehat{X}}(\hat{\mathbf{x}}(t)) = \sum_{i=0}^{k-1} \int c_i \prod_{Y \in S} \hat{\mathbf{x}}(t)(Y)^{n_{i,Y}} dt + C$$

and therefore $H(\mathbf{u})(t)(\widehat{X}) = P_{\widehat{X}}(\hat{\mathbf{x}}(t))$ for all $t \in [0,\infty)$. Thus the CRN $\widehat{N}$ and solution $\hat{\mathbf{x}}$ weakly CRN-compute the signal $H(\mathbf{u})$. □

The rest of the closure properties we prove follow a similar pattern, therefore we grouped them in the following theorem. We also include the closure properties we have already proven so as to have a single place to look up the closure properties.

Theorem 3.24. The set **WCRN** of weakly CRN-computable signals is closed under the following operators.

Addition. $H : C_{\mathbb{R}}[\{X,Y\}] \to C_{\mathbb{R}}[\{Z\}]$ defined by

$$H(\mathbf{w})(t)(Z) = \mathbf{w}(t)(X) + \mathbf{w}(t)(Y). \tag{3.14}$$

Multiplication. $H : C_{\mathbb{R}}[\{X,Y\}] \to C_{\mathbb{R}}[\{Z\}]$ defined by

$$H(\mathbf{w})(t)(Z) = \mathbf{w}(t)(X) \cdot \mathbf{w}(t)(Y). \tag{3.15}$$

Differentiation. $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{Y\}]$ defined by

$$H(\mathbf{w})(t)(Y) = \mathbf{w}'(t)(X). \tag{3.16}$$

Integration. $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{Y\}]$ defined by

$$H(\mathbf{w})(t)(Y) = \int \mathbf{w}(t)(X)dt + C \tag{3.17}$$

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

Reciprocal. $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{Y\}]$ defined by

$$H(\mathbf{w})(t)(Y) = \frac{1}{\mathbf{w}(t)(X)} \tag{3.18}$$

where $\mathbf{w}(0)(X) \neq 0$.

Exponentiation. $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{Y\}]$ defined by

$$H(\mathbf{w})(t)(Y) = e^{\mathbf{w}(t)(X)}. \tag{3.19}$$

Logarithm. $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{Y\}]$ defined by

$$H(\mathbf{w})(t)(Y) = \log\big(\mathbf{w}(t)(X)\big) \tag{3.20}$$

where $\mathbf{w}(0)(X) > 0$.

Power. $H : C_{\mathbb{R}}[\{X, Y\}] \to C_{\mathbb{R}}[\{Z\}]$ defined by

$$H(\mathbf{w})(t)(Z) = \mathbf{w}(t)(X)^{\mathbf{w}(t)(Y)} \tag{3.21}$$

where $\mathbf{w}(0)(X) \neq 0$ or $\mathbf{w}(0)(Y) \geq 0$.

Sine. $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{Y\}]$ defined by

$$H(\mathbf{w})(t)(Y) = \sin\big(\mathbf{w}(t)(X)\big). \tag{3.22}$$

Arctangent. $H : C_{\mathbb{R}}[\{X\}] \to C_{\mathbb{R}}[\{Y\}]$ defined by

$$H(\mathbf{w})(t)(Y) = \arctan\big(\mathbf{w}(t)(X)\big). \tag{3.23}$$

Proof. The first four closure properties have already been shown in Observation 3.19, Lemma 3.20, and Lemma 3.23; closure under reciprocals is obvious from Theorems 3.9 and 3.21; and closure under logarithms and arctangent are natural consequences of closure under addition, multiplication, reciprocals, and integration. The rest of the closure properties require a little more work, and each property depends on the reciprocal theorem in some way.

To show that **WCRN** is closed under exponentiation, let H be defined as given in equation (3.19), and let $\mathbf{w} \in C_{\mathbb{R}}[\{X\}]$ be weakly CRN-computable by the CRN $N = (S, R)$ and polynomial map $P_X : [0,\infty)^S \to \mathbb{R}$. Now let $\widehat{P}_X : [0,\infty)^S \to \mathbb{R}$ be the derivative of P_X defined by

$$\widehat{P}_X(\mathbf{x}(t)) = \frac{d}{dt} P_X(\mathbf{x}(t))$$

for all $\mathbf{x} \in C[S]$ and $t \in [0,\infty)$. As we showed earlier, $\widehat{P}_X$ is also a polynomial map.

Let $\widehat{N} = (\widehat{S}, \widehat{R})$ be a CRN such that $N \trianglelefteq \widehat{N}$, $\widehat{S} = S \cup \{Y, \overline{Y}\}$, and the deterministic mass action rate functions $\widehat{F}_Y$ and $\widehat{F}_{\overline{Y}}$ in $\widehat{N}$ are defined by

$$\widehat{F}_Y(\hat{\mathbf{x}}) = \hat{\mathbf{x}}(Y)^2 \hat{\mathbf{x}}(\overline{Y}) \widehat{P}(\hat{\mathbf{x}}_S)$$
$$\widehat{F}_{\overline{Y}}(\hat{\mathbf{x}}) = -\hat{\mathbf{x}}(\overline{Y})^2 \widehat{F}(\hat{\mathbf{x}}).$$

This construction is possible because $\widehat{F}_Y$ and $\widehat{F}_{\overline{Y}}$ are valid rate functions.

Now let $\mathbf{x} \in \mathbf{sol}(N)$ be a solution that computes $\mathbf{w}$, and let $\hat{\mathbf{x}} \in \mathbf{sol}(\widehat{N})$ such that $\hat{\mathbf{x}}_S = \mathbf{x}$ and $\hat{\mathbf{x}}(0)(Y) = \frac{1}{\hat{\mathbf{x}}(0)(\overline{Y})} = e^{\mathbf{w}(0)(X)}$. Then by the reciprocal theorem, $\hat{\mathbf{x}}(t)(\overline{Y}) =$

$\frac{1}{\hat{\mathbf{x}}(t)(Y)}$ for all $t \in [0, \infty)$, therefore the derivative of $\hat{\mathbf{x}}(t)(Y)$ is

$$\hat{\mathbf{x}}'(t)(Y) = \hat{\mathbf{x}}(t)(Y)\widehat{P}_X(\hat{\mathbf{x}}_S(t)) = \hat{\mathbf{x}}(t)(Y)\mathbf{w}'(t)(X).$$

By separation of variables and integration, we obtain

$$\log(\hat{\mathbf{x}}(t)(Y)) = \mathbf{w}(t)(X) + C$$

for some $C \in \mathbb{R}$. Therefore $\hat{\mathbf{x}}(t)(Y) = e^{\mathbf{w}(t)(X)+C}$, and by the initial condition, we see that $C = 0$. This shows that $\hat{\mathbf{x}}(t)(Y) = H(\mathbf{w})(t)(Y)$, and concludes the proof that **WCRN** is closed under exponentiation.

Closure under powers is a consequence of closure under exponentiation, multiplication, and logarithms. This can bee seen from the equality $f(t)^{g(t)} = e^{g(t)\log(f(t))}$ for any two continuous functions f and g.

It remains to be shown that **WCRN** is closed under the sine operator. Let H be defined as given in equation (3.24), let $\mathbf{w} \in C_{\mathbb{R}}[\{X\}]$ be weakly CRN-computable by the CRN $N = (S, R)$ and polynomial map $P_X : [0, \infty)^S \to \mathbb{R}$, and let $\widehat{P}_X : [0, \infty)^S \to \mathbb{R}$ be the derivative of P_X as done before.

Let $\widehat{N} = (\widehat{S}, \widehat{R})$ be a CRN such that $N \trianglelefteq \widehat{N}$, $\widehat{S} = S \cup \{Y, \overline{Y}, Z, \overline{Z}\}$, and where the deterministic mass action rate functions $\widehat{F}_Y$, $\widehat{F}_{\overline{Y}}$, $\widehat{F}_Z$, and $\widehat{F}_{\overline{Z}}$ in $\widehat{N}$ are defined by

$$\begin{aligned}
\widehat{F}_Y(\hat{\mathbf{x}}) &= \hat{\mathbf{x}}(Y)\hat{\mathbf{x}}(\overline{Y})\left(\widehat{P}_X(\hat{\mathbf{x}}_S)\big(\hat{\mathbf{x}}(Z) - 2\big)\right) \\
\widehat{F}_{\overline{Y}}(\hat{\mathbf{x}}) &= -\hat{\mathbf{x}}(\overline{Y})^2\widehat{F}_Y(\hat{\mathbf{x}}) \\
\widehat{F}_Z(\hat{\mathbf{x}}) &= \hat{\mathbf{x}}(Z)\hat{\mathbf{x}}(\overline{Z})\left(\widehat{P}_X(\hat{\mathbf{x}}_S)\big(\hat{\mathbf{x}}(Y) - 2\big)\right) \\
\widehat{F}_{\overline{Z}}(\hat{\mathbf{x}}) &= -\hat{\mathbf{x}}(\overline{Z})^2\widehat{F}_Z(\hat{\mathbf{x}})
\end{aligned}$$

This construction is possible because $\widehat{F}_Y$, $\widehat{F}_{\overline{Y}}$, $\widehat{F}_Z$, and $\widehat{F}_{\overline{Z}}$ are valid rate functions.

Now let $\mathbf{x} \in \mathbf{sol}(N)$ be a solution that computes $\mathbf{w}$, and let $\hat{\mathbf{x}} \in \mathbf{sol}(\widehat{N})$ such that $\hat{\mathbf{x}}_S = \mathbf{x}$ and

$$\hat{\mathbf{x}}(0)(Y) = \frac{1}{\hat{\mathbf{x}}(0)(\overline{Y})} = \sin\big(\mathbf{w}(0)(X)\big) + 2$$
$$\hat{\mathbf{x}}(0)(Z) = \frac{1}{\hat{\mathbf{x}}(0)(\overline{Z})} = \cos\big(\mathbf{w}(0)(X)\big) + 2.$$

Then by the reciprocal theorem, $\hat{\mathbf{x}}(t)(\overline{Y}) = \frac{1}{\hat{\mathbf{x}}(t)(Y)}$ and $\hat{\mathbf{x}}(t)(\overline{Z}) = \frac{1}{\hat{\mathbf{x}}(t)(Z)}$ for all $t \in [0, \infty)$. Therefore, the derivatives of $\hat{\mathbf{x}}(t)(Y)$ and $\hat{\mathbf{x}}(t)(Z)$ can be written a more intuitive way as

$$\frac{dy}{dt} = \frac{dx}{dt}(z-2), \text{ and} \qquad \frac{dz}{dt} = -\frac{dx}{dt}(y-2),$$

respectively. These pair of ODEs are easily solvable with solutions

$$\hat{\mathbf{x}}(t)(Y) = \sin\big(\mathbf{w}(t)(X)\big) + 2$$
$$\hat{\mathbf{x}}(t)(Z) = \cos\big(\mathbf{w}(t)(X)\big) + 2$$

for all $t \in [0, \infty)$. Therefore $H(\mathbf{w})(t)(X) = \hat{\mathbf{x}}(t)(Y) - 2$.

Since $\hat{\mathbf{x}}(t)(Y) - 2$ can be written in terms of a polynomial map, we have shown that **WCRN** is indeed closed under the sine operator. □

CHAPTER 4. SIGNAL RESTORATION ALGORTIHMS*

In this chapter we present and analyze two signal restoration algorithms used in CRNs under deterministic mass action semantics. Signal restoration concerns the prevention of memory corruption caused by noise and other factors present in the system. In CRNs, a bit of memory can be stored by the contrast of high and low concentrations of a species (similar to how circuits use high and low voltages). The two CRN signal restoration algorithms investigated here aim to prevent high concentrations from becoming low, and vice versa, even in the presence of adversarial reactions.

The two algorithms are presented in sections 4.1 and 4.2. The first algorithm consists of two termolecular reactions (reactions with three reactants) and is the simpler of the two algorithms [40, 37]. The second signal restoration algorithm consists of four bimolecular reactions and was developed in 2008 by Angluin, Aspnes, and Eisenstat for population protocols [23, 3]. For both algorithms, we introduce new reactions to simulate the presence of noise from other reactions and analyze the behavior and runtime of the algorithms.

4.1 Termolecular Signal Restoration

The termolecular signal restoration algorithm presented here has the structure

$$2X + Y \xrightarrow{1} 3X \tag{4.1}$$

$$2Y + X \xrightarrow{1} 3Y. \tag{4.2}$$

* An earlier version of most of the material in this chapter will appear in [39].

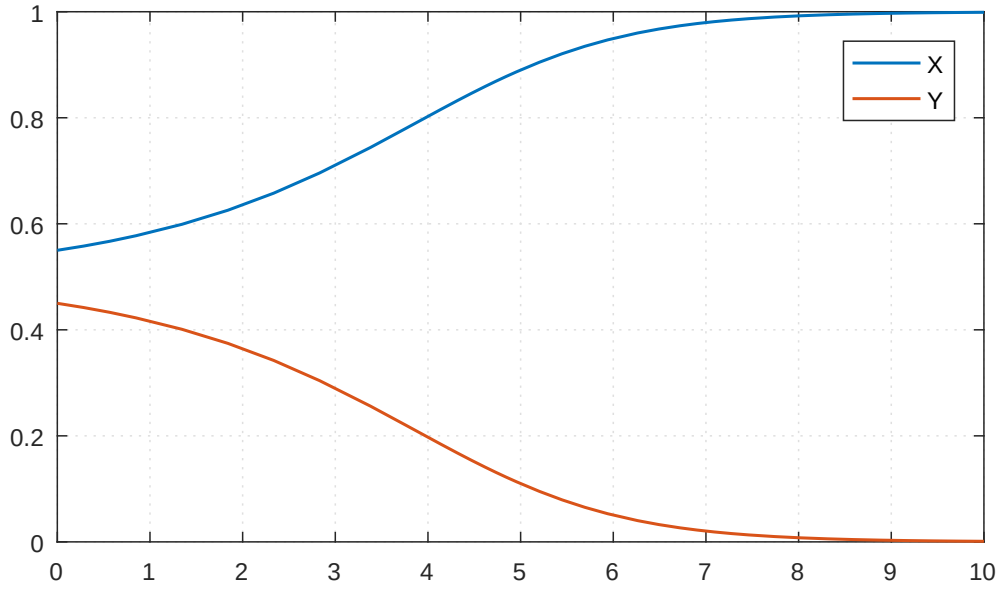


Figure 4.1: Plot of the solution generated by reactions (4.1) and (4.2) and the initial condition $x(0) = 0.55$ and $y(0) = 0.45$

Intuitively, the species X and Y are competing for total majority by converting the opposing species to itself. The algorithm is designed so that when the system is initialized with some concentrations of X and Y, the species with the initial majority will asymptotically annihilate the other. This is demonstrated in figure 4.1.

If $N = (S, R)$ is the CRN consisting of the reactions (4.1) and (4.2) and $\mathbf{x} \in \mathbf{sol}(N)$, then using equation (2.4) we see that the system consists of the two ODEs

$$\mathbf{x}'(t)(X) = \mathbf{x}(t)(X)^2\mathbf{x}(t)(Y) - \mathbf{x}(t)(X)\mathbf{x}(t)(Y)^2 \tag{4.3}$$

$$\mathbf{x}'(t)(Y) = \mathbf{x}(t)(Y)^2\mathbf{x}(t)(X) - \mathbf{x}(t)(Y)\mathbf{x}(t)(X)^2, \tag{4.4}$$

or more simply

$$\frac{dx}{dt} = x^2y - xy^2 \tag{4.5}$$

$$\frac{dy}{dt} = y^2x - yx^2. \tag{4.6}$$

This system is simple enough that it can be solved explicitly. Since $\frac{dx}{dt} + \frac{dy}{dt} = 0$, if $x(0) + y(0) = 1$ then $x(t) + y(t) = 1$ for all $t \in [0, \infty)$. Therefore equation (4.5) can be rewritten

$$\frac{dx}{dt} = x^2(1-x) - x(1-x)^2. \tag{4.7}$$

By separation of variables and integration, it is easy to show that the solution $\mathbf{x}$ can be written analytically as

$$x(t) = \begin{cases} \frac{1}{2}(1-u), & \text{if } x(0) < \frac{1}{2} \\ \frac{1}{2}, & \text{if } x(0) = \frac{1}{2} \\ \frac{1}{2}(1+u), & \text{if } x(0) > \frac{1}{2} \end{cases} \tag{4.8}$$

$$y(t) = 1 - x(t), \tag{4.9}$$

where $u = \frac{1}{\sqrt{1+C \cdot e^{-t}}}$ and $C = \frac{4x(0)y(0)}{(x(0)-y(0))^2}$.

By inspection of (4.8) and (4.9), we can see that the CRN has the behavior we expect. The species with the majority initial concentration eliminates the minority at an exponential rate.

The drawback to this solution is that it depends on the following assumptions.

1. The rate constants of the reactions are *exactly* equal.
2. The initial concentrations sum *exactly* to 1.
3. No other reactions interfere with the system.

In practice, we do not have infinitely precise control over rate constants and initial conditions. Furthermore, the algorithm is most useful in systems with other reactions affecting X and Y. We now specify a more general system that avoids these dependencies.

Construction 4.1. Given the real valued constants $a > 0$, $b > 0$, and $c > 0$, let $N = (S, R)$ be a CRN with $S = \{X, Y\}$ and R consisting of the reactions

$$2X + Y \xrightarrow{a} 3X \tag{4.10}$$

$$2Y + X \xrightarrow{b} 3Y \tag{4.11}$$

$$X \xrightarrow{c} Y. \tag{4.12}$$

This CRN has unrelated rate constants and one additional reaction (4.12) that models interference that biases the system toward Y. Without this extra reaction, our analysis would only be modeling the system in isolation. We do not include a reaction $Y \to X$ because of the symmetry of the species X and Y. Therefore this system captures the worst case scenario for the species X.

Theorem 4.2. If $a > 0$, $b > 0$, and $c > 0$ are real valued constants, $N = (S, R)$ is constructed according to Construction 4.1, $\mathbf{x}_0 \in [0, \infty)^S$ is an initial state of N, and $c < \frac{p^2 a^2}{4(a+b)}$ where $p = \mathbf{x}_0(X) + \mathbf{x}_0(Y)$, then the initial value problem generated by the ICRN $(N, \mathbf{x}_0)$ has exactly three equilibrium points $\hat{\mathbf{x}}_0$, $\hat{\mathbf{x}}_1$, and $\hat{\mathbf{x}}_2$ defined by

$$\begin{aligned} \hat{\mathbf{x}}_0(X) &= 0, & \hat{\mathbf{x}}_0(Y) &= p, \\ \hat{\mathbf{x}}_1(X) &= E_1, & \hat{\mathbf{x}}_1(Y) &= p - E_1, \\ \hat{\mathbf{x}}_2(X) &= E_2, & \hat{\mathbf{x}}_2(Y) &= p - E_2, \end{aligned}$$

where E_1 and E_2 are the real valued constants

$$E_1 = p\left(\frac{b}{a+b}\right) + A, \qquad E_2 = p - A,$$

and where $A = \frac{p}{2}\left(\frac{a}{a+b}\right)\left(1 - \sqrt{1 - c^*}\right)$ and $c^* = c \cdot \frac{4(a+b)}{p^2 a^2}$. Moreover, $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{x}}_2$ are stable, and $\hat{\mathbf{x}}_1$ is unstable.

Proof. Assume the hypothesis. Then the ODEs for the species X and Y can be derived from the reactions (4.10)-(4.12) and are

$$\frac{dx}{dt} = ax^2y - bxy^2 - cx, \tag{4.13}$$

$$\frac{dy}{dt} = by^2x - ayx^2 + cx, \tag{4.14}$$

respectively. Note that $\frac{dx}{dt} + \frac{dy}{dt} = 0$, so the concentrations $x(t)$ and $y(t)$ differ by a constant. Since $p = x(0) + y(0)$, it is easy to show that for all $t \in [0, \infty)$

$$y(t) = p - x(t). \tag{4.15}$$

It immediately follows from (4.13) that

$$\frac{dx}{dt} = ax^2(p - x) - bx(p - x)^2 - cx. \tag{4.16}$$

The equilibrium points of $(N, \mathbf{x}_0)$ can now be derived from the roots of the right-hand side of (4.16). It is routine to verify that these roots are 0, E_1, and E_2, and since $c < \frac{p^2a^2}{4(a+b)}$ these roots are distinct and real. It follows from (4.15) that $\hat{\mathbf{x}}_0$, $\hat{\mathbf{x}}_1$, and $\hat{\mathbf{x}}_2$ are the equilibrium points of $(N, \mathbf{x}_0)$.

We now examine the stability of the points. It is well known that the stability of an equilibrium point can be determined by examining the eigenvalues of the Jacobian matrix evaluated at that point [28, 64]. A point is stable if each eigenvalue has a negative real part, and it is unstable if each eigenvalue has a positive real part.

Since (4.16) is 1-dimensional, there is only a single eigenvalue for each equilibrium point. It is not difficult to show that the eigenvalues for the points $\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2$ are

$$\lambda_0 = (a + b)(-E_1E_2),$$

$$\lambda_1 = (a + b)E_1(E_2 - E_1),$$

$$\lambda_2 = (a + b)E_2(E_1 - E_2),$$

respectively. Since $a > 0$, $b > 0$, and $E_2 > E_1 > 0$, we see that λ_0 and λ_2 are negative and λ_1 is positive. It immediately follows that $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{x}}_2$ are stable and $\hat{\mathbf{x}}_1$ is unstable. □

The above theorem shows that the CRN is indeed bistable if c is sufficiently small. We also see that E_1 is the deciding threshold of the system, i.e., if $x(0) > E_1$ then $x(t)$ converges to E_2, and if $x(0) < E_1$ then $x(t)$ converges to 0.

Observation 4.3. In Theorem 4.2,

$$E_1 < p\left(\frac{b}{a+b}\right) + \frac{2c}{pa}, \text{ and } \quad E_2 > p - \frac{2c}{pa}.$$

Proof. This follows immediately from the fact that

$$A < \frac{p}{2}\left(\frac{a}{a+b}\right)(1-(1-c^*)) = \frac{2c}{pa}.$$ □

Theorem 4.4. Under the assumptions of Theorem 4.2, if x_1, x_2, and t are constants such that $t \geq 0$ and $E_1 < x_1 < x_2 < E_2$, with $x(0) = x_1$ and $x(t) = x_2$, then

$$t = \frac{1}{ap\sqrt{1-c^*}}\left(\frac{1}{E_2}\log u + \frac{1}{E_1}\log v\right), \tag{4.17}$$

where $u = \frac{x_2(E_2-x_1)}{x_1(E_2-x_2)}$ and $v = \frac{x_1(x_2-E_1)}{x_2(x_1-E_1)}$.

Proof. Assume the hypothesis. It is not difficult to show that the ODE (4.16) of $x(t)$ can be rewritten

$$\frac{dx}{dt} = (a+b)x(x-E_1)(E_2-x). \tag{4.18}$$

By separation of variables and integration, we have

$$\int_0^t dt = \frac{1}{a+b}\int_{x_1}^{x_2} \frac{1}{x(x-E_1)(E_2-x)}dx,$$

whence

$$t = \frac{1}{a+b}\left(\frac{1}{E_1E_2(E_2-E_1)}\right)\log\left(u^{E_1}v^{E_2}\right).$$

The theorem immediately follows from the fact that $E_2 - E_1 = \frac{ap}{a+b}\sqrt{1-c^*}$. □

The above theorem gives the exact time required to move from an inital concentration $x(0) > E_1$ to a concentration closer to the equilibrium point E_2. It is useful to note that the only components in equation (4.17) that are not bounded by a constant are $\frac{1}{x_1-E_1}$ and $\frac{1}{E_2-x_2}$. Therefore it is clear that the concentration of X converges to its stable equilibrium in logarithmic time.

Corollary 4.5. Under the hypothesis of Theorem 4.4,

$$t < \frac{a+b}{abp^2(1-c^*)}\log u, \tag{4.19}$$

where $u = \frac{(x_2-E_1)(E_2-x_1)}{(x_1-E_1)(E_2-x_2)}$.

Proof. Since $E_1 < E_2$, Theorem 4.4 tells us that

$$t < \frac{1}{apE_1\sqrt{1-c^*}}\log u. \tag{4.20}$$

Since $c^* \in (0,1)$, we have $1 - c^* < \sqrt{1-c^*}$ and

$$E_1 - \frac{bp}{a+b} = A > \frac{ap}{2(a+b)} > 0,$$

whence

$$\frac{1}{apE_1\sqrt{1-c^*}} < \frac{a+b}{abp^2(1-c^*)}. \tag{4.21}$$

The corollary follows immediately from (4.20) and (4.21). □

4.2 Bimolecular Signal Restoration

The bimolecular signal restoration algorithm we present here was originally given by Angluin et al. in [5] as an approximate majority algorithm for population protocols. In the context of CRNs, the algorithm consists of the four reactions

$$X + Z \xrightarrow{1} 2X \tag{4.22}$$

$$X + Y \xrightarrow{1} X + Z \tag{4.23}$$

$$Y + Z \xrightarrow{1} 2Y \tag{4.24}$$

$$X + Y \xrightarrow{1} Y + Z. \tag{4.25}$$

Again, the species X and Y are opposed to one another and are competing for total majority. The additional species Z is intuitively a *neutral* species that acts as a buffer between X and Y. Angluin et al. proved that this algorithm effeciently computes approximate majority in the context of population protocols, and here we prove that it can be effeciently used for signal restoration in deterministic CRNs. Figure 4.2 demonstrates the behavior of the algorithm.

The CRN $N = (S, R)$ consisting of reactions (4.22)-(4.25) gives the system of ODEs

$$\frac{dx}{dt} = xz - xy \tag{4.26}$$

$$\frac{dy}{dt} = yz - xy \tag{4.27}$$

$$\frac{dz}{dt} = 2xy - xz - xy. \tag{4.28}$$

This system is already considerably more complex than the original termolecular system in section 4.1, and a solution cannot be trivially found. The task becomes especially difficult with arbitrary rate constants and interference from other reactions. However, if the rate constants meet a few constraints, the behavior of this algorithm is extremely similar to

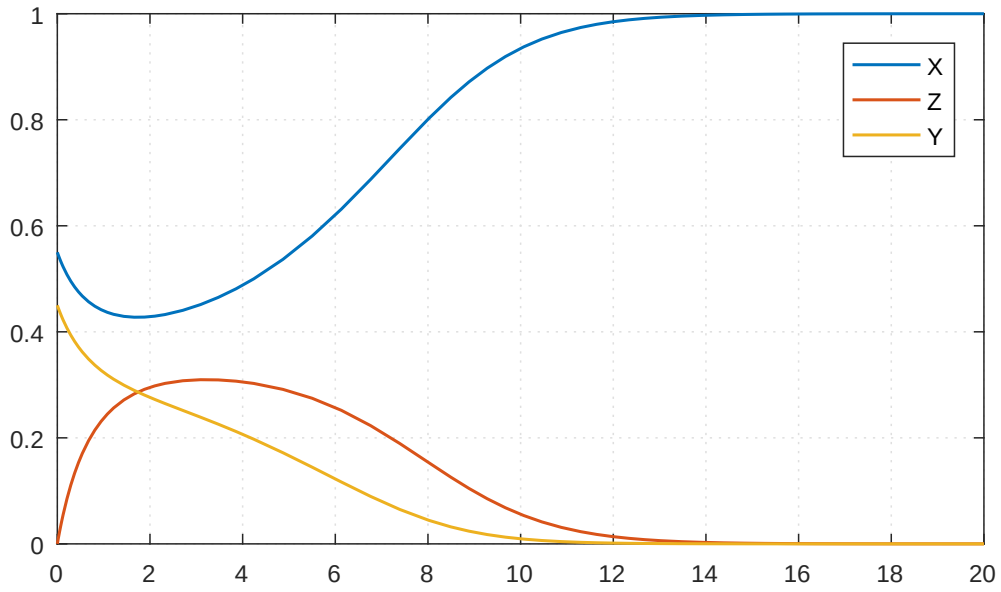


Figure 4.2: Plot of the solution generated by reactions (4.22)-(4.25) and the initial condition $x(0) = 0.55$, $y(0) = 0.45$, and $z(0) = 0$

that of the termolecular algorithm from section 4.1. We leverage this relationship in the fully general proof.

Construction 4.6. Given the strictly positive, real valued constants $k_1, \ldots, k_6$, let $N = (S, R)$ be a CRN with $S = \{X, Y\}$ and R consisting of the reactions

$$X + Z \xrightarrow{k_1} 2X \tag{4.29}$$

$$X + Y \xrightarrow{k_2} X + Z \tag{4.30}$$

$$Y + Z \xrightarrow{k_3} 2Y \tag{4.31}$$

$$X + Y \xrightarrow{k_4} Y + Z \tag{4.32}$$

$$Z \xrightarrow{k_5} Y \tag{4.33}$$

$$X \xrightarrow{k_6} Z. \tag{4.34}$$

This fully generalized CRN contains the new reactions (4.33) and (4.34). These reactions serve the same purpose of representing outside interference from other reactions

in the system. The new reactions still bias the algorithm in favor of Y, but do so in two steps instead of one. Rather than X molecules being converted immediately into Y molecules, they are converted to Z molecules first.

Before we analyze this CRN in its full generality, we assume that the rate constants have the following relationships. For $a > 0$, $b > 0$, and $c > 0$, the rate constants $k_1, \ldots, k_6$ are related by

$$a = k_1 = \frac{k_2}{2} \tag{4.35}$$

$$b = k_3 = \frac{k_4}{2} \tag{4.36}$$

$$c = k_5 = \frac{k_6}{2}. \tag{4.37}$$

If these relationships are satisfied, the system of ODEs for $N = (S, R)$ from Construction 4.6 are

$$\frac{dx}{dt} = axz - 2bxy - 2cx \tag{4.38}$$

$$\frac{dy}{dt} = byz - 2axy + cz \tag{4.39}$$

$$\frac{dz}{dt} = 2axy + 2bxy - axz - byz + 2cx - cz. \tag{4.40}$$

Note that $\frac{dx}{dt} + \frac{dy}{dt} + \frac{dz}{dt} = 0$, so the solutions to $x(t)$, $y(t)$, and $z(t)$ differ only by a constant. If $p = x(0) + y(0) + z(0)$, then it is clear that for all $t \in [0, \infty)$

$$z(t) = p - x(t) - y(t). \tag{4.41}$$

To help us analyze a system of this complexity, we introduce the following four variables. Let $\overline{x}(t)$, $\overline{y}(t)$, $v(t)$, and $w(t)$ be the functions defined by

$$\overline{x}(t) = \sqrt{x(t)}, \tag{4.42}$$

$$\overline{y}(t) = \sqrt{y(t)}, \tag{4.43}$$

$$v(t) = 2\overline{x}(t)\overline{y}(t) - z(t), \tag{4.44}$$

$$w(t) = \overline{x}(t) + \overline{y}(t). \tag{4.45}$$

Lemma 4.7. The ODEs of $\overline{x}$, $\overline{y}$, v, and w are

$$\frac{d\overline{x}}{dt} = \overline{x}\overline{y}\left(a\overline{x} - b\overline{y} - \frac{c}{\overline{y}}\right) - \frac{v}{2}a\overline{x} \tag{4.46}$$

$$\frac{d\overline{y}}{dt} = -\overline{x}\overline{y}\left(a\overline{x} - b\overline{y} - \frac{c}{\overline{y}}\right) - \frac{v}{2}\left(b\overline{y} + \frac{c}{\overline{y}}\right) \tag{4.47}$$

$$\frac{dv}{dt} = -v(\overline{x} + \overline{y})\left(a\overline{x} + b\overline{y} + \frac{c}{\overline{y}}\right) \tag{4.48}$$

$$\frac{dw}{dt} = -\frac{v}{2}\left(a\overline{x} + b\overline{y} + \frac{c}{\overline{y}}\right). \tag{4.49}$$

Proof. Since $\overline{x} = \sqrt{x}$, its derivative is

$$\frac{d\overline{x}}{dt} = \frac{1}{2\overline{x}} \cdot \frac{dx}{dt},$$

and from (4.38) we obtain

$$\frac{d\overline{x}}{dt} = \frac{a}{2}\overline{x}z - b\overline{x}\overline{y}^2 - c\overline{x}.$$

Since $v = 2\overline{x}\overline{y} - z$, it follows that

$$\frac{d\overline{x}}{dt} = \frac{a\overline{x}}{2}(2\overline{x}\overline{y} - v) - b\overline{x}\overline{y}^2 - c\overline{x},$$

and therefore (4.46) holds. By symmetry, (4.47) also holds.

Since $v = 2\overline{x}\overline{y} - z$, its derivative is

$$\frac{dv}{dt} = 2\left(\overline{y}\frac{d\overline{x}}{dt} + \overline{x}\frac{d\overline{y}}{dt}\right) - \frac{dz}{dt}.$$

By substituting (4.40), (4.46), (4.44), and (4.47) into the above equation, we obtain (4.48).

Finally, since $w = \overline{x} + \overline{y}$, its derivative is

$$\frac{dw}{dt} = \frac{d\overline{x}}{dt} + \frac{d\overline{y}}{dt},$$

which clearly simplifies to (4.49) from (4.46) and (4.47). □

Lemma 4.8. If $\alpha = \min\{a, b\}$ and $\hat{v}$ is the function

$$\hat{v}(t) = \frac{p \cdot v(0)}{(p + v(0))e^{\alpha p t} - v(0)}, \tag{4.50}$$

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

$$v(t) \geq \hat{v}(t), \qquad \text{if } v(0) < 0 \tag{4.51}$$

$$v(t) \leq \hat{v}(t), \qquad \text{if } v(0) > 0 \tag{4.52}$$

$$v(t) = 0, \qquad \text{if } v(0) = 0. \tag{4.53}$$

Proof. Assume the hypothesis. By (4.48), we see that if $v(t) = 0$ then $\frac{dv}{dt} = 0$. It follows that (4.53) holds.

If $v(0) < 0$, then

$$\frac{dv}{dt} \geq -v(\overline{x} + \overline{y})(a\overline{x} + b\overline{y}) \geq -v\alpha(\overline{x} + \overline{y})^2$$

Since $p = \overline{x}^2 + \overline{y}^2 + z$ and $v = 2\overline{xy} - z$,

$$(\overline{x} + \overline{y})^2 = \overline{x}^2 + \overline{y}^2 + 2\overline{xy} = p - z + 2\overline{xy} = p + v,$$

and therefore

$$\frac{dv}{dt} \geq -v\alpha(p + v).$$

This ODE can be trivially solved with separation of variables and has solution $\hat{v}(t)$. Therefore $v(t) \geq \hat{v}(t)$ for all $t \in [0, \infty)$, so (4.51) holds.

Similarly, if $v(t) > 0$, then

$$\frac{dv}{dt} \leq -v\alpha(p + v),$$

and so (4.52) holds. □

Corollary 4.9. The limits of $v(t)$ and $w(t)$ are

$$\lim_{t\to\infty} v(t) = 0 \tag{4.54}$$

$$\lim_{t\to\infty} w(t) = \sqrt{p}. \tag{4.55}$$

Theorem 4.10. If $a > 0$, $b > 0$, and $c > 0$ are real valued constants, $N = (S, R)$ is constructed according to Construction 4.6 such that constraints (4.35)-(4.37) are satisfied, $\mathbf{x}_0 \in [0,\infty)^S$ is an initial state, and $c < \frac{pa^2}{4(a+b)}$ where $p = \mathbf{x}_0(X) + \mathbf{x}_0(Y) + \mathbf{x}_0(Z)$, then the IVP generated by $(N, \mathbf{x}_0)$ has the equilibrium points $\hat{\mathbf{x}}_0$, $\hat{\mathbf{x}}_1$, and $\hat{\mathbf{x}}_2$ defined by

$$\begin{aligned}
&\hat{\mathbf{x}}_0(X) = 0, && \hat{\mathbf{x}}_0(Y) = p, && \hat{\mathbf{x}}_0(Z) = 0,\\
&\hat{\mathbf{x}}_1(X) = E_1^2, && \hat{\mathbf{x}}_1(Y) = (\sqrt{p} - E_1)^2, && \hat{\mathbf{x}}_1(Z) = p - \hat{\mathbf{x}}_i(X) - \hat{\mathbf{x}}_i(Y),\\
&\hat{\mathbf{x}}_2(X) = E_2^2, && \hat{\mathbf{x}}_2(Y) = (\sqrt{p} - E_2)^2, && \hat{\mathbf{x}}_2(Z) = p - \hat{\mathbf{x}}_i(X) - \hat{\mathbf{x}}_i(Y),
\end{aligned}$$

where E_1 and E_2 are real valued constants defined by

$$E_1 = \sqrt{p}\left(\frac{b}{a+b}\right) + A, \qquad E_2 = \sqrt{p} - A,$$

and where $A = \frac{\sqrt{p}}{2}\left(\frac{a}{a+b}\right)\left(1 - \sqrt{1-c^*}\right)$, and $c^* = c \cdot \frac{4(a+b)}{pa^2}$. Moreover, $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{x}}_2$ are stable, and $\hat{\mathbf{x}}_1$ is a saddle node.

Proof. Assume the hypothesis. We prove the theorem using the variables $\overline{x}(t)$ and $w(t)$. By Lemma 4.7 and the fact that $v = w^2 - p$, the ODEs for $\overline{x}$ and w can be written as

$$\frac{d\overline{x}}{dt} = a\overline{x}^2(w - \overline{x}) - b\overline{x}(w - \overline{x})^2 - c\overline{x} - \frac{a}{2}\overline{x}(w^2 - p)\overline{x} \tag{4.56}$$

$$\frac{dw}{dt} = \frac{p - w^2}{2}\left(a\overline{x} + b(w - \overline{x}) + \frac{c}{w - \overline{x}}\right), \tag{4.57}$$

respectively. Corollary 4.9 tells us that all equilibrium points are states in which $w = \sqrt{p}$. Therefore we need only find the values for $\overline{x}$ by finding the roots of the right-hand side of

$$\frac{d\overline{x}}{dt} = a\overline{x}^2(\sqrt{p} - \overline{x}) - b\overline{x}(\sqrt{p} - \overline{x})^2\overline{y} - c\overline{x}. \tag{4.58}$$

The ODE (4.46) is identical in structure to (4.16), and it follows that these roots are 0, E_1, and E_2. Since $x = \overline{x}^2$, we have $\hat{\mathbf{x}}_0(X) = 0$, $\hat{\mathbf{x}}_1(X) = E_1^2$, and $\hat{\mathbf{x}}_2(X) = E_2^2$. Similarly, since $\overline{y} = w - \overline{x}$ and $y = \overline{y}^2$, it follows that $\hat{\mathbf{x}}_0(Y) = p$, $\hat{\mathbf{x}}_1(Y) = (\sqrt{p} - E_1)^2$, and $\hat{\mathbf{x}}_2(Y) = (\sqrt{p} - E_2)^2$. Finally, since $z = p - x - y$, it follows that $\hat{\mathbf{x}}_i(Z) = p - \hat{\mathbf{x}}_i(X) - \hat{\mathbf{x}}_i(Y)$ for $0 \leq i \leq 2$.

What remains to be shown is the stability of the three points. Using the ODEs (4.56) and (4.49), it is routine to show that the two eigenvalues of the Jacobian matrix evaluated at $\hat{\mathbf{x}}_0$ are $\lambda_1 = \lambda_2 = -c - bp$. Since both are negative, it follows that $\hat{\mathbf{x}}_0$ is exponentially stable.

Similarly, the eigenvalues at $\hat{\mathbf{x}}_1$ are

$$\begin{aligned}
\lambda_3 &= \frac{a\sqrt{p}}{2(a+b)}\sqrt{1-c^*}\left(a(1-\sqrt{1-c^*}) + 2b\right), \\
\lambda_4 &= -\frac{a\sqrt{p}}{a+b}\left(a(1-\sqrt{1-c^*}) + 2b\right),
\end{aligned}$$

and since $0 \leq \sqrt{1-c^*} < 1$, we know that $\lambda_3 > 0$ and $\lambda_4 < 0$. Therefore $\hat{\mathbf{x}}_1$ is a saddle node.

Finally, the eigenvalues of $\hat{\mathbf{x}}_2$ are

$$\begin{aligned}
\lambda_5 &= -\frac{ap}{2(a+b)}\sqrt{1-c^*}\left(a(1+\sqrt{1-c^*}) + 2b\right), \\
\lambda_6 &= -\frac{ap}{a+b}\left(a(1-\sqrt{1+c^*}) + 2b\right),
\end{aligned}$$

which are both clearly negative, so $\hat{\mathbf{x}}_2$ is exponentially stable. □

The above theorem not only demonstrates that the bimolecular algorithm has bistable behavior but that its behavior is closely related to the termolecular algorithm from section 4.1. In fact, if the CRN is initialized such that $z(0) = 2\sqrt{x(0)y(0)}$, then the solutions have a precise relationship.

From the theorem it is also clear that if $x(0) > E_1^2$ and $y(0) < (\sqrt{p} - E_1)^2$, then with certainty the system converges to the equilibrium point $\hat{\mathbf{x}}_2$.

We conclude this section with the following theorem that bounds the fully general bimolecular algorithm.

Theorem 4.11. If $N = (S, R)$ is constructed according to Construction 4.6 with the strictly positive real-valued constants $k_1, \ldots, k_6$ and $\widehat{N} = (\widehat{S}, \widehat{R})$ is constructed according to Construction 4.6 with the constants $\hat{k}_1, \ldots, \hat{k}_6$ defined by $\hat{k}_1 = \frac{\hat{k}_2}{2} = \min\{k_1, \frac{k_2}{2}\}$, $\hat{k}_3 = \frac{\hat{k}_4}{2} = \max\{k_3, \frac{k_4}{2}\}$, and $\hat{k}_5 = \frac{\hat{k}_6}{2} = \max\{k_5, \frac{k_6}{2}\}$, then the behavior of X in N is lower bounded by $\widehat{X}$ in $\widehat{N}$.

Proof. The theorem immediately follows from the fact that we are only reducing the rate constants responsible for generating X and increasing the rate constants responsible for destroying X. □

The above theorem shows that we can bound the behavior of a fully general CRN from Construction 4.6 by a slightly modified CRN. Modifying the CRN in this way may not always be possible, but in the most common uses of the algorithm, the rate constants $k_1, \ldots, k_6$ are "close" to satisfying the constraints (4.35)-(4.37), therefore the modifications are minor.

CHAPTER 5. INPUT/OUTPUT CRNS*

The deterministic CRN model that we have been investigating specifies systems which are autonomous and receive no input other than what is encoded into the initial state. However, in chapter 3 we introduced a notion of concentration signal operators which transform one concentration signal into another. We have computed these operators by extending one CRN (which generates the input signal) into a new CRN that uses the original CRN to compute the new signal. However, many of the CRNs that compute these operators have the following undesirable features.

1. The structure of the enclosing CRN depends on the structure of the sub-CRN that generates the input signal.

2. The initial state of the enclosing CRN depends on a precise relationship to the initial state of the sub-CRN.

In practice, it is desirable to design CRNs that act only on the input signal directly and not depend on how the signal is produced. Likewise, it is desirable to have a single initial state of a CRN that handles all valid input signals rather than have the initial state depend on the input signal directly.

In this chapter, we introduce a new model called *input/output chemical reaction networks* which is an extension of the CRN model we have been investigating previously. This new model provides a more intuitive and robust means of computing concentration

* The material in this chapter is joint work with Jim Lathrop and Jack Lutz that has been presented briefly in the posters at the *21st Conference on DNA Computing and Molecular Programming, 2015* and the *Molecular Programming Project 2016 Annual Workshop*. This work will also appear in a forthcoming extension of [40].

signal operators that avoid the undesirable features above and also promotes the modular design of chemical systems. The model was inspired by the input/output systems of control theory [9, 22], and we borrow terminology and notation to remain as consistent as possible.

Formally, 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 U and S are called *input species* and *state species* of N, respectively.

We now specify the *deterministic mass action semantics* of I/O CRNs. These semantics are similar to the those we reviewed in chapter 2 with the exception that I/O CRNs need to be provided with an input signal and an output interface. Therefore we define a *context* of an *I/O* CRN $N = (U, R, S)$ as 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 $h : [0, \infty)^{S \cup U} \to [0, \infty)^V$ is an *output function*. We write $\mathcal{C}_N$ for the set of all contexts of N. Intuitively, the semantics of an I/O CRN N specify how it 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.

For each reaction $\rho \in R$ and each $(\mathbf{x}, \mathbf{u}) \in [0, \infty)^{S \cup U}$, we define the *rate of ρ in the global state* $(\mathbf{x}, \mathbf{u})$ to be

$$\text{rate}_{\mathbf{x},\mathbf{u}}(\rho) = k(\rho) \prod_{Y \in S \cup U} (\mathbf{x}, \mathbf{u})(Y)^{\mathbf{r}(\rho)(Y)}. \tag{5.1}$$

Notice that equation (5.1) has the same structure as equation (2.2), therefore if ρ is the reaction $X+Y \xrightarrow{k} X+2Z$, where $X \in U$ and $Y, Z \in S$, then $\text{rate}_{\mathbf{x},\mathbf{u}}(\rho) = k \cdot \mathbf{u}(X) \cdot \mathbf{x}(Y)$.

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

$$F_Y(\mathbf{x}, \mathbf{u}) = \sum_{\rho \in R} \text{rate}_{\mathbf{x},\mathbf{u}}(\rho) \Delta\rho(Y) \tag{5.2}$$

for all $\mathbf{x} \in [0,\infty)^S$ and $\mathbf{u} \in [0,\infty)^U$, noting its similarity to equation (2.3).

Now let $(\mathbf{u}, V, h)$ be a context of the I/O CRN N. Then if the state of N is $\mathbf{x}(t) \in [0,\infty)^S$ at time t, then the concentration of each state species $Y \in S$ must obey the ordinary differential equation (ODE)

$$y'(t) = F_Y(\mathbf{x}(t), \mathbf{u}(t)). \tag{5.3}$$

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

$$F(\mathbf{x}, \mathbf{u}) = (F_Y(\mathbf{x}, \mathbf{u}) \mid Y \in S) \tag{5.4}$$

for all $\mathbf{x} \in [0,\infty)^S$ and $\mathbf{u} \in [0,\infty)^U$, then the deterministic mass action system of N in the context $(\mathbf{u}, V, h)$ is the vector

$$\mathbf{x}'(t) = F(\mathbf{x}(t), \mathbf{u}(t)). \tag{5.5}$$

We also note that equations (5.3)-(5.5) are symmetrical to equations (2.4)-(2.6).

If an I/O CRN N is initialized to a state $\mathbf{x}_0 \in [0,\infty)^S$ at time 0 in the context $(\mathbf{u}, V, h)$, then its state evolves according to the mass action system (5.5). 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 (5.5) together with the initial value condition $y(0) = \mathbf{x}_0(Y)$ for each $Y \in S$. By the standard existence-uniqueness

theory for ODEs [7, 64], 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.

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

$$h_0(\mathbf{x}, \mathbf{u})(Y) = \mathbf{x}(Y) \tag{5.6}$$

for all $Y \in V$. If $\mathbf{x}(t)$ is a solution to the IVP as described in the preceding paragraph, then the *output signal* of the I/O CRN N in the context $\mathbf{c} = (\mathbf{u}, V, h)$ 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

$$N_{\mathbf{c},\mathbf{x}_0}(t) = h(\mathbf{x}(t), \mathbf{u}(t)) \tag{5.7}$$

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

In the language of control theory, an *input/output system* is a system of the form (5.5), 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*.

By comparing the equations (5.1)-(5.5) with equations (2.2)-(2.6), it is clear that a CRN $N = (S, R)$ is equivalent to the I/O CRN $\widehat{N} = (\emptyset, R, S)$. It is also useful to note that the I/O CRNs considered here have mass action systems that are neither autonomous nor polynomial.

We also note 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

$$\bigsqcup \mathcal{N} = (U^* \setminus S^*, R^*, S^*), \tag{5.8}$$

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 $\mathcal{N}$.

5.1 I/O Time-Dependent CRNs

We now briefly introduce a variant of the I/O CRN model called *I/O time-dependent CRNs* (*I/O tdCRNs*). This model is used in section 5.2 to define what it means for an I/O CRN to be robust with respect to rate constants. Intuitively, an I/O tdCRN is simply an I/O CRN in which the "rate constants" of the reactions are allowed to vary over time.

Formally, a *time-dependent reaction* over a finite set $S \subseteq \mathbf{S}$ is a triple $\rho = (\mathbf{r}, \mathbf{p}, \hat{k})$ where $\mathbf{r}, \mathbf{p} \in \mathbb{N}^S$ and $\hat{k} : [0, \infty) \to [0, \infty)$ is a continuous function. 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

$$\widehat{N} = (U, \widehat{R}, S),$$

where U and S are as in the I/O CRN definition and $\widehat{R}$ is a finite set of time-dependent reactions over S. The deterministic mass action semantics of an I/O tdCRN $\widehat{N} = (U, \widehat{R}, S)$

is defined in the obvious way, rewriting (5.1)–(5.5) as

$$\text{rate}_{\mathbf{x},\mathbf{u}}(\rho)(t) = \hat{k}(\rho)(t)(\mathbf{x},\mathbf{u})^{\mathbf{r}(\rho)}, \tag{5.9}$$

$$F_Y(\mathbf{x},\mathbf{u},t) = \sum_{\rho\in\widehat{R}} \text{rate}_{\mathbf{x},\mathbf{u}}(\rho)(t)\Delta_\rho(Y), \tag{5.10}$$

$$y'(t) = F_Y(\mathbf{x}(t),\mathbf{u}(t),t), \tag{5.11}$$

$$(\mathcal{E}_Y \mid Y \in S), \tag{5.12}$$

$$F(\mathbf{x},\mathbf{u},t) = (F_Y(\mathbf{x},\mathbf{u},t) \mid Y \in S), \tag{5.13}$$

$$\mathbf{x}'(t) = F(\mathbf{x}(t),\mathbf{u}(t),t). \tag{5.14}$$

The *output signal* $\widehat{N}_{\mathbf{c},\mathbf{x}_0}$ of an I/O tdCRN $\widehat{N}$ 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 $\widehat{N} = (U, \widehat{R}, S)$ in which $\widehat{R}$ 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

$$|\hat{k}(t) - k| \leq \delta \tag{5.15}$$

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

5.2 Robustness of I/O CRNs

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

$$\alpha(\mathbf{c}) \implies \phi(\mathbf{u}, N_{\mathbf{c},\mathbf{x}_0}) \tag{5.16}$$

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 (5.16) 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 $B_\epsilon(f)$ of all $g \in C[W]$ such that $\|g - f\| \leq \epsilon$.

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

$$\alpha(\mathbf{c}) \implies (\exists \mathbf{v} \in B_\epsilon(N_{\mathbf{c},\mathbf{x}_0}))\phi(\mathbf{u}, \mathbf{v}) \tag{5.17}$$

holds for every context $\mathbf{c} = (\mathbf{u}, h) \in \mathcal{C}_N$. 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 (5.16) and (5.17) 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 "$\widehat{N}$ close to N," the right-hand side of (5.17) holds with $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ 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 $f, g : [0,\infty)^W \to [0,\infty)^{W'}$ and the closed ball of radius ϵ about f. 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.

Finally, we say that $\widehat{N}$ is δ-*close* to N if $\widehat{N}$ is a δ-perturbation of N.

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 $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4) \in (0,\infty)^4$ 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 $N, \mathbf{x}_0 \models_\epsilon^{\boldsymbol{\delta}} \Phi$, if, for every $\mathbf{c} = (\mathbf{u}, V, h) \in \mathcal{C}_N$ satisfying $\alpha(\mathbf{c})$, every $\hat{\mathbf{c}}$ that is (δ_1, δ_2)-close to $\mathbf{c}$, every $\hat{\mathbf{x}}_0$ that is δ_3-close to $\mathbf{x}_0$, and every $\widehat{N}$ that is δ_4-close to N, there exists $\mathbf{v} \in B_\epsilon(\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0})$ such that $\phi(\mathbf{u}, \mathbf{v})$ holds. Finally, we say that N *$\boldsymbol{\delta}$-robustly ϵ-satisfies* Φ, and we write $N \models_\epsilon^{\boldsymbol{\delta}} \Phi$, if there exists $\mathbf{x}_0 \in [0,\infty)^S$ such that $N, \mathbf{x}_0 \models_\epsilon^{\boldsymbol{\delta}} \Phi$.

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

CHAPTER 6. ROBUST SIGNAL CONDITIONING*

Signal conditioning is a general term referring to preparing a signal for the next stage of processing. In this chapter, we study a specific type of signal conditioner that serves to help remove noise from a particular input signal. The types of signals we concern ourselves with here are those that encode a binary sequence by contrasting "low" and "high" concentrations. This is a common technique of encoding a binary sequence and is used in electronics by the contrast of low and high voltages in a circuit. Ideally, we would want a binary sequence to be encoded into a concentration signal as a square wave where the high corresponds to a 1 and the low corresponds to a 0. In reality, we can only approximate this type of encoding, and it is common for a signal to degrade during various computations so that the differences between high and low become indistiguishable. In chapter 4 we studied two signal restoration algorithms that help prevent a certain type of degrading over time. However, these algorithms are designed to help store a single bit of memory indefinitely rather than clean up a sequence of time-varying bits encoded in a concentration signal. The signal conditioning I/O CRN presented here, however, is designed to copy an input signal and literally just make the high points higher and the low points lower.

We begin by formally stating the requirement of our signal conditioning system. Let $\tau > 0$, and let $X \in \mathbf{S}$ be a species. Define the requirement $\Phi^{(X)} = \Phi^{(X)}(\tau) = (\alpha, \phi)$ as

* The material in this chapter is joint work with Jim Lathrop and Jack Lutz that has been presented briefly in the posters at the *21st Conference on DNA Computing and Molecular Programming, 2015* and the *Molecular Programming Project 2016 Annual Workshop*. This work will also appear in a forthcoming extension of [40].

follows. The context assumption $\alpha : C[\{X\}] \to \{\mathbf{false}, \mathbf{true}\}$ of $\Phi^{(X)}$ is defined by

$$\alpha(\mathbf{u}, V, h) \equiv \Big[V = \{X^*, \overline{X}^*\} \text{ and } h = h_0\Big]. \tag{6.1}$$

Notice that this context assumption simply requires that the I/O CRN has one input species X and two output species X^* and $\overline{X}^*$. The restriction that the output function is the zero-error projection function (5.6) is a necessary because we are only concerning ourselves with basic measurement functions that are trying to measure the concentrations of the output species exactly. Noise will be introduced into this output function when we show that this requirement is robustly satisfied.

The I/O requirement ϕ of $\Phi^{(X)}$ requires a little more work, and we begin by defining some terminology and notation. If $I \subseteq [0, \infty)$ is a closed interval, we write $|I|$ to denote the length of the interval. If $I = [a, b]$ and $|I| \geq \tau$, we define the subinterval $I_\tau = [a + \tau, b]$ which is simply the interval I with τ sliced off the left-hand side.

Now 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 $|I| \geq \tau$, 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 simply a segment of the input signal which has length at least τ and presents a single bit for the duration of the interval.

Similarly, 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.

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**. An output event (b, I) is a segment of the output signal that presents the bit b in a dual-rail format, i.e., X^* matches the bit b and $\overline{X}^*$ matches the bit $1 - b$ during the interval.

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

$$\phi(\mathbf{u}, \mathbf{v}) \equiv \big[(b, I) \in \mathbf{IEV} \implies (b, I_\tau) \in \mathbf{OEV}\big]. \tag{6.2}$$

This completes our specification of $\Phi^{(X)}$.

Intuitively, the requirement $\Phi^{(X)}$ requires an I/O CRN to have one input species X, two output species $X^*, \overline{X}^*$, and requires that whenever the input signal has exactly concentration $b \in \{0, 1\}$, the output signal needs to, within τ time, make the species X^* become b and the species $\overline{X}^*$ become $1 - b$. Therefore the output species encode both the original bit b and its compliment $1 - b$.

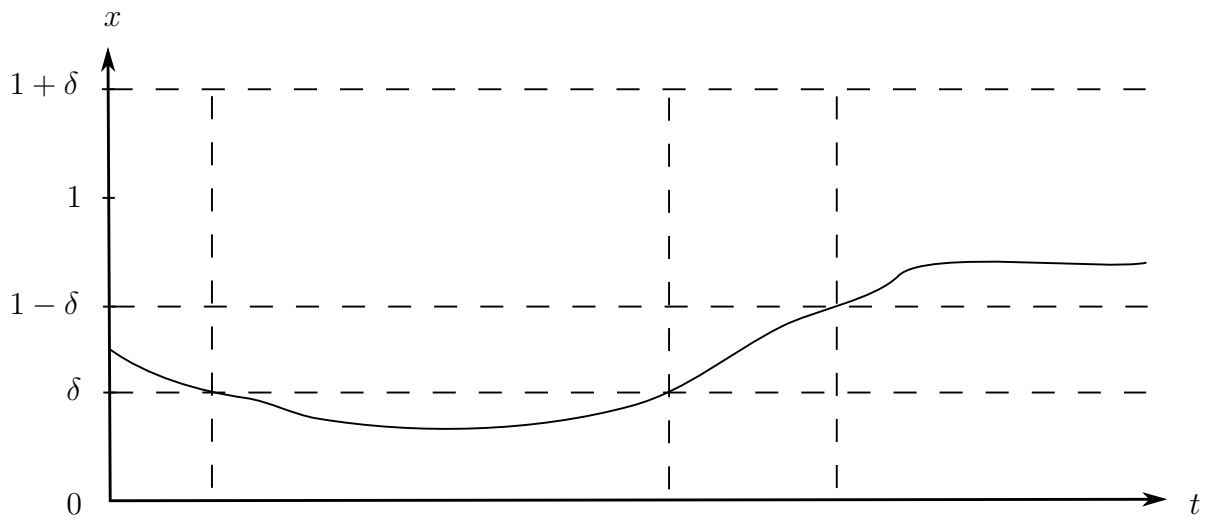
By inspecting equations (6.1) and (6.2), we can see that the requirement matches this intuition. As an example, suppose an input signal $x(t)$ contains input events which only get within $\delta \in (0, \frac{1}{2})$ of the bits it is encoding. Then, what we desire is an I/O CRN which is capable of *improving* this signal so that it gets within $\epsilon < \delta$ of the bits it is encoding and only introducing τ amount of delay. Figure 6.1 depicts what the output of such an I/O CRN would be given such a perturbed input signal.

We now give a construction that specifies a family of I/O CRNs that are capable of robustly satisfying the requirement $\Phi^{(X)}$. We first state the construction formally, and afterwards we give an intuitive overview of how it works.

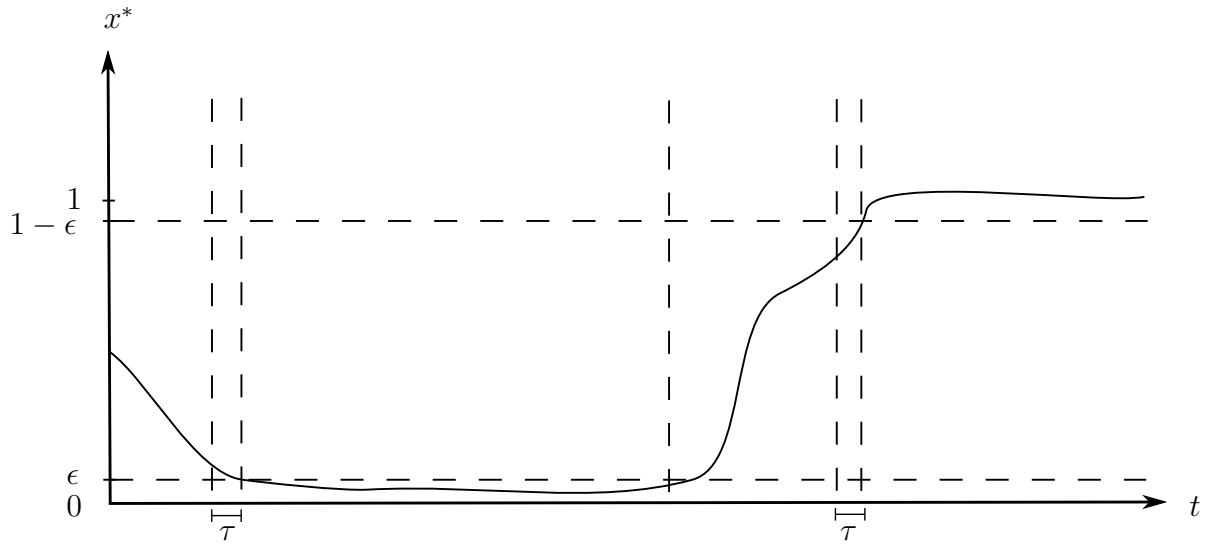
Construction 6.1. Given the real numbers $\tau > 0$, $\epsilon > 0$, and $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$ where $\delta_1 \in (0, \frac{1}{3})$, $\delta_2 \in [0, \epsilon)$, $\delta_3 \in (0, \frac{1}{2})$, and $\delta_4 > 0$, let $b = \frac{1-\delta_1}{2\delta_1}$ and $n = \lceil 2 \log_b(\frac{8}{\epsilon - \delta_2}) \rceil$.

Define the I/O CRN $N^{(X)} = N^{(X)}(\tau, \epsilon, \boldsymbol{\delta}) = (U, R, S)$ where

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



(a) An input signal $x(t)$ with two input events.



(b) An output signal $x^*(t)$ that satisfies the I/O requirement ϕ.

Figure 6.1: 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 a "cleaner" version of the input signal $x(t)$. Note that other output signal $\overline{x}^*(t)$ is not depicted.

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

$$\begin{aligned} k_1 &= 2\delta_4 + \frac{2n\log(2n)}{\tau(1-\delta_1)} + \frac{2}{\tau}\log\left(10\left(\frac{8}{\epsilon-\delta_2}\right)^2\left(\frac{2}{1-\delta_1}\right)^n\right) + \frac{\delta_4(2+\delta_1)}{\delta_1}, \\ k_2 &= \frac{2}{\tau}\log\left(\frac{3}{\epsilon-\delta_2}\right) + 4\delta_4. \end{aligned}$$

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

$$\begin{aligned} \mathbf{x}_0^{(X)}(X^*) &= 0, \\ \mathbf{x}_0^{(X)}(\overline{X}^*) &= 1 + \delta_3, \\ \mathbf{x}_0^{(X)}(X_0) &= \frac{10}{\epsilon-\delta_2}\left(\frac{2}{1-\delta_1}\right)^n + \delta_3, \\ \mathbf{x}_0^{(X)}(X_i) &= 0 \quad (\forall\, 0 < i \le n). \end{aligned}$$

The above construction looks overwhelmingly complicated, but the intuition behind the construction is simple. The parameter τ corresponds to the delay, ϵ corresponds to how well the output needs to be "cleaned up," and $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$ corresponds to how robust the I/O CRN needs to be. These parameters tune the construction so as to satisfy the requirement $\Phi^{(X)}$ in a robust way.

The species of the I/O CRN are naturally separated into two parts. The first part is the cascade of species $X_0, \ldots, X_n$ where the length of the cascade n is defined in the

construction. This cascade is designed so that every species X_i "falls down" to the bottom of the cascade 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 the input species 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 I/O CRN $N^{(X)}$ consists of the species X^* and $\overline{X}^*$ which are the output species. These species always sum to a concentration of 1, 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 $\overline{X}^*$ both collaborate to clean up the input signal.

The length of the cascade, the rate constants, and the initial concentration of X_0 are all carefully set and depend on the parameters of $N^{(X)}$. The more robust and the more cleaning up the CRN has to do, the longer the cascade must be, etc. The reason for their complexity becomes apparent in the proof of the main theorem of this chapter, stated below.

Theorem 6.2. If $\tau > 0$, $\epsilon \in (0, \frac{1}{2})$, $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$ with $\delta_1 \in (0, \frac{1}{3})$, $\delta_2 \in (0, \epsilon)$, $\delta_3 \in (0, \frac{1}{2})$, $\delta_4 > 0$, and $N^{(X)} = N^{(X)}(\tau, \epsilon, \boldsymbol{\delta})$ and $\mathbf{x}_0^{(X)}$ are constructed according to Construction 6.1, then

$$N^{(X)}, \mathbf{x}_0^{(X)} \models_\epsilon^{\boldsymbol{\delta}} \Phi^{(X)}(\tau). \tag{6.3}$$

The rest of this chapter is devoted to proving Theorem 6.2. This proof is rather extensive and is partitioned into two parts. Section 6.1 is dedicated to the analysis of the ODEs generated by the cascade of species of $N^{(X)}$, and section 6.2 presents a complete proof of Theorem 6.2.

6.1 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 6.1. 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 6.3. 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

$$\frac{dx_0}{dt} = \sum_{i=1}^{n} bx_i - fx_0, \tag{6.4}$$

$$\frac{dx_i}{dt} = fx_{i-1} - (f+b)x_i \quad \text{for } 0 < i < n, \tag{6.5}$$

$$\frac{dx_n}{dt} = fx_{n-1} - bx_n. \tag{6.6}$$

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

$$F_i(t) = p\left(\frac{f}{f+b}\right)^i e^{-(f+b)t} \sum_{k=i}^{\infty} \frac{t^k (f+b)^k}{k!}. \tag{6.7}$$

Observation 6.5. If $f > 0$, $b > 0$, $p > 0$, $i \in \mathbb{N}$ and F_i is constructed according to Construction 6.4, then

$$\int e^{(f+b)t} \cdot F_i(t) dt = \frac{1}{f} e^{(f+b)t} F_{i+1}(t) + C \tag{6.8}$$

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

Proof. Assume the hypothesis. Then by the definition of F_i from equation (6.7),

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

Since

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

for some $C_1 \in \mathbb{R}$, we see that

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

for some $C \in \mathbb{R}$. □

Lemma 6.6. 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 6.3 and 6.4 such that $x_0(0) = p$ and $x_i(0) = 0$ for all $0 < i \leq n$, then for all $t \in [0, \infty)$,

$$x_i(t) = F_i(t) - F_{i+1}(t) \quad \text{for } 0 \leq i < n, \tag{6.9}$$

$$x_n(t) = F_n(t). \tag{6.10}$$

Proof. Assume the hypothesis. We begin by proving (6.9) 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 (6.4) 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 $0 \leq i < n-1$. By (6.5), the derivative of x_{i+1} is

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

This ODE is of the form

$$\frac{dy}{dt} + f(t) \cdot y = g(t),$$

which can be solved using the integrating factor method. Using this method, 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 6.5 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 (6.9) holds.

It remains to be shown that (6.10) holds. Since $\sum_{i=0}^{n} x_i(t) = p$ for all $t \in [0, \infty)$,

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

Lemma 6.7. Under the assumptions of Lemma 6.6,

$$F_n(t) > p \left(\frac{f}{f+b}\right)^n \left(1 - ne^{-\frac{1}{n}(f+b)t}\right) \tag{6.11}$$

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

Proof. It suffices to show that

$$e^{-(f+b)t}\sum_{k=n}^{\infty}\frac{t^k(f+b)^k}{k!} > 1 - ne^{-\frac{1}{n}(f+b)t}. \tag{6.12}$$

The left-hand side of the (6.12) is related to the incomplete gamma function. In fact,

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

where $\gamma(a,x)$ is the lower incomplete gamma function. The incomplete gamma function is well understood and many useful bounds exist. One particularly useful bound by Alzer [1, 30] is

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

for $a \geq 1$ 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 $n^n \geq n!$, we know $s_n > \frac{1}{n}$, whence

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

Corollary 6.8. Under the assumptions of Lemma 6.6 with $(f+b)t \geq n\log(2n)$,

$$F_n(t) > \frac{p}{2}\left(\frac{f}{f+b}\right)^n \tag{6.13}$$

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

Lemma 6.9. 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 6.3 and 6.4 such that $x_i(0) = 0$ for all $0 \leq i < n$

and $x_n(0) = p$, then for all $t \in [0, \infty)$,

$$x_i(t) = \frac{b}{f} F_{i+1}(t) \quad \text{for } 0 \leq i < n. \tag{6.14}$$

Proof. Assume the hypothesis. We prove (6.14) 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 (6.4) can be simplified to

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

and has solution

$$\begin{aligned} x_0(t) &= p\left(\frac{b}{f+b}\right)\left(1 - e^{-(f+b)t}\right) \\ &= \frac{b}{f}\left(p\left(\frac{f}{f+b}\right) e^{-(f+b)t} \sum_{k=1}^{\infty} \frac{t^k (f+b)^k}{k!}\right) = \frac{b}{f} \cdot F_1(t). \end{aligned}$$

For the induction step, assume that $x_i(t) = \frac{b}{f} F_{i+1}(t)$ for some $0 \leq i < n-1$. By (6.5), 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

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

It follows from Observation 6.5 that

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

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

Lemma 6.10. Under the assumptions of Lemma 6.9,

$$x_n(t) < pe^{-bt} + p\left(\frac{f}{f+b}\right)^n \left(1 - e^{-bt}\right) \tag{6.15}$$

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

Proof. Assume the hypothesis. By equation (6.6), the derivative of x_n is

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

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 6.9, we know that $x_{n-1}(t) = \frac{b}{f} F_n(t)$, therefore

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

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

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

Since

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

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

$$\begin{aligned} x_n(t) &= bp \left(\frac{f}{f+b}\right)^n e^{-bt} \left[\frac{1}{b} e^{bt} - \sum_{k=0}^{n-1} \frac{(f+b)^k}{k!} \left(-\frac{k!}{f^{k+1}} e^{-ft} \sum_{i=0}^{k} \frac{t^i f^i}{i!}\right) + C_3\right] \\ &= p \left(\frac{f}{f+b}\right)^n + p\frac{b}{f} e^{-bt} \sum_{k=0}^{n-1} \left(\frac{f}{f+b}\right)^{n-k-1} e^{-ft} \sum_{i=0}^{k} \frac{t^i f^i}{i!} + C_4 e^{-bt} \end{aligned}$$

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_4, we obtain

$$\begin{aligned} x_n(t) &= pe^{-bt} + p \left(\frac{f}{f+b}\right)^n \left(1 - e^{-bt}\right) + p\frac{b}{f} e^{-bt} \sum_{k=0}^{n-1} \left(\frac{f}{f+b}\right)^{n-k-1} \left(e^{-ft} \sum_{i=0}^{k} \frac{t^i f^i}{i!} - 1\right) \\ &= pe^{-bt} + p \left(\frac{f}{f+b}\right)^n \left(1 - e^{-bt}\right) - p\frac{b}{f} e^{-(f+b)t} \sum_{k=0}^{n-1} \left(\frac{f}{f+b}\right)^{n-k-1} \sum_{i=k+1}^{\infty} \frac{t^i f^i}{i!} \\ &< pe^{-bt} + p \left(\frac{f}{f+b}\right)^n \left(1 - e^{-bt}\right). \end{aligned}$$

□

At this point, we have derived the solutions and bounds necessary for the cascade of species $X_0, \ldots, X_n$ from Construction 6.1. 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 6.11. Given $f > 0$ and $b > 0$, let $x, \overline{x} : [0, \infty) \to [0, \infty)$ be functions that satisfy the ODEs

$$\frac{dx}{dt} = f\overline{x} - bx, \tag{6.16}$$

$$\frac{d\overline{x}}{dt} = bx - f\overline{x}. \tag{6.17}$$

Lemma 6.12. If x and $\overline{x}$ are functions constructed according to Construction 6.11 with $f > 0$ and $b > 0$, then for all $t \in [0, \infty)$,

$$x(t) = p\left(\frac{f}{f+b}\right)\left(1 - e^{-(f+b)t}\right) + x(0) \cdot e^{-(f+b)t} \tag{6.18}$$

$$\overline{x}(t) = p - x(t), \tag{6.19}$$

(6.20)

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 (6.19) holds.

To show (6.18) holds, we solve the ODE (6.16) which can be simplified to

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

which has solution (6.18). □

Lemma 6.13. If $\epsilon \in (0, \frac{1}{2})$, $\tau > 0$, and $x, \overline{x}$ are constructed according to Construction 6.3 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 for all $t \geq \tau$,

$$x(t) > p - \epsilon, \text{ and } \qquad \overline{x}(t) < \epsilon.$$

Proof. Assume the hypothesis. Then by Lemma 6.12, for all $t \geq \tau$,

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

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

$$x(t) \geq p\left(\frac{1}{1+\frac{\epsilon}{2p}}\right)\left(1-\frac{\epsilon}{2p}\right).$$

Since $\frac{\epsilon}{2p} < \frac{\epsilon}{2p-\epsilon}$, Since $\frac{\epsilon}{2p} < \frac{\epsilon}{2p-\epsilon}$,

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

6.2 Proof of Theorem

Assume the hypothesis of Theorem 6.2. Then $\tau > 0$, $\epsilon \in (0, \frac{1}{2})$, $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$ with $\delta_1 \in (0, \frac{1}{3})$, $\delta_2 \in (0, \epsilon)$, $\delta_3 \in (0, \frac{1}{2})$, $\delta_4 > 0$, and $N(X) = N^{(X)}(\tau, \epsilon, \boldsymbol{\delta})$ and $\mathbf{x}_0^{(X)}$ are constructed according to Construction 6.1. Then we need only 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_1, δ_2)-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_3-close to $\mathbf{x}_0^{(X)}$, let $\widehat{N}$ be δ_4-close to $N^{(X)}$, and let p and p^* be the constants

$$p = \sum_{i=0}^{n} \hat{\mathbf{x}}_0(X_i),$$

$$p^* = \hat{\mathbf{x}}_0(X^*) + \hat{\mathbf{x}}_0(\overline{X}^*).$$

It now suffices to show that $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ 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 $\widehat{N}$ 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 $\widehat{N}$ can be derived from the reactions in Construction 6.1 along with the perturbed mass action function from equation (5.11) and are

$$\frac{dx_0}{dt} = \sum_{i=1}^{n} \hat{k}_1 x_i - (\hat{k}_1 x) x_0, \tag{6.21}$$

$$\frac{dx_i}{dt} = (\hat{k}_1 x) x_{i-1} - (\hat{k}_1 x + \hat{k}_1) x_i \quad \text{for } 0 < i < n, \tag{6.22}$$

$$\frac{dx_n}{dt} = (\hat{k}_1 x) x_{n-1} - \hat{k}_1 x_n. \tag{6.23}$$

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

$$\frac{dx^*}{dt} = (\hat{k}_2 x_n)\overline{x}^* - \hat{k}_2 x^*, \tag{6.24}$$

$$\frac{d\overline{x}^*}{dt} = \hat{k}_2 x^* - (\hat{k}_2 x_n)\overline{x}^*. \tag{6.25}$$

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 $\mathbf{u}$, and let $I = [t_1, t_2]$. Since the input signal can be perturbed by δ_1, it follows that $x(t) > 1 - \delta_1$ for all $t \in I$. We also know that the rate constants can be perturbed by δ_4. 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 6.6, 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 $f = (k_1 - \delta_4)(1 - \delta_1)$ and $b = k_1 + \delta_4$. 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 6.7, 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 $k_1 > \delta_4 + \frac{2n}{\tau(1-\delta_1)} \log(2n)$, Corollary 6.8 tells us

$$x_n(t) > p\left(\frac{f}{f+b}\right)^n \left(\frac{1}{2}\right) = \frac{p}{2}\left(\frac{(k_1 - \delta_4)(1-\delta_1)}{(k_1 - \delta_4)(1-\delta_1) + k_1 + \delta_4}\right)^n = \frac{p}{2}\left(\frac{1-\delta_1}{1-\delta_1+u}\right)^n,$$

where $u = \frac{k_1+\delta_4}{k_1-\delta_4}$. Since $k_1 > 2\frac{\delta_4(2+\delta_1)}{\delta_1}$, we know that $u < 1 + \delta_1$ and therefore

$$x_n(t) > \frac{p}{2}\left(\frac{1-\delta_1}{2}\right)^n.$$

Since the initial condition can be perturbed by at most δ_3, $p > \frac{10}{\epsilon - \delta_2}\left(\frac{2}{1-\delta_1}\right)^n$, therefore

$$x_n(t) > \frac{5}{\epsilon - \delta_2},$$

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

Recall that the ODEs for X^* and $\overline{X}^*$ are (6.24) and (6.25). 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 6.12, 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 $f = (k_2 - \delta_4)\frac{5}{\epsilon - \delta_2}$ and $b = k_2 + \delta_4$. Since $k_2 > 4\delta_4$, then $\frac{k_2+\delta_4}{k_2-\delta_4} < \frac{5}{3}$ and therefore

$$b = k_2 + \delta_4 < \frac{\epsilon - \delta_2}{3}\left[(k_2 - \delta_4)\frac{5}{\epsilon - \delta_2}\right] = \frac{\epsilon - \delta_2}{3} f,$$

whence

$$b < \frac{\epsilon - \delta_2}{2p^*} f.$$

Since $\frac{5}{\epsilon - \delta_2} > 1$ and $k_2 = \frac{2}{\tau}\log\left(\frac{3}{\epsilon - \delta_2}\right) + 4\delta_4$,

$$f = (k_2 - \delta_4)\frac{5}{\epsilon - \delta_2} > \frac{2}{\tau}\log\left(\frac{3}{\epsilon - \delta_2}\right),$$

whence

$$f > \frac{2}{\tau}\log\left(\frac{2p^*}{\epsilon - \delta_2}\right).$$

By Lemma 6.13, for all $t \in [t_1 + \tau, t_2]$,

$$x^*(t) > p^* - \epsilon + \delta_2.$$

Since the initial state can only be perturbed by at most δ_3 and the output function can only introduce δ_2 error, it follows that

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

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

It remains to be shown that $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ 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 $x(t) < \delta_1$ for all $t \in I$. Similar to the above argument, by Lemma 6.10, 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 $f = (k_1 + \delta_4)\delta_1$ and $b = k_1 - \delta_4$. Since this function is monotonically decreasing, for all $t \in [t_1 + \frac{\tau}{2}, t_2]$,

$$\begin{aligned} x_n(t) &< p\left(\frac{f}{f+b}\right)^n + pe^{-b\frac{\tau}{2}} \\ &= p\left(\frac{(k_1+\delta_4)\delta_1}{(k_1+\delta_4)\delta_1 + k_1 - \delta_4}\right)^n + pe^{-b\frac{\tau}{2}} \\ &= p\left(\frac{\delta_1}{\delta_1 + u}\right)^n + pe^{-b\frac{\tau}{2}}, \end{aligned}$$

where $u = \frac{k_1 - \delta_4}{k_1 + \delta_4}$. Since $k_1 > \frac{\delta_4(2-\delta_1)}{\delta_1}$, we know that $u > 1 - \delta_1$, whence for all $t \in [t_1 + \frac{\tau}{2}, t_2]$,

$$x_n(t) < p\delta_1^n + pe^{-b\frac{\tau}{2}}.$$

Since $p < \frac{10}{\epsilon - \delta_2}\left(\frac{2}{1-\delta_1}\right)^n + 2\delta_3$,

$$\begin{aligned} x_n(t) &< \frac{10}{\epsilon - \delta_2}\left(\frac{2\delta_1}{1-\delta_1}\right)^n + \delta_1^n + pe^{-b\frac{\tau}{2}} \\ &< \frac{10 + \epsilon - \delta_2}{\epsilon - \delta_2}\left(\frac{2\delta_1}{1-\delta_1}\right)^n + pe^{-b\frac{\tau}{2}} \\ &< \frac{32}{3(\epsilon - \delta_2)}\left(\frac{1-\delta_1}{2\delta_1}\right)^{-n} + pe^{-b\frac{\tau}{2}}. \end{aligned}$$

Since $n \geq \log_{\left(\frac{1-\delta_1}{2\delta_1}\right)}\left(\frac{64}{(\epsilon-\delta_2)^2}\right)$,

$$\begin{aligned} x_n(t) &< \frac{32}{3(\epsilon - \delta_2)}\left(\frac{(\epsilon - \delta_2)^2}{64}\right) + pe^{-b\frac{\tau}{2}} \\ &= \frac{\epsilon - \delta_2}{6} + pe^{-b\frac{\tau}{2}}. \end{aligned}$$

As we showed before, $p < \frac{32}{3(\epsilon-\delta_2)}\left(\frac{2}{1-\delta_1}\right)^n + 2\delta_3$, whence

$$x_n(t) < \frac{\epsilon - \delta_2}{6} + \frac{32}{3(\epsilon - \delta_2)}\left(\frac{4}{1-\delta_1}\right)^n e^{-b\frac{\tau}{2}} + 2\delta_3 e^{-b\frac{\tau}{2}}.$$

Since $b = k_1 - \delta_4 > \frac{2}{\tau} \log\left(\frac{640}{(\epsilon-\delta_2)^2}\left(\frac{2}{1-\delta_1}\right)^n\right)$,

$$x_k(t) < \frac{\epsilon-\delta_2}{6} + \frac{\epsilon-\delta_2}{60} + 2\delta_3 e^{-b\frac{\tau}{2}} < \frac{\epsilon-\delta_2}{6} + \frac{\epsilon-\delta_2}{30},$$

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

$$x_n(t) < \frac{\epsilon-\delta_2}{5}.$$

We now bound the concentration of X^* and $\overline{X}^*$. If $f = (k_2+\delta_4)\frac{\epsilon-\delta_2}{5}$ and $b = k_2 - \delta_4$, then by Lemma 6.12

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

for all $t \in [t_1 + \tau, t_2]$. Since $\frac{k_2-\delta_4}{k_2+\delta_4} > \frac{3}{5}$,

$$f = (k_2+\delta_4)\frac{\epsilon-\delta_2}{5} < \frac{\epsilon-\delta_2}{3}(k_2-\delta_4) < \frac{\epsilon-\delta_2}{2p^*}b.$$

Since $b = k_2 - \delta_4 > \frac{2}{\tau}\log\left(\frac{3}{\epsilon-\delta_2}\right) > \frac{2}{\tau}\log\left(\frac{2p^*}{\epsilon-\delta_2}\right)$, by Lemma 6.13, for all $t \in [t_1+\tau, t_2]$,

$$\overline{x}^*(t) > p^* - \epsilon + \delta_2.$$

Since $x^*(t) + \overline{x}^*(t) = p^*$ for all $t \in [0,\infty)$, it follows that $x^*(t) < \epsilon - \delta_2$ for all $t \in I_\tau$. Since $p^* > 1 + \delta_3$, it follows that $\overline{x}^*(t) > 1 - \epsilon + \delta_2$ for all $t \in I_\tau$.

Finally, since the output function can at most deviate by δ_2 from the solutions of $x^*(t)$ and $\overline{x}^*(t)$, it is clear that $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to having $(0, I_\tau)$ as a valid output event. $\square$

CHAPTER 7. ROBUST FINITE AUTOMATA*

In this chapter we give a uniform translation of a nondeterministic finite automaton 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 chapter [44].

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*. 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 $|\Delta|$.

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.

* The material in this chapter is joint work with Jim Lathrop and Jack Lutz that has been presented briefly in the posters at the *21st Conference on DNA Computing and Molecular Programming, 2015* and the *Molecular Programming Project 2016 Annual Workshop*. This work will also appear in a forthcoming extension of [40].

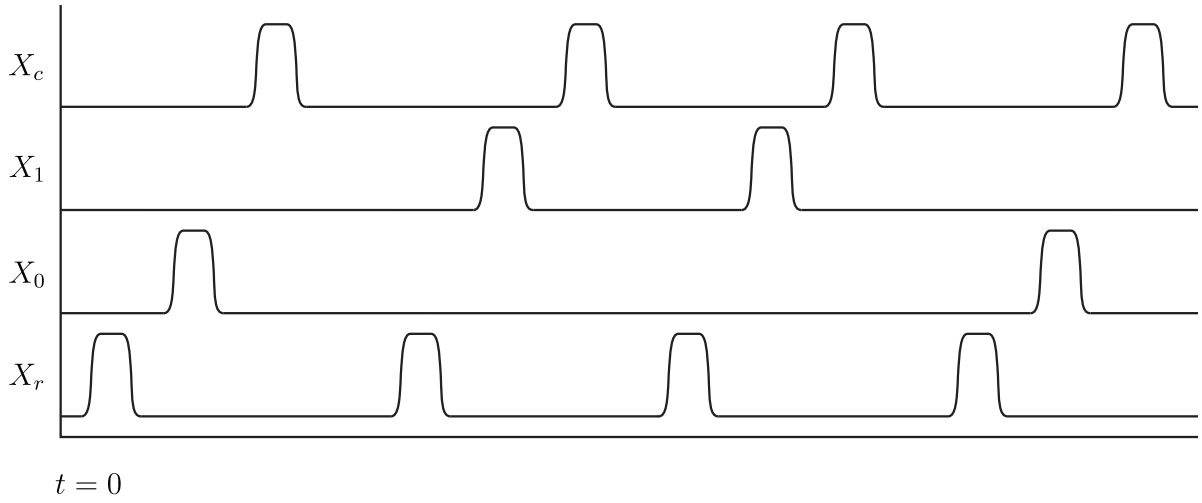


Figure 7.1: Example input signal for 0110.

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 now define the set of input species of N to be

$$U = \{X_a \mid a \in \Sigma\} \cup \{X_r, X_c\}, \tag{7.1}$$

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

An input $w \in \Sigma^*$ is presented to N as a sequence of pulses in the concentrations of the $|\Sigma| + 2$ input species. Each character $a \in \Sigma$ 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 Figure 7.1. 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 and the sequence $(I_i \mid 0 \leq i < k)$ testifies to this fact where I_i is an a_i-event for each $0 \leq i < k$, and if $t \in [0, \infty)$, 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 $I_j < t$.

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 $\tau(\mathbf{u}) = \mathit{if}\ k = 0\ \mathit{then}\ 1\ \mathit{else}$ 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

$$\alpha(\mathbf{u}, V, h) \equiv \big[\mathbf{u} \text{ is terminal and } V = \{Y_q \mid q \in F\} \text{ and } h = h_0\big], \tag{7.2}$$

where h_0 is the zero-error projection function (5.6).

The I/O requirement ϕ of Φ is defined by

$$\phi(\mathbf{u}, \mathbf{v}) \equiv \psi_1 \text{ and } \psi_2, \tag{7.3}$$

where ψ_1 and ψ_2 are the formulas

$$\psi_1 \equiv \big[M \text{ accepts } w(\mathbf{u}) \implies (\forall t > \tau(\mathbf{u}))(\exists Y \in V)\mathbf{v}(t)(Y) = 1\big], \tag{7.4}$$

$$\psi_2 \equiv \big[M \text{ rejects } w(\mathbf{u}) \implies (\forall t > \tau(\mathbf{u}))(\forall Y \in V)\mathbf{v}(t)(Y) = 0\big]. \tag{7.5}$$

The two parts ψ_1 and ψ_2 of the I/O requirement simply 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.

At this time 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: a signal conditioning module from chapter 6 that cleans up the input, and a module responsible for the NFA logic. We begin by defining the I/O CRN that computes the main logic of the NFA, and later we join this I/O CRN with the signal conditioning module.

Construction 7.1. Given an NFA $M = (Q, \Sigma, \Delta, I, F)$ and strictly positive real numbers ϵ and $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$, 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 (7.1) 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

$$X_r^* + Z_q \xrightarrow{k_1} X_r^* + \overline{Z}_q. \tag{7.6}$$

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

$$X_a^* + Y_q + \overline{Z}_r \xrightarrow{k_1} X_a^* + Y_q + Z_r. \tag{7.7}$$

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

$$X_c^* + Z_q + \overline{Y}_q \xrightarrow{k_2} X_c^* + Z_q + Y_q, \tag{7.8}$$

$$X_c^* + \overline{Z}_q + Y_q \xrightarrow{k_2} X_c^* + \overline{Z}_q + \overline{Y}_q. \tag{7.9}$$

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

$$2Y_q + \overline{Y}_q \xrightarrow{k_2} 3Y_q \tag{7.10}$$

$$2\overline{Y}_q + Y_q \xrightarrow{k_2} 3\overline{Y}_q. \tag{7.11}$$

   Note these reactions are an implementation of the termolecular signal restoration algorithm analyzed in section 4.1.

The rate constants k_1 and k_2 are defined by

$$k_1 = \frac{30|Q|}{\epsilon - \delta_2 - \delta_3}, \tag{7.12}$$

$$k_2 = 18 \log(\frac{20|Q|}{\epsilon - \delta_2 - \delta_3}). \tag{7.13}$$

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 (7.6). 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 (7.7). Therefore, after this pulse is processed, the portal species Z_q will be close to 1 if and only if $q \in \widehat{\Delta}(I, wa)$ and close to 0 otherwise.

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 (7.8) and (7.9). 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 (7.10) and (7.11) are present to prevent the values stored in the state species from degrading over time. These reactions ensure that in the absence of a symbol event, the state will remain valid indefinitely.

We now add the signal conditioning module in the following construction. Recall that our set of input species 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 signal conditioning I/O CRN for each input species, so that every input species is conditioned to be an approximate square wave. Therefore, the I/O CRN that we will actually prove satisfies the requirement is shown in the following construction.

Construction 7.2. Given NFA $M = (Q, \Sigma, \Delta, I, F)$ and strictly positive real numbers ϵ and $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$, we define the family of I/O CRNs $\mathcal{N} = \mathcal{N}(M, \epsilon, \boldsymbol{\delta})$ by

$$\mathcal{N} = \{N^*\} \cup \left\{N^{(X_a)} \mid a \in \Sigma \cup \{r, c\}\right\}, \tag{7.14}$$

where $N^* = N^*(M, \epsilon, \boldsymbol{\delta})$ is constructed according to Construction 7.1 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 6.1 where

$$\gamma = \frac{\epsilon - \delta_2 - \delta_3}{(34|Q|)^4} \tag{7.15}$$

and $\boldsymbol{\delta}^* = (\delta_1, 0, \delta_3, \delta_4)$.

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

$$N = \bigsqcup \mathcal{N}. \tag{7.16}$$

We note that N from Construction 7.2 is indeed an I/O CRN because the family $\mathcal{N}$ is a *modular* family of I/O CRNs, and we also note that the set of input species U matches equation (7.1) defined earlier.

Theorem 7.3. If $M = (Q, \Sigma, \Delta, I, F)$ is an NFA and ϵ, $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$ are strictly positive real numbers satisfying

$$\delta_1, \delta_2, \delta_3, \delta_4 < \frac{1}{20}, \tag{7.17}$$

$$\delta_2 + \delta_3 < \epsilon, \tag{7.18}$$

and $N = N(M, \epsilon, \boldsymbol{\delta})$ is constructed according to Construction 7.2, then

$$N \models_{\epsilon}^{\boldsymbol{\delta}} \Phi(M). \tag{7.19}$$

The rest of this chapter is devoted to proving the above theorem. The proof is rather extensive and split up into many lemmas.

Assume the hypothesis of Theorem 7.3. We begin by constructing an initial state for N that we will use to simulate M. Since N consists of signal conditioning I/O CRNs from chapter 6, we want this initial state of N to initialize each of its signal conditioning

modules according to Construction 6.1. Therefore, let $N^{(X)} = (U^{(X)}, R^{(X)}, S^{(X)})$ be the signal conditioning module of N for the input species $X \in U$ and let $\mathbf{x}_0^{(X)}$ be the initial state of $N^{(X)}$ constructed according to Construction 6.1. 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

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

Therefore the initial state $\mathbf{x}_0$ ensures that every signal conditioning module $N^{(X)}$ is initialized properly . The initial state $\mathbf{x}_0$ is also defined such that the set of state species encode the set of start states I and the portal species Z_q 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 $q \in Q$.

It is important to notice that each signal conditioning module $N^{(X)} = N^{(X)}(\frac{1}{2}, \gamma, \boldsymbol{\delta}^*)$ is constructed with the parameter $\frac{1}{2}$ corresponding to the delay introduced by the module, γ (which is defined in Construction 7.2) corresponding to how cleaned up the signal becomes, and $\boldsymbol{\delta}^* = (\delta_1, 0, \delta_3, \delta_4)$ being the robustness parameters[2].

Observation 7.4. $N, \mathbf{x}_0 \models_{\gamma}^{\boldsymbol{\delta}^*} \Phi^{(X)}$ for each $X \in U$ where $\Phi^{(X)} = \Phi^{(X)}(\frac{1}{2})$ is the signal conditioning requirement from chapter 6.

Proof. This follows from Theorem 6.2 which says that $N^{(X)}, \mathbf{x}_0^{(X)} \models_{\gamma}^{\boldsymbol{\delta}^*} \Phi^{(X)}$ along with the fact that N is a modular composition I/O CRNs which includes $N^{(X)}$ for each $X \in U$. □

This observation simply states that the signal conditioning modules satisfy their requirements. Therefore within $\frac{1}{2}$ time, each module will clean up an input event with

[2]The reason $\boldsymbol{\delta}^*$ has a measurement function perturbation of zero is because its output signal is not being measured but instead is being passed to the I/O CRN N^*.

up to δ_1 amount of error to have at most γ amount of error. This is an important step for N^* to appropriately compute the logic of the NFA because N^* uses the preprocessed input species as catalysts for the computation.

We now enumerate the ODEs generated by the I/O CRN N. Using the mass action function (5.11), for each $q \in Q$, the ODEs of the species $Y_q, Z_q, \overline{Y}_q, \overline{Z}_q$ of N are

$$\frac{dy_q}{dt} = k_2 x_c^* z_q \overline{y}_q - k_2 x_c^* \overline{z}_q y_q + k_2 y_q^2 \overline{y}_q - k_2 y_q \overline{y}_q^2, \tag{7.20}$$

$$\frac{dz_q}{dt} = -k_1 x_r^* z_q + \sum_{(s,a,q)\in\Delta} k_1 x_a^* y_s \overline{z}_q, \tag{7.21}$$

$$\frac{d\overline{y}_q}{dt} = -\frac{dy_q}{dt}, \tag{7.22}$$

$$\frac{d\overline{z}_q}{dt} = -\frac{dz_q}{dt}, \tag{7.23}$$

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 always a constant and the sum of the concentrations of Z_q and $\overline{Z}_q$ is always a constant. Unfortunately, we cannot assume that these sums are equal to 1 (even though that is how they were initialized) because we must take into account the initial state perturbation. Therefore, for each $q \in Q$ we define the constants $p(Y_q)$ and $p(Z_q)$ to be the sums of the concentrations of the corresponding state species and portal species, respectively.

We prove that $N, \mathbf{x}_0 \models^{\boldsymbol{\delta}}_{\epsilon} \Phi$ by showing that N and $\mathbf{x}_0$ robustly satisfy a family of weaker requirements. To formally state these new requirements, a bit 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.

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

$$\alpha_w(\mathbf{u}, V, h) \equiv \big[\alpha(\mathbf{u}, Y_F, h) \text{ and } w(\mathbf{u}) = w \text{ and } V = Y_Q\big], \tag{7.24}$$

and where ϕ_w is defined by

$$\phi(\mathbf{u}, \mathbf{v}) \equiv\ (\forall t \geq \tau(\mathbf{u}))\big[Y_Q \text{ encodes } \widehat{\Delta}(I, w(\mathbf{u})) \text{ in } \mathbf{v}(t)\big]. \tag{7.25}$$

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 a complete encoding of $\widehat{\Delta}(I, w)$.

Lemma 7.5. If η is a strictly positive real number such that $\eta < \epsilon - \delta_2 - \delta_3$ and $N, \mathbf{x}_0 \models_\eta^{\boldsymbol{\delta}^*} \Phi_w$ for all $w \in \Sigma^*$, then

$$N, \mathbf{x}_0 \models_\epsilon^{\boldsymbol{\delta}} \Phi(M).$$

Proof. 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_1, δ_2)-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_3-close to $\mathbf{x}_0$, let $\widehat{N}$ be δ_4-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 $\widehat{N}_{\hat{\mathbf{c}}, \hat{\mathbf{x}}_0}$ is ϵ-close to a signal $\mathbf{v} \in C[V]$ such that $\Phi(\mathbf{u}, \mathbf{v})$ is satisfied.

By the hypothesis, we know that $N, \mathbf{x}_0 \models_\eta^{\boldsymbol{\delta}^*} \Phi_w$. It follows that Y_Q η-encodes $\widehat{\Delta}(I, w)$ in $\widehat{N}_{\hat{\mathbf{c}}_w, \hat{\mathbf{x}}_0}(t)$ 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 $\widehat{N}_{\hat{\mathbf{c}}_w, \hat{\mathbf{x}}_0}(t)(Y_q) > p(Y_q) - \eta$ for all $t \geq \tau(\mathbf{u})$.

Since the perturbed initial state $\hat{\mathbf{x}}_0$ is δ_3-close to $\mathbf{x}_0$, it follows that $p(Y_q) > 1 - \delta_3$. Similarly, the measurement function can at most introduce δ_2 error, therefore

$$\widehat{N}_{\hat{\mathbf{c}}_w,\hat{\mathbf{x}}_0}(t)(Y_q) > 1 - \delta_2 - \delta_3 - \eta$$

for all $t \geq \tau(\mathbf{u})$. Since $\eta < \epsilon - \delta_2 - \delta_3$, it follows that $\widehat{N}_{\hat{\mathbf{c}}_w,\hat{\mathbf{x}}_0}(t)(Y_q) > 1 - \epsilon$, and since $Y_q \in V$, the function $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ 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 $t \geq \tau(\mathbf{u})$, $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}(t)(Y) < \eta + \delta_2 < \epsilon$. Therefore $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to satisfying ψ_2 of ϕ. It follows that $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is ϵ-close to a function $\mathbf{v} \in C[V]$ such that the I/O requirement $\phi(\mathbf{u}, \mathbf{v})$ holds. Therefore $N, \mathbf{x}_0 \models_{\epsilon}^{\boldsymbol{\delta}} \Phi$. □

To finish the proof of Theorem 7.3, it now suffices to show that $N, \mathbf{x}_0 \models_{\eta}^{\boldsymbol{\delta}^*} \Phi_w$ holds for all $w \in \Sigma^*$ for some $\eta < \epsilon - \delta_2 - \delta_3$. We prove this via induction over the strings $w \in \Sigma^*$. For the remainder of the proof, we explicitly define the constant

$$\eta = \frac{\epsilon - \delta_2 - \delta_3}{(80|Q|)^2} \tag{7.26}$$

noting that indeed $\eta < \epsilon - \delta_2 - \delta_3$.

Before we begin the induction, we first prove some necessary inequalities used throughout the proof. These inequalities are left in their most general form so that they can easily be referenced when needed. We also note that our specification for the rate constants k_1 and k_2 in Construction 7.1 and the constraints on the parameters ϵ and $\boldsymbol{\delta}$ in Theorem 7.3 were selected so as to satisfy these inequalities. Tighter bounds on these parameters may exist that also satisfy these inequalities, but in this chapter we aim only to demonstrate that our NFA construction is indeed robust.

Observation 7.6. The following inequalities hold where the constant $n^* = \frac{1}{20}$.

$$\frac{\delta_3(1+\delta_3)}{(1-\delta_3)^2} < \frac{\left(1-\frac{\delta_4}{k_2}\right)^2}{8\left(1+\frac{\delta_4}{k_2}\right)} \tag{7.27}$$

$$\gamma < \frac{\eta\left(1-\frac{\delta_4}{k_2}\right)^2(1-\delta_3)^2}{16\left(1+\frac{\delta_4}{k_2}\right)(1+\delta_3)} \tag{7.28}$$

$$\frac{8(1+\delta_3)\delta_3}{7\left(\frac{1-\delta_3}{2}\left(1-\frac{\delta_4}{k_2}\right)-\frac{9\delta_3}{8}\right)} < 1 \tag{7.29}$$

$$\gamma < \frac{\eta}{4k_1|Q|(1+\delta_3)\left(1+\frac{\delta_4}{k_1}\right)} \tag{7.30}$$

$$\frac{\gamma(k_1+\delta_4)(1+\delta_3)}{(k_1-\delta_4)(1-\gamma)(1-\eta^*)} < \frac{\eta}{4} \tag{7.31}$$

$$(k_1+\delta_4)\gamma(1+\delta_3) < \frac{\eta}{12} \tag{7.32}$$

$$\frac{\left(1-\frac{\delta_4}{k_2}\right)(1-\gamma)(1-\delta_3-\eta^*)-\frac{1}{6}(1+\delta_3)^2\left(1+\frac{\delta_4}{k_2}\right)^3}{\left(1-\frac{\delta_4}{k_2}\right)(1-\gamma)(1-\delta_3-\eta^*)+\left(1+\frac{\delta_4}{k_2}\right)(1+\delta_3)\eta^*} > \frac{2}{3}\left(1+\frac{\delta_4}{k_2}\right) \tag{7.33}$$

Proof. It is routine to verify that these inequalities hold. The equations (7.12), (7.13), (7.15), and (7.26) specify the constants k_1, k_2, γ, and η, respectively, and the constraints on ϵ and $\delta_1, \delta_2, \delta_3, \delta_4$ can be found in Theorem 7.3. □

We now prove the base case of our induction—an encoding of the empty string λ. This base case is one of the most difficult proofs of the main theorem because of the complexity of the signal restoration algorithm. We simply need to show that the signal restoration algorithm is capable of keeping the encoding of the set I indefinitely, but as we will see, this is not trivial even with our previous analysis from chapter 4.

Lemma 7.7. $N, \mathbf{x}_0 \models_{\eta}^{\boldsymbol{\delta}^*} \Phi_\lambda$.

Proof. 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 $(\delta_1, 0)$-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_3-close to $\mathbf{x}_0$, and let $\widehat{N}$ be δ_4-close to N. It suffices to show that $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is

η-close to a function $\mathbf{v} \in C[V]$ which satisfies $\phi_\lambda(\mathbf{u}, \mathbf{v})$. Therefore we must show that for all $t \geq 1$, Y_Q η-encodes I at time t.

We begin by showing that for $q \in I$, $y_q(t) > p(Y_q) - \eta$ for all $t \geq 1$, therefore we assume that $q \in I$. The ODE for Y_q in $\widehat{N}$ is

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

where $\hat{k}_2$ is a function of t that is a perturbation of the true rate constant k_2.

Since $\mathbf{u}$ encodes the empty string, it contains no symbol events, and since $\hat{\mathbf{u}}$ is δ_1-close to $\mathbf{u}$, the input species X_c has concentration less than δ_1 for all $t \in [0, \infty)$. Therefore the preprocessed input species X_c^* has concentration less than γ for all $t \geq \frac{1}{2}$. Since the species X_c^* is initialized to 0 and $\hat{\mathbf{x}}_0$ is δ_3-close to $\mathbf{x}_0$, it is easy to show that X_c^* has concentration less than δ_3 for all $t \in [0, \frac{1}{2}]$.

We first show that during the time interval $[0, \frac{1}{2}]$ the concentration of Y_q is unaffected by noise from unwanted reactions. We do this by assuming the worst possible scenarios and bounding how much Y_q can change even under those conditions. We begin by noting that the ODE for Y_q during the interval $[0, \frac{1}{2}]$ is bounded by

$$\frac{dy_q}{dt} > \hat{k}_2 y_q^2 \overline{y}_q - \hat{k}_2 y_q \overline{y}_q^2 - \hat{k}_2 \delta_3 \overline{z}_q y_q.$$

Since $\hat{k}_2$ is δ_4-close to the constant k_2, we have the bound

$$\frac{dy_q}{dt} > (k_2 - \delta_4) y_q^2 \overline{y}_q - (k_2 + \delta_4) y_q \overline{y}_q^2 - (k_2 + \delta_4) \delta_3 \overline{z}_q y_q.$$

The concentration of $\overline{Z}_q$ is bounded by $p(Z_q)$, so

$$\frac{dy_q}{dt} > (k_2 - \delta_4) y_q^2 \overline{y}_q - (k_2 + \delta_4) y_q \overline{y}_q^2 - (k_2 - \delta_4) \delta_3 p(Z_q) y_q.$$

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

$$\frac{dy_q}{dt} > (k_2 - \delta_4)y_q^2(p(Y_q) - y_q) - (k_2 + \delta_4)y_q(p(Y_q) - y_q)^2 - (k_2 - \delta_4)\delta_3 p(Z_q)y_q.$$

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

$$a = k_2 - \delta_4, \qquad b = k_2 + \delta_4, \qquad c = (k_2 - \delta_4)p(Z_q)\delta_3, \qquad p = p(Y_q),$$

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 equation (4.16) from the termolecular signal restoration algorithm. This means that if the inequality $c < \frac{p^2a^2}{4(a+b)}$ holds, we can make use of Theorem 4.2 to bound the concentration of Y_q during the interval $[0, \frac{1}{2}]$.

To show that $c < \frac{p^2a^2}{4(a+b)}$, it suffices to show that $\frac{4(a+b)c}{p^2a^2} < 1$. By expanding the constants, we know

$$\frac{4(a+b)c}{p^2a^2} = \frac{4(2k_2)(k_2 + \delta_4)p(Z_q)\delta_3}{p(Y_q)^2(k_2 - \delta_4)^2} = 8\delta_3 \frac{\left(1 + \frac{\delta_4}{k_2}\right)}{\left(1 - \frac{\delta_4}{k_2}\right)^2}\left(\frac{p(Z_q)}{p(Y_q)^2}\right).$$

Since the input can be perturbed by at most δ_3, we know $\frac{p(Z_q)}{p(Y_q)^2} < \frac{1+\delta_3}{(1-\delta_3)^2}$, so

$$\frac{4(a+b)c}{p^2a^2} < \frac{8\delta_3(1 + \delta_3)}{(1 - \delta_3)^2} \frac{\left(1 + \frac{\delta_4}{k_2}\right)}{\left(1 - \frac{\delta_4}{k_2}\right)^2}.$$

It then follows from the inequality (7.27) from Observation 7.6 that $\frac{4(a+b)c}{p^2a^2} < 1$.

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 be the constant constructed with a, b, c, and p according to Theorem 4.2. Recall that this constant corresponds to the unstable equilibrium point of the signal restoration algorithm. Therefore, if $y_q(0) > E_1$, then Y_q will converge to the high equilibrium point, whereas if $y_q(0) < E_1$ then it would converge to 0.

By Observation 4.3 and the fact that $c < \frac{p^2a^2}{4(a+b)}$, we know that

$$E_1 < p\left(\frac{b}{a+b}\right) + \frac{2c}{pa} < p\left(\frac{b}{a+b}\right) + \frac{2}{pa}\left(\frac{p^2a^2}{4(a+b)}\right) = p - \frac{p}{2}\left(\frac{a}{a+b}\right).$$

By expanding the constants a, b, and p, we obtain

$$E_1 < p(Y_q) - \frac{p(Y_q)}{2}\left(\frac{k_2 - \delta_4}{2k_2}\right) < p(Y_q) - \frac{1-\delta_3}{4}\left(1 - \frac{\delta_4}{k_2}\right).$$

The inequality (7.27) from Observation 7.6 tells us

$$\delta_3 < \frac{(1-\delta_3)^2\left(1-\frac{\delta_4}{k_2}\right)^2}{8(1+\delta_3)\left(1+\frac{\delta_4}{k_2}\right)} < \frac{1-\delta_3}{4}\left(1-\frac{\delta_4}{k_2}\right),$$

therefore it is clear that $E_1 < p(Y_q) - \delta_3 < y_q(0)$.

Since E_1 is the decision point of the signal restoration algorithm, Theorem 4.2 tells us that the concentration of Y_q will converge away from the constant E_1 to the constant E_2. It is now easy to show that for all $t \in [0, \frac{1}{2}]$, $y_q(t) > y_q(0)$. Therefore we know that $y_q(\frac{1}{2}) > y_q(0)$.

We now focus on the behavior of Y_q during the time interval $[\frac{1}{2}, \infty)$. Recall that the signal conditioner for X_c is guaranteed to activate at time $\frac{1}{2}$. Therefore, for all $t \geq \frac{1}{2}$ we

know that $x_c^*(t) < \gamma$. Taking this into account, we can bound the ODE for Y_q during the interval $[\frac{1}{2}, \infty)$ by

$$\begin{aligned}\frac{dy_q}{dt} &> \hat{k}_2 y_q^2 \overline{y}_q - \hat{k}_2 y_q \overline{y}_q^2 - \hat{k}_2 \gamma \overline{z}_q y_q, \\ &> (k_2 - \delta_4) y_q^2 (p(Y_q) - y_q) - (k_2 + \delta_4) y_q (p(Y_q) - y_q)^2 - (k_2 + \delta_4)\gamma p(Z_q) y_q.\end{aligned}$$

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

$$a = k_2 - \delta_4, \qquad b = k_2 + \delta_4, \qquad c = (k_2 + \delta_4)p(Z_q)\gamma, \qquad p = p(Y_q),$$

then we can rewrite the above ODE as

$$\frac{dy_q}{dt} > a y_q^2 (p - y_q) - b y_q (p - y_q)^2 - c y_q.$$

We again show that $c < \frac{p^2 a^2}{4(a+b)}$ holds with these new constants with

$$\frac{4(a+b)c}{p^2 a^2} = \frac{4(2k_2)(k_2 + \delta_4)p(Z_q)\gamma}{p(Y_q)^2 (k_2 - \delta_4)^2} < 8\gamma \frac{\left(1 + \frac{\delta_4}{k_2}\right)}{\left(1 - \frac{\delta_4}{k_2}\right)^2} \left(\frac{1 + \delta_3}{(1 - \delta_3)^2}\right),$$

and therefore by inequality (7.28) from Observation 6.12 we know that

$$\frac{4(a+b)c}{p^2 a^2} < \frac{1}{2}. \tag{7.34}$$

We now redefine the constants E_1 and E_2 to be constructed with the new constants a, b, c, and p according to Theorem 4.2 as before. It is easy to show that $E_1 < y_q(\frac{1}{2})$, so we know 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 $p(Y_q) - \eta$. First we note that

$$\frac{2c}{pa} = \frac{2\gamma(k_2 + \delta_4)p(Z_q)}{p(Y_q)(k_2 - \delta_4)} > p(Y_q) - 2\gamma \left(\frac{1 + \frac{\delta_4}{k_2}}{1 - \frac{\delta_4}{k_2}}\right)\left(\frac{1 + \delta_3}{1 - \delta_3}\right),$$

and so with inequality (7.28) we know that

$$\frac{2c}{pa} > \frac{\eta}{8}. \tag{7.35}$$

It follows from Observation 4.3 that

$$E_2 > p - \frac{2c}{pa} > p(Y_q) - \frac{\eta}{8} > p(Y_q) - \eta.$$

Therefore we can use Corollary 4.5 to bound the amount of time it takes the signal restoration reactions to restore the concentration of Y_q to $p(Y_q) - \eta$ from $p(Y_q) - \delta_3$ with

$$t < \frac{a + b}{abp^2(1 - c\frac{4(a+b)}{p^2a^2})} \log u,$$

where $u = \frac{(p-\eta-E_1)(E_2-p+\delta_3)}{(p-\delta_3-E_1)(E_2-p+\eta)}$ and where t in this case is the amount of time required for the concentration of Y_q to reach $p(Y_q) - \eta$. This is a complicated equation, therefore we simplify some of its parts.

It follows from equation (7.34) that

$$\frac{a + b}{abp^2(1 - c\frac{4(a+b)}{p^2a^2})} < \frac{2(a + b)}{abp^2}.$$

Since $p > E_2$, we can also bound the inside of the logarithm by

$$u = \frac{(p - \eta - E_1)(E_2 - p + \delta_3)}{(p - \delta_3 - E_1)(E_2 - p + \eta)} < \frac{p\delta_3}{(p - \delta_3 - E_1)(E_2 - p + \eta)},$$

and by Observation 4.3 and equation (7.35), we know that

$$u < \frac{p\delta_3}{(p(\frac{a}{a+b}) - \frac{2c}{pa} - \delta_3)(\frac{7}{8}\eta)} < \frac{8p\delta_3}{7\eta(p(\frac{a}{a+b}) - \frac{\eta}{8} - \delta_3)}$$
$$< \frac{8p(Y_q)\delta_3}{7\eta\left(\frac{p(Y_q)}{2}\left(1 - \frac{\delta_4}{k_2}\right) - \frac{\delta_3}{8} - \delta_3\right)} = \frac{8(1+\delta_3)\delta_3}{7\eta\left(\frac{1-\delta_3}{2}\left(1 - \frac{\delta_4}{k_2}\right) - \frac{9\delta_3}{8}\right)},$$

and by the inequality (7.29) from Observation 7.6 we know that $u < \frac{1}{\eta}$.

This simplifies our bound for the amount of time to reach concentration $p(Y_q) - \eta$ to

$$t < \frac{2(a+b)}{abp^2}\log\frac{1}{\eta} = \frac{-4k_2\log\eta}{(k_2-\delta_4)(k_2+\delta_4)p(Y_q)^2} < \frac{-4\log\eta}{k_2\left(1-\frac{\delta_4}{k_2}\right)\left(1+\frac{\delta_4}{k_2}\right)(1-\delta_3)^2}.$$

Using the definition of k_2 from equation (7.13), it is easy to verify that

$$k_2 > \frac{-8\log\eta}{\left(1-\frac{\delta_4}{k_2}\right)\left(1+\frac{\delta_4}{k_2}\right)(1-\delta_3)^2}.$$

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

This finishes one half of the proof, namely, that if $q \in I$ that the I/O CRN N robustly keeps the value of Y_q η-close to $p(Y_q)$. It remains to be shown that for $q \in Q \setminus I$, $y_q(t) < \eta$ for all $t \geq 1$. This follows by the symmetry of Y_q and $\overline{Y}_q$ and is omitted. □

The above lemma shows that the base case of our induction holds. To complete the proof of the theorem, it suffices to show that

$$N, \mathbf{x}_0 \models_\eta^{\boldsymbol{\delta}^*} \Phi_w \implies N, \mathbf{x}_0 \models_\eta^{\boldsymbol{\delta}^*} \Phi_{wa}, \tag{7.36}$$

where $w \in \Sigma^*$ and $a \in \Sigma$.

For the rest of this chapter, let $w \in \Sigma^*$ and $a \in \Sigma$ and assume the inductive hypothesis $N, \mathbf{x}_0 \models_\eta^{\boldsymbol{\delta}^*} \Phi_w$ 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

$(\delta_1, 0)$-close to $\mathbf{c}$, let $\hat{\mathbf{x}}_0$ be δ_3-close to $\mathbf{x}_0$, and let $\widehat{N}$ be δ_4-close to N. It suffices to show that $\widehat{N}_{\hat{\mathbf{c}},\hat{\mathbf{x}}_0}$ is η-close to a function $\mathbf{v} \in C[V]$ which satisfies $\phi_{wa}(\mathbf{u}, \mathbf{v})$. Therefore 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 $\tau(\mathbf{u}) = b + 12$. We also know by the inductive hypothesis that the following observation holds.

Observation 7.8. *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 $\mathbf{u}^*(t) = \mathbf{0}$[3] for all $t > b$. Then $w(\mathbf{u}^*) = w$ and $\tau(\mathbf{u}^*) \leq b$. Since $N, \mathbf{x}_0 \models_\eta^{\delta^*} \Phi_w$ by our inductive hypothesis, we know that Y_Q η-encodes $\widehat{\Delta}(I, w)$ 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_1 during the interval $[b, b+8]$, so the values of the species in Y_Q will be maintained by the signal restoration reactions as shown in the proof of Lemma 7.7. □

Observation 7.9. *$x_d^*(b) < \gamma$ for each symbol $d \in \Sigma \cup \{r, c\}$.*

Proof. This follows from the fact that for all $d \in \Sigma \cup \{r, c\}$ the concentration of input X_d is less than δ_1 in the interval $[b - \frac{1}{2}, b]$. Therefore the preprocessed input species X_d has time to drop below γ before time b. □

Lemma 7.10. *Z_Q η-encodes $\emptyset$ at time $b + 4$.*

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

$$\begin{aligned}\frac{dz_q}{dt} &= -\hat{k}_1 x_r^* z_q + \sum_{(s,a,q)\in\Delta} \hat{k}_1 x_a^* y_s \overline{z}_q \\ &< -(k_1 - \delta_4) x_r^* z_q + \sum_{(s,a,q)\in\Delta} (k_1 + \delta_4) x_a^* p(Y_r) \overline{z}_q.\end{aligned}$$

[3] We use $\mathbf{0}$ to denote the state vector containing all zeros.

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

$$\begin{aligned}\frac{dz_q}{dt} &< -(k_1-\delta_4)x_r^* z_q + \sum_{(s,a,q)\in\Delta} p(Y_r)(k_1+\delta_4)\gamma\overline{z}_q \\ &< -(k_1-\delta_4)x_r^* z_q + |Q|(1+\delta_3)(k_1+\delta_4)\gamma\overline{z}_q.\end{aligned}$$

During the interval $[b+1, b+2]$, the species X_r is δ_1-close to one. Therefore during the interval $[b+1.5, b+2]$ the species X_r^* is γ-close to 1. Thus

$$\frac{dz_q}{dt} < -(k_1-\delta_4)(1-\gamma)z_q + |Q|(1+\delta_3)(k_1+\delta_4)\gamma\overline{z}_q.$$

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

$$\hat{f} = |Q|(1+\delta_3)(k_1+\delta_4)\gamma \qquad \hat{b} = (k_1-\delta_4)(1-\gamma), \qquad \hat{p} = p(Z_q),$$

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 6.11. By Lemma 6.12, 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 (7.12), it is easy to show that $k_1 > \delta_4 + \frac{2}{1-\gamma}\log\left(\frac{4(1+\delta_3)}{\eta}\right)$, therefore

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

Equation (7.30) from Observation 7.6 tells us that $\gamma < \frac{\eta}{4k_1|Q|(1+\delta_3)\left(1+\frac{\delta_4}{k_1}\right)}$, therefore $\hat{p}\hat{f} > \frac{\eta}{4}$, 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 $z_q(b+4) < \eta$. □

Lemma 7.11. *Z_Q η^*-encodes $\widehat{\Delta}(I, wa)$ during the interval $[b+8, b+12]$ where $\eta^* = \frac{1}{20}$.*

Proof. We prove this in two steps. First we prove that if $q \in \widehat{\Delta}(I, wa)$, then for all $t \in [b+8, b+12]$ $z_q(t) > p(Z_q) - \eta^*$, and second we prove that if $q \notin \widehat{\Delta}(I, wa)$, then for all $t \in [b+8, b+12]$ $z_q(t) < \eta^*$.

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 (7.7) and Construction 7.1 that computes the transition $(s, a, q) \in \Delta$. Therefore we can bound the ODE corresponding to Z_q by

$$\begin{aligned}
\frac{dz_q}{dt} &= -\hat{k}_1 x_r^* z_q + \sum_{(s,a,q)\in\Delta} \hat{k}_1 x_a^* y_s \overline{z}_q, \\
&> -\hat{k}_1 x_r^* z_q + \hat{k}_1 x_a^* y_s \overline{z}_q, \\
&> -(k_1 + \delta_4) x_r^* z_q + (k_1 - \delta_4) x_a^* y_s \overline{z}_q.
\end{aligned}$$

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

$$\frac{dz_q}{dt} > -(k_1+\delta_4)\gamma z_q + (k_1-\delta_4)(1-\gamma)y_s\overline{z}_q.$$

By Observation 7.8, 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

$$\frac{dz_q}{dt} > -(k_1+\delta_4)\gamma z_q + (k_1-\delta_4)(1-\gamma)(1-\eta)\overline{z}_q.$$

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

$$\hat{f} = (k_1-\delta_4)(1-\gamma)(1-\eta) \qquad \hat{b} = (k_1+\delta_4)\gamma \qquad \hat{p} = p(Z_q),$$

so that

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

Then by Lemma 6.12, 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 (7.12), it is easy to show that $k_1 > \delta_4 + \frac{2}{(1-\gamma)(1-\eta)}\log\left(\frac{4(1+\delta_3)}{\eta}\right)$. Therefore we know that

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

By inequality (7.31) from Observation 7.6, we know that

$$\hat{p}\frac{\hat{b}}{\hat{f}} = p(Z_q)\frac{\gamma(k_1+\delta_4)}{(k_1-\delta_4)(1-\gamma)(1-\eta^*)} < \frac{\gamma(k_1+\delta_4)(1+\delta_3)}{(k_1-\delta_4)(1-\gamma)(1-\eta^*)} < \frac{\eta}{4}.$$

therefore we have the bound

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

Finally, by inequality (7.32) of Observation 7.6, we know that Zq is bounded during the interval $[b+6, b+12]$, Z_q by

$$\frac{dz_q}{dt} > -\hat{b}\hat{p} = -(k_1+\delta_4)\gamma p(Z_q) > -(k_1+\delta_4)\gamma(1+\delta_3) > -\frac{\eta}{12},$$

therefore at most $\frac{\eta}{2}$ of Z_q can be destroyed in this interval. Therefore for all $t \in [b+8, b+12]$, $z_q(t) > p(Z_q) - \eta > p(Z_q) - \eta^*$.

It remains to be shown that if $q \notin \widehat{\Delta}(I, wa)$, then for all $t \in [b+8, b+12]$ $z_q(t) < \eta^*$. 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 $[b+4, b+12]$.

$$\begin{aligned}\frac{dz_q}{dt} &= -\hat{k}_1 x_r^* z_q + \sum_{(s,a,q)\in\Delta} \hat{k}_1 x_a^* y_s \overline{z}_q < \sum_{(s,a,q)\in\Delta} \hat{k}_1 x_a^* y_s \overline{z}_q \\ &< |Q|(k_1+\delta_4)(1+\delta_3)\eta(1+\delta_3) = |Q|(k_1+\delta_4)(1+\delta_3)^2\eta.\end{aligned}$$

Therefore

$$\begin{aligned}z_q(b+12) &< z_q(b+4) + 8|Q|(k_1+\delta_4)(1+\delta_3)^2\eta \\ &< \eta + 8|Q|(k_1+\delta_4)(1+\delta_3)^2\eta < \eta^*.\end{aligned}$$

□

Lemma 7.12. Y_Q β-encodes $\widehat{\Delta}(I, wa)$ at time $b+10$ where $\beta = \frac{1}{20}$.

Proof. Recall that the ODE for Y_q 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 the signal restoration part of the ODE with

$$\begin{aligned} B &> y_q \overline{y}_q \big((k_2 - \delta_4) y_q - (k_2 + \delta_4) \overline{y}_q \big) \\ &= -k_2 y_q \overline{y}_q \left(2\overline{y}_q - p(Y_q) \left(1 - \frac{\delta_4}{k_2} \right) \right). \end{aligned}$$

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

$$B > -\frac{k_2}{6} p(Y_q)^3 \left(1 + \frac{\delta_4}{k_2} \right)^3.$$

Now let $q \in \widehat{\Delta}(I, wa)$. Then by Lemma 7.11, $z_q(t) > p(Z_q) - \eta^*$ 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

$$\begin{aligned} \frac{dy_q}{dt} &> \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q - \frac{k_2}{6} p(Y_q)^3 \left(1 + \frac{\delta_4}{k_2} \right)^3 \\ &= \hat{a}(\hat{p} - y_q) - \hat{b} y_q - \hat{c} \end{aligned}$$

where

$$\begin{aligned} \hat{a} &= k_2 \left(1 - \frac{\delta_4}{k_2} \right) (1 - \gamma)(1 - \delta_3 - \eta^*) \\ \hat{b} &= k_2 \left(1 + \frac{\delta_4}{k_2} \right) (1 + \delta_3) \eta^* \\ \hat{c} &= \frac{k_2}{6} p(Y_q)^3 \left(1 + \frac{\delta_4}{k_2} \right)^3 \\ \hat{p} &= p(Y_q). \end{aligned}$$

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

$$y_q(b+9.75) > \frac{\hat{p}\hat{a}-\hat{c}}{\hat{a}+\hat{b}}\left(1-e^{-(\hat{a}+\hat{b})\frac{1}{4}}\right). \tag{7.37}$$

After expanding the expression $\frac{\hat{p}\hat{a}-\hat{c}}{\hat{a}+\hat{b}}$ we obtain

$$\frac{\hat{p}\hat{a}-\hat{c}}{\hat{a}+\hat{b}} > p(Y_q)\left(\frac{\left(1-\frac{\delta_4}{k_2}\right)(1-\gamma)(1-\delta_3-\eta^*)-\frac{1}{6}(1+\delta_3)^2\left(1+\frac{\delta_4}{k_2}\right)^3}{\left(1-\frac{\delta_4}{k_2}\right)(1-\gamma)(1-\delta_3-\eta^*)+\left(1+\frac{\delta_4}{k_2}\right)(1+\delta_3)\eta^*}\right)$$

By inequality (7.33) from Observation 7.6 we obtain

$$\frac{\hat{p}\hat{a}-\hat{c}}{\hat{a}+\hat{b}} > \frac{2}{3}p(Y_q)\left(1+\frac{\delta_4}{k_2}\right). \tag{7.38}$$

Using the specification of k_2 from equation (7.13), it is easy to show that $k_2 > \delta_4 + \frac{4\log 4}{(1-\gamma)(1-\delta_3-\eta^*)}$. 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 (7.38) into equation (7.37), we obtain

$$y_q(b+9.75) > \frac{2}{3}p(Y_q)\left(1+\frac{\delta_4}{k_2}\right)\left(\frac{3}{4}\right) = \frac{p(Y_q)}{2}\left(1+\frac{\delta_4}{k_2}\right).$$

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

$$\begin{aligned}\frac{dy_q}{dt} &> \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q - k_2 y_q \overline{y}_q\left(2\overline{y}_q - p(Y_q)\left(1-\frac{\delta_4}{k_2}\right)\right)\\ &= \hat{k}_2 x_c^* z_q \overline{y}_q - \hat{k}_2 x_c^* \overline{z}_q y_q + k_2 y_q \overline{y}_q\left(2y_q - p(Y_q)\left(1+\frac{\delta_4}{k_2}\right)\right).\end{aligned}$$

Since the concentration of Y_q is greater than $\frac{p(Y_q)}{2}\left(1+\frac{\delta_4}{k_2}\right)$ 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 6.12 we obtain the bound

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

Since $\frac{\beta}{2} > \frac{\left(1+\frac{\delta_4}{k_2}\right)(1+\delta_3)^2\eta^*}{\left(1-\frac{\delta_4}{k_2}\right)(1-\gamma)(1-\delta_3-\eta^*)}$,

$$\hat{p}\frac{\hat{b}}{\hat{a}} = p(Y_q)\frac{\left(1+\frac{\delta_4}{k_2}\right)(1+\delta_3)\eta^*}{\left(1-\frac{\delta_4}{k_2}\right)(1-\gamma)(1-\delta_3-\eta^*)} < \frac{\left(1+\frac{\delta_4}{k_2}\right)(1+\delta_3)^2\eta^*}{\left(1-\frac{\delta_4}{k_2}\right)(1-\gamma)(1-\delta_3-\eta^*)} < \frac{\beta}{2}.$$

and since $k_2 > \delta_4 + \frac{4\log\left(\frac{2(1+\delta_3)}{\beta}\right)}{(1-\gamma)(1-\delta_3-\eta^*)}$

$$\hat{p}e^{-\hat{a}\frac{1}{4}} < \frac{\beta}{2},$$

so we have the bound $y_q(b+10) > p(Y_q) - \beta$.

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

Lemma 7.13. For all $t \geq \tau(\mathbf{u})$, Y_Q η-encodes $\widehat{\Delta}(I, wa)$ at time t.

Proof. This lemma holds by a similar argument of the proof of Lemma 7.7. Lemma 7.12 shows that Y_Q β-encodes $\widehat{\Delta}(I, wa)$ at time $b+10$, and the signal restoration reactions restore the concentration to η by time $b+12$ and maintain it indefinitely. □

CHAPTER 8. CONCLUSION

In this thesis, we introduced two new notions for modular development of chemical reaction networks (CRNs): closed sub-CRNs and input/output CRNs. Closed sub-CRNs were defined using a Kuratowski closure operator and proved to be a useful tool for modularly extending and composing CRNs. In chapter 3, we explored the full generality of these extensions and demonstrated that a large class of concentration signals can be constructed using these extensions. Using closed sub-CRNs, it is more natural to design CRNs to produce a specific concentration signal by breaking the signal into components and composing them together. One drawback to these extensions is that some of them produce CRNs that are not well-behaved if their rate constants or initial concentrations are not set precisely. Moreover, many of the extensions are tightly coupled to the underlying structure of their closed sub-CRNs rather than acting on their concentration signals alone.

The I/O CRN model and the notion of robustness presented in chapter 5 require that these drawbacks are avoided. We investigated the capacity for robust computation and modularity of I/O CRNs in chapter 7 and showed that they are capable of robustly simulating any nondeterministic finite automata. This construction consisted of several modules including the CRNs studied in chapters 4 and 6 and demonstrates that I/O CRNs are capable of robustly computing the regular languages.

Future research of modular and robust development of CRNs is likely to include investigating if deterministic CRNs can robustly simulate a Turing machine (with unbounded memory). Under deterministic semantics, it is not even known if CRNs are capable of *unrobust* Turing machine computation. However, understanding exactly what can and

cannot be robustly computed with deterministic CRNs is an important boundary to explore.

Another research direction will include investigating which CRN extension operators can be *robustly* implemented with closed sub-CRNs and I/O CRNs. Many of the most general CRN extension operators demonstrated in chapter 3 are not robust with respect to rate constants and initial states, and developing an analogous extension method which preserves robustness is necessary to make the method practical.

BIBLIOGRAPHY

[1] Horst Alzer. On some inequalities for the incomplete gamma function. *Mathematics of Computation*, 66(218):771–778, 1997.

[2] David F. Anderson, Germán A. Enciso, and Matthew D. Johnston. Stochastic analysis of biochemical reaction networks with absolute concentration robustness. *Journal of The Royal Society Interface*, 11(93), 2014.

[3] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006.

[4] Dana Angluin, James Aspnes, and David Eisenstat. Stably computable predicates are semilinear. In *Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing*, pages 292–299. ACM, 2006.

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

[6] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.

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

[8] Rutherford Aris. Prolegomena to the rational analysis of systems of chemical reactions. *Archive for Rational Mechanics and Analysis*, 19(2):81–99, 1965.

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

[10] Zamir Bavel. The source as a tool in automata. *Information and Control*, 18(2):140–155, 1971.

[11] Erik Benson, Abdulmelik Mohammed, Johan Gardell, Sergej Masich, Eugen Czeizler, Pekka Orponen, and Björn Högberg. DNA rendering of polyhedral meshes at the nanoscale. *Nature*, 523(7561):441–444, 2015.

[12] Filippo Bonchi and Damien Pous. Checking NFA equivalence with bisimulations up to congruence. In *Proceedings of the 40th Symposium on Principles of Programming Languages*, pages 457–468. ACM, 2013.

[13] Filippo Bonchi and Damien Pous. Hacking nondeterminism with induction and coinduction. *Communications of the ACM*, 58(2):87–95, 2015.

[14] Luca Cardelli. Morphisms of reaction networks that couple structure to function. *BMC Systems Biology*, 8:84, 2014.

[15] Ho-Lin Chen, Rachel Cummings, David Doty, and David Soloveichik. Speed faults in computation by chemical reaction networks. In *Proceedings of the 28th International Symposium on Distributed Computing*, volume 8784 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2014.

[16] Ho-Lin Chen, David Doty, and David Soloveichik. Deterministic function computation with chemical reaction networks. In *Proceedings of the 18th International Conference on DNA Computing and Molecular Programming*, volume 7433 of *Lecture Notes in Computer Science*, pages 25–42. Springer, 2012.

[17] Ho-Lin Chen, David Doty, and David Soloveichik. Rate-independent computation in continuous chemical reaction networks. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, pages 313–326. ACM, 2014.

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

[19] Juan Cheng, Sarangapani Sreelatha, Ruizheng Hou, Artem Efremov, Ruchuan Liu, Johan R. C. van der Maarel, and Zhisong Wang. Bipedal nanowalker by pure physical mechanisms. *Physical Review Letters*, 109:238104, 2012.

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

[21] Rachel Cummings, David Doty, and David Soloveichik. Probability 1 computation with chemical reaction networks. In *Proceedings of the 20th International Conference on DNA Computing and Molecular Programming*, volume 8727 of *Lecture Notes in Computer Science*, pages 37–52. Springer, 2014.

[22] Domitilla Del Vecchio and Richard M Murray. *Biomolecular Feedback Systems*. Princeton University Press, 2014.

[23] David Doty. Timing in chemical reaction networks. In *Proceedings of the 25th Symposium on Discrete Algorithms*, pages 772–784, 2014.

[24] David Doty, Jack H. Lutz, Matthew J. Patitz, Scott M. Summers, and Damien Woods. Random number selection in self-assembly. In *Proceedings of the 8th International Conference on Unconventional Computation*, volume 5715 of *Lecture Notes in Computer Science*, pages 143–157. Springer, 2009.

[25] Shawn M. Douglas, Ido Bachelet, and George M. Church. A logic-gated nanorobot for targeted transport of molecular payloads. *Science*, 335(6070):831–834, 2012.

[26] Samuel J. Ellis, Eric R. Henderson, Titus H. Klinge, James I. Lathrop, Jack H. Lutz, Robyn R. Lutz, Divita Mathur, and Andrew S. Miner. Automated requirements analysis for a molecular watchdog timer. In *Proceedings of the 29th International Conference on Automated Software Engineering*, pages 767–778. ACM, 2014.

[27] Samuel Jay Ellis. Designing a molecular watchdog timer for safety critical systems. Master's thesis, Iowa State University, 2014.

[28] Irving Robert Epstein and John Anthony Pojman. *An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves, Patterns, and Chaos*. Oxford University Press, 1998.

[29] Péter Érdi and János Tóth. *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models*. Manchester University Press, 1989.

[30] Walter Gautschi. The incomplete gamma functions since Tricomi. In Francesco Giacomo Tricomi, editor, *Tricomi's Ideas and Contemporary Applied Mathematics*, volume 147, pages 203–237. Accademia Nazionale dei Lincei, 1998.

[31] Daniel T Gillespie. The deterministic limit of stochastic chemical kinetics. *The Journal of Physical Chemistry B*, 113(6):1640–1644, 2009.

[32] Daniel S. Graça, Manuel L. Campagnolo, and Jorge Buescu. Computability with polynomial differential equations. *Advances in Applied Mathematics*, 40(3):330–349, 2008.

[33] Jeremy Gunawardena. Chemical reaction network theory for in-silico biologists, 2003. http://www.jeremy-gunawardena.com/papers.

[34] Vera Hárs and János Tóth. On the inverse problem of reaction kinetics. In *Colloquia Mathematica Societatis János Bolyai, 30: Qualitative Theory of Differential Equations*, pages 363–379, 1981.

[35] Monika Heiner, David Gilbert, and Robin Donaldson. Petri nets for systems and synthetic biology. In Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016, pages 215–264. Springer, 2008.

[36] Thomas A Henzinger and Jean-François Raskin. The equivalence problem for finite automata: technical perspective. *Communications of the ACM*, 58(2):86–86, 2015.

[37] Thomas Hinze, Raffael Fassler, Thorsten Lenser, and Peter Dittrich. Register machine computations on binary numbers by oscillating and catalytic chemical reactions modelled using mass-action kinetics. *International Journal of Foundations of Computer Science*, 20(3):411–426, 2009.

[38] Yonggang Ke, Luvena L. Ong, William M. Shih, and Peng Yin. Three-dimensional structures self-assembled from DNA bricks. *Science*, 338(6111):1177–1183, 2012.

[39] Titus H. Klinge. Robust signal restoration in chemical reaction networks. In *Proceedings of the 3rd Annual International Conference on Nanoscale Computing and Communication*. ACM, 2016. To appear.

[40] Titus H. Klinge, James I. Lathrop, and Jack H. Lutz. Robust biomolecular finite automata. Technical Report 1505.03931, arXiv.org e-Print archive, 2015.

[41] Wlodzimierz Klonowski. Simplifying principles for chemical and enzyme reaction kinetics. *Biophysical Chemistry*, 18(2):73–87, 1983.

[42] M Korzuhin. Oscillatory processes in biological and chemical systems, 1967.

[43] S. Rao Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *Proceedings of the 14th ACM Symposium on Theory of Computing*, pages 267–281. ACM, 1982.

[44] Dexter Kozen. *Automata and Computability*. Springer, 1997.

[45] Thomas G. Kurtz. The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics*, 57(7):2976–2978, 1972.

[46] Matthew R Lakin, Simon Youssef, Filippo Polo, Stephen Emmott, and Andrew Phillips. Visual DSD: a design and analysis tool for DNA strand displacement systems. *Bioinformatics*, 27(22):3211–3213, 2011.

[47] Gábor Lente. *Deterministic Kinetics in Chemistry and Systems Biology: The Dynamics of Complex Reaction Networks*. Springer, 2015.

[48] Marcelo O. Magnasco. Chemical kinetics is Turing universal. *Physical Review Letters*, 78(6):1190–1193, 1997.

[49] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[50] B. O. Nash. Reachability problems in vector addition systems. *American Mathematical Monthly*, 80(3):292–295, 1973.

[51] Lulu Qian, David Soloveichik, and Erik Winfree. Efficient Turing-universal computation with DNA polymers. In Yasubumi Sakakibara and Yongli Mi, editors, *Proceedings of the 16th International Conference on DNA Computing and Molecular Programming*, volume 6518 of *Lecture Notes in Computer Science*, pages 123–140. Springer, 2011.

[52] Lulu Qian and Erik Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–1201, 2011.

[53] Lulu Qian and Erik Winfree. A simple DNA gate motif for synthesizing large-scale circuits. *Journal of the Royal Society Interface*, 8(62):1281–1297, 2011.

[54] Lulu Qian, Erik Winfree, and Jehoshua Bruck. Neural network computation with DNA strand displacement cascades. *Nature*, 475(7356):368–372, 2011.

[55] Michael O Rabin and Dana Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

[56] Paul W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[57] Nicholas Schiefer and Erik Winfree. Universal computation and optimal construction in the chemical reaction network-controlled tile assembly model. In *Proceedings of the 21st International Conference on DNA Computing and Molecular Programming*, pages 34–54. Springer, 2015.

[58] Nadrian C. Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Biology*, 99(2):237 – 247, 1982.

[59] Jong-Shik Shin and Niles A Pierce. A synthetic dna walker for molecular transport. *Journal of the American Chemical Society*, 126(35):10834–10835, 2004.

[60] Guy Shinar and Martin Feinberg. Structural sources of robustness in biochemical reaction networks. *Science*, 327(5971):1389–1391, 2010.

[61] Wagish Shukla and Arun K. Srivastava. A topology for automata: A note. *Information and Control*, 32(2):163–168, 1976.

[62] David Soloveichik. Robust stochastic chemical reaction networks and bounded tau-leaping. *Journal of Computational Biology*, 16(3):501–522, 2009.

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

[64] Gerald Teschl. *Ordinary Differential Equations and Dynamical Systems*, volume 140 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

[65] Chris Thachuk, Erik Winfree, and David Soloveichik. Leakless DNA strand displacement systems. In *Proceedings of the 21st International Conference on DNA Computing and Molecular Programming*, pages 133–153. Springer, 2015.

[66] Anthony S. Walsh, HaiFang Yin, Christoph M. Erben, Matthew J. A. Wood, and Andrew J. Turberfield. DNA cage delivery to mammalian cells. *ACS Nano*, 5(7):5427–5432, 2011.

[67] Thomas Wilhelm. Chemical systems consisting only of elementary steps – a paradigma for nonlinear behavior. *Journal of Mathematical Chemistry*, 27(1):71–88, 2000.

[68] Erik Winfree. *Algorithmic self-assembly of DNA*. PhD thesis, California Institute of Technology, 1998.

[69] Bernard Yurke, Andrew J. Turberfield, Allen P. Mills, Friedrich C. Simmel, and Jennifer L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406(6796):605–608, 2000.

[70] David Yu Zhang and Erik Winfree. Control of DNA strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society*, 131(47):17303–17314, 2009.