Robust Biomolecular Finite Automata*

Titus H. Klinge, James I. Lathrop, and Jack H. Lutz
Department of Computer Science
Iowa State University
Ames, IA 50011 USA

Abstract

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

1 Introduction

Molecular programming combines computer science principles with the information processing capabilities of DNA and other biomolecules in order to control the structure and behavior of matter at the nanoscale. Molecular programming has its origin in Seeman's development of DNA nanotechnology in the 1980s [28] (indeed, "molecular programming" and "DNA nanotechnology" are still nearly synonymous), but the field has made progress in the present century at a rate whose increase is reminiscent of Moore's law. The achievements of molecular programming are far too numerous to survey here, but they include the self-assembly of virtually any two- or three-dimensional nanoscale structure that one wants to prescribe [27, 14, 18, 20, 34], DNA strand displacement networks that simulate logic circuits and neural networks [24, 23, 25], and molecular robots that perform various functions while either walking on nanoscale tracks or floating free in solution [29, 33, 38, 9, 13, 11, 37]. All this has been achieved in real laboratory experiments, and applications to synthetic biology, medicine, and computer electronics are envisioned. Theoretical progress includes

*This research was supported in part by National Science Foundation Grant 1247051.

demonstrations that various molecular programming paradigms are, in principle, Turing universal [35, 1, 30, 10, 12, 36], thereby indicating that the full generality and creativity of algorithmic computation may be deployed in molecular and biological arenas.

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

In this paper we give a uniform method for translating nondeterministic finite automata to biochemical reaction networks that implement them robustly. Nondeterministic finite automata (NFAs) are over half a century old [26] and far from Turing universal, but they have many applications and remain an active research topic [4, 19, 5]. Applications of NFAs that are likely to extend to molecular programming include their uses in monitoring and parsing large data streams and in implementing and verifying secure and/or safe communication protocols. Biochemical reaction networks (BRNs) are also at least half a century old [3]. Originally–and still often–called chemical reaction networks (CRNs), their role in molecular programming did not become fully apparent until recently, when Soloveichik, Seelig, and Winfree [31] showed that there is a systematic method for translating an arbitrary BRN, which is an abstract mathematical object, into a set of DNA strands and complexes that simulates the BRN via toehold-mediated strand displacement. This method has been refined and formulated as a compiler [8], and BRNs are now the programming language of choice for many molecular programming investigations. (The fact that this systematic implementation–as opposed to earlier *ad hoc* implementations of individual reaction networks–uses DNA has probably contributed to the CRN-to-BRN terminology shift that seems to be underway.) The two most widely used semantics (operational meanings) for BRNs are deterministic mass action semantics and stochastic mass action semantics. In this paper we use deterministic mass action, which implies that the state of a BRN at any time is determined by the *real-valued concentrations* of its molecular species at that time.

An NFA is a real-time device that reads its input string *sequentially*, left to right, changing states appropriately in response to each symbol prior to reading the next symbol. Accordingly, we translate each NFA to a BRN that receives the NFA’s input string formatted as a continuous time *concentration signal* consisting of concentrations of input species that vary to represent the input string in a natural way. Using the inherent parallelism of chemical kinetics, our BRN implements the NFA in *real time*, processing each input symbol before the next one arrives, and it does so with a number of molecular species that is *linear* in the number of states of the NFA that it implements. Specifically, if the NFA has q states, s symbols, and d transitions, then our BRN has $4q + s + 2$ species and $5q + d$ reactions. The compiler of [8] would then translate this into a DNA strand displacement network consisting of $28q + 4d + 2s + 6$ single strands of DNA. Our translation thus appears to make small NFA implementable in laboratories now and NFAs of modest size implementable in the near future.

Most importantly, our BRN's correct implementation of the NFA is robust with respect to small perturbations of four things, namely, its input signal, the initial concentrations of its species, its output signal, and the rate constants of its reactions (acceptance or rejection of its input string). One key to achieving this robustness is a memory-refresh technique akin to the approximate majority algorithm of [1, 7, 6].

The NFA and BRN models and their semantics are reviewed in detail in section 2. The robustness properties of our BRNs are explained in section 3. Our translation from NFAs to BRNs is specified in section 4, and section 5 gives the proof that our BRNs are robustly correct.

2 Review of NFAs and BRNs

Finite automata are ubiquitus in computer science, but details and notation vary from context to context, so we briefly review the specific model used in this paper. (See, e.g., [21].)

A *nondeterministic finite automaton (NFA)* is an ordered 5-tuple

$$M = (Q, \Sigma, \Delta, I, F),$$

where Q is a finite set of *states*; Σ is a finite *input alphabet*; $I \subseteq Q$ is the set of *initial states*; $F \subseteq Q$ is the set of *accepting states*; and $\Delta : Q \times \Sigma \to \mathcal{P}(Q)$ is the *transition function.* Here we are using the notation $\mathcal{P}(Q)$ for the *power set* of Q, i.e., the set of all subsets of Q. When convenient we identify the transition function Δ with the ternary relation

$$\Delta = \{(q, a, r) \in Q \times \Sigma \times Q \mid r \in \Delta(q, a)\},$$

which is the set of all *transitions* of M. 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

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

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.

Biochemical reaction networks are not as widely known in computer science, so we review them less tersely.

Formally, we fix a countably infinite set $\mathbf{S} = \{X_0, X_1, X_2, \ldots\}$, whose elements we call *species*. Informally, we regard each species as an abstract name of a type of molecule, and we avoid excessive subscripts by writing elements of $\mathbf{S}$ in other ways, e.g., X, Y, Z, $\widehat{X}$, $\widetilde{X}$, etc.

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

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

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

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

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

for the reaction $(\mathbf{r}, \mathbf{p}, k)$, where $\mathbf{r}$, $\mathbf{p} : S \to \mathbb{N}$ are defined by

$$\begin{aligned} &\mathbf{r}(X) = \mathbf{r}(Z) = 1, \quad \mathbf{r}(Y) = 0, \\ &\mathbf{p}(X) = 0, \quad \mathbf{p}(Y) = 2, \quad \mathbf{p}(Z) = 1. \end{aligned}$$

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

A *biochemical reaction network* (BRN) is an ordered pair

$$N = (S, R)$$

where $S \subseteq \mathbf{S}$ is finite, and R is a finite set of reactions over S.

There are various *semantics* (i.e., operational meanings) that can be usefully ascribed to a BRN $N = (S, R)$. In the *deterministic mass action semantics* (also called *deterministic mass action kinetics*) that we use in this paper, a *state* of the BRN N is a vector $\mathbf{x} \in [0, \infty)^S$. For each $Y \in S$ we call the nonnegative real number $\mathbf{x}(Y)$ the *concentration* of the species Y in the state $\mathbf{x}$.

In deterministic mass action semantics, the state of a BRN, and hence the concentrations of its species, vary continuously over time. This makes the following notational convention convenient. Concentrations of species X, Y, etc., are denoted by the corresponding lower-case letters x, y, etc. Thus, for example, we write $y(t)$ for the concentration of Y at time t. This convention allows us to write a state of a BRN $N = (S, R)$ as a vector $\mathbf{x} = (y \mid Y \in S) \in [0, \infty)^S$.

Let $N = (S, R)$ be a BRN. For each $\rho \in R$ and $\mathbf{x} \in [0, \infty)^S$, the (*deterministic mass action*) *rate of* ρ *in state* $\mathbf{x}$ is

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

where we write

$$\mathbf{x}^{\mathbf{r}(\rho)} = \prod_{Y \in S} \mathbf{x}(Y)^{\mathbf{r}(\rho)(Y)}.$$

For example, if ρ is the reaction $X + Z \xrightarrow{k} 2Y + Z$, then $\mathrm{rate}_{\mathbf{x}}(\rho) = k\mathbf{x}(X)\mathbf{x}(Z)$. Intuitively, the frequency with which an X and a Z encounter one another is proportional to $\mathbf{x}(X)\mathbf{x}(Z)$, and the constant of proportionality k summarizes other factors, not depending on $\mathbf{x}$ (e.g., temperature, salinity of solution, properties of X and Z), that also govern the rate at which ρ occurs.

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

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

for all $\mathbf{x} \in [0, \infty)^S$. Then $F_Y(\mathbf{x})$ is the total rate at which the concentration of Y is changing in state $\mathbf{x}$. Hence, if at time t the BRN N is in state $\mathbf{x}(t)$, then the concentration $y(t)$ of each species Y must obey the ordinary differential equation (ODE)

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

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

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

of ODEs. If we define the vector-valued function $F : [0, \infty)^S \to \mathbb{R}^S$ 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 mass action system (2.4) can also be written in the vector form

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

It is clear by inspection of (2.1)-(2.6) that the mass action system (2.6) is (i) *autonomous*, meaning that its right-hand side only depends upon the time t indirectly, via the concentrations $y(t)$, and (ii) *polynomial*, meaning that its right-hand side is a vector of functions that are polynomial in their arguments $y(t)$.

In deterministic mass action semantics a BRN N is initialized to some state at time 0, and this state then evolves according to the mass action system (2.6). Accordingly, we define an *initialized biochemical reaction network* (*IBRN*) to be an ordered pair $(N, \mathbf{x}_0)$, where N is a BRN and $\mathbf{x}_0$ is a state of N. The (*deterministic*) *mass action initial value problem* (*mass action IVP*) of an IBRN $(N, \mathbf{x}_0)$ is the initial value problem (IVP) consisting of the mass action system (2.6) of N together with the *initial value condition*

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

By the standard existence-uniqueness theory for ODEs [2, 32], 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 deterministic mass action semantics, this solution $\mathbf{x} : [0, b) \to [0, \infty)^S$ is the operational meaning of the IBRN $(N, \mathbf{x}_0)$, and the family of such solutions, as $\mathbf{x}_0$ ranges over $[0, \infty)^S$, is the operational meaning of the BRN N. The BRNs defined in this paper are all very well behaved, so that $b = \infty$, i.e., $\mathbf{x}(t)$ is well defined for all $t \in [0, \infty)$ and all initial values $\mathbf{x}_0$ considered here.

Further discussions of biochemical reaction networks with deterministic mass action semantics appear in [16, 15, 17, 22].

3 Robustness

The objective of this paper is to translate an arbitrary NFA $M = (Q, \Sigma, \Delta, I, F)$ into a BRN $N = (S, R)$ that is robust with respect to four types of perturbations, namely, perturbations of the input signal, the initial concentrations of species, the output (decision) of the BRN, and the rate constants of reactions. The present section describes these types of robustness separately.

3.1 The input signal. An NFA $M = (Q, \Sigma, \Delta, I, F)$ receives its input $w \in \Sigma^*$ from an external agency (user) that is not part of the NFA model. This input w is presented to M *sequentially*, reading left-to-right, with M changing states appropriately in response to each symbol of w before reading the next symbol. Accordingly, the BRN $N = (S, R)$ that simulates M receives the input string $w \in \Sigma^*$ formatted by an external agency as a continuous time signal consisting of concentrations of *input species* that vary to represent w in a natural way. This *input signal* and the perturbations of it with respect to which our BRN is robust require some detailed explanation.

It is the user's responsibility to provide the input signal, i.e., to control the concentrations of the input species. This, in turn, implies that our BRN should not interfere with the user's control. Hence *the input species only appear as catalysts* of the reactions in our BRN.

Our BRN N has $|\Sigma| + 2$ input species, namely, a species X_a for each input symbol $a \in \Sigma$ and two special input species, X_r ("reset") and X_c ("copy"). Intuitively and idealistically, N would receive an input string $w = a_0 a_1 \cdots a_{n-1}$, where each $a_i \in \Sigma$, as a 0/1-valued "square-wave" (i.e., step-function) representation of the "padded" string $\hat{x} = a_0 c r a_1 c r \cdots a_{n-1} c r$, where the square-wave representation of $\hat{x}$ is as depicted in Figure 1.

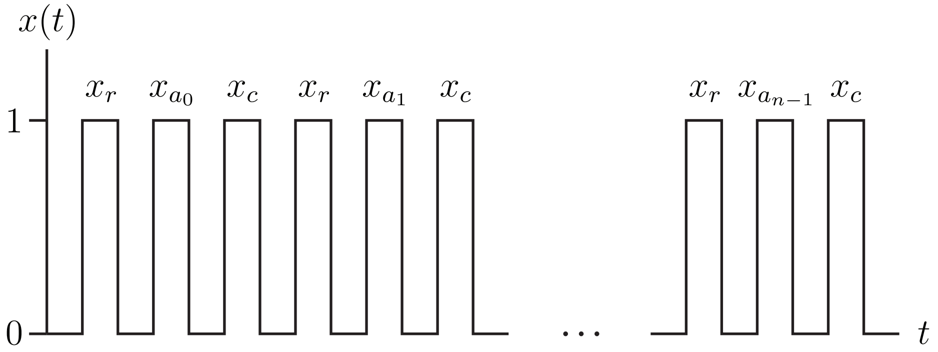

Figure 1: Idealized square-wave format of padded version $\hat{w} = r a_0 c r a_1 c \cdots r a_{n-1} c$ of input string $w = a_0 a_1 \cdot a_{n-1} \in \Sigma^*$. All input species concentrations are 0 at all times when the figure does not indicate that they are 1.

There are two things wrong with this idealization. First, step functions are not continuous, and it is unrealistic to hypothesize an agency that varies concentrations discontinuously. More importantly, it is unrealistic to hypothesize an external agency that varies concentrations according to *any* scheme that does not tolerate small errors. We thus allow the following relaxed input format. An *input signal* is an assignment of a continuous concentration function $x : [0, \infty) \to [0, \infty)$ to each of the $|\Sigma| + 2$ input species X. In the following we fix a *phase duration* $\tau > 0$. Let $\epsilon \in (0, \frac{1}{2})$. An input signal is an ϵ*-admissible encoding* of an input string $w = a_0 a_1 \cdots a_{n-1}$ for M if it satisfies the following conditions for all $k \in \mathbb{N}$. (See Figure 2.)

(1) All input species have concentrations less than $1 + \epsilon$ at all times.

(2) All input species have concentrations less than ϵ at time $k\tau$.

(3) All but at most one input species X have the property that, at all times $t \in [k\tau, (k + 1)\tau]$, the concentration $x(t)$ is less than ϵ. Such species are *not present during phase k*.

(4) If any input species is an exception to (3), then its concentration exceeds $1 - \epsilon$ throughout the time interval $[k\tau + \frac{\tau}{3}, k\tau + \frac{2\tau}{3}]$.

(5) If $k = 3i$, where $0 \leq i < n$, then input species X_r is present during phase k.

(6) If $k = 3i + 1$, where $0 \leq i < n$, then input species X_{a_i} is present during phase k.

(7) If $k = 3i + 2$, where $0 \leq i < n$, then input species X_c is present during phase k.

(8) If $k \geq 3n$, then no input species is present during phase k.

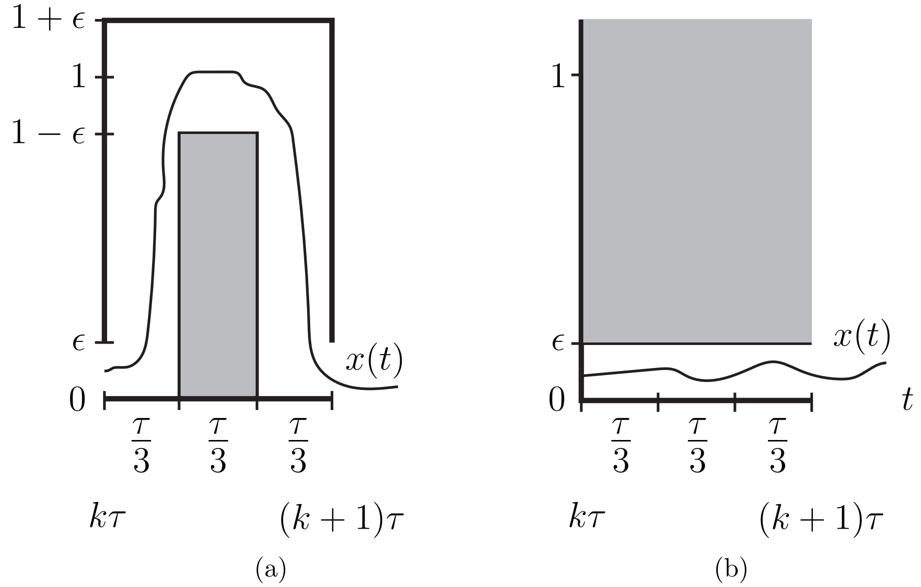

Figure 2: ϵ-admissible input signals. In (a) the input species X is present during phase k. In (b) the input species X is not present during phase k. In either case, *any* concentration signal confined to the interior of the white region is ϵ-admissible

.

3.2 Initial concentrations. Given two initial states $\mathbf{x}_0, \mathbf{x}_0^* \in [0,\infty)^S$ of a BRN $N = (S, R)$, we define the *distance* between these to be

$$\|\mathbf{x}_0^* - \mathbf{x}_0\| = \max_{Y \in S} |\mathbf{x}_0^*(Y) - \mathbf{x}_0(Y)|.$$

Intuitively, this is the "amount by which $\mathbf{x}_0$ has been perturbed" if $\mathbf{x}_0$ is replaced by $\mathbf{x}_0^*$.

3.3 Output. As seen in subsection 3.1 above, our BRN N receives an input $x = a_0 a_1 \cdots a_{n-1} \in \Sigma^*$ for M as an ϵ-admissible signal that arrives in $3n$ phases, each of duration τ. After this time, no input species is present, i.e., all input species have concentrations less than ϵ. At any time after one additional phase, i.e., any $t \geq (3n+1)\tau$, a user of our BRN N may observe one or more of the concentrations $y_q(t)$ of those species Y_q that correspond to states $q \in Q$ of the NFA M. This observation itself is subject to error, i.e., it may be a value $\hat{y}_q(t)$ that differs from $y_q(t)$. Let $\eta \in (0, \frac{1}{2})$. We say that a *state observation scheme* $y_q \mapsto \hat{y}_q$ is *η-admissible* if, for every $t \geq (3n+1)\tau$ and every state $q \in Q$ of M, the observed concentration $\hat{y}_q(t)$ satisfies

$$|\hat{y}_q(t) - y_q(t)| \leq \eta.$$

3.4 Rate constants. In order to state the sense in which our BRN's simulation of an NFA is robust with respect to perturbations of rate constants, we need a slight generalization of the BRN definition.

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

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

where $C([0,\infty),(0,\infty))$ is the set of continuous functions from $[0,\infty)$ to $(0,\infty)$. As before, we write $\mathbf{r}(\rho) = \mathbf{r}$, $\mathbf{p}(\rho) = \mathbf{p}$, and $k^*(\rho) = k^*$, and we use more intuitive notions like

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

remembering that k^* is now a function of time, rather than a constant. A *time-dependent BRN* (*tdBRN*) is then an ordered pair

$$N^* = (S, R^*),$$

where $S \subseteq \mathbf{S}$ is finite, and R^* is a finite set of time-dependent reactions over S. The deterministic mass action semantics of a tdBRN $N^* = (S, R^*)$ are then defined in the obvious way, rewriting (2.1)–(2.6) as

$$\text{rate}_{\mathbf{x}}(\rho)(t) = k^*(\rho)(t)\mathbf{x}^{\mathbf{r}(\rho)}, \tag{3.1}$$

$$F_Y(\mathbf{x}, t) = \sum_{\rho \in R^*} \text{rate}_{\mathbf{x}}(\rho)(t)\Delta_\rho(Y), \tag{3.2}$$

$$y'(t) = F_Y(\mathbf{x}(t), t), \tag{3.3}$$

$$(\mathcal{E}_Y \mid Y \in S), \tag{3.4}$$

$$F(\mathbf{x}, t) = (F_Y(\mathbf{x}, t) \mid Y \in S), \tag{3.5}$$

$$x'(t) = F(\mathbf{x}(t), t). \tag{3.6}$$

Note that (3.6) is *not* autonomous.

Let $N = (S, R)$ be a BRN, and let $\delta > 0$ be a positive real number. A *δ-perturbation* of N is a tdBRN $N^* = (S, R^*)$ in which R^* is exactly like R, except that each reaction $(\mathbf{r}, \mathbf{p}, k)$ is replaced by a time-dependent reaction $(\mathbf{r}, \mathbf{p}, k^*)$ satisfying

$$|k^*(t) - k| \leq \delta \tag{3.7}$$

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

Intuitively, a δ-perturbation of N is a tdBRN N^* in which an *adversary* is allowed to *vary* each rate constant k, subject to the constraint (3.7).

Putting this all together, let $\epsilon, \eta \in (0, \frac{1}{2})$ and $\delta > 0$; let $M = (Q, \Sigma, \Delta, I, F)$ be an NFA; and let $(N, \mathbf{x}_0)$ be an IBRN, where $N = (S, R)$. We say that $(N, \mathbf{x}_0)$ (ϵ, η, δ)-*robustly simulates* M if the following conditions hold. (Recall that we have fixed a phase duration $\tau > 0$.)

(i) N has the $|\Sigma| + 2$ input species described in subsection 3.1, and these only appear as catalysts of reactions in R.

(ii) There is a *state species* $Y_q \in S$ for each state $q \in Q$.

(iii) For each $q \in I$, $\mathbf{x}_0(Y_q) = 1$. For each $q \in Q \setminus I$, $\mathbf{x}_0(Y_q) = 0$.

(iv) For every δ-perturbation N^* of N, every input string $w \in \Sigma^*$ for M, every ϵ-admissible encoding of w as an input signal, every initial state $\mathbf{x}_0^*$ of N^* satisfying $\|\mathbf{x}_0^* - \mathbf{x}_0\| < \epsilon$, every η-admissible state observation scheme $y_q \mapsto \hat{y}_q$, and every time $t \geq (3n+1)\tau$, the observed concentrations $\hat{y}_q(t)$ in $(\mathbf{x}_0^*, N^*)$ satisfy

$$\hat{y}_q(t) > \frac{2}{3} \iff q \in \hat{\Delta}(I, w)$$

and

$$\hat{y}_q(t) < \frac{1}{3} \iff q \notin \hat{\Delta}(I, w).$$

4 Translating NFAs to BRNs

We now describe our method for translating an arbitrary NFA $M = (Q, \Sigma, \Delta, I, F)$ into a BRN $N = (S, R)$.

4.1 Species Our translation uses exactly $4|Q|+|\Sigma|+2$ species. Of these species, there are four species *types* that are useful to distinguish.

1. For each state, $q \in Q$, there is a *state species* Y_q whose presence in the BRN corresponds to the NFA being in state q.

2. For each state, $q \in Q$, there is a *portal species*, Z_q, which is used as a buffer to facilitate state transitions in the BRN.

3. In order to support ϵ-admissible encodings of input strings, there are exactly $|\Sigma| + 2$ *input species*. There is a species X_a for each symbol $a \in \Sigma$, and the two special input species X_r and X_c. The species X_r intuitively is used to “reset” the BRN in order to prepare to process the next input symbol. The species X_c is used along with the portal species in the BRN to “copy” the current value of each portal species, Z_q, into the corresponding state species, Y_q.

4. For each state species Y_q and portal species Z_q, there are *dual species* $\bar{Y}_q$ and $\bar{Z}_q$. We refer to the species Y_q, Z_q as *basic species* in order to further distiguish them from their duals $\bar{Y}_q$, $\bar{Z}_q$. Intuitively, a dual of a basic species is one that has exactly the opposite operational meaning, i.e., when Y_q has high concentration, $\bar{Y}_q$ has low concentration and vice versa.

4.2 Reactions Our translation uses exactly $5|Q| + |\Delta|$ reactions and they are enumerated below.

- $\forall q \in Q$

$$X_r + Z_q \xrightarrow{k_3} X_r + \bar{Z}_q \tag{4.1}$$

- $\forall (s, a, q) \in \Delta$

$$X_a + Y_s + \bar{Z}_q \xrightarrow{k_1} X_a + Y_s + Z_q \tag{4.2}$$

- $\forall q \in Q$

$$X_c + Z_q + \bar{Y}_q \xrightarrow{k_2} X_c + Z_q + Y_q \tag{4.3}$$

$$X_c + \bar{Z}_q + Y_q \xrightarrow{k_2} X_c + \bar{Z}_q + \bar{Y}_q \tag{4.4}$$

- $\forall q \in Q$

$$2Y_q + \bar{Y}_q \xrightarrow{k_4} 3Y_q \tag{4.5}$$

$$2\bar{Y}_q + Y_q \xrightarrow{k_4} 3\bar{Y}_q \tag{4.6}$$

Ideally, our BRN would perform three different computations in sequence.

1. Reset the portal species to 0 (using reactions (4.1))
2. Compute the state transition function and store the result in the portal species concentrations (using reactions (4.2))
3. Copy the value of the portal species into the state species (using reactions (4.3)-(4.4))

In reality, all the reactions are active simultaneously and cause errors to accumulate to catastrophic levels. Hence the reactions (4.5)-(4.6) are included to mitigate these errors. These reactions compute an approximate majority algorithm to keep the concentrations of the state species and their duals at appropriate distances.

4.3 Example. Finally we give a small example to demonstrate the transformation from a small NFA that decides if a binary string's second to last bit is a 1. (See Figure 3.) The NFA has 3 states and 5 transitions, so the BRN contains $4 \cdot 4 + 2 + 2 = 16$ species, and $5 \cdot 3 + 5 = 20$ reactions, and is shown in Figure 4 below. Note the species are not listed explicitly but are implicitly listed within the reactions.

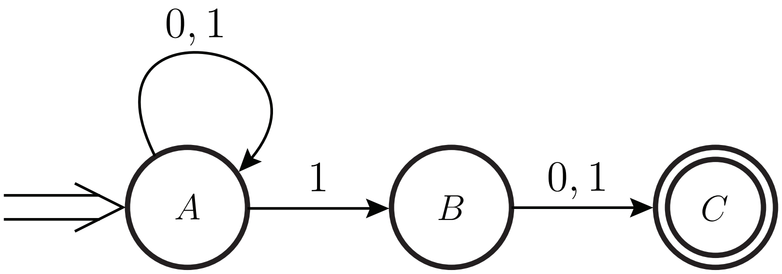

Figure 3: Example NFA with 3 states and 5 transitions that decides if the second to the last bit is a 1.

$$\begin{aligned}
&X_r + Z_A \xrightarrow{k_3} X_r + \bar{Z}_A && \text{from (4.1)}\\
&X_r + Z_B \xrightarrow{k_3} X_r + \bar{Z}_B &&\\
&X_r + Z_C \xrightarrow{k_3} X_r + \bar{Z}_C &&
\end{aligned}$$

$$\begin{aligned}
&X_0 + Y_A + \bar{Z}_A \xrightarrow{k_1} X_0 + Y_A + Z_A && \text{from (4.2)}\\
&X_1 + Y_A + \bar{Z}_A \xrightarrow{k_1} X_1 + Y_A + Z_A &&\\
&X_1 + Y_A + \bar{Z}_B \xrightarrow{k_1} X_1 + Y_A + Z_B &&\\
&X_0 + Y_B + \bar{Z}_C \xrightarrow{k_1} X_0 + Y_B + Z_C &&\\
&X_1 + Y_B + \bar{Z}_C \xrightarrow{k_1} X_1 + Y_B + Z_C &&
\end{aligned}$$

$$\begin{aligned}
&X_c + Z_A + \bar{Y}_A \xrightarrow{k_2} X_c + Z_A + Y_A && \text{from (4.3)}\\
&X_c + Z_B + \bar{Y}_B \xrightarrow{k_2} X_c + Z_B + Y_B &&\\
&X_c + Z_C + \bar{Y}_C \xrightarrow{k_2} X_c + Z_C + Y_C &&
\end{aligned}$$

$$\begin{aligned}
&X_c + \bar{Z}_A + Y_A \xrightarrow{k_2} X_c + \bar{Z}_A + \bar{Y}_A && \text{from (4.4)}\\
&X_c + \bar{Z}_B + Y_B \xrightarrow{k_2} X_c + \bar{Z}_B + \bar{Y}_B &&\\
&X_c + \bar{Z}_C + Y_C \xrightarrow{k_2} X_c + \bar{Z}_C + \bar{Y}_C &&
\end{aligned}$$

$$\begin{aligned}
&2Y_A + \bar{Y}_A \xrightarrow{k_4} 3Y_A && \text{from (4.5)}\\
&2Y_B + \bar{Y}_B \xrightarrow{k_4} 3Y_B &&\\
&2Y_C + \bar{Y}_C \xrightarrow{k_4} 3Y_C &&
\end{aligned}$$

$$\begin{aligned}
&2\bar{Y}_A + Y_A \xrightarrow{k_4} 3\bar{Y}_A && \text{from (4.6)}\\
&2\bar{Y}_B + Y_B \xrightarrow{k_4} 3\bar{Y}_B &&\\
&2\bar{Y}_C + Y_C \xrightarrow{k_4} 3\bar{Y}_C &&
\end{aligned}$$

Figure 4: BRN reactions for example NFA.

5 Robust correctness of translation

We now give a complete proof that for an arbitrary NFA M our BRN translation N robustly simulates the NFA, provided some modest constraints are placed on the parameters. We first state the main theorem here with proof using lemmas that follow.

Theorem 5.1. Let $\epsilon, \eta \in (0, \frac{1}{2})$, let $\tau, \delta > 0$, and let $M = (Q, \Sigma, \Delta, I, F)$ be an NFA. Let $N = (S, R)$ be the BRN constructed from M in section 4, and let $\mathbf{x}_0$ be a state with

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

If the constraints from lemmas 5.27 and 5.28 hold, and the helper constant γ is bounded such that $1 - \gamma \geq \frac{2}{3} + \eta$ and $\gamma \leq \frac{1}{3} - \eta$ then $(N, \mathbf{x}_0)$ (ϵ, η, δ)-robustly simulates M.

Proof. Assume the hypothesis. By lemmas 5.27 and 5.28 we have that for all $w \in \Sigma^*$ the inductive hypothesis (defined below) $\Phi(w) = \text{true}$. Since $1 - \gamma > \frac{2}{3} + \eta$ and $\gamma < \frac{1}{3} - \eta$, the theorem holds. □

The rest of this section is devoted to proving the lemmas required for theorem 5.1.

Let N^* be a δ-perturbation of N, and let $\mathbf{x}_0^*$ be a state such that $\|\mathbf{x}_0^* - \mathbf{x}_0\| < \epsilon$. Let $w \in \Sigma^*$ be an input string of M, and let the input species of N conform to an ϵ-admissible encoding of w.

Recall that the tdIBRN $(N^*, \mathbf{x}_0^*)$ specifies an IVP that has a unique solution. It is easy to verify that the set of ODEs generated by N^* are as follows.

$$\frac{dy_q}{dt} = k_2^* x_c \left[z_q \bar{y}_q - \bar{z}_q y_q\right] + k_4^* y_q \bar{y}_q \left[y_q - \bar{y}_q\right] \tag{5.1}$$

$$\frac{dz_q}{dt} = \sum_{(s,a,q)\in\Delta} k_1^* x_a y_s \bar{z}_q - k_3^* x_r z_q \tag{5.2}$$

$$\frac{d\bar{z}_q}{dt} = -\frac{dz_q}{dt} \tag{5.3}$$

$$\frac{d\bar{y}_q}{dt} = -\frac{dy_q}{dt} \tag{5.4}$$

In order to prove that N (ϵ, η, δ)-robustly simulates M, it suffices to show that for all $t \geq (3|w| + 1)\tau$, the solution to $(N^*, \mathbf{x}_0^*)$ satisfies

$$\forall q \in \widehat{\Delta}(I, w), \quad y_q(t) > \frac{2}{3} + \eta, \tag{5.5}$$

$$\forall q \notin \widehat{\Delta}(I, w), \quad y_q(t) < \frac{1}{3} - \eta. \tag{5.6}$$

We prove this in two steps. First, we prove by induction on $w \in \Sigma^*$ that at time $t = 3|w|\tau$, the solution to $(N^*, \mathbf{x}_0^*)$ "almost" satisfies the statements (5.5)-(5.6). Second, we prove that the approximate majority structure within N^* achieves the necessary bounds by time $t = (3|w| + 1)\tau$ and maintains them indefinitely. Together, these two parts trivially satisfy the statements (5.5)-(5.6).

For clarity, we state our induction hypothesis as the Boolean function $\Phi : \Sigma^* \to \{\text{true}, \text{false}\}$. Let $\gamma \in (0, \frac{1}{2})$ be a constant. Define $\Phi(w) = \text{true}$ if and only if the following two conditions hold.

$$\forall q \in \widehat{\Delta}(I, w), \quad y_q(3|w|\tau) \geq 1 - \gamma, \tag{5.7}$$

$$\forall q \notin \widehat{\Delta}(I, w), \quad y_q(3|w|\tau) \leq \gamma. \tag{5.8}$$

Lemma 5.2. If $\gamma \geq \epsilon$, $\Phi(\lambda)$ is true.

Proof. Recall that $\widehat{\Delta}(I, \lambda) = I$. Therefore, it suffices to show

$$\begin{aligned} &\forall q \in I, && y_q(0) \geq 1 - \gamma, \\ &\forall q \in Q \setminus I, && y_q(0) \leq \gamma. \end{aligned}$$

Since $\|\mathbf{x}_0^* - \mathbf{x}_0\|$ and $\gamma \geq \epsilon$ we have

$$\forall q \in I, \quad y_q(0) \geq 1 - \epsilon \geq 1 - \gamma$$

and

$$\forall q \in Q \setminus I, \quad y_q(0) \leq \epsilon \leq \gamma$$

□

To prove that $\Phi(w)$ holds for every string $w \in \Sigma^*$, it remains to be shown that for every string $w \in \Sigma^*$ and for every symbol $a \in \Sigma$, the following implication holds.

$$\Phi(w) \implies \Phi(wa) \tag{5.9}$$

The majority of this section is devoted to the proof of (5.9). We begin by proving a series of lemmas in support of this proof.

For each basic species Y_q, Z_q, let $c(Y_q), c(Z_q)$ be the following constants.

$$c(Y_q) = y_q(0) + \bar{y}_q(0) \tag{5.10}$$

$$c(Z_q) = z_q(0) + \bar{z}_q(0) \tag{5.11}$$

Lemma 5.3. For all $t \in [0, \infty)$ and for all $q \in Q$

(a) $y_q(t) + \bar{y}_q(t) = c(Y_q)$

(b) $z_q(t) + \bar{z}_q(t) = c(Z_q)$

Proof. Recall the ODEs for $y_q, \bar{y}_q, z_q$ and $\bar{z}_q$ have the following relationships.

$$\frac{dy_q}{dt} = -\frac{d\bar{y}_q}{dt} \qquad \frac{dz_q}{dt} = -\frac{d\bar{z}_q}{dt}$$

Integration and solving for the constants trivially confirms (a) and (b) are true. □

Lemma 5.4. For all $q \in Q$, $c(Y_q), c(Z_q) \in [1 - \epsilon, 1 + 2\epsilon]$.

Proof. Recall every basic species Y_q, Z_q in the state $\mathbf{x}_0$ is set to 0 or 1 and its $\bar{Y}_q, \bar{Z}_q$ dual is set to the appropriate 0 or 1 value. Since $\|\mathbf{x}_0^* - \mathbf{x}_0\| \leq \epsilon$, it is clear that $c(Y_q), c(Z_q) \leq 1 + 2\epsilon$. Since no species can be negative, it is clear that $c(Y_q), c(Z_q) \geq 1 - \epsilon$. □

Construction 5.5. Let $a > 0$, $b > 0$, $c > 0$, and $p > 0$ be constants and let $u(t)$ and $\bar{u}(t)$ be functions defined by

$$\frac{du}{dt} = u\bar{u}(au - b\bar{u}) - cu, \tag{5.12}$$

where $\bar{u}(t) = p - u(t)$ for all $t \in [0, \infty)$.

Lemma 5.6. In construction 5.5, if $c < \frac{p^2a^2}{4(a+b)}$, then the function $u(t)$ has the following equilibrium points.

$$E_1 = 0, \tag{5.13}$$

$$E_2 = \frac{p(a+2b) - A}{2(a+b)}, \tag{5.14}$$

$$E_3 = \frac{p(a+2b) + A}{2(a+b)}, \tag{5.15}$$

where

$$A = \sqrt{p^2a^2 - 4c(a+b)}.$$

Of these three equilibrium points, (5.13) and (5.15) are exponentially stable, and (5.14) is unstable.

Proof. The equilibrium points of $u(t)$ can be found by solving the equation.

$$u\bar{u}(au - b\bar{u}) - cu = 0 \tag{5.16}$$

It is routine to show that the roots of (5.16) are indeed the points (5.13)-(5.15).

Note that $u(t)$ has three distinct real roots if and only if A is a non-zero real. The constant A is real and non-zero if and only if

$$p^2a^2 > 4c(a+b),$$

whence,

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

What remains to be shown is the stability of the three points. It is well known in dynamical systems that if each eigenvalue of the Jacobian matrix at an equilibrium point has negative real part, then the point is exponentially stable. Likewise, if each eigenvalue has a positive real part, then it is unstable. Since the system in construction (5.5) is 1-dimensional, there is only one eigenvalue.

Equilibrium point (5.13) has the eigenvalue $\lambda_1 = -(c + bp^2)$ which is always negative. Therefore it is exponentially stable.

Equilibrium point (5.14) has the eigenvalue $\lambda_2 = \frac{A}{2(a+b)}[p(a+2b) - A]$. Since $c > 0$, we know $0 < A < pa$, therefore $\lambda_2 > \frac{A}{2(a+b)}[p(2b)]$ which is always positive. Therefore it is unstable.

Equilibrium point (5.15) has the eigenvalue $\lambda_3 = -\lambda_2$. Since $\lambda_2 > 0$, we know that $\lambda_3 < 0$. Therefore it is exponentially stable. □

Corollary 5.7. For construction 5.5 and $t_0 \geq 0$, if $u(t_0) \geq E_2$, then for all $t \geq 0$, $u(t_0 + t) \geq u(t_0)$.

Lemma 5.8. Assuming construction 5.5, let u_1, u_2 be constants such that $E_3 > u_2 > u_1 > E_2$, let $t_0 \geq 0$, and let $t > 0$ such that $u(t_0) = u_1$ and $u(t_0 + t) = u_2$. Then

$$t = \frac{1}{A}\left(\frac{1}{E_3}\log\left[\frac{u_2(E_3 - u_1)}{u_1(E_3 - u_2)}\right] + \frac{1}{E_2}\log\left[\frac{u_1(u_2 - E_2)}{u_2(u_1 - E_2)}\right]\right). \tag{5.17}$$

Proof. Recall that the ODE for $u(t)$ is

$$\begin{aligned}\frac{du}{dt} &= u\bar{u}(au - b\bar{u}) - cu \\ &= u(p-u)[au - b(p-u)] - cu,\end{aligned}$$

and the roots of the ODE are E_1, E_2, and E_3. It can be shown that

$$\frac{du}{dt} = (a+b)u(u - E_2)(E_3 - u).$$

By separation of variables and integration, we have

$$\frac{1}{a+b}\int_{u_1}^{u_2} \frac{1}{u(u-E_2)(E_3-u)}du = \int_{t_0}^{t_0+t} dt.$$

Therefore,

$$t = \frac{1}{a+b}\left(\frac{1}{E_2E_3(E_2-E_3)}\log\left[\left(\frac{u}{u-E_2}\right)^{E_3}\left(\frac{u-E_3}{u}\right)^{E_2}\right]\right)\Bigg|_{u_1}^{u_2}$$

$$= \frac{1}{a+b}\left(\frac{1}{E_2E_3(E_2-E_3)}\log\left[\left(\frac{u_2(u_1-E_2)}{u_1(u_2-E_2)}\right)^{E_3}\left(\frac{u_1(u_2-E_3)}{u_2(u_1-E_3)}\right)^{E_2}\right]\right)$$

$$= \frac{1}{a+b}\left(\frac{1}{E_2-E_3}\right)\left(\frac{1}{E_2}\log\left[\frac{u_2(u_1-E_2)}{u_1(u_2-E_2)}\right]+\frac{1}{E_3}\log\left[\frac{u_1(u_2-E_3)}{u_2(u_1-E_3)}\right]\right)$$

$$= \frac{1}{a+b}\left(-\frac{a+b}{A}\right)\left(\frac{1}{E_2}\log\left[\frac{u_2(u_1-E_2)}{u_1(u_2-E_2)}\right]+\frac{1}{E_3}\log\left[\frac{u_1(E_3-u_2)}{u_2(E_3-u_1)}\right]\right)$$

$$= \frac{1}{A}\left(\frac{1}{E_2}\log\left[\frac{u_1(u_2-E_2)}{u_2(u_1-E_2)}\right]+\frac{1}{E_3}\log\left[\frac{u_2(E_3-u_1)}{u_1(E_3-u_2)}\right]\right)$$

□

Construction 5.9. Let $a > 0$, $b > 0$, $c > 0$, and $p > 0$ be constants and let $u(t)$ and $\bar{u}(t)$ be functions defined by

$$\frac{du}{dt} = u\bar{u}(au - b\bar{u}) + c\bar{u}, \tag{5.18}$$

where $\bar{u}(t) = p - u(t)$ for all $t \in [0, \infty)$.

Lemma 5.10. In construction 5.9, if $c < \frac{p^2b^2}{4(a+b)}$, then the function $u(t)$ has the following equilibrium points.

$$E_1^* = \frac{bp - A^*}{2(a+b)}, \tag{5.19}$$

$$E_2^* = \frac{bp + A^*}{2(a+b)}, \tag{5.20}$$

$$E_3^* = p, \tag{5.21}$$

where

$$A^* = \sqrt{p^2b^2 - 4c(a+b)}.$$

Of these three equilibrium points, (5.19) and (5.21) are exponentially stable, and (5.20) is unstable.

Proof. Using a similar argument as the proof of lemma 5.6, it is easy to show that this lemma also holds. □

Corollary 5.11. For construction 5.9 and $t_0 \geq 0$, if $u(t_0) \leq E_2^*$, then for all $t \geq 0$, $u(t_0 + t) \leq u(t_0)$.

Lemma 5.12. Assuming construction 5.9, let u_1, u_2 be constants such that $E_2^* > u_1 > u_2 > E_1^*$, let $t_0 \geq 0$, and let $t > 0$ such that $u(t_0) = u_1$ and $u(t_0 + t) = u_2$. Then

$$t = \frac{1}{A^*} \left(\frac{1}{E_1^*} \log \left[\frac{u_2(u_1 - E_1^*)}{u_1(u_2 - E_1^*)} \right] + \frac{1}{E_2^*} \log \left[\frac{u_1(E_2^* - u_2)}{u_2(E_2^* - u_1)} \right] \right). \tag{5.22}$$

Proof. Using a similar argument as the proof of lemma 5.8, it is easy to show that this lemma also holds. □

Construction 5.13. Let $a > 0$, $b > 0$, and $p > 0$ be constants, and let $u(t)$ and $\bar{u}(t)$ be functions defined by

$$\frac{du}{dt} = a\bar{u} - bu, \tag{5.23}$$

where $\bar{u}(t) = p - u(t)$ for all $t \in [0, \infty)$.

Lemma 5.14. In construction 5.13, if $t_0, t \geq 0$, then

$$u(t_0 + t) = u(t_0)e^{-(a+b)t} + \frac{ap}{a+b}\left(1 - e^{-(a+b)t}\right). \tag{5.24}$$

Proof. The ODE (5.23) is trivially solvable via separation of variables and integration. It is routine to show that its solution matches that of equation (5.24). □

Corollary 5.15. In construction 5.13,

(a) if $a = 0$, then
$$u(t_0 + t) = u(t_0)e^{-bt},$$

(b) if $b = 0$, then
$$u(t_0 + t) = u(t_0)e^{-at} + p(1 - e^{-at}).$$

Corollary 5.16. In construction 5.13, for $t_0, t \geq 0$, regardless of the initial condition $u(t_0)$, the function $u(t)$ is bounded in the following way.

$$u(t_0 + t) \leq p\left(\frac{a}{b} + e^{-bt}\right) \tag{5.25}$$

$$u(t_0 + t) \geq p - \frac{b}{a} - e^{-at} \tag{5.26}$$

Proof. To show (5.25), we have

$$\begin{aligned} u(t_0 + t) &= u(t_0)e^{-(a+b)t} + \frac{ap}{a+b}\left(1 - e^{-(a+b)t}\right) \\ &\leq pe^{-bt} + \frac{ap}{a+b}(1) \\ &\leq p\left(e^{-bt} + \frac{a}{b}\right) \end{aligned}$$

To show (5.26), we have

$$\begin{aligned}
u(t_0+t) &= u(t_0)e^{-(a+b)t} + \frac{ap}{a+b}\left(1-e^{-(a+b)t}\right) \\
&\geq p\left(\frac{a}{a+b}\right)\left(1-e^{-at}\right) \\
&\geq p\left(1-\frac{b}{a+b}\right)\left(1-e^{-at}\right) \\
&\geq p-\frac{b}{a}-e^{-at}
\end{aligned}$$

□

We now begin proving lemmas showing bounds of the solution to $(N^*, \mathbf{x}_0^*)$ while processing input. Let $d = |\Delta|$.

Construction 5.17. Let X be an input species in $(N^*, \mathbf{x}_0^*)$ that is present during the interval $[k\tau, (k+1)\tau]$.

Lemma 5.18. In construction 5.17, if $X = X_r$, then for all $q \in Q$,

$$z_q((k+1)\tau) \leq 2\left(\frac{a}{b} + e^{-b\tau/3}\right) + a\tau \tag{5.27}$$

where

$$\begin{aligned}
a &= 2d(k_1+\delta)\epsilon \\
b &= (k_3-\delta)(1-\epsilon)
\end{aligned}$$

Proof. Recall the ODE for z_q is

$$\frac{dz_q}{dt} = \sum_{(s,a,q)\in\Delta} k_1^* x_a y_s \bar{z}_q - k_3^* x_r z_q.$$

Since $d = |\Delta|$, there can be at most d reactions contributing to the sum, and by lemma 5.4 we know that the species Y_q has at most a concentration of $1+2\epsilon$. Therefore, we have

$$\begin{aligned}
\frac{dz_q}{dt} &\leq dk_1^* x_a(1+2\epsilon)\bar{z}_q - k_3^* x_r z_q \\
&\leq 2dk_1^* x_a \bar{z}_q - k_3^* x_r z_q
\end{aligned}$$

The constraints on the δ-perturbated rate constants ensure that for all $t \in [0,\infty)$, $|k^*(t) - k| \leq \delta$, whence

$$\frac{dz_q}{dt} \leq 2d(k_1+\delta)x_a\bar{z}_q - (k_3-\delta)x_r z_q.$$

Since X_r is present, we know that during the interval $[k\tau + \frac{\tau}{3}, k\tau + \frac{2\tau}{3}]$ $x_r \geq 1 - \epsilon$ and all other input species are less than ϵ. Therefore, we have

$$\begin{aligned} \frac{dz_q}{dt} &\leq 2d(k_1 + \delta)\epsilon \bar{z}_q - (k_3 - \delta)(1 - \epsilon) z_q \\ &= a\bar{z}_q - bz_q \end{aligned}$$

which by corollary 5.16 yields

$$\begin{aligned} z_q(k\tau + \frac{2\tau}{3}) &\leq (1 + 2\epsilon)\left(\frac{a}{b} + e^{-b\tau/3}\right) \\ &\leq 2\left(\frac{a}{b} + e^{-b\tau/3}\right) \end{aligned}$$

Finally, during the time interval $[k\tau + \frac{2\tau}{3}, (k+1)\tau]$ the ODE for z_q is bounded by

$$\begin{aligned} \frac{z_q}{dt} &\leq a\bar{z}_q \\ &\leq a(1 + 2\epsilon) \\ &\leq 2a \end{aligned}$$

and therefore

$$\begin{aligned} z_q((k+1)\tau) &\leq z_q(k\tau + \frac{2\tau}{3}) + 2a\left(\frac{\tau}{3}\right) \\ &\leq 2\left(\frac{a}{b} + e^{-b\tau/3}\right) + a\tau \end{aligned}$$

□

Construction 5.19. For $q \in Q$, let a, b, c, and p be constants such that

$$\begin{aligned} a &= k_4 - \delta, & c &= 2\epsilon(k_2 + \delta), \\ b &= k_4 + \delta, & p &= c(Y_q). \end{aligned}$$

With respect to the above constants, let E_1, E_2, E_3, and A be defined as in lemma 5.6.

Lemma 5.20. Given construction 5.17 with $X \neq X_c$ and construction 5.19 for $q \in Q$, let y_1, y_2 be constants such that $E_3 > y_2 > y_1 > E_2$ and $y_q(k\tau) \geq y_1$. Then if

$$\tau \geq \frac{1}{A}\left(\frac{1}{E_3}\log\left[\frac{y_2(E_3 - y_1)}{y_1(E_3 - y_2)}\right] + \frac{1}{E_2}\log\left[\frac{y_1(y_2 - E_2)}{y_2(y_1 - E_2)}\right]\right),$$

then

$$y_q((k+1)\tau) \geq y_2.$$

Proof. Recall that the ODE for Y_q is

$$\begin{aligned}\frac{dy_q}{dt} &= k_2^* x_c \left[z_q \bar{y}_q - \bar{z}_q y_q\right] + k_4^* y_q \bar{y}_q \left[y_q - \bar{y}_q\right] \\ &= k_2^* x_c z_q \bar{y}_q - k_2^* x_c \bar{z}_q y_q + k_4^* y_q^2 \bar{y}_q - k_4^* y_q \bar{y}_q^2 \\ &\geq -(k_2+\delta)\epsilon(1+2\epsilon) y_q + (k_4-\delta) y_q^2 \bar{y}_q - (k_4+\delta) y_q \bar{y}_q^2 \\ &\geq -2\epsilon(k_2+\delta) y_q + (k_4-\delta) y_q^2 \bar{y}_q - (k_4+\delta) y_q \bar{y}_q^2 \\ &= y_q \bar{y}_q \left[a y_q - b \bar{y}_q\right] - c y_q.\end{aligned}$$

Let $\tau^* > 0$ be a constant such that $y_q(k\tau + \tau^*) = y_2$. Then by lemma 5.8, we know that

$$\begin{aligned}\tau^* &\leq \frac{1}{A}\left(\frac{1}{E_3}\log\left[\frac{y_2(E_3-y_1)}{y_1(E_3-y_2)}\right] + \frac{1}{E_2}\log\left[\frac{y_1(y_2-E_2)}{y_2(y_1-E_2)}\right]\right) \\ &\leq \tau\end{aligned}$$

Therefore, it is clear that $y_q((k+1)\tau) \geq y_2$. □

Construction 5.21. For $q \in Q$, let a, b, c, and p be constants such that

$$\begin{aligned} a &= k_4 + \delta, & c &= 2(k_2+\delta)\epsilon, \\ b &= k_4 - \delta, & p &= c(Y_q).\end{aligned}$$

With respect to the above constants, let E_1^*, E_2^*, E_3^*, and A^* be defined as in lemma 5.10.

Lemma 5.22. Given construction 5.17 with $X \neq X_c$ and construction 5.21 for $q \in Q$, let y_1, y_2 be constants such that $E_2^* > y_1 > y_2 > E_1^*$ and $y_q(k\tau) \leq y_1$. Then if

$$\tau \geq \frac{1}{A^*}\left(\frac{1}{E_1^*}\log\left[\frac{y_2(y_1-E_1^*)}{y_1(y_2-E_1^*)}\right] + \frac{1}{E_2^*}\log\left[\frac{y_1(E_2^*-y_2)}{y_2(E_2^*-y_1)}\right]\right),$$

then

$$y_q((k+1)\tau) \leq y_2.$$

Proof. Recall that the ODE for Y_q is

$$\begin{aligned}\frac{dy_q}{dt} &= k_2^* x_c \left[z_q \bar{y}_q - \bar{z}_q y_q\right] + k_4^* y_q \bar{y}_q \left[y_q - \bar{y}_q\right] \\ &= k_2^* x_c z_q \bar{y}_q - k_2^* x_c \bar{z}_q y_q + k_4^* y_q^2 \bar{y}_q - k_4^* y_q \bar{y}_q^2 \\ &\leq (k_2+\delta)\epsilon(1+2\epsilon) \bar{y}_q + (k_4+\delta) y_q^2 \bar{y}_q - (k_4-\delta) y_q \bar{y}_q^2 \\ &\leq 2(k_2+\delta)\epsilon \bar{y}_q + (k_4+\delta) y_q^2 \bar{y}_q - (k_4-\delta) y_q \bar{y}_q^2 \\ &= y_q \bar{y}_q \left[a y_q - b \bar{y}_q\right] + c \bar{y}_q.\end{aligned}$$

Let $\tau^* > 0$ be a constant such that $y_q(k\tau + \tau^*) = y_2$. Then by lemma 5.12, we know that

$$\begin{aligned}\tau^* &\leq \frac{1}{A^*}\left(\frac{1}{E_1^*}\log\left[\frac{y_2(y_1 - E_1^*)}{y_1(y_2 - E_1^*)}\right] + \frac{1}{E_2^*}\log\left[\frac{y_1(E_2^* - y_2)}{y_2(E_2^* - y_1)}\right]\right)\\ &\leq \tau\end{aligned}$$

Therefore, it is clear that $y_q((k+1)\tau) \leq y_2$. □

Lemma 5.23. Given construction 5.17 with $X = X_a$, let y_0, α, and β be constants such that

$$\begin{aligned}&y_0 \geq E_2 \text{ (from construction 5.19)},\\ &\alpha = (k_1 - \delta)(1 - \epsilon)y_0,\\ &\beta = (k_3 + \delta)\epsilon.\end{aligned}$$

If $(s, a, q) \in \Delta$ and $y_s(k\tau) \geq y_0$, then

$$z_q((k+1)\tau) \geq 1 - \epsilon - \frac{\beta}{\alpha} - e^{-\alpha\frac{\tau}{3}} - 2\beta\tau.$$

Proof. Recall the ODE for y_s when X_c is not present is

$$\frac{dy_s}{dt} \geq y_s\bar{y}_s[ay_s - b\bar{y}_s] - cy_s.$$

By corollary 5.7 we know that for all $t \in [k\tau, (k+1)\tau]$, $y_s(t) \geq y_0$. The ODE for z_q is

$$\frac{dz_q}{dt} = \sum_{(s,a,q)\in\Delta} k_1^* x_a y_s \bar{z}_q - k_3^* x_r z_q.$$

Since $(s, a, q) \in \Delta$,

$$\begin{aligned}\frac{dz_q}{dt} &\geq k_1^* x_a y_s \bar{z}_q - k_3^* x_r z_q\\ &\geq (k_1 - \delta)x_a y_0 \bar{z}_q - (k_3 + \delta)\epsilon z_q\end{aligned}$$

During phase $[k\tau + \frac{\tau}{3}, k\tau + \frac{2\tau}{3}]$, $x_a \geq 1 - \epsilon$, so

$$\begin{aligned}\frac{dz_q}{dt} &\geq (k_1 - \delta)(1 - \epsilon)y_0\bar{z}_q - (k_3 + \delta)\epsilon z_q\\ &= \alpha\bar{z}_q - \beta z_q.\end{aligned}$$

By corollary 5.16 we have

$$z_q(k\tau + \frac{2\tau}{3}) \geq c(Z_q) - \frac{\beta}{\alpha} - e^{-\alpha\frac{\tau}{3}},$$

and in the interval $[k\tau + \frac{2\tau}{3}, (k+1)\tau]$, we have the inequalities

$$\begin{aligned}\frac{dz_q}{dt} &\geq -\beta z_q \\ &\geq -\beta(1+2\epsilon) \\ &\geq -2\beta.\end{aligned}$$

Whence, at most $2\beta\frac{\tau}{3}$ of z_q is lost during that time interval. Therefore,

$$z_q((k+1)\tau) \geq 1 - \epsilon - \frac{\beta}{\alpha} - e^{-\alpha\frac{\tau}{3}} - 2\beta\tau.$$

□

Lemma 5.24. Given construction 5.17 with $X = X_a$, let y_0 and z_0 be constants such that $y_0 \leq E_2^*$ from construction 5.21. If for every $(s,a,q) \in \Delta$, $y_s(k\tau) \leq y_0$, and $z_q(k\tau) \leq z_0$ then

$$z_q((k+1)\tau) \leq z_0 + 4d(k_1+\delta)y_0\tau.$$

Proof. Recall the ODE for y_s when X_c is not present is

$$\frac{dy_s}{dt} \leq y_s\bar{y}_s[ay_s - b\bar{y}_s] + c\bar{y}_s.$$

By corollary 5.11 we know that for all $t \in [k\tau, (k+1)\tau]$, $y_s(t) \leq y_0$. The ODE for z_q is

$$\begin{aligned}\frac{dz_q}{dt} &= \sum_{(s,a,q)\in\Delta} k_1^* x_a y_s \bar{z}_q - k_3^* x_r z_q \\ &\leq d(k_1+\delta)(1+2\epsilon)y_0(1+2\epsilon) \\ &\leq 4d(k_1+\delta)y_0\end{aligned}$$

Therefore

$$z_q((k+1)\tau) \leq z_0 + 4d(k_1+\delta)y_0\tau$$

□

Lemma 5.25. Given construction 5.17 with $X = X_c$, if $z_q(k\tau) \geq z_0$, then

$$y_q(k\tau + \frac{2\tau}{3}) \geq \frac{\alpha}{\alpha+\beta}\left(1 - \epsilon - e^{-(\alpha+\beta)\frac{\tau}{3}}\right),$$

where

$$\begin{aligned}&\alpha = (k_3-\delta)(1-\epsilon)(z_0 - 2\epsilon(k_3+\delta)), \text{ and} \\ &\beta = (k_3+\delta)(1+2\epsilon)(1+2\epsilon+2\epsilon(k_3+\delta)-z_0) + 4(k_4+\delta).\end{aligned}$$

Proof. Recall the ODE for z_q is

$$\begin{aligned}\frac{dz_q}{dt} &= \sum_{(s,a,q)\in\Delta} k_1^* x_a y_s \bar{z}_q - k_3^* x_r z_q \\ &\geq -k_3^* x_r z_q \\ &\geq -(k_3+\delta)\epsilon(1+2\epsilon) \\ &\geq -2\epsilon(k_3+\delta).\end{aligned}$$

Therefore, at most $2\epsilon(k_3+\delta)\frac{2\tau}{3}$ can be lost during the interval $[k\tau, k\tau+\frac{2\tau}{3}]$. Thus, for all $t \in [k\tau, k\tau+\frac{2\tau}{3}]$,

$$z_q(t) \geq z_0 - 2\epsilon(k_3+\delta)\tau.$$

The ODE for y_q in the interval $[k\tau+\frac{\tau}{3}, k\tau+\frac{2\tau}{3}]$ is

$$\begin{aligned}\frac{dy_q}{dt} &= k_3^* x_c[z_q\bar{y}_q - \bar{z}_q y_q] + k_4^* y_q \bar{y}_q[y_q - \bar{y}_q]] \\ &\geq (k_3-\delta)(1-\epsilon)(z_0 - 2\epsilon(k_3+\delta))\bar{y}_q \\ &\quad - (k_3+\delta)(1+2\epsilon)(1+2\epsilon+2\epsilon(k_3+\delta)-z_0)y_q \\ &\quad - (k_4+\delta)(1+2\epsilon)^2 y_q \\ &\geq \alpha\bar{y}_q - \beta y_q.\end{aligned}$$

Now by lemma 5.14 we have

$$\begin{aligned}y_q(k\tau+\frac{2\tau}{3}) &\geq y_q(k\tau+\frac{\tau}{3})e^{-(\alpha+\beta)\frac{\tau}{3}} + c(Y_q)(\frac{\alpha}{\alpha+\beta})(1-e^{-(\alpha+\beta)\frac{\tau}{3}}) \\ &\geq c(Y_q)(\frac{\alpha}{\alpha+\beta})(1-e^{-(\alpha+\beta)\frac{\tau}{3}}) \\ &\geq (1-\epsilon)(\frac{\alpha}{\alpha+\beta})(1-e^{-(\alpha+\beta)\frac{\tau}{3}}) \\ &\geq \frac{\alpha}{\alpha+\beta}(1-\epsilon-e^{-(\alpha+\beta)\frac{\tau}{3}}).\end{aligned}$$

□

Lemma 5.26. Given construction 5.17 with $X = X_c$, if $z_q(k\tau) \leq z_0$, then

$$y_q(k\tau+\frac{2\tau}{3}) \leq \frac{2}{\alpha+\beta}\left(\beta e^{-(\alpha+\beta)\frac{\tau}{3}} + \alpha\right),$$

where

$$\begin{aligned}&\alpha = (k_3+\delta)(1+2\epsilon)(z_0+4d(k_1+\delta))\epsilon + 4(k_4+\delta), \text{ and} \\ &\beta = (k_3-\delta)(1-\epsilon)(1-\epsilon-z_0-4d(k_1+\delta)\epsilon).\end{aligned}$$

Proof. Recall the ODE for z_q is

$$\begin{aligned}\frac{dz_q}{dt} &= \sum_{(s,a,q)\in\Delta} k_1^* x_a y_s \bar{z}_q - k_3^* x_r z_q \\ &\leq d(k_1+\delta)\epsilon(1+2\epsilon)(1+2\epsilon) \\ &\leq 4d(k_1+\delta)\epsilon\end{aligned}$$

Therefore, at most $4d(k_1+\delta)\epsilon\frac{2\tau}{3}$ can be gained during the interval $[k\tau, k\tau+\frac{2\tau}{3}]$. Thus, for all $t \in [k\tau, k\tau+\frac{2\tau}{3}]$,

$$z_q(t) \leq z_0 + 4d(k_1+\delta)\epsilon\tau.$$

The ODE for y_q in the interval $[k\tau+\frac{\tau}{3}, k\tau+\frac{2\tau}{3}]$ is

$$\begin{aligned}\frac{dy_q}{dt} &= k_3^* x_c[z_q\bar{y}_q - \bar{z}_q y_q] + k_4^* y_q \bar{y}_q[y_q - \bar{y}_q]] \\ &\leq (k_3+\delta)(1+2\epsilon)(z_0+4d(k_1+\delta)\epsilon\tau)\bar{y}_q \\ &\quad - (k_3-\delta)(1-\epsilon)(1-\epsilon-z_0-4d(k_1+\delta)\epsilon\tau)y_q \\ &\quad + (k_4+\delta)(1+2\epsilon)^2 y_q \\ &\leq \alpha\bar{y}_q - \beta y_q.\end{aligned}$$

Now by lemma 5.14 we have

$$\begin{aligned}y_q(k\tau+\frac{2\tau}{3}) &\leq y_q(k\tau+\frac{\tau}{3})e^{-(\alpha+\beta)\frac{\tau}{3}} + c(Y_q)(\frac{\alpha}{\alpha+\beta})(1-e^{-(\alpha+\beta)\frac{\tau}{3}}) \\ &\leq c(Y_q)e^{-(\alpha+\beta)\frac{\tau}{3}} + c(Y_q)(\frac{\alpha}{\alpha+\beta})(1-e^{-(\alpha+\beta)\frac{\tau}{3}}) \\ &= c(Y_q)\left(e^{-(\alpha+\beta)\frac{\tau}{3}} + \frac{\alpha}{\alpha+\beta} - \frac{\alpha}{\alpha+\beta}e^{-(\alpha+\beta)\frac{\tau}{3}}\right) \\ &= c(Y_q)\left(\frac{\beta}{\alpha+\beta}e^{-(\alpha+\beta)\frac{\tau}{3}} + \frac{\alpha}{\alpha+\beta}\right) \\ &\leq (1+2\epsilon)\left(\frac{\beta}{\alpha+\beta}e^{-(\alpha+\beta)\frac{\tau}{3}} + \frac{\alpha}{\alpha+\beta}\right) \\ &\leq \frac{2}{\alpha+\beta}\left(\beta e^{-(\alpha+\beta)\frac{\tau}{3}} + \alpha\right)\end{aligned}$$

□

Lemma 5.27. Let $\Phi(w) = \Phi_1(w)$ and $\Phi_2(w)$ where Φ_1 and Φ_2 correspond to the first and second clause of the inductive hypothesis Φ. If constants ϵ, δ, η, τ, γ have the constraints below for some $\epsilon < \gamma^* < \gamma$,

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

2. $$\tau \geq \frac{1}{A}\left(\frac{1}{E_3}\log\left[\frac{(1-\gamma^*)(E_3-1+\gamma)}{(1-\gamma)(E_3-1+\gamma^*)}\right] + \frac{1}{E_2}\log\left[\frac{(1-\gamma)(1-\gamma^*-E_2)}{(1-\gamma^*)(1-\gamma-E_2)}\right]\right),$$

3. $$\gamma \geq \frac{\beta}{\alpha+\beta} + \epsilon + e^{-(\alpha+\beta)\frac{\tau}{3}},$$

where the constants a, b, c, p, E_1, E_2, E_3, A are from construction 5.19 and where

$$\alpha = (k_3-\delta)(1-\epsilon)(z_0 - 2\epsilon(k_3+\delta)),$$
$$\beta = (k_3+\delta)(1+2\epsilon)(1+2\epsilon+2\epsilon(k_3+\delta)-z_0) + 4(k_4+\delta),$$
$$z_0 = 1-\epsilon-\frac{(k_3+\delta)\epsilon}{(k_1-\delta)(1-\epsilon)(1-\gamma^*)} - e^{-(k_1-\delta)(1-\epsilon)(1-\gamma^*)\frac{\tau}{3}} - 2(k_3+\delta)\epsilon\tau,$$

then

$$\Phi(w) \implies \Phi_1(wa).$$

Proof. Assume $\Phi(w)$ and let $(s,a,q) \in \Delta$. It suffices to show that $y_q(3|wa|\tau) \geq 1-\gamma$. Let the constants a, b, c, p, E_1, E_2, E_3, A be as given in construction 5.19, and let γ^* be a constant such that $\gamma > \gamma^* > \epsilon$. By lemma 5.20 and constraints (1) and (2),

$$y_s((3|w|+1)\tau) \geq 1-\gamma^*.$$

By lemma 5.23 and constraint (1)

$$z_q((3|w|+2)\tau) \geq z_0,$$

where z_0 is defined above. By lemma 5.25,

$$y_q((3|w|+2)\tau + \frac{2\tau}{3}) \geq \frac{\alpha}{\alpha+\beta} - \epsilon - e^{-(\alpha+\beta)\frac{\tau}{3}},$$

and by constraint (3)

$$y_q((3|w|+2)\tau + \frac{2\tau}{3}) \geq 1-\gamma.$$

Finally, by corollary 5.7,

$$y_q(3|wa|\tau) \geq 1-\gamma.$$

□

Lemma 5.28. If the following constraints are satisfied with $\gamma > \gamma^* > E_1^*$,

1. $$\tau \geq \frac{1}{A^*}\left(\frac{1}{E_1^*}\log\left[\frac{\gamma^*(\gamma-E_1^*)}{\gamma(\gamma^*-E_1^*)}\right] + \frac{1}{E_2^*}\log\left[\frac{\gamma(E_2^*-\gamma^*)}{\gamma^*(E_2^*-\gamma)}\right]\right),$$

2. $$\gamma \geq \frac{2}{\alpha+\beta}(\beta e^{-(\alpha+\beta)\frac{\tau}{3}} + \alpha),$$

where

$$\alpha = (k_3+\delta)(1+2\epsilon)(z_0+4d(k_1+\delta))\epsilon + 4(k_4+\delta),$$

$$\beta = (k_3-\delta)(1-\epsilon)(1-\epsilon-z_0-4d(k_1+\delta)\epsilon),$$

$$z_0 = 2\left(\frac{2d(k_1+\delta)\epsilon}{(k_3-\delta)(1-\epsilon)} + e^{-(k_3-\delta)(1-\epsilon)\frac{\tau}{3}}\right) + 2d(k_1+\delta)\epsilon\tau,$$

then

$$\Phi(w) \implies \Phi_2(wa).$$

Proof. Assume the constraints and $\Phi(w)$. Let $q \notin \widehat{\delta}(I, wa)$. By lemma 5.18

$$\begin{aligned} z_q((3|w|+1)\tau) &\leq 2\left(\frac{2d(k_1+\delta)\epsilon}{(k_3-\delta)(1-\epsilon)} + e^{-(k_3-\delta)(1-\epsilon)\frac{\tau}{3}}\right) + 2d(k_1+\delta)\epsilon\tau \\ &= z_0. \end{aligned}$$

By lemma 5.22 and constraint (1),

$$y_q((3|w|+1)\tau) \leq \gamma^*.$$

By lemma 5.24 we have

$$z_q((3|w|+2)\tau) \leq z_0 + 4d(k_1+\delta)\gamma^*\tau.$$

By lemma 5.26

$$\begin{aligned} y_q((3|w|+2)\tau + \frac{2\tau}{3}) &\leq \frac{2}{\alpha+\beta}(\beta e^{-(\alpha+\beta)\frac{\tau}{3}} + \alpha) \\ &\leq \gamma. \end{aligned}$$

Finally, by corollary 5.11,

$$y_q(3|wa|\tau) \leq \gamma.$$

□

Acknowledgments

We thank students and faculty auditors in our spring, 2015, molecular programming course for helpful and challenging discussions.

References

[1] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.

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

[3] R. Aris. Prolegomena to the rational analysis of systems of chemical reactions. *Archive for Rational Mechanics and Analysis*, 19(2):81–99, 1965.

[4] F. Bonchi and D. 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.

[5] F. Bonchi and D. Pous. Hacking nondeterminism with induction and coinduction. *Communications of the ACM*, 58(2):87–95, 2015.

[6] L. Cardelli. Morphisms of reaction networks that couple structure to function. *BMC Systems Biology*, 8:84, 2014.

[7] L. Cardelli and A. Csikász-Nagy. The cell cycle switch computes approximate majority. *Scientific Reports*, 2, 2012.

[8] Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig. Programmable chemical controllers made from DNA. *Nature Nanotechnology*, 8(10):755–762, 2013.

[9] J. Cheng, S. Sreelatha, R. Hou, A. Efremov, R. Liu, J. R. C. van der Maarel, and Z. Wang. Bipedal nanowalker by pure physical mechanisms. *Physical Review Letters*, 109:238104, 2012.

[10] M. Cook, D. Soloveichik, E. Winfree, and J. Bruck. Programmability of chemical reaction networks. In A. Condon, D. Harel, J. N. Kok, A. Salomaa, and E. Winfree, editors, *Algorithmic Bioprocesses*, Natural Computing Series, pages 543–584. Springer, 2009.

[11] F. Dannenberg, M. Kwiatkowska, C. Thachuk, and A. J. Turberfield. DNA walker circuits: Computational potential, design, and verification. In *Proceedings of the 19th International Conference on DNA Computing and Molecular Programming*, volume 8141 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2013.

[12] D. Doty, J. H. Lutz, M. J. Patitz, R. T. Schweller, S. M. Summers, and D. Woods. The tile assembly model is intrinsically universal. In *Proceedings of the 53rd Symposium on Foundations of Computer Science*, pages 302–310. IEEE, 2012.

[13] S. M. Douglas, I. Bachelet, and G. M. Church. A logic-gated nanorobot for targeted transport of molecular payloads. *Science*, 335(6070):831–834, 2012.

[14] S. M. Douglas, A. H. Marblestone, S. Teerapittayanon, A. Vazquez, G. M. Church, and W. M. Shih. Rapid prototyping of 3D DNA-origami shapes with caDNAno. *Nucleic Acids Research*, pages 1–6, 2009.

[15] I. R. Epstein and J. A. Pojman. *An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves, Patterns, and Chaos*. Oxford University Press, 1998.

[16] P. Érdi and J. Tóth. *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models*. Manchester University Press, 1989.

[17] J. Gunawardena. Chemical reaction network theory for in-silico biologists, 2003. `http://www.jeremy-gunawardena.com/papers`.

[18] D. Han, S. Pal, J. Nangreave, Z. Deng, Y. Liu, and H. Yan. DNA origami with complex curvatures in three-dimensional space. *Science*, 332(6027):342–346, 2011.

[19] T. A. Henzinger and J.-F. Raskin. The equivalence problem for finite automata: technical perspective. *Communications of the ACM*, 58(2):86–86, 2015.

[20] Y. Ke, L. L. Ong, W. M. Shih, and P. Yin. Three-dimensional structures self-assembled from DNA bricks. *Science*, 338(6111):1177–1183, 2012.

[21] D. Kozen. *Automata and Computability*. Springer, 1997.

[22] G. Lente. *Deterministic Kinetics in Chemistry and Systems Biology: The Dynamics of Complex Reaction Networks*. Springer, 2015.

[23] L. Qian and E. Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–1201, 2011.

[24] L. Qian and E. Winfree. A simple DNA gate motif for synthesizing large-scale circuits. *Journal of the Royal Society Interface*, 8(62):1281–1297, 2011.

[25] L. Qian, E. Winfree, and J. Bruck. Neural network computation with DNA strand displacement cascades. *Nature*, 475(7356):368–372, 2011.

[26] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

[27] P. W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[28] N. C. Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Biology*, 99(2):237 – 247, 1982.

[29] L. M. Smith. Nanotechnology: Molecular robots on the move. *Nature*, 465(7295):167–168, 2010.

[30] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck. Computation with finite stochastic chemical reaction networks. *Natural Computing*, 7(4):615–633, 2008.

[31] D. Soloveichik, G. Seelig, and E. Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398, 2010.

[32] G. Teschl. *Ordinary Differential Equations and Dynamical Systems*, volume 140 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

[33] A. S. Walsh, H. Yin, C. M. Erben, M. J. A. Wood, and A. J. Turberfield. DNA cage delivery to mammalian cells. *ACS Nano*, 5(7):5427–5432, 2011.

[34] B. Wei, M. Dai, and P. Yin. Complex shapes self-assembled from single-stranded DNA tiles. *Nature*, 485(7400):623–626, 2012.

[35] E. Winfree. *Algorithmic self-assembly of DNA*. PhD thesis, California Institute of Technology, 1998.

[36] D. Woods. Intrinsic universality and the computational power of self-assembly. In *Proceedings of the International Conference on Machines, Computations and Universality*, volume 128 of *Electronic Proceedings in Theoretical Computer Science*, pages 16–22. Open Publishing Association, 2013.

[37] B. Yurke, A. J. Turberfield, A. P. Mills, F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406(6796):605–608, 2000.

[38] D. Y. Zhang and G. Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature chemistry*, 3(2):103–113, 2011.