Computing with Chemical Reaction Networks

Nikhil Gopalkrishnan

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)

Real programmers code in CHEMISTRY

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Chemical Reaction Networks (CRN)

syntax:

$$A \xrightarrow{0.03} 2A$$

$$2A \xrightarrow{5\times10^4} A$$

$$B + A \xrightarrow{10^5} 2B$$

$$B \xrightarrow{0.01}$$

$$A + C \xrightarrow{10^5}$$

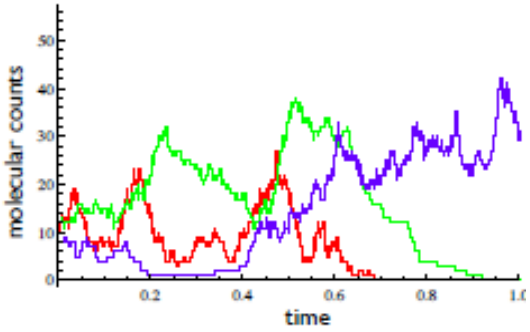
$$C \xrightarrow{0.0165} 2C$$

$$2C \xrightarrow{5\times10^4} C$$

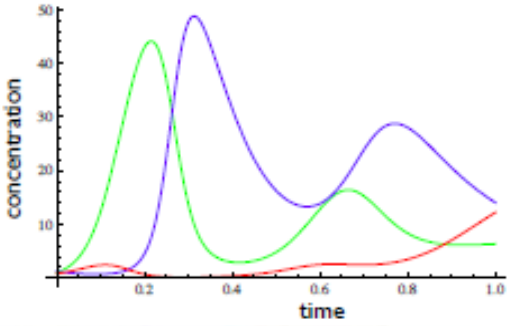
two possible semantics:

stochastic: discrete state space, continuous time Poisson process

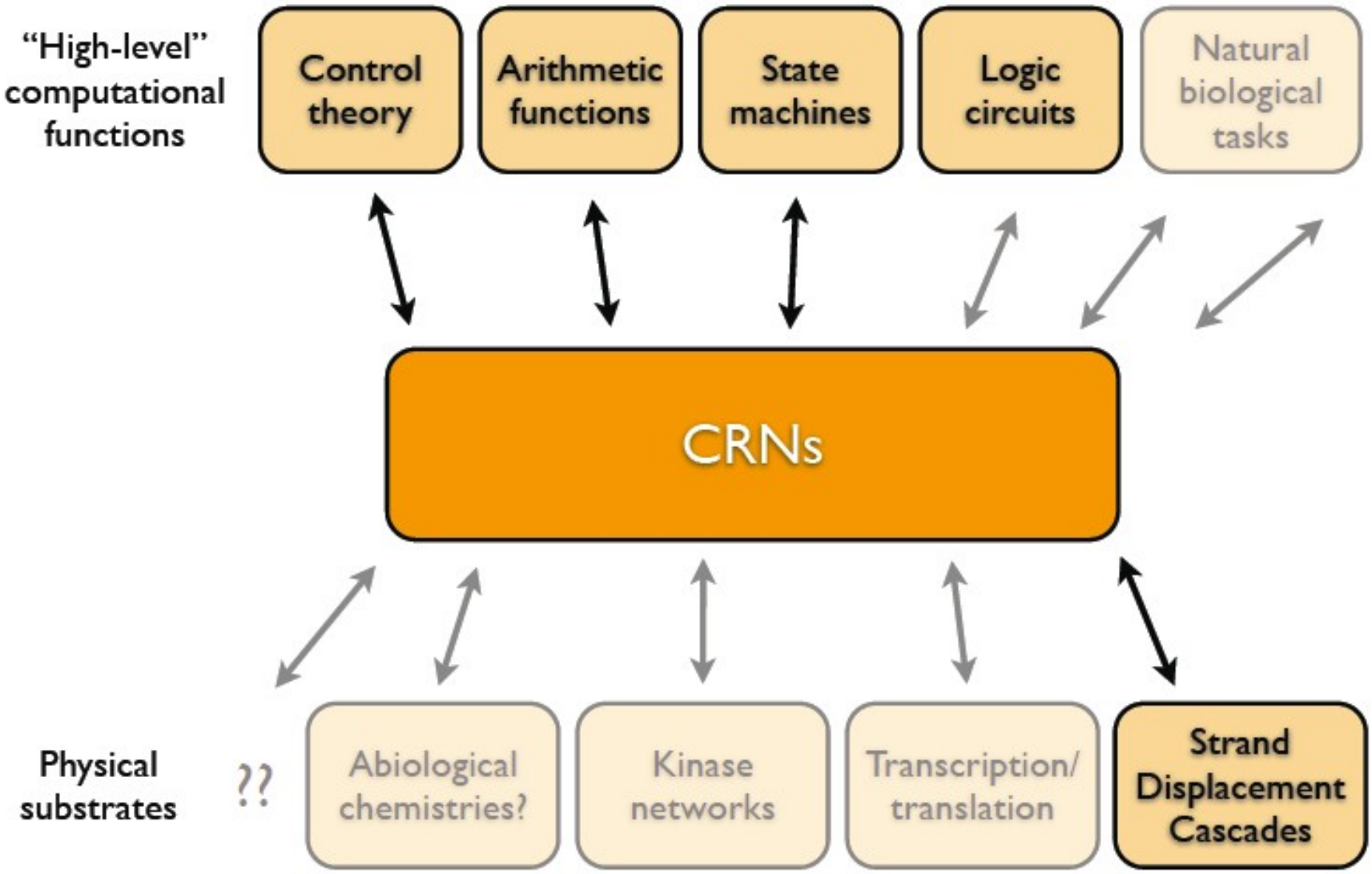
we use only stochastic CRNs in this talk



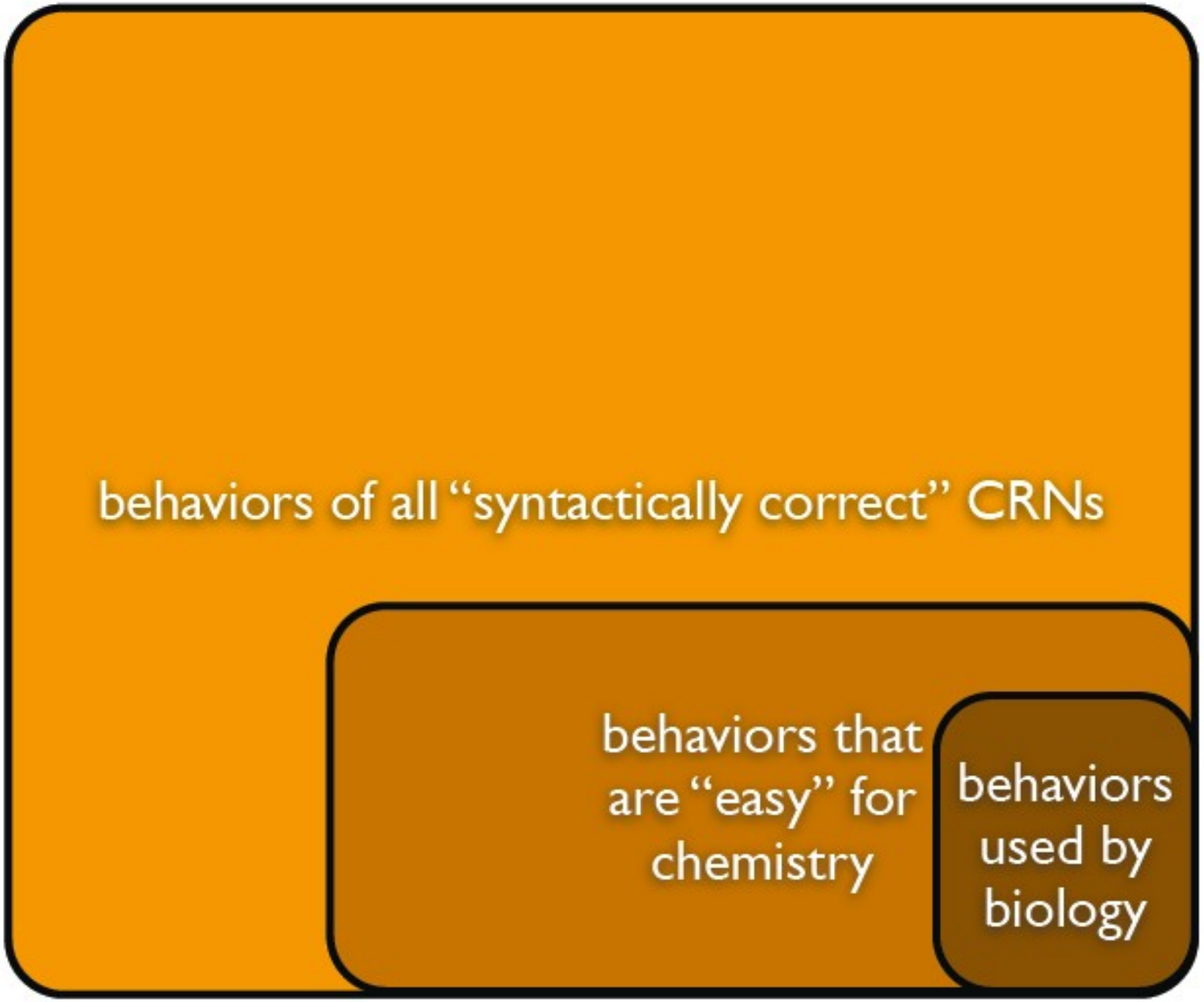
mass-action: continuous ODEs



(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)



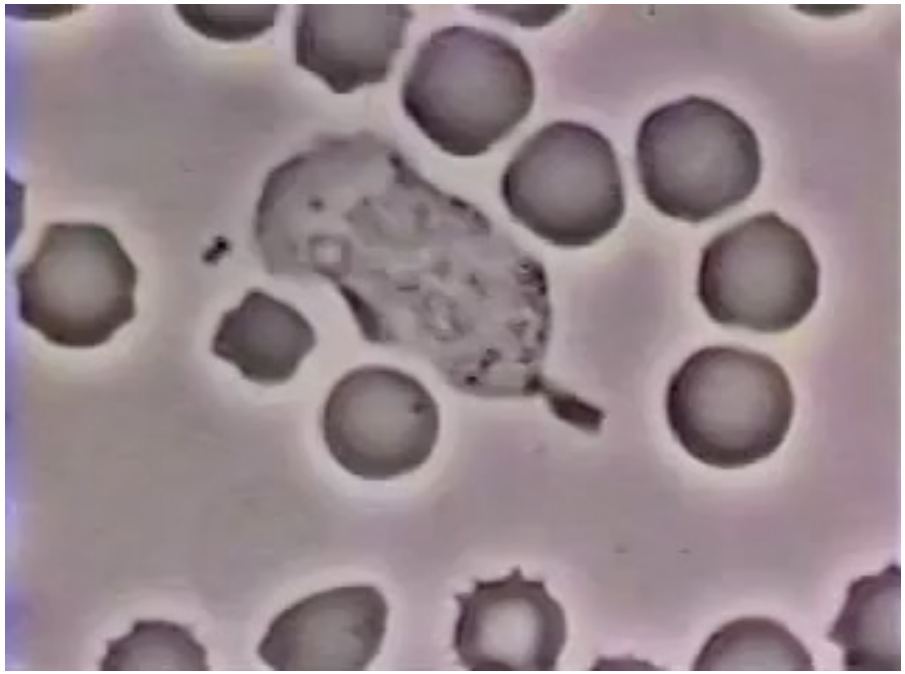
(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)



(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Cells are smart: controlled by signaling and regulatory networks

Human neutrophil chasing a bacterium through red blood cells



source: David Rogers, Vanderbilt University

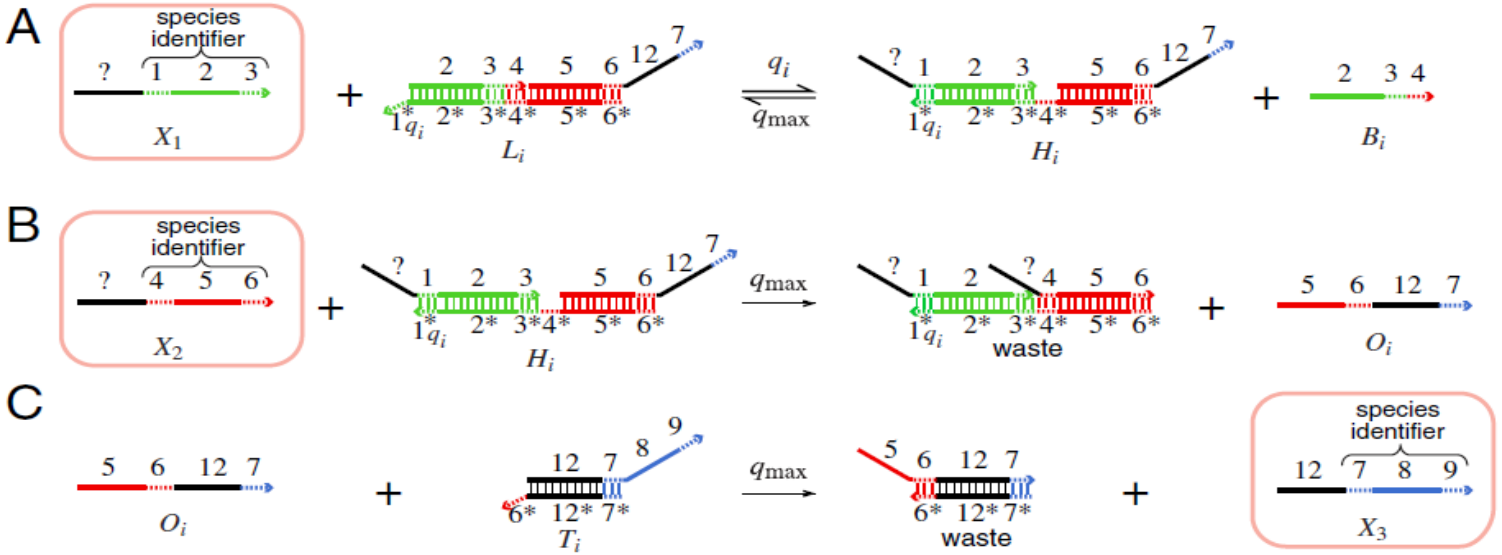
Want to understand principles of chemical computation

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

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

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*



(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Formal Definition of Discrete (Stochastic) CRN Model

- Finite set of **species** {X, Y, Z, …}
- 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.

  $X \rightarrow W + Y + Z$

  $A + B \rightarrow C$

- (assume all rate constants are 1, and all reactions are unimolecular or bimolecular)

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Stochastic Chemical Reaction Network

Stochastic Chemical Reaction Network(SCRN): a finite set of d reactions acting on a finite number m of species.

The **stoichiometry** is the nonnegative number of copies of each species required for a reaction to take place, or produced when the reaction does take place.

A reaction α is defined as:

- **Stoichiometry of the reactants: $r_\alpha = (r_{\alpha,1},\ldots, r_{\alpha,m})$.**
  - A vector of nonnegative integers.
- **Stoichiometry of the products, $p_\alpha = (p_{\alpha,1}, \ldots, p_{\alpha,m})$.**
  - A vector of nonnegative integers.

Stochastic Chemical Reaction Network

- We will use **capital letters** to refer to various **species** and we will use standard chemical notation to describe reactions.
- **Example:**
- **Reaction A + D → A + 2E**
  - **Consumes 1 molecule of species A and 1 molecule of species D**
  - **Produces 1 molecule of species A and 2 molecules of species E.**
- In this reaction, A acts catalytically because it must be present for the reaction to occur, but its number is unchanged when the reaction does occur.

Stochastic Chemical Reaction Network

State of the network: $A = (q_1, \ldots, q_m)$.

- a vector of nonnegative integers specifying the quantities present of each species.

For reaction α:

- **Stoichiometry of reactants: $r_\alpha = (r_{\alpha,1}, \ldots, r_{\alpha,m})$.**
- **Stoichiometry of the products, $p_\alpha = (p_{\alpha,1}, \ldots, p_{\alpha,m})$.**

Reaction α is possible in state A:

- Only if there are enough reactants present, that is, $\forall i,\ q_i \geq r_{\alpha,i}$.

When reaction α occurs in state A:

- The reactant molecules are used up and the products are produced.
- New state: $B = (q_1 - r_{\alpha,1} + p_{\alpha,1}, \ldots, q_m - r_{\alpha,m} + p_{\alpha,m})$.

Stochastic Chemical Reaction Network

Reaction Notation:

- We write A→ B if there is some reaction in the Stochastic Chemical Reaction Network C that can change A to B.
- We write $\rightarrow^*$ for the reflexive transitive closure of →.

Probabilistic Reactions:

- We write Pr[A → B] to indicate the probability that, given that the state is initially A, the next reaction will transition to the state to B.

Discrete (Stochastic) CRN Model

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

Reaction *j*	**Propensity ρ_j**	
• $A \rightarrow \ldots$	$\#A$	
• $A + B \rightarrow \ldots$	$(1/v)\ \#A\ \#B$	*where* 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 $\rho_k / \Sigma_j \rho_j$ that the next reaction is the kth reaction.

Stochastic Chemical Reaction Networks Papers

Computation With Finite Stochastic Chemical Reaction Networks (Soloveichik,Cook, Winfree, Bruck, 1985)

DNA as a universal substrate for chemical kinetics(Soloveichik, Seelig, Winfree, PNAS2010)

Programmability of Chemical Reaction Networks(Chen, Doty, Soloveichik)

Simulation of Boolean circuits using CRNs

Programmability of Chemical Reaction Networks(Chen, Doty, Soloveichik)

Given: Boolean Circuit: **The Boolean Circuit suffices to use only one type of Boolean operation: NAND:**

$$x_k = x_i \text{ NAND } x_j$$

Construct: Simulating CRN:

$$A_i + A_j \rightarrow A_i + A_j + B_k,$$

$$A_i + B_j \rightarrow A_i + B_j + B_k,$$

$$B_i + A_j \rightarrow B_i + A_j + B_k,$$

$$B_i + B_j \rightarrow B_i + B_j + A_k.$$

- **It deterministically computes the same function as the given Boolean circuit, despite the uncontrollable order in which reactions occur.**
- **The presence of a single A_i molecule represents that $x_i = 0$, the presence of a single B_i molecule represents that $x_i = 1$, and the presence of neither indicates that x_i has not yet been computed.**
- **If one starts with a single A or B molecule for each input variable, then with probability 1 the correct species will be eventually produced for each output variable.**

Computations using CRNs

Programmability of Chemical Reaction Networks(Chen, Doty, Soloveichik)

Four representations of the same computation:

Starting with 1 A and n C's, the maximum number of D's that can be produced is 2n.

a)

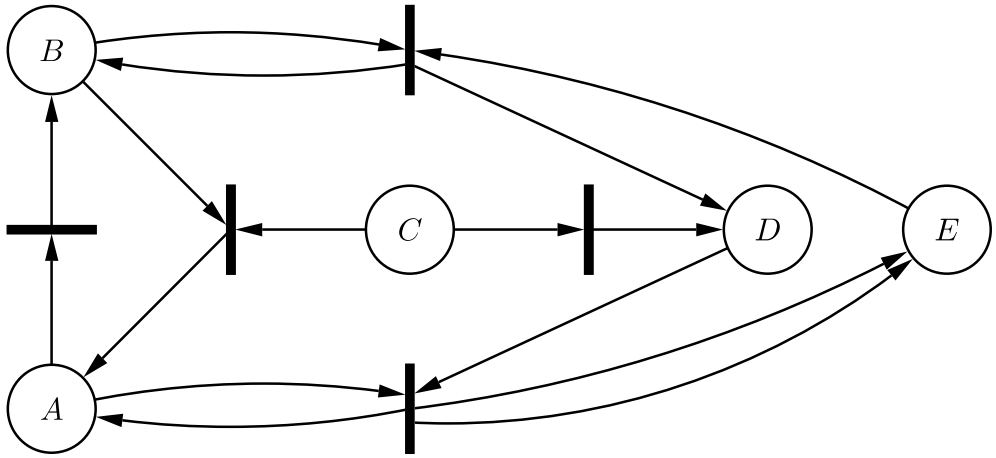
(a) A Stochastic Chemical Reaction Network:

$$\begin{aligned} A &\rightarrow B \\ C &\rightarrow D \\ B + C &\rightarrow A \\ A + D &\rightarrow A + 2E \\ B + E &\rightarrow B + D \end{aligned}$$

b)

(b) A Petri net:

Each circle corresponds to a *place* (a molecular species), and each black bar corresponds to a *transition* (a reaction).



c)

(c) A Vector Addition System: **Note that dimensions F and G must be added to the Vector Addition System to capture the two reactions that are catalyzed by A and B.**

$$\begin{array}{ccccccc} A & B & C & D & E & F & G \\ \hline \langle -1 & 1 & 0 & 0 & 0 & 0 & 0 \rangle \\ \langle 0 & 0 & -1 & 1 & 0 & 0 & 0 \rangle \\ \langle 1 & -1 & -1 & 0 & 0 & 0 & 0 \rangle \\ \langle -1 & 0 & 0 & -1 & 0 & 1 & 0 \rangle \\ \langle 1 & 0 & 0 & 0 & 2 & -1 & 0 \rangle \\ \langle 0 & -1 & 0 & 0 & -1 & 0 & 1 \rangle \\ \langle 0 & 1 & 0 & 1 & 0 & 0 & -1 \rangle \end{array}$$

(d) A Fractran program:

- **The numerators correspond to the reaction products, and the denominators correspond to the reactants.**
- **The first seven prime numbers are used here in correspondence to the letters A through G in the other examples.**
- **As in the previous example, F (13) and G (17) must be introduced to avoid unreduced fractions for the catalyzed reactions**

d)

$$\frac{3}{2} \quad \frac{7}{5} \quad \frac{2}{15} \quad \frac{13}{14} \quad \frac{242}{13} \quad \frac{17}{33} \quad \frac{21}{17}$$

Deterministic Function Computation with Chemical Reaction Networks

Ho-Lin Chen, David Doty, and David Soloveichik

Deterministic Function Computation with CRNs

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

- **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)

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

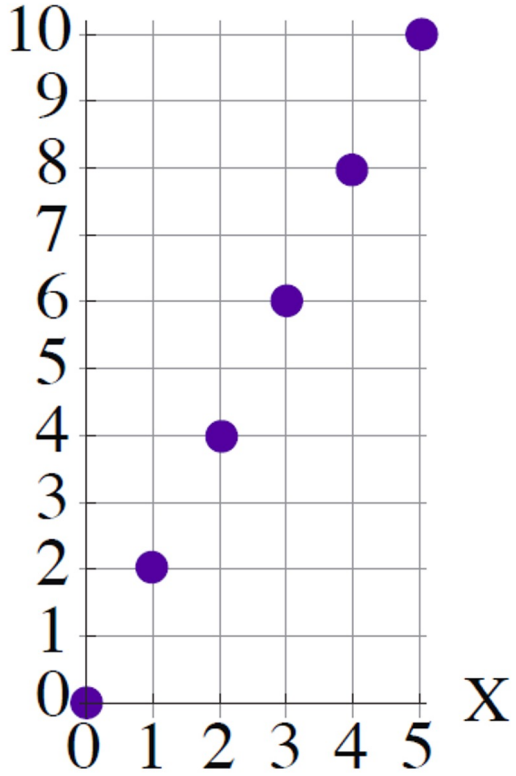
Example 1 of Deterministic Function Computation with CRNs

Task is to Compute: f(x) = 2x

Start with input amount x of *X*

Reaction: X → Z + Z

Output z

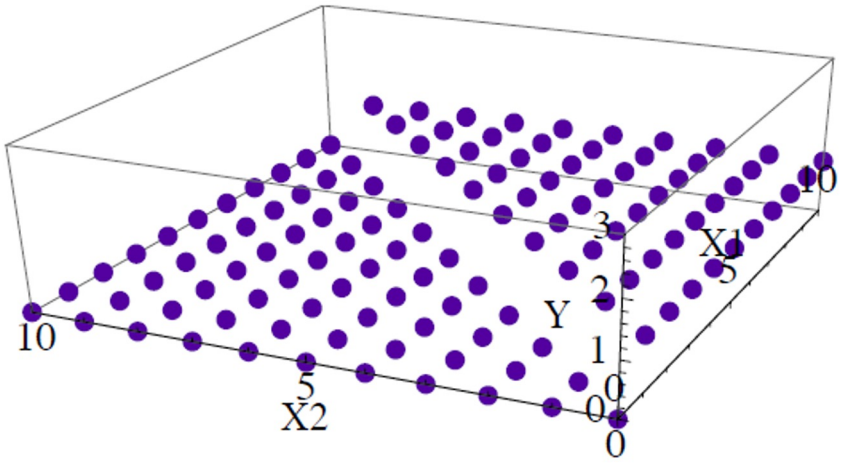


(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Example 2 of Deterministic Function Computation with CRNs

Task is to Compute:

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



start with 1 N and input amounts of X_1,X_2

$X_1 + N \rightarrow Y$

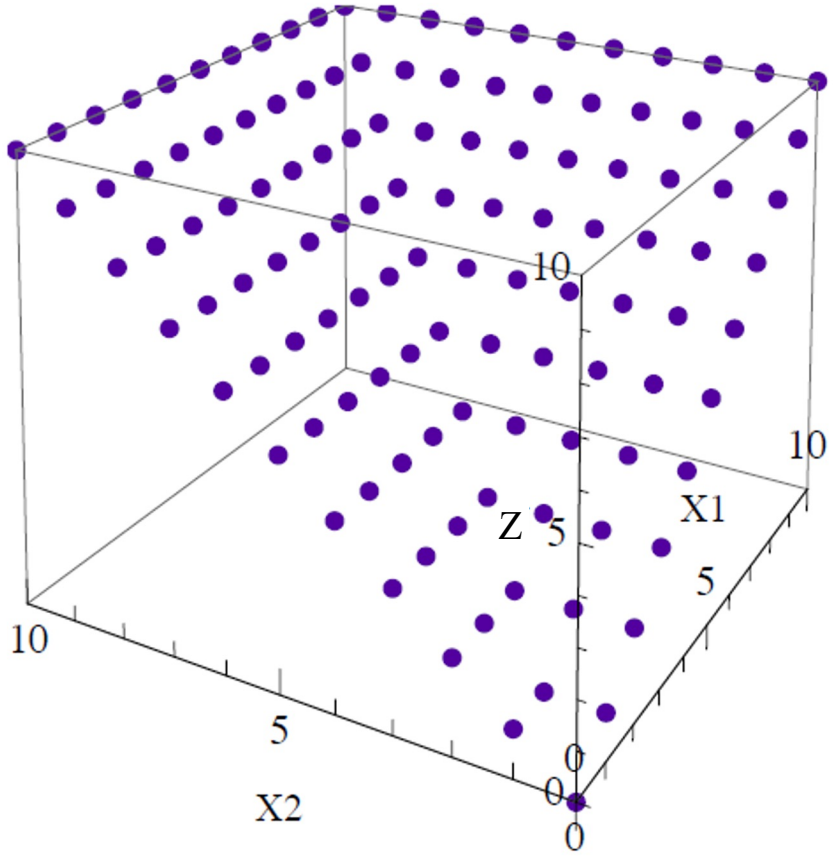
$X_2 + Y \rightarrow N$

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Example 3 of Deterministic Function Computation with CRNs

Task is to Compute:

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



start with input amounts of X_1,X_2

$X_1 \rightarrow Z_1 + Z$
$X_2 \rightarrow Z_2 + Z$
$Z_1 + Z_2 \rightarrow K$
$K + Z \rightarrow \emptyset$

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Other Deterministic Functions: Can they be computed with CRNs?

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

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Deterministic Functions Computed with CRNs

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.

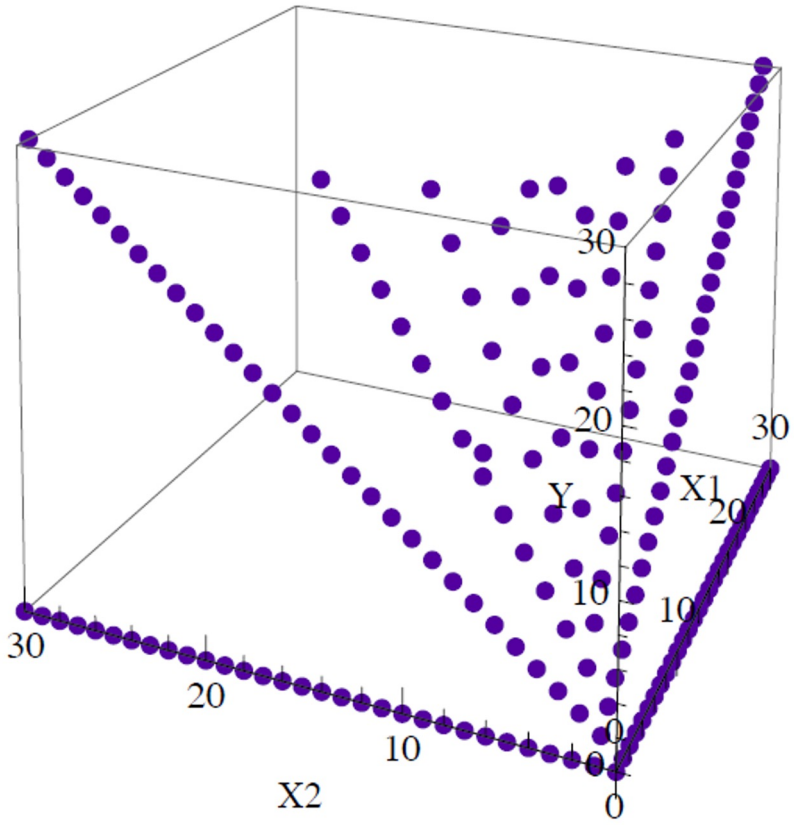
Intuitively, semilinear functions are "piecewise linear functions" with a finite number of pieces

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Non-semilinear examples

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

no finite union of linear sets



Other functions not semilinear:

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

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

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)

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Deterministic predicate computation with stochastic CRNs:

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**: $S_{yes} = \{ \mathbf{x} \in \mathbb{N}^k \mid \varphi(\mathbf{x}) = \text{yes} \}$

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

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

Two directions of proof

Only semilinear functions can be computed:

- *f* computed by CRN *C* ⇒ graph(*f*) decided by CRN *D*

All semilinear functions can be computed:

- graph(*f*) decided by CRN *D* ⇒ *f* computed by CRN *C*

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

f computed by CRN *C* ⇒ 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:

  $A + B \rightarrow Z + W$ becomes $A + B \rightarrow Z + W + Z_P$

  $A + Z \rightarrow B$ becomes $A + Z \rightarrow B + Z_C$

- Initial state has *z* copies of Z_C

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

  $Y + N \rightarrow Y$ $\quad$ $Z_C + Y \rightarrow Z_C + N$

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

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

If neither is left over, change answer to YES

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

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 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

$$N \longrightarrow N + Z_P + Z$$
$$N + Z \longrightarrow N + Z_C$$

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

CRN Simulations of: Minsky's register machine (RM)

A Finite State Machine with a fixed number of registers

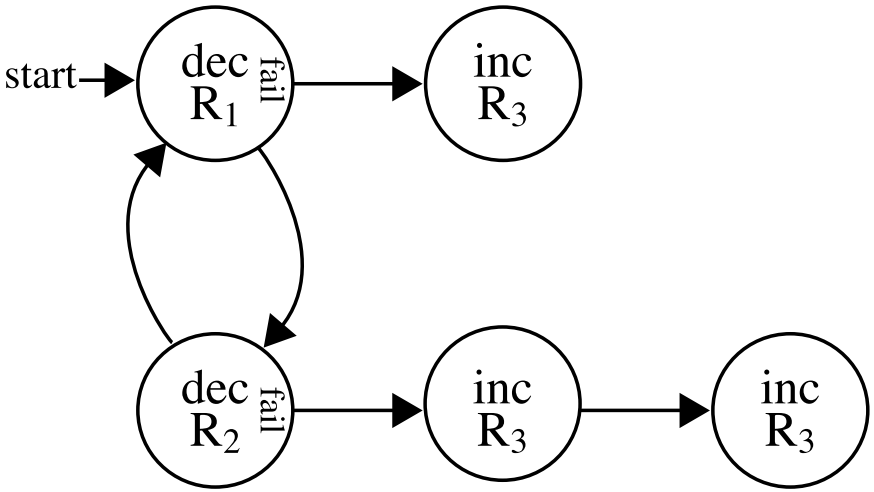
- Each register can store a non-negative integer
- Inc(i,r,j) : Increment register r and move from state i to j
- Dec(I,r,j,k): Decrement register r if r > 0 and move from state i to j; else move to state k

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Minsky's register machine (RM):

A Finite state machine with a fixed number of registers

- Each register can store a non-negative integer
- Inc(i,r,j) : Increment register r and move from state i to j
- Dec(I,r,j,k): Decrement register r if r > 0 and move from state i to j; else move to state k



A register machine comparing the value of register R_1 to R_2. If $R_1 \leq R_2$, then it outputs 1 in register R_3 . If $R_1 > R_2$ then it outputs 2 in register R_3 .

The start state is indicated with "start" and the halting states are those without outgoing arrows

(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Computations using CRNs

Programmability of Chemical Reaction Networks(Chen, Doty, Soloveichik)

Simulating a register machine:

(a) The communication between the clock and the register logic modules is through single molecules of species C and T .

(**b**) The **clock module** is responsible for producing a C molecule once every so often.

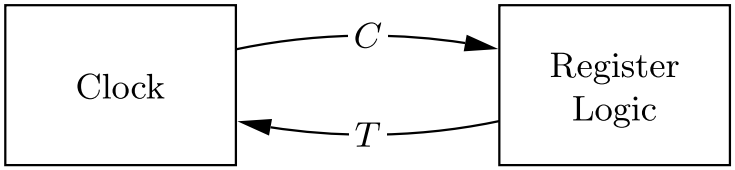
- The clock module is designed so that the length of time between receiving a T and producing a C slowly increases throughout the computation, thus slowing down the register logic module to help it avoid error.
- The more A's there are, the longer the delay.
- The clock starts out with n_0 A's and one each of B, B', and B'' and T .
- Every clock cycle not only produces a C, but increases the number of A's by one.
- Thus, at the beginning of the kth cycle, there are n = k + n_0 molecules of A.

(c) The register logic module simulates the register machine state transitions.

- The register logic module starts out with quantities of molec
- ules of R_i indicating the starting value of register i, and a single molecule of species S_a where a is the start state of the register machine.
- Note that at all times the entire system contains at most a single molecule of any species other than the A and R_i species.

All rate constants are 1 (The construction will work with any rate constants)

(a) The Clock and Register Logic Modules



(b) Clock

$$
\begin{aligned}
T + B &\to T' + B \\
T' + A &\to T + A \\
T' + B' &\to T'' + B' \\
T'' + A &\to T' + A \\
T'' + B'' &\to C + B'' + A
\end{aligned}
$$

(c) Register Logic

In state a increment register i and go to state b $\Rightarrow$ $S_a + C \to S_b + R_i + T$

In state a decrement register i and go to state b, or if the register is empty go to state c

$$
\Rightarrow \quad \begin{aligned} S_a + R_i &\to S'_a \\ S'_a + C &\to S_b + T \\ S_a + C &\to S_c + T \end{aligned}
$$

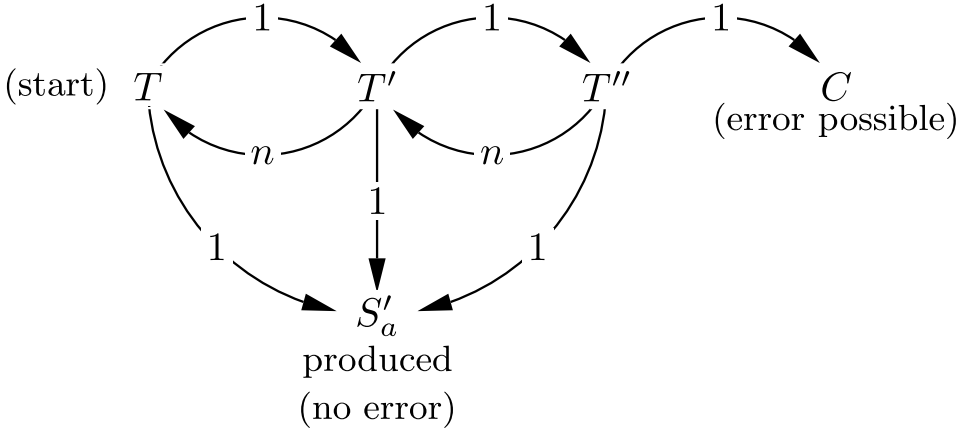
Computations using CRNs

Programmability of Chemical Reaction Networks(Chen, Doty, Soloveichik)

Simulating a register machine:

The state diagram for a single decrement operation when:

- There are n A's and the register to be decremented holds the value 1, and
- The corresponding system of differential equations governing the instantaneous probabilities of being in a given state.



The *numbers on the arrows:* are the transition rates.

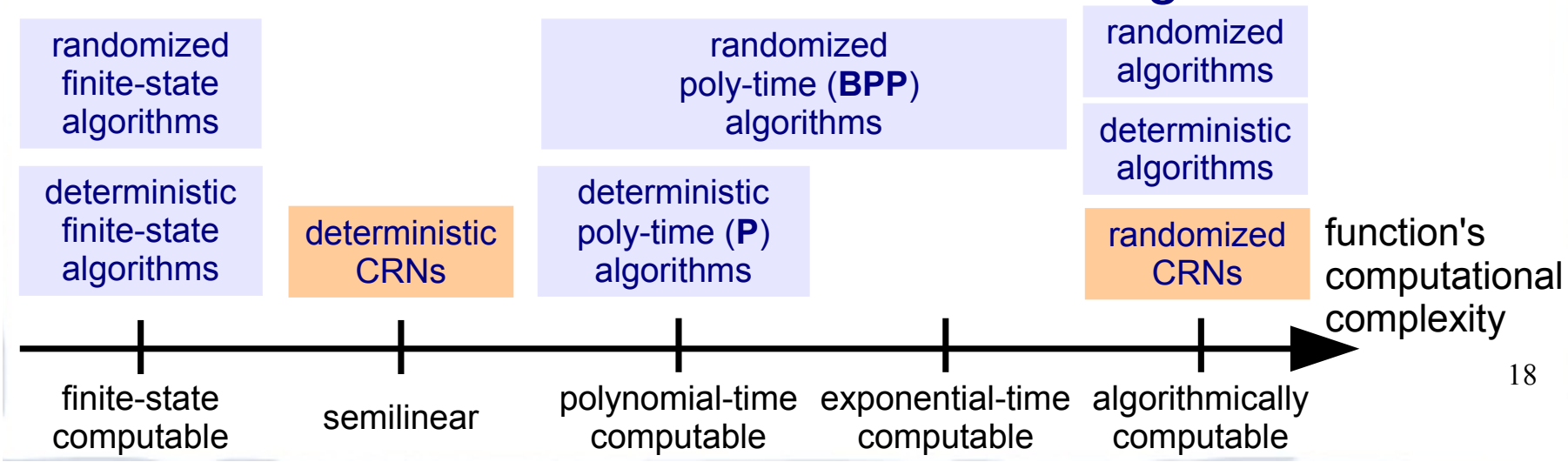
instantaneous probabilities:

- The instantaneous probability of being in state T is s, in state T ′ is s′, and in state T ″ is s″.
- The instantaneous probability of being in the error-possible state is p and
- The probability of being in the no-error state is q

$$\frac{d}{dt}\begin{bmatrix} s \\ s' \\ s'' \\ p \\ q \end{bmatrix} = \begin{bmatrix} -2 & n & 0 & 0\ 0 \\ 1 & -2-n & n & 0\ 0 \\ 0 & 1 & -2-n & 0\ 0 \\ 0 & 0 & 1 & 0\ 0 \\ 1 & 1 & 1 & 0\ 0 \end{bmatrix} \cdot \begin{bmatrix} s \\ s' \\ s'' \\ p \\ q \end{bmatrix}$$

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



(From presentation of Chen, Doty, Soloveichik, Deterministic Function Computation with Chemical Reaction Networks.)

Computation With Finite Stochastic Chemical Reaction Networks

(Soloveichik,Cook, Winfree, Bruck, 1985)

- **Turing-universal computation using molecular counts**
- **Fast and reliable**
- **Small number of distinct molecular species**
- **Probability of error can be made arbitrarily small (> 0) by increasing molecular counts**
- **But require assumption:**

  - fast reactions are guaranteed to occur before any slow reaction.

Computation With Finite Stochastic Chemical Reaction Networks

(Soloveichik,Cook, Winfree, Bruck, 1985)

- **SCRNs are Turing universal** and thus can compute any computable function without error, *assuming fast reactions are guaranteed to occur before any slow reaction.*

- **SCRNs can compute any computable function** with probability of error less than ε for any ε > 0, but *for ε = 0 universal computation is impossible*.

- **SCRNs are NOT capable of universal computation with any fixed bounded probability of success**, if *each reaction's probability of occuring depends only on what reactions are possible (but not on the concentrations).*

- **SCRNs are capable of computing exactly the class of primitive recursive functions** without error, if we take the result of the *longest possible sequence of reactions as the answer*.

- The **time and space requirements** for Stochastic Chemical Reaction Networks doing computation, compared to a Turing Machine, are a *simple polynomial slowdown in time, but an exponential increase in space*.

Naive Simulation of Register Machine (RM) by Stochastic CRN

$inc(i, r, j)$ $\quad S_i \rightarrow S_j + M_r$

$dec(i, r, j, k)$ $\quad dec_1: S_i + M_r \rightarrow S_j$

$dec_2: S_i \rightarrow S_k$

Error Worse when $M_r = 1$

Error per step = $K_2 / (K_1 / v + K_2)$

K_1 = rate constant for dec_1

K_2 = rate constant for dec_2

Improved Bounded RM Simulation

	Rxn	Logical function
(*inc*)	$C + S_i \rightarrow S_j + M_r + C$	$inc(i, r, j)$
(dec_1)	$S_i + M_r \rightarrow S_j$	$dec(i, r, j, k)$
(dec_2)	$C_1 + S_i \rightarrow S_k + C_l$	

Bounded RM simulation:

Species C (#C = 1) acts a dummy catalyst to ensure that all reactions are bimolecular, simplifying the analysis of how the simulation scales with the volume.

Initial molecular counts are:

- if i is the start state then $\#S_i = 1$,
- $\#S_j = 0$ for $j \neq i$, and
- $\#M_r$ is the initial value of register r.

Improved Bounded RM Simulation

Rxn	Catalysts
$C_l \rightarrow C_{l-1}$	A^*
$C_{l-1} \rightarrow C_l$	A
$\vdots$	$\vdots$
$C_3 \rightarrow C_2$	A^*
$C_2 \rightarrow C_3$	A
$C_2 \rightarrow C_1$	A^*
$C_1 \rightarrow C_2$	A

Clock module for the RM simulation:

Dummy Catalyst A∗ : acts as a dummy catalyst to ensure that all reactions in the clock module are bimolecular and thus scale equivalently with the volume, ensuring that the error probability is independent of the volume.

Clock Module: The clock module maintains the average concentration of C_i at approximately $(\#A^*)^i/(\#A)^{i-1}$.

Initial molecular counts are: $\#C_i = 1$, and $\#C_1 = \cdots = \#C_{i-1} = 0$.

For the RM simulation $\#A^* = 1$, and $\#A = \Theta(1/\varepsilon^{i/(i-1)})$.