Safety-Aware Cyber-Molecular Systems

Robyn R. Lutz (PI), Eric R. Henderson, James I. Lathrop, Jack H. Lutz
Iowa State University

Robust Biomolecular Finite Automata*

Overview

We introduce **input/output chemical reaction networks** (**I/O CRNs**) which generalize the behavior of CRNs under deterministic mass-action semantics to allow input to be provided over time via a chemical signal.

We present a uniform **compiler** for translating an arbitrary nondeterministic finite automaton (NFA) into an I/O CRN that simulates it.

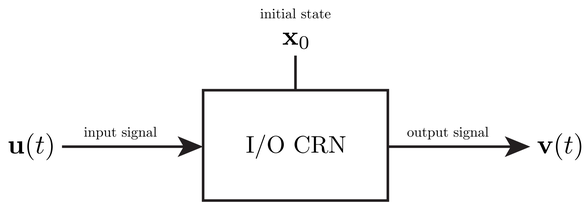
The I/O CRN exploits the inherent parallelism of chemical kinetics to simulate the NFA in **real time** with a number of chemical species that is **linear** in the number of states of the NFA.

We show this translation is **robust** with respect to perturbations of the (i) initial state; (ii) input signal; (iii) output signal; and (iv) rate constants.

Robustness is a **necessary** condition for safety.

Input / Output Chemical Reaction Networks

I/O CRNs are a generalization of CRNs that receive input via **signals** over time.



$N = (S, R, U, V)$ is an I/O CRN where

- S is a set of **state species**;
- U is a set of **input species**;
- $V \subseteq S$ is a set of **output species**;
- R is a set of **reactions** over $S \cup U$;

Species in U are only used as catalysts in N.

Starting from an initial state, $\mathbf{x}_0$, the I/O CRN N continuously reads the input signal $\mathbf{u}(t)$ and produces an output signal $\mathbf{v}(t)$.

The concentration signals $\mathbf{u}(t)$ and $\mathbf{v}(t)$ are simply vectors of concentrations of the species in U and V, respectively.

Robustness in I/O CRNs

We define four ways an I/O CRN $N = (S, R, U, V)$ can be robust.

Inital state:	works for all $\mathbf{x}_0^*$ near $\mathbf{x}_0$
Input signal:	works for all $\mathbf{u}^*(t)$ near $\mathbf{u}(t)$
Output signal:	works for all $\mathbf{v}^*(t)$ near $\mathbf{v}(t)$
Rate constants:	works for all $k^*(t)$ near k

Note that $k^*(t)$ is a function of time! Our notion of robustness with respect to rate constants must account for **adversaries** capable of changing the rate constants over time.

Howver, we still use the intuitive notation

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

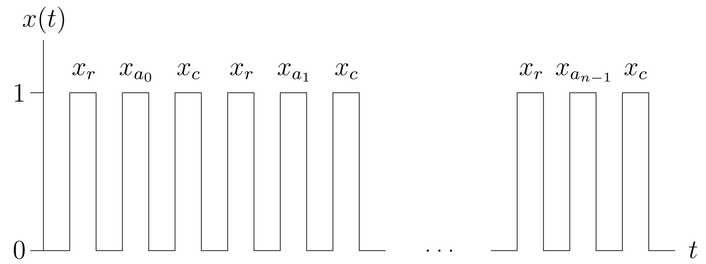
Encoding Strings into Signals

Suppose we wish to encode the string $w = a_0 a_1 a_2 \cdots a_{n-1}$ over an alphabet Σ as a concentration signal.

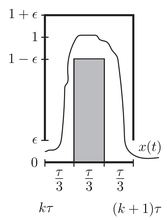
It is useful to introduce two new symbols, r and c, and replace each a_i with the padded ra_ic string yielding

$$\hat{w} = ra_0cra_1cra_2c \cdots ra_{n-1}c.$$

Ideally we would like to encode the modified string $\hat{w}$ as a square wave.



Robustness with respect to the input signal must allow for approximations to the ideal input.



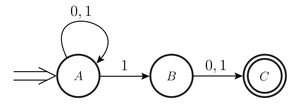
Any signal $x(t)$ that stays within the bounds is close enough.

Nondeterministic Finite Automata

An NFA is a **state transition system** that reads an input string and outputs "accept" or "reject."

$M = (Q, \Sigma, \Delta, I, F)$ is an NFA where

- Q is a set of **states**
- Σ is an **input alphabet**
- Δ is a **transition function**
- $I \subseteq Q$ is a set of **initial states**
- $F \subseteq Q$ is a set of **final states**



This NFA "accepts" bit strings with a "1" as the second-to-last bit.

Compiling NFAs into I/O CRNs

We construct an I/O CRN ($N = (S, R, U, V)$ from an NFA $M = (Q, \Sigma, \Delta, I, F)$. The input string is provided to N as a signal, and N outputs its decision via the concentration of its output species after reading the input.

We show that N is robust with respect to all four types of robustness described.

Input Species (U)

We use $|\Sigma| + 2$ input species.

- X_a for each $a \in \Sigma$;
- X_r for **setup**;
- X_c for **teardown**.

The input signal always comes in triples: $X_r \to X_a \to X_c$.

State / Output Species (S, V)

We use $4|Q|$ state species. For $q \in Q$

- $Y_q, \bar{Y}_q$ to **store** the current state;
- $Z_q, \bar{Z}_q$ to **compute** next states.

Y_q and Z_q are **dual railed** with $\bar{Y}_q$ and $\bar{Z}_q$.

Output species: each Y_q for $q \in F$.

Reactions (R)

We use $5|Q| + |\Delta|$ reactions

- $|Q|$ reactions use X_r to **setup** to receive the next symbol.
- $|\Delta|$ reactions use X_a to **compute** the transition function.
- $2|Q|$ reactions use X_c to **store** the next states.
- $2|Q|$ reactions use approximate majority to **maintain** values.

Example: for each transition $(s, a, q) \in \Delta$, we have

$$X_a + Y_s + \bar{Z}_q \xrightarrow{k} X_a + Y_s + Z_q$$

* T. H. Klinge, J. I. Lathrop, and J. H. Lutz, arxiv.org/abs/1505.03931

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