Microsoft Research

UNIVERSITY OF OXFORD

Programming with Chemical Reactions

Luca Cardelli, Microsoft Research & Oxford University

IMT Lucca, 2017-03-30

Credit: David Soloveichik

Why are chemical reactions interesting?

$$X + Y \xrightarrow{r} Z + W$$

- A fundamental model of kinetics (i.e. "behavior") in the natural sciences
- A fundamental mathematical structure, rediscovered in many forms
  - Vector Addition Systems, Petri Nets, Bounded Context-Free Languages, Population Protocols, …
- A programming language (coded up in the genome) by which living things manage the processing of matter and information

#1 Discrete (-state) Semantics

- A *state* of the system is a finite multiset of molecules; each molecule belongs to one of a finite set of *species*.
- A fixed finite set of *reactions* over species performs multiset-rewriting over those states.
- Reactions have rates: the state space is a Continuous-Time Markov Chain (a labeled transition system where labels are transition speeds).
- Hence the semantics is discrete and stochastic = atomic theory of matter.

Programming Examples

spec	*program*
Y = 2X	X -> Y + Y
Y = ⌊X/2⌋	X + X -> Y
Y = X1 + X2	X1 -> Y X2 -> Y
Y = min(X1, X2)	X1 + X2 -> Y

Advanced Programming Examples

spec

Y = max(X1, X2)

program

X1 -> L1 + Y
X2 -> L2 + Y
L1 + L2 -> K
Y + K -> 0

max(X1,X2)=
(X1+X2)-min(X1,X2)

(but is not computed "sequentially")

Approximate Majority

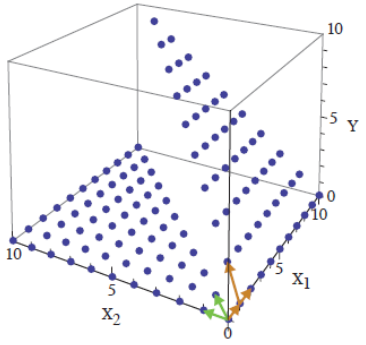
(X,Y) :=
if X≥Y then (X+Y, 0)
if Y≥X then (0, X+Y)

X + Y -> Y + B
Y + X -> X + B
B + X -> X + X
B + Y -> Y + Y

What can we compute this way?

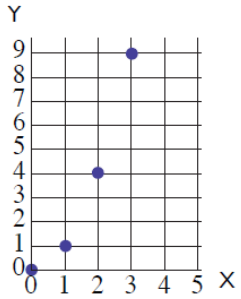
- The semilinear functions
  - Those whose graph is a finite union of linearly-bounded regions

$f(x_1, x_2)=x_2$ if $x_1>x_2$ and 0 otherwise



$\{n_1 \cdot (1,1,0) + n_2 \cdot (0,1,0) \mid n_1, n_2 \in \mathbb{N}\} \cup$
$\{(1,0,0) + n_1 \cdot (1,1,1) + n_2 \cdot (1,0,0) \mid n_1, n_2 \in \mathbb{N}\}$

$f(x) = X^2$



not semilinear

Chen, Doty, Soloveichik, "Deterministic Function Computation with Chemical Reaction Networks" (2013)

But also Register Machines (almost…)

i: INC R_1; JMP j	$PC_i \rightarrow R_1 + PC_j$
i: DEC R_1; JMP j	$PC_i + R_1 \rightarrow PC_j$
i: IF R_2>0 {INC R_1; JMP j}	$PC_i + R_2 \rightarrow R_2 + R_1 + PC_j$
i: IF R_2=0 …	??? Whatever trick we use will have some error

- Turing-complete up to an arbitrarily small error
  - The error bound is set in advance uniformly for any computation of arbitrary length (because we cannot know how long the computation will last), and the machine will progressively "slow down" to always stay below that bound.

- David Soloveichik, Matt Cook, Erik Winfree, Shuki Bruck, "Computation with Finite Stochastic Chemical Reaction Networks".
  [Natural Computing, (online Feb 2008), or Technical Report: CaltechPARADISE:2007.ETR085: .pdf]

#2 Continuous (-state) Semantics

- A state of the system is a (real-valued) concentration for each species.
- A fixed finite set of reactions act (continuously) on such states.
- The Law of Mass Action describes how the system evolves in continuous time.
  - Each reaction acts with a "speed" that is proportional to the product of the concentrations on its left-hand-side, multiplied by its rate.
  - Each species concentration increases or decreases according to the sum of the effects of all the reactions.

Steady-State Multiply (and Divide)

[X] := [A]*[B] (at steady state)

$$A + B \xrightarrow{k_1} A + B + X,$$

$$X \xrightarrow{k_2}.$$

H. J. Buisman et al. Computing Algebraic Functions with Biochemical Reaction Networks

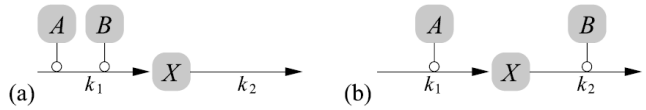


Figure 2. Catalytic reaction networks for (a) multiplication and (b) division.

$$\dot{x} = k_1 ab - k_2 x,$$

whose solution is

$$x = \frac{k_1 a_0 b_0 - (k_1 a_0 b_0 - k_2 x_0) e^{-k_2 t}}{k_2},$$

with stable steady state

$$\hat{x} = \lim_{t \to \infty} x = \frac{k_1}{k_2} a_0 b_0.$$

Computing Algebraic Functions

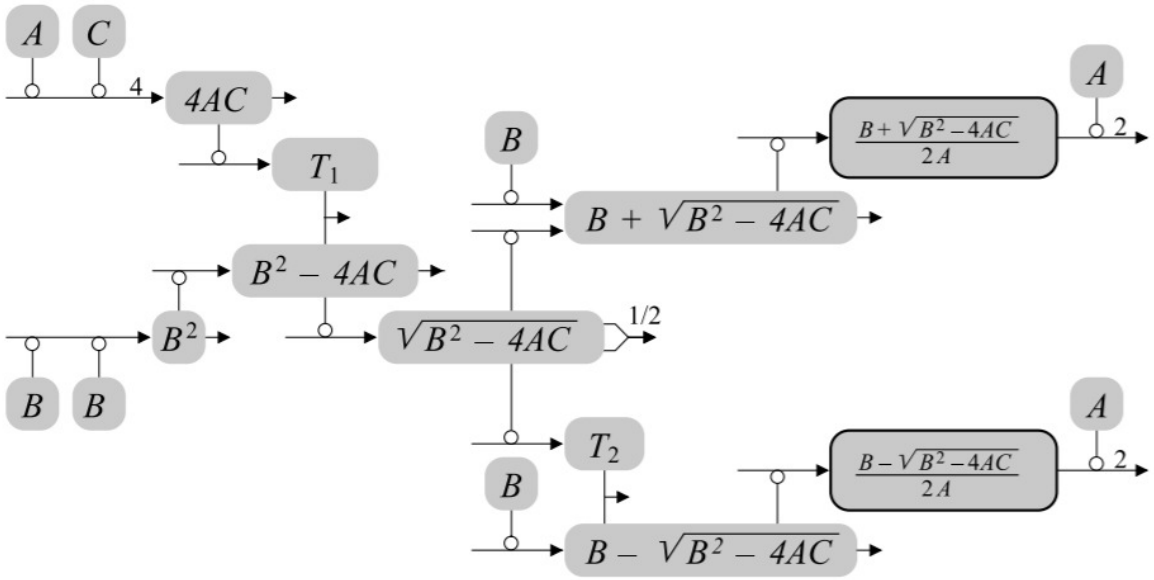
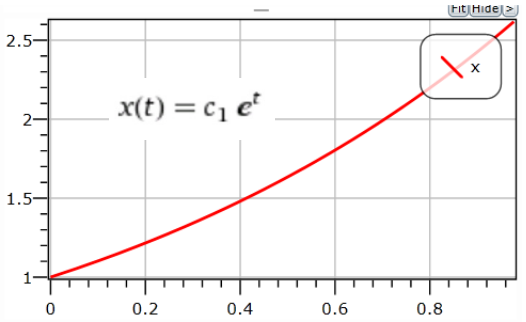


Figure 8. The quadratic formula for finding (the positive real parts of) the roots of $ax^2 - bx + c = 0$. Each of the species in the network has been given a name that represents its steady state concentration. The output species of the computation are highlighted with a black border.

Some Bad Programs

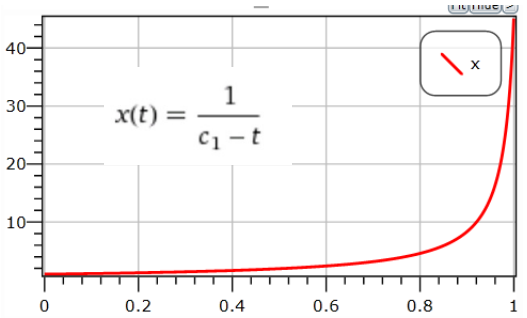
X -> X + X

Violates "only" conservation of mass. (No biggie.)



X + X -> X + X + X

Violates "finite density". (This is bad.)



Invariance from Initial Conditions

X + Y -> Y + Y

Y -> X

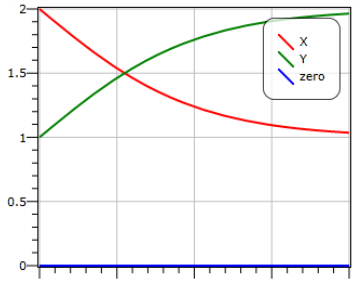
Will produce some X-Y equilibrium, which usually depends on initial values.

But here, for any initial values of X and Y (above 1) the value of X gets fixed to 1
(in general to the ratio of the second reaction rate over the first)

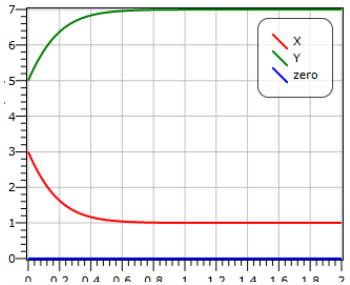
There is a static analysis that will tell you that:

Structural Sources of Robustness in Biochemical Reaction Networks

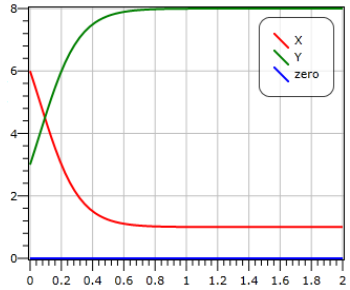
Guy Shinar[1] and Martin Feinberg[2,*]



X=2, Y=1



X=3, Y=5

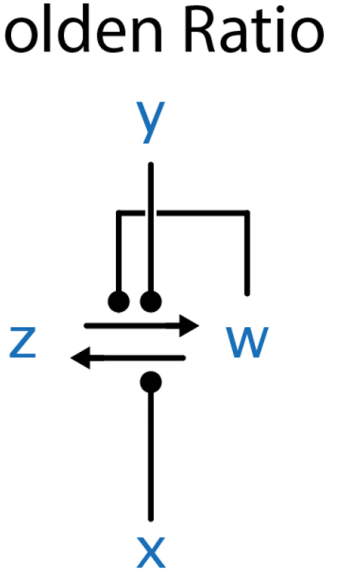


X=6, Y=3

Golden Ratio (conjugate)

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

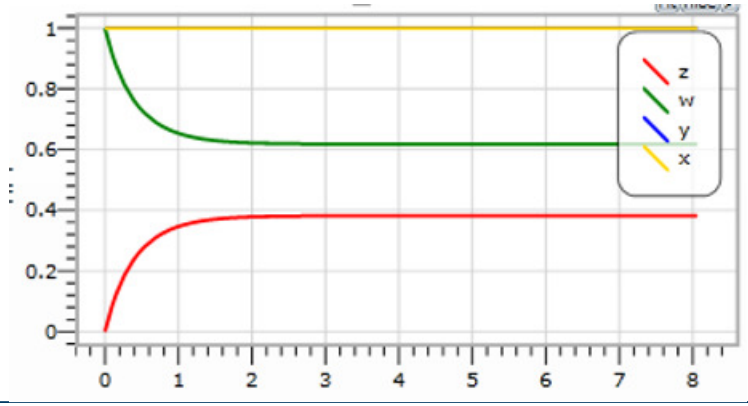
Init x=y=w=1.0
Init z = 0.0
all rates 1.0



Then (we can show analytically by the law of mass action that) at steady state:

$$w = 1/\varphi = \varphi - 1 = 0.61803$$

All algebraic equations can be solved [Ref]



#3 Wait, there are *two* semantics?

- In a given volume are there
  - (A) A finite number of molecules? or
  - (B) A continuous concentration of <something>?

- Does it make a difference?
  - Related by Avogadro's number: #molecules = concentration * Avogadro
  - But finite density issues: concentration is not unbounded in the discrete model

Are these programs equivalent? (YES!)

AM with 4 reactions

X + Y -> Y + B
Y + X -> X + B
B + X -> X + X
B + Y -> Y + Y

AM with 3 reactions

X + Y -> B + B
B + X -> X + X
B + Y -> Y + Y

Same *identical* ODEs => EQUIVALENT

dX/dt = -XY + BX
dY/dt = -YX + BY
dB/dt = 2XY - BX - BY

Are these programs equivalent? (NO!)

- With 3 reactions:
  - {X, Y} -> {B, B} in one step, then stop

```
X + Y -> B + B
B + X -> X + X
B + Y -> Y + Y
```

- With 4 reactions:
  - {X, Y} -> ({X, B} or {Y, B}) -> ({X, X} or {Y, Y}), then stop
  - (no {B, B} final state)

```
X + Y -> Y + B
Y + X -> X + B
B + X -> X + X
B + Y -> Y + Y
```

- Different final states => NOT EQUIVALENT
  - The 3-reaction version fails the requirement that in the end one of the outputs should be the sum of the inputs.

Who is right?

- #1: Believe the discrete nature of atoms (and cells): there are no continuous concentrations
- #2: Believe the analytical power of calculus: a useful approximation in appropriate conditions
- Biology has (quite recently) discovered that #1 must be taken seriously, because of advances in laboratory equipment that allow examining single molecules and single cells.