Deterministic Function Computation with Chemical Reaction Networks

Ho-Lin Chen[1], David Doty[2], David Soloveichik[3]

[1]National Taiwan University [2]Caltech [3]UCSF

The programming language of chemical kinetics

Use the language of coupled chemical reactions *prescriptively* as a “programming language” for engineering new systems (rather than *descriptively* as a modeling language for existing systems)

These gloves came free with my toilet brush!

Real programmers code in CHEMISTRY

Cells are smart: controlled by signaling and regulatory networks

Human neutrophil chasing a bacterium through red blood cells

source: David Rogers, Vanderbilt University

Want to understand principles of chemical computation

Engineer embedded controllers for biochemical systems, “wet robots”, smart drugs, etc.

Chemical Reaction Networks (CRN)

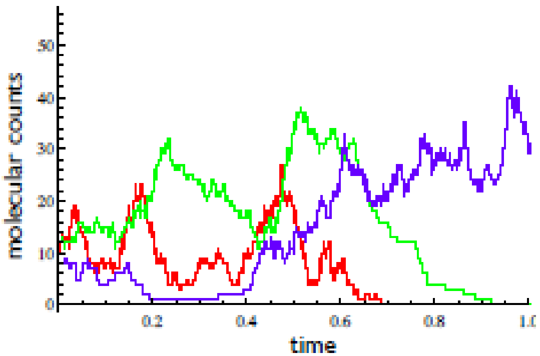
syntax:

$$
\begin{aligned}
A &\xrightarrow{0.03} 2A \\
2A &\xrightarrow{5\times10^4} A \\
B + A &\xrightarrow{10^5} 2B \\
B &\xrightarrow{0.01} \\
A + C &\xrightarrow{10^5} \\
C &\xrightarrow{0.0165} 2C \\
2C &\xrightarrow{5\times10^4} C
\end{aligned}
$$

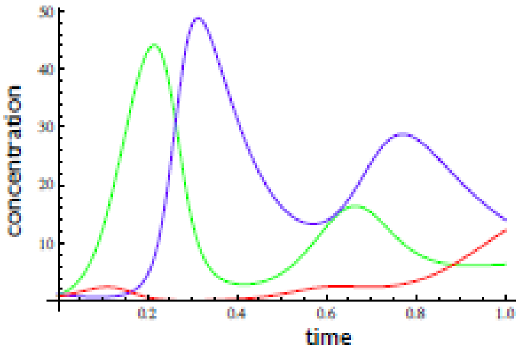
two possible semantics:

stochastic: discrete state space, continuous time Poisson process

we use only stochastic CRNs in this talk



mass-action: continuous ODEs



Discrete (Stochastic) CRN Model

- Finite set of **species** $\{X, Y, Z, \ldots\}$
- A **state** is a nonnegative integer vector **c** indicating the *count* (number of molecules) of each species: write counts as $\#_{\mathbf{c}}X$, $\#_{\mathbf{c}}Y$, …
- Finite set of **reactions**: e.g.

$$X \rightarrow W + Y + Z$$

$$A + B \rightarrow C$$

(in our paper, all rate constants are 1, and all reactions are unimolecular or bimolecular)

Discrete (Stochastic) CRN Model

System evolves via a **continuous time Poisson process:**

reaction j propensity ρ_j

- $A \rightarrow \ldots$ $\#A$
- $A + B \rightarrow \ldots$ $(1/v)\ \#A\ \#B$ v = volume
- $A + A \rightarrow \ldots$ $(1/v)\ \#A\ (\#A - 1) / 2$

time until next reaction is exponential random variable with rate $\Sigma_j \rho_j$ (and expected value $1 / \Sigma_j \rho_j$)

probability that the next reaction is j^* is $\rho_{j^*} / \Sigma_j \rho_j$

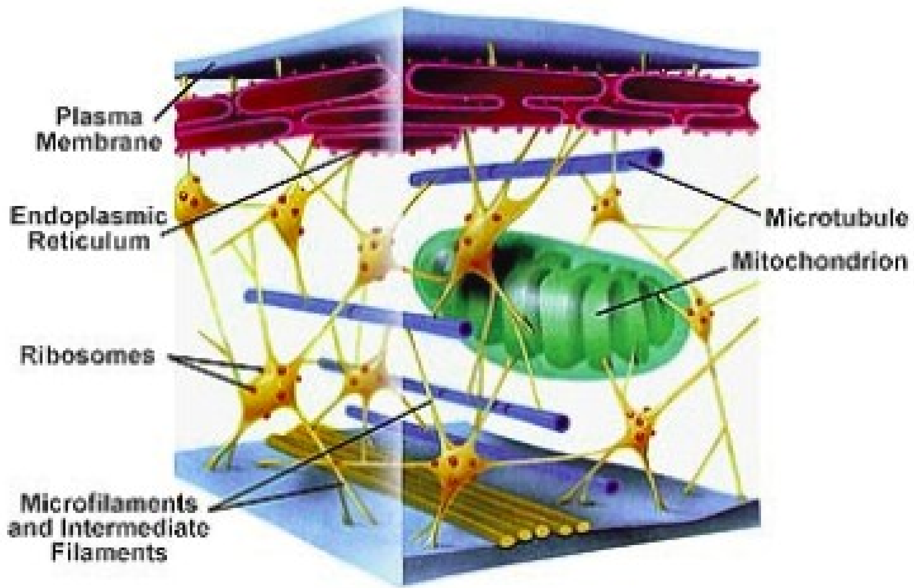
Objections?

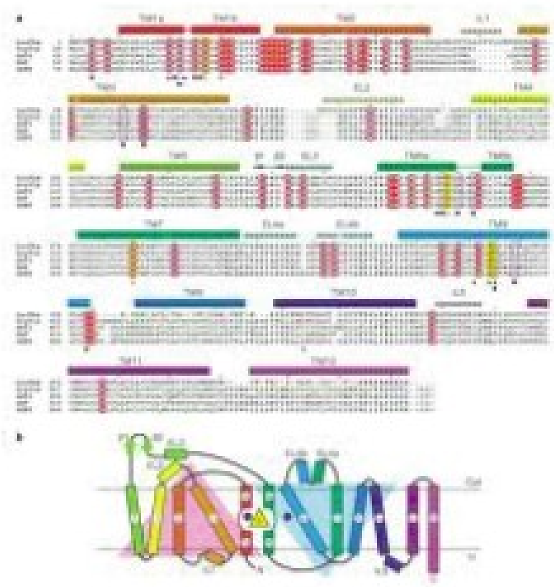
What is not captured?

Localization, space, assembly/
disassembly, movement

Combinatorial species

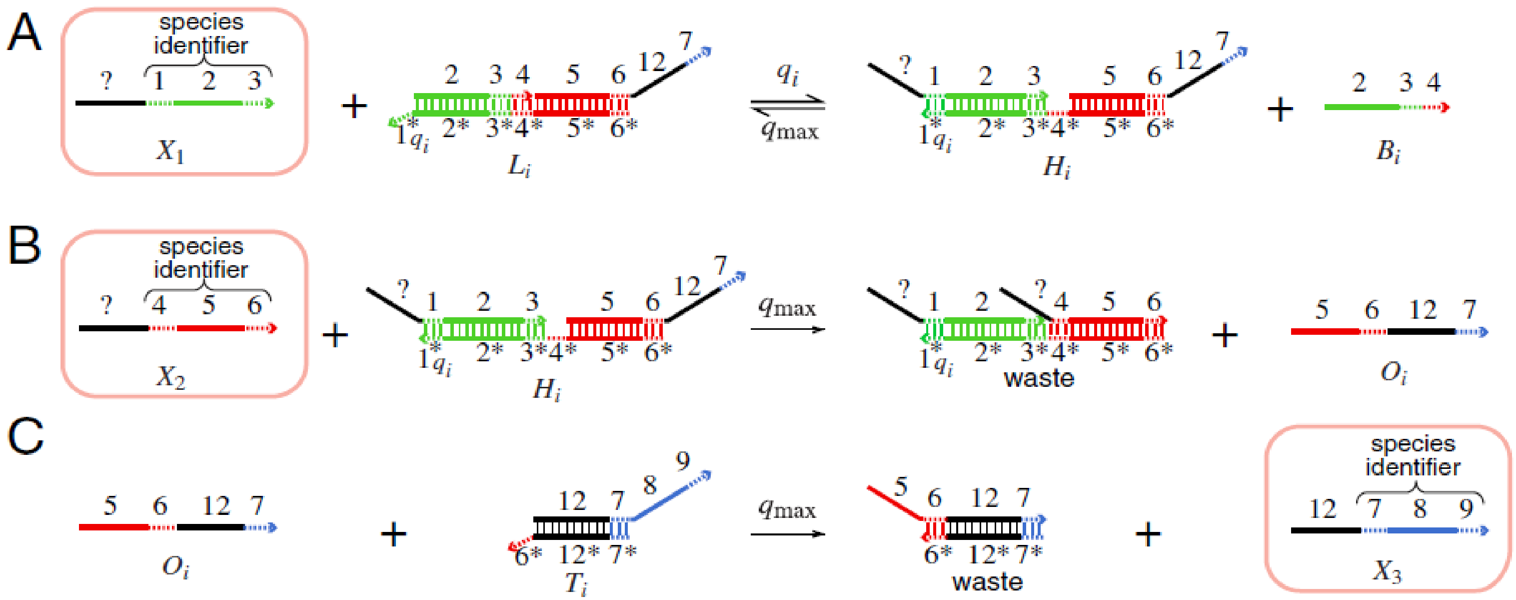
Biological examples





Are CRNs an "implementable" programming language?

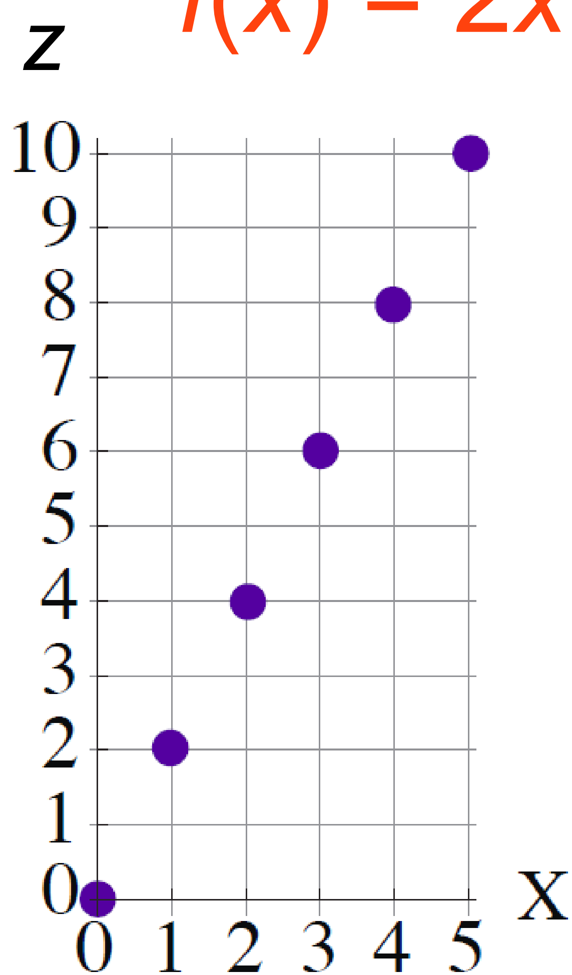
- "I don't believe that every crazy CRN you write down actually describes real chemicals!"
- **Response to objection**: Soloveichik, Seelig, Winfree [*PNAS* 2010] found a physical implementation (high-accuracy approximation) of any CRN, using *nucleic-acid strand displacement cascades*



Deterministic Function Computation with CRNs

Deterministic function computation with CRNs (example 1)

$f(x) = 2x$

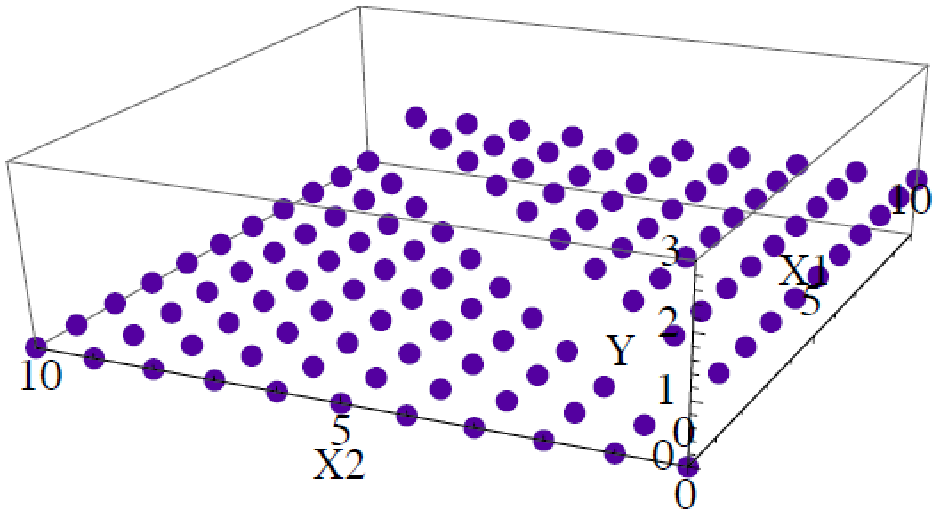


start with x (input amount) of X

$X \rightarrow Z + Z$

Deterministic function computation with CRNs (example 2)

$f(x_1,x_2)$ = if $x_1 > x_2$ then $y = 1$ else $y = 0$



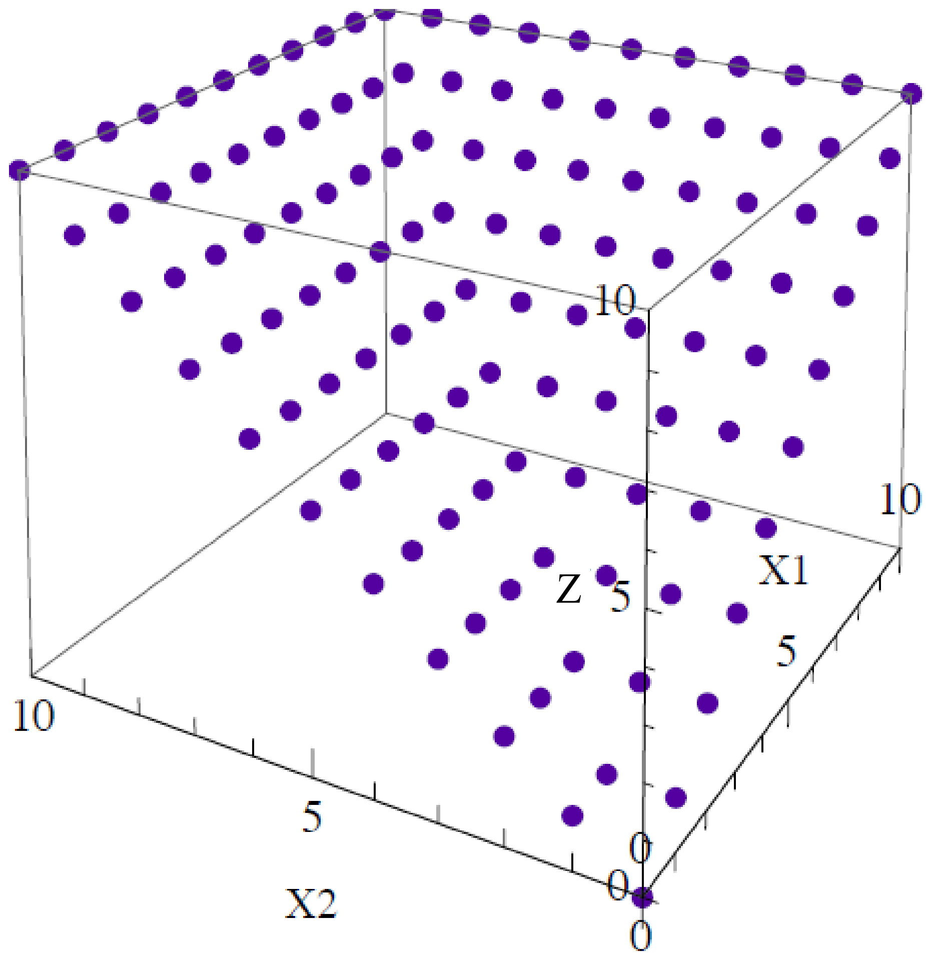
start with 1 N and input amounts of X_1,X_2

$X_1 + N \rightarrow Y$

$X_2 + Y \rightarrow N$

Deterministic function computation with CRNs (example 3)

$f(x_1,x_2) = \max\{x_1,x_2\}$



start with input amounts of X_1,X_2

$X_1 \rightarrow Z_1 + Z$

$X_2 \rightarrow Z_2 + Z$

$Z_1 + Z_2 \rightarrow K$

$K + Z \rightarrow \emptyset$

Deterministic function computation with CRNs (definition)

task: compute function $\mathbf{z} = f(\mathbf{x})$ $(\mathbf{x} \in \mathbb{N}^k, \mathbf{z} \in \mathbb{N}^l)$

- **initial state**: input counts $X_1, X_2, \ldots, X_k$ (and fixed counts of non-input species)
- **output**: counts of $Z_1, Z_2, \ldots, Z_l$
- **output-stable state**: all states reachable from it have same counts of $Z_1, Z_2, \ldots, Z_l$
- **deterministic computation**: a correct output-stable state "always reached in the limit $t \to \infty$" (infinitely often reachable states are infinitely often reached)

Other functions?

- $f(x) = x/2$?
- $f(x) = x^2$?
- $f(x_1,x_2) = x_1 \cdot x_2$?
- $f(x) = 2^x$?

Main result

Theorem: Functions f: $\mathbb{N}^k \to \mathbb{N}^l$ deterministically computable by CRNs are precisely those with a *semilinear graph*. graph(f) = $\{ (\mathbf{x},\mathbf{z}) \in \mathbb{N}^{k+l} \mid f(\mathbf{x})=\mathbf{z} \}$

$A \subseteq \mathbb{N}^{k+l}$ is linear if there are vectors $\mathbf{b}$, $\mathbf{u}_1$, …, $\mathbf{u}_p$

so that $A = \{ \mathbf{b} + n_1 \cdot \mathbf{u}_1 + \ldots + n_p \cdot \mathbf{u}_p \mid n_1,\ldots,n_p \in \mathbb{N} \}$

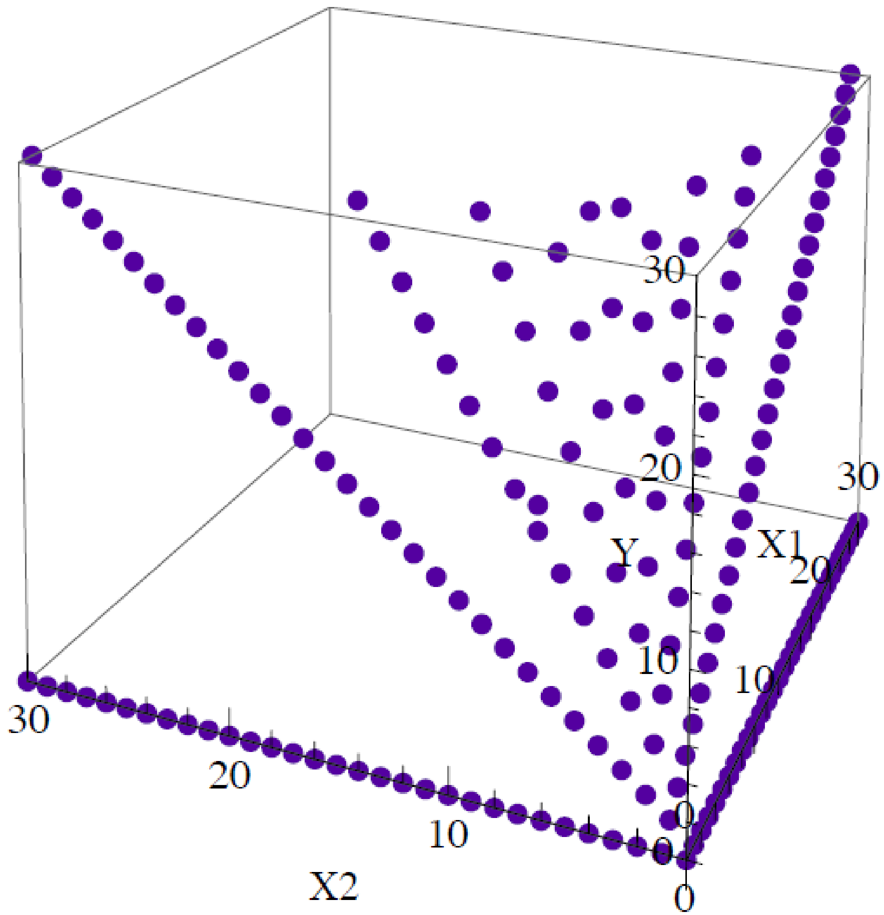
A is semilinear if it is a finite union of linear sets.

Intuitively, semilinear functions are "piecewise linear functions" with a finite number of pieces

Non-semilinear examples

$f(x_1,x_2) = x_1 \cdot x_2$

no finite union of linear sets

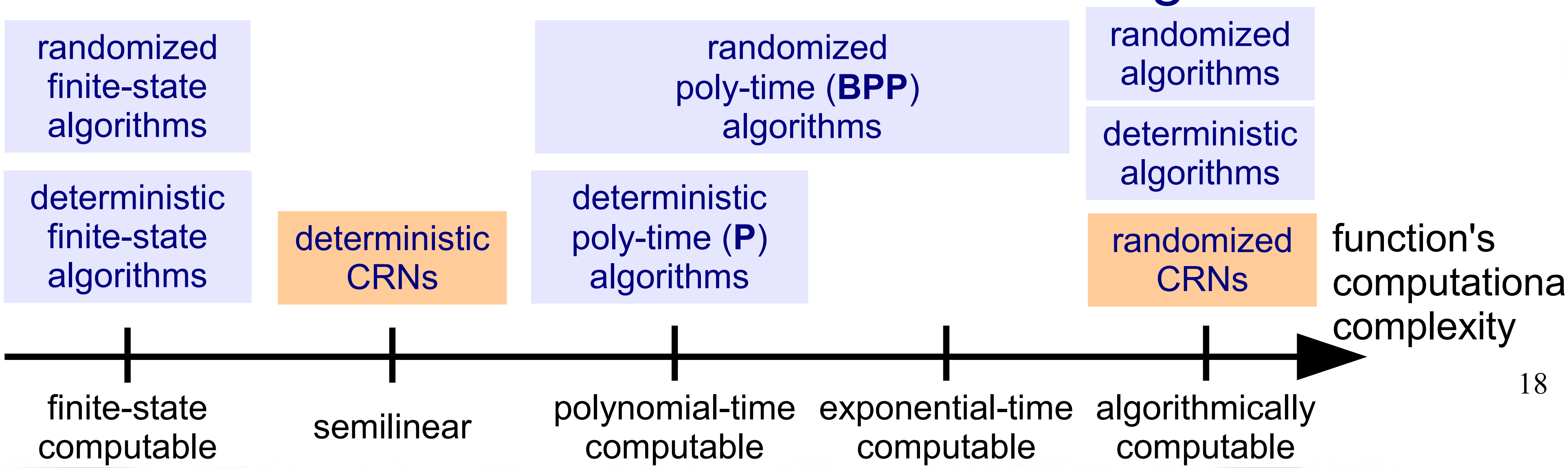


Others:

- $f(x) = x^2$
- $f(x) = 2^x$

What if we allow error?

- Any function computable by an algorithm is computable by a randomized CRN with arbitrarily small positive probability of error.
  - [Soloveichik, Cook, Winfree, Bruck, *Natural Computing* 2008]
  - [Angluin, Aspnes, Eisenstat, *Distributed Computing* 2006]
- **Moral**: disallowing error hurts chemical algorithms much more than it hurts conventional algorithms



How do we show this?

Theorem [Angluin, Aspnes, Eisenstat, PODC 2006]: The *predicates* decidable by CRNs are precisely the semilinear predicates.

We connect computation of *functions* (integer output) to computation of *predicates* (YES/NO output)

Deterministic predicate computation with stochastic CRNs (definition)

task: decide predicate $b = \varphi(\mathbf{x})$ ($\mathbf{x} \in \mathbb{N}^k$, $b \in \{\text{yes,no}\}$)

- **initial state**: input counts $X_1, X_2, \ldots, X_k$ (and fixed counts of non-input species)
- **output**: either $\#Y > 0$ and $\#N = 0$ (yes)
  or $\#Y = 0$ and $\#N > 0$ (no)
- **output-stable state**: all states reachable from it have same yes/no answer
- **set decided by CRN**: $S_{yes} = \{ \mathbf{x} \in \mathbb{N}^k \mid \varphi(\mathbf{x}) = \text{yes} \}$

Two directions to proof

(reminder) **Theorem** [Angluin, Aspnes, Eisenstat, PODC 2006]: The sets decidable by CRNs are precisely the semilinear sets.

- Only semilinear functions can be computed:

  f computed by CRN $C \Rightarrow$ graph(f) decided by CRN D
- All semilinear functions can be computed:

  graph(f) decided by CRN $D \Rightarrow f$ computed by CRN C

f computed by CRN $C \Rightarrow$ graph(f) decided by CRN D

- Want to decide, given input (x,z), is $f(x) = z$?
- Keep track of total number of Z's ever produced or consumed:

$$A + B \rightarrow Z + W \text{ becomes } A + B \rightarrow Z + W + Z_P$$

$$A + Z \rightarrow B \qquad \text{becomes } A + Z \rightarrow B + Z_C$$

- Initial state has z copies of Z_C

Eventually all Z_P and Z_C go away (if equal) or one is left over (if unequal)

$$Z_P + Z_C \rightarrow Y \qquad Z_P + Y \rightarrow Z_P + N$$

$$Y + N \rightarrow Y \qquad Z_C + Y \rightarrow Z_C + N$$

If Z_P or Z_C are left over, change answer to NO

If neither is left over, change answer to YES

graph(f) decided by CRN $D \Rightarrow$ f computed by CRN C

- Want: given x copies of X, produce $f(x)$ copies of Z
- If graph(f) = $\{ (x,z) \in \mathbb{N}^2 \mid f(x) = z \}$ is semilinear, then so is the the set

$$F_{\text{diff}} = \{ (x,z_P,z_C) \in \mathbb{N}^3 \mid f(x) = z_P - z_C \}$$

- So some CRN D_{diff} decides F_{diff}
- Start with 0 of Z, Z_P, Z_C, and add to D_{diff} the reactions

N only present when D_{diff} thinks answer is NO

$$N \rightarrow N + Z_P + Z$$
$$N + Z \rightarrow N + Z_C$$

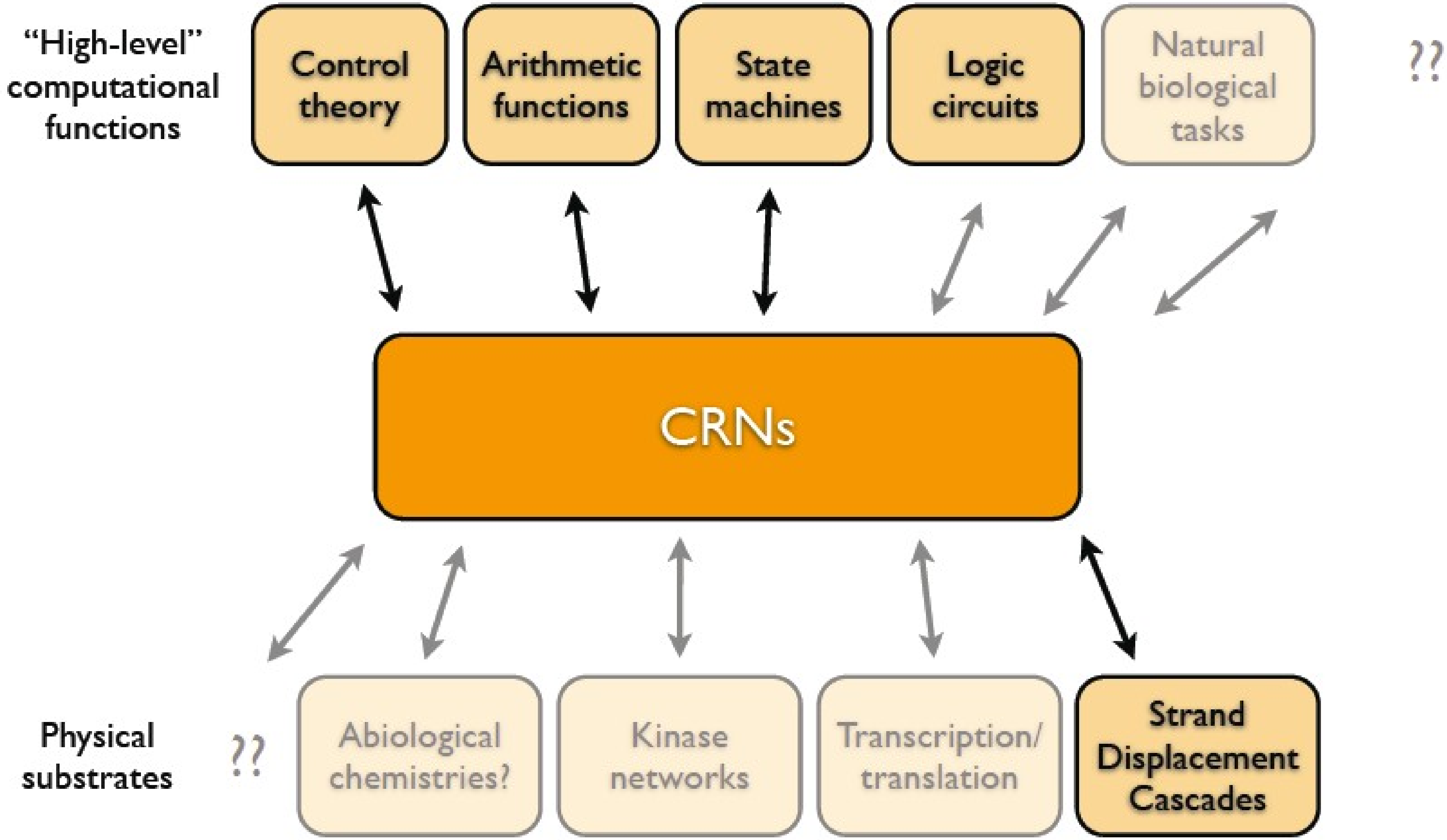
This is *really* slow!

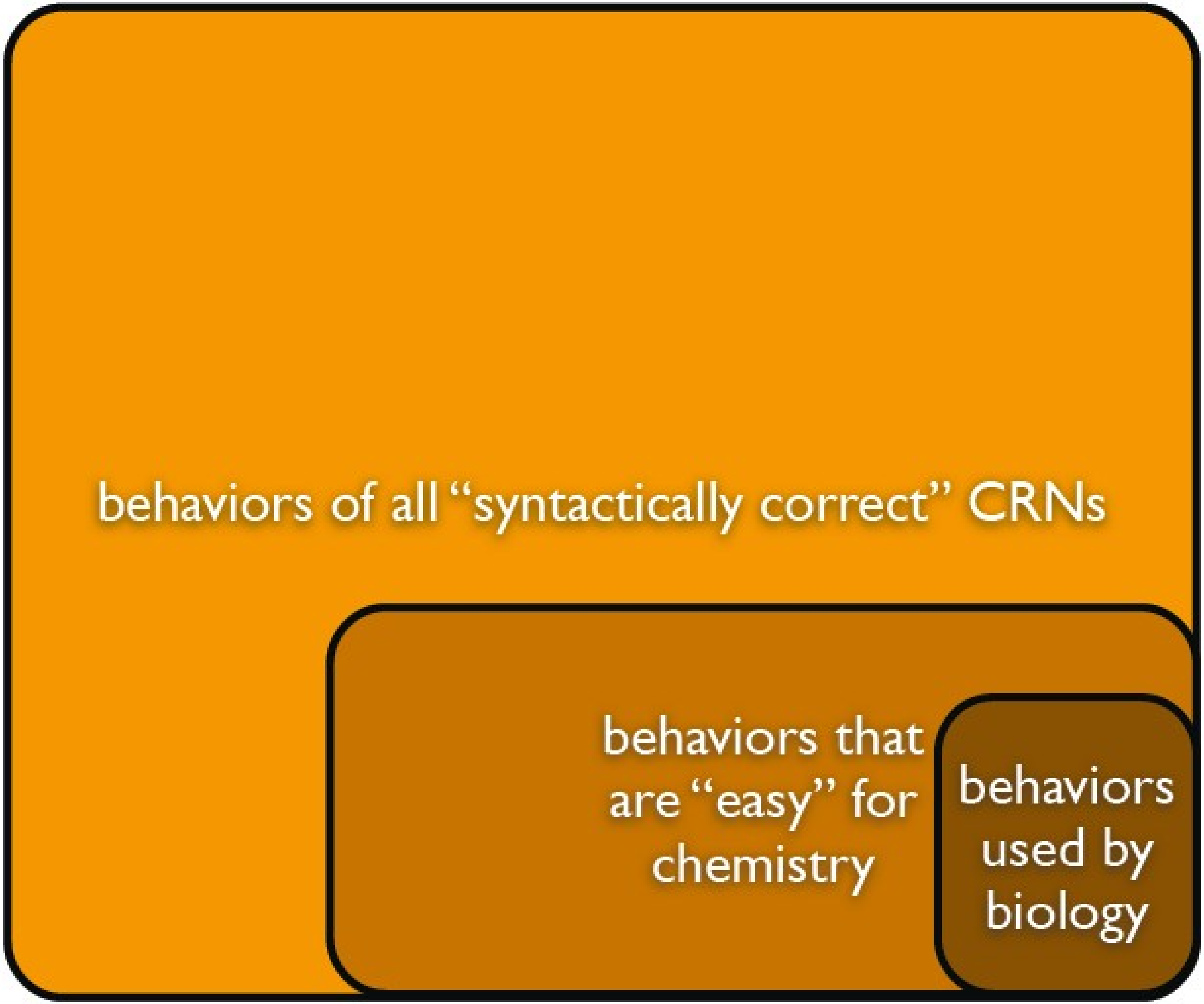
How fast can semilinear functions be computed?

Theorem: Every semilinear function f can be computed by a CRN on input **x** in expected time $O(\log^5 ||\mathbf{x}||)$. $||\mathbf{x}|| = \Sigma_i\, \mathbf{x}(i)$

i.e., in time $O(n^5)$, where n is the number of bits needed to write **x** in binary

Proof: MATH.


"High-level" computational functions
Control theory
Arithmetic functions
State machines
Logic circuits
Natural biological tasks
??
CRNs
Physical substrates
??
Abiological chemistries?
Kinase networks
Transcription/ translation
Strand Displacement Cascades


behaviors of all "syntactically correct" CRNs
behaviors that are "easy" for chemistry
behaviors used by biology

Acknowledgments

Ho-Lin Chen

David Soloveichik

Damien Woods

UCSF
center for systems & synthetic biology
an NIGMS national systems biology center

NSF

COMPUTING RESEARCH ASSOCIATION CRA

COMPUTING COMMUNITY CONSORTIUM CCC

Niranjan Srinivas

Thank you!