On Process Rate Semantics
Representing Biochemical Systems as Collectives of Interacting Automata

Luca Cardelli
Microsoft Research

MFPS 24, Philadelphia, 2008-05-24

http://LucaCardelli.name
Motivation: Cells Compute

- **No survival without computation!**
  - Finding food
  - Avoiding predators

- **How do they compute?**
  - Unusual computational paradigms.
  - Proteins: do they work like electronic circuits?
  - Genes: what kind of software is that?

- **Signaling networks**
  - Clearly “information processing”
  - They are “just chemistry”: molecule interactions
  - But what are their principles and algorithms?

- **Complex, higher-order interactions**
  - MAPKKK = MAP Kinase Kinase Kinase:
    that which operates on that which operates on that which operates on protein.

- **General models of biological computation**
  - What are the appropriate ones?

---

Aims

- **Connections between modeling approaches**
  - Connecting the discrete/concurrent/stochastic/molecular approach
  - to the continuous/sequential/deterministic/population approach

- **Connecting syntax with semantics**
  - Syntax = model presentation (equations/programs/diagrams/blobs etc.)
  - Semantics = state space (generated by the syntax)

- **Ultimately, connections between analysis techniques**
  - We need (and sometimes have) good semantic techniques to analyze state spaces (e.g. calculus, but also increasingly modelchecking)
  - But we need equally good syntactic techniques to structure complex models (e.g. compositionality) and analyze them (e.g. process algebra)
(Macro)Molecules as Interacting Automata

- Concurrent (math is based on processes, not functions)
- Asynchronous (no global clock)
- Stochastic (or nondeterministic)
- Stateful (e.g. phosphorylation state)
- Discrete (transitions between states)
- Interacting (an “interaction” can be pretty much anything you want that changes molecular state)

- Based on work on process algebra and biological modeling; see references in related papers.
“Collective”:
- A large set of interacting finite state automata:
  - Not quite language automata ("large set")
  - Not quite cellular automata ("interacting" but not on a grid)
  - Not quite process algebra ("collective behavior")
  - Cf. multi-agent systems and swarm intelligence

“Stochastic”:
- Interactions have rates
  - Not quite discrete (hundreds or thousands of components)
  - Not quite continuous (non-trivial stochastic effects)
  - Not quite hybrid (no "switching" between regimes)

Very much like biochemistry
- Which is a large set of stochastically interacting molecules/proteins
- Are proteins finite state and subject to automata-like transitions?
  - Let's say they are, at least because:
    - Much of the knowledge being accumulated in Systems Biology is described as state transition diagrams [Kitano].
Interacting Automata

A_1 is a state

a is a channel i.e. a named interaction interface (e.g. a surface patch)

?,! indicate any complementarity of interaction (e.g. charge)

?a, !a indicate complementary actions,

@r, @s are rates
Interacting Automata

Kinetic laws:

Two complementary actions may result in an interaction.

A₁ is a state

a is a channel i.e. a named interaction interface (e.g. a surface patch)

?,! indicate any complementarity of interaction (e.g. charge)

?a, !a indicate complementary actions, joined by an interaction arrow

@r, @s are rates
**Interacting Automata**

**Kinetic laws:**

Two complementary actions may result in an interaction.

A decay may happen spontaneously.

- $A_1$ is a state
- $a$ is a channel, i.e., a named interaction interface (e.g., a surface patch)
- $?,!$ indicate any complementarity of interaction (e.g., charge)
- $?a, !a$ indicate complementary actions, joined by an interaction arrow $\cdots\rightarrow$
- $@r, @s$ are rates
Interacting Automata

Communication channels

Automata

The system and initial state

Interactions have rates. Actions DO NOT have rates.

The equivalent process algebra model

new a@r_1
new b@r_2
new c@r_3

A_1 = ?a; A_2
A_2 = !c; A_3
A_3 = τ@λ_5; A_1

B_1 = τ@λ_2; B_2 + !a; B_3
B_2 = τ@λ_1; B_1
B_3 = ?b; B_2

C_1 = !b; C_2 + ?c; C_3
C_2 = τ@λ_3; C_1
C_3 = τ@λ_4; C_2

A_1 | B_1 | C_1
Suppose this is the next interaction (stochastically chosen).

One lonely automaton cannot interact.
Interactions in a Population
Interactions in a Population

All-A stable population
Suppose this is the next interaction.
Interactions in a Population (2)

All-B stable population

Nondeterministic population behavior ("multistability")
CTMC Semantics

CTMC (homogeneous) Continuous Time Markov Chain
- directed graph with no self loops
- nodes are system states
- arcs have transition rates

Probability of holding in state A:
\[ \Pr(H_A > t) = e^{-rt} \]
in general, \( \Pr(H_A > t) = e^{-Rt} \) where \( R \) is the sum of all the exit rates from A
Chemistry vs. Automata

A process algebra (chemistry)

\[ \begin{align*}
  r &: A + B \rightarrow_{k_1} C + D \\
  s &: C + D \rightarrow_{k_2} A + B
\end{align*} \]

A different process algebra (automata)

Reaction oriented

1 line per reaction

Does A become C or D?

Interaction oriented

1 line per component

A Petri-Net-like representation. Precise and dynamic, but not modular, scalable, or maintainable.

A compositional graphical representation (precise, dynamic and modular) and the corresponding calculus.

\[ \begin{align*}
  A &= !r_{k_1} : C \\
  C &= ?s_{k_2} : A \\
  B &= ?r_{k_1} : D \\
  D &= !s_{k_2} : B
\end{align*} \]
Groupies and Celebrities
Groupies and Celebrities

**Groupie** (wants to be like somebody different)

```plaintext
directive sample 1.0 1000
directive plot A(); B()
new a@1.0:chan()
new b@1.0:chan()
let A() = do !a; A() or ?b; B() and B() = do !b; B() or ?a; A()
run 100 of (A() | B())
```

**Celebrity** (does not want to be like somebody else)

```plaintext
directive sample 1.0 1000
directive plot A(); B()
new a@1.0:chan()
new b@1.0:chan()
let A() = do !a; A() or ?a; B() and B() = do !b; B() or ?b; A()
run 100 of (A() | B())
```

A stochastic collective of celebrities:

A stochastic collective of groupies:

Unstable because within an A majority, an A has difficulty finding a B to emulate, but the few B’s have plenty of A’s to emulate, so the majority may switch to B. Leads to deadlock when everybody is in the same state and there is nobody different to emulate.

Stable because as soon as an A finds itself in the majority, it is more likely to find somebody in the same state, and hence change, so the majority is weakened.

Both Together

A way to break the deadlocks: Groupies with just a few Celebrities

Luca Cardelli
**Hysteric Groupies**

We can get more regular behavior from groupies if they “need more convincing”, or “hysteresis” (history-dependence), to switch states.

Regular behavior can arise not far from chaos.

N.B.: It will not oscillate without doping (noise).

(With doping to break deadlocks)

```
directive sample 10.0 1000
directive plot Ga(); Gb()
new a@1.0:chan()
new b@1.0:chan()
let Ga() = do !a; Ga() or ?b; ?b
Gb() and Gb() = do !b; Gb() or ?a; ?a
Ga()
let Da() = !a; Da() and Db() = !b; Db()
run 100 of (Ga() | Gb())
run 1 of (Da() | Db())
```

```
directive sample 10.0 1000
directive plot Ga(); Gb()
new a@1.0:chan()
new b@1.0:chan()
let Ga() = do !a; Ga() or ?b; ?b
Gb() and Gb() = do !b; Gb() or ?a; ?a
Ga()
let Da() = !a; Da() and Db() = !b; Db()
run 100 of (Ga() | Gb())
run 1 of (Da() | Db())
```

```
0 1 2 3 4 5 6 7 8 9 10
0 20 40 60 80 100 120 140 160
1 sample orbit A vs. B
```

```
0 1 2 3 4 5 6 7 8 9 10
0 20 40 60 80 100 120 140 160
1 sample orbit A vs. B
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
Semantics of Collective Behavior
These diagrams commute via appropriate maps.

L. Cardelli: “On Process Rate Semantics” (TCS)

L. Cardelli: “A Process Algebra Master Equation” (QEST’07)
Quantitative Process Semantics

Continuous-state Semantics (Mass Action Kinetics)

\[ \frac{d[X]}{dt} = \left( \sum_{Y \in E} \text{Accr}_E(Y,X)[Y] \right) - \text{Depl}_E(X)[X] \text{ for all } X \in E \]

Discrete-state Semantics (Chemical Master Equation)

\[ \frac{\partial pr(p,t)}{\partial t} = \sum_{\iota \in \mathcal{I}} a_\iota (p-v_\iota) pr(p-v_\iota,t) - a_\iota (p) pr(p,t) \text{ for all } p \in \text{States}(E) \]

Discrete Chemistry

Continuous Chemistry

Process Rate Equation

Nondeterministic Semantics

Stochastic Semantics

Process Master Equation

Accretion

Depletion

Defined over the syntax of processes

Interactions

Propensity

Luca Cardelli
2008-05-24
Stochastic Processes & Discrete Chemistry
## Chemical Reactions

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Description</th>
<th>Rate Law</th>
</tr>
</thead>
<tbody>
<tr>
<td>A \rightarrow^{r} B_1 + \ldots + B_n (n \geq 0)</td>
<td>Unary Reaction</td>
<td>( \frac{d[A]}{dt} = -r[A] )</td>
</tr>
<tr>
<td>A_1 + A_2 \rightarrow^{r} B_1 + \ldots + B_n (n \geq 0)</td>
<td>Hetero Reaction</td>
<td>( \frac{d[A_i]}{dt} = -r[A_1][A_2] )</td>
</tr>
<tr>
<td>A + A \rightarrow^{r} B_1 + \ldots + B_n (n \geq 0)</td>
<td>Homeo Reaction</td>
<td>( \frac{d[A]}{dt} = -2r[A]^2 )</td>
</tr>
</tbody>
</table>

(assuming \( A \neq B_i \neq A_j \) for all \( i,j \))

### No other reactions!

---

**Trimolecular reactions:**

\[ A + B + C \rightarrow^{r} D \]

The measured \( "r" \) is an (imperfect) aggregate of e.g.:  
\[ A + B \leftrightarrow AB \]
\[ AB + C \rightarrow D \]

---

**Enzymatic reactions:**

\[ S \xrightarrow{E \rightarrow P} \]

the "r" is given by Michaelis-Menten (approximated steady-state) laws:

\[ E + S \leftrightarrow ES \]
\[ ES \rightarrow P + E \]

---

**Chapter IV: Chemical Kinetics**

[David A. Reckhow, CEE 572 Course]

... reactions may be either elementary or non-elementary. **Elementary reactions** are those reactions that occur exactly as they are written, without any intermediate steps. These reactions almost always involve just one or two reactants. **Non-elementary reactions** involve a series of two or more elementary reactions. Many complex environmental reactions are non-elementary. In general, reactions with an overall reaction order greater than two, or reactions with some non-integer reaction order are non-elementary.

**The chemical Langevin equation**

Genuinely trimolecular reactions do not physically occur in dilute fluids with any appreciable frequency. **Apparently trimolecular reactions** in a fluid are usually the combined result of two bimolecular reactions and one monomolecular reaction, and involve an additional short-lived species.

---

**The Collision Theory of Reaction Rates**

www.chemguide.co.uk

The chances of all this happening if your reaction needed a collision involving more than 2 particles are remote. All three (or more) particles would have to arrive at exactly the same point in space at the same time, with everything lined up exactly right, and having enough energy to react. That's not likely to happen very often!

---

**Reactions have rates. Molecules do not have rates.**
Chemical Ground Form (CGF)

\[
\begin{align*}
E & ::= 0 : X=M, E \\
M & ::= 0 : \pi;P \oplus M \\
P & ::= 0 : X \parallel P \\
\pi & ::= \tau(r) : ?a(r) : !a(r) \\
CGF & ::= E, P
\end{align*}
\]

- Reagents
- Molecules
- Solutions
- Actions (delay, input, output)
- Reagents plus Initial Conditions

⊕ is stochastic choice (vs. + for chemical reactions)
0 is the null solution (P|0 = 0|P = P)
and null molecule (M⊕0 = 0⊕M = M)
Each X in E is a distinct species
Each name a is assigned a fixed rate r: a(r)

(To translate chemistry to processes we need a bit more than interacting automata: we may have “+” on the right of →, that is we may need “|” after π.)

A stochastic subset of CCS
(no values, no restriction)

Interacting Automata + dynamic forking

Ex: Interacting Automata
(= finite-control CGFs: they use “|” only in initial conditions):

A = !a;A ⊕ ?b;B
B = !b;B ⊕ ?a;A
A|A|B|B

Automaton in state A
Automaton in state B
Initial conditions: 2A and 2B
### Interacting Automata

<table>
<thead>
<tr>
<th>Initial States</th>
<th>Initial Quantities</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \mid A \mid \ldots \mid A$</td>
<td>$#A_0$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A $\rightarrow_r$ A'</th>
<th>A $\rightarrow_r$ A'</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>A $\rightarrow_r$ A'</th>
<th>A + B $\rightarrow_r$ A' + B'</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>A $\rightarrow_r$ A'</th>
<th>A + A $\rightarrow_{2r}$ A' + A''</th>
</tr>
</thead>
</table>

### Discrete Chemistry

<table>
<thead>
<tr>
<th>ODE</th>
<th>ODE</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Continuous Chemistry</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Process Algebra</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CTMC</th>
<th>CTMC</th>
</tr>
</thead>
</table>
From Reactions to Reagents (by example)

1: Fill the matrix by columns:

Degradation reaction \( v_i : X \rightarrow_{k_i} P_{i} \)
add \( \tau;P_{i} \) to \( <X,v_i> \).

Hetero reaction \( v_i : X+Y \rightarrow_{k_i} P_{i} \)
add \( ?;P_{i} \) to \( <X,v_i> \) and \( !;0 \) to \( <Y,v_i> \).

Homeo reaction \( v_i : X+X \rightarrow_{k_i} P_{i} \)
add \( ?;P_{i} \) and \( !;0 \) to \( <X,v_i> \).

2: Read the result by rows:

\[
\begin{align*}
A &= ?v_{1(k1)}(C|C) \oplus ?v_{2(k2)}D \\
B &= !v_{1(k1)}0 \\
C &= !v_{2(k2)}0 \oplus \tau_{k3}(E|F) \\
D &= 0 \\
E &= 0 \\
F &= ?v_{4(k4/2)}B \oplus !v_{4(k4/2)}0
\end{align*}
\]
Entangled vs Detangled

Entangled: Two reactions on one channel

- \( \text{a: } A+B \rightarrow^r A+B' \)
- \( \text{a: } A+C \rightarrow^r A+C' \)

\( \text{A} = !a;A \)
\( \text{B} = ?a;B' \)
\( \text{C} = ?a;C' \)
\( \text{B'} = 0 \)
\( \text{C'} = 0 \)

Entangled: Two reactions on one channel

- \( \text{b: } A+B \rightarrow^r A+B' \)
- \( \text{c: } A+C \rightarrow^r A+C' \)

\( \text{A} = !b;A \oplus !c;A \)
\( \text{B} = ?b;B' \)
\( \text{C} = ?c;C' \)
\( \text{B'} = 0 \)
\( \text{C'} = 0 \)

Detangled: Two reactions on two separate channels

We need a semantics of automata that identifies automata that have the “same chemistry”.
No process algebra equivalence is like this!

Entangled automata lead to more compact models than in chemistry.
Detangled automata are in simple correspondence with chemistry.
Could chemistry itself be that semantics?

No: different sets of reactions can have the same behavior!

Different reactions, but they induce the same ODEs.
Discrete-State Semantics

Continuous Chemistry

Discrete Chemistry

Process Algebra

CTMC

CTMC
Discrete Semantics of Reactions

Syntax:

\[ A + B \rightarrow^r A + A \]
\[ A + B \rightarrow^r B + B \]
\[ A + B + B \]

Semantics:

\[ \{3B\} \xrightarrow{2r_b} \{2A,1B\} \xrightarrow{2r_a} \{3A\} \]
Discrete Semantics of Reagents

Syntax:

Semantics:

CTMC
Discrete State Equivalence

- **Def:** $\equiv$ is equivalent CTMC's (isomorphic graphs with same rates).
- **Thm:** $E \equiv \text{Ch}(E)$
- **Thm:** $C \equiv \text{Pi}(C)$

For each $E$ there is an $E' \equiv E$ that is detangled ($E' = \text{Pi}(\text{Ch}(E))$)

For each $E$ in automata form there is an $E' \equiv E$ that is detangled and in automata form ($E' = \text{Detangle}(E)$).
This is enough to establish that the process algebra is really faithful to the chemistry.

But CTMC are not the “ultimate semantics” because there are still questions of when two different CTMCs are actually equivalent (e.g. “lumping”).

The “ultimate semantics” of chemistry is the Chemical Master Equation (derivable from the Chapman-Kolmogorov equation of the CTMC).
Continuous-State Semantics
(short version)

[Diagram]

- ODE = ODE
- Continuous Chemistry
- Discrete Chemistry
- CTMC = CTMC
- Process Algebra
The Gillespie\(^{(\text{?})}\) Conversion

<table>
<thead>
<tr>
<th>Discrete Chemistry</th>
<th>Continuous Chemistry</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\gamma = N_A V)</td>
<td>(\gamma = N_A V)</td>
</tr>
<tr>
<td>(\gamma = N_A V)</td>
<td>(\gamma = N_A V)</td>
</tr>
<tr>
<td>(\gamma = N_A V)</td>
<td>(\gamma = N_A V)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>initial quantities</th>
<th>initial concentrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(#A_0)</td>
<td>([A]_0)</td>
</tr>
<tr>
<td>([A]_0)</td>
<td>([A]_0)</td>
</tr>
<tr>
<td>([A]_0)</td>
<td>([A]_0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(A \rightarrow_r A')</th>
<th>(A \rightarrow^k A')</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A \rightarrow^k A')</td>
<td>(k = r)</td>
</tr>
<tr>
<td>(A \rightarrow^k A')</td>
<td>(k = r)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(A+B \rightarrow_r A'+B')</th>
<th>(A+B \rightarrow^k A'+B')</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A+B \rightarrow^k A'+B')</td>
<td>(k = r\gamma)</td>
</tr>
<tr>
<td>(A+B \rightarrow^k A'+B')</td>
<td>(k = r\gamma)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(A+A \rightarrow_r A'+A'')</th>
<th>(A+A \rightarrow^k A'+A'')</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A+A \rightarrow^k A'+A'')</td>
<td>(k = r\gamma/2)</td>
</tr>
<tr>
<td>(A+A \rightarrow^k A'+A'')</td>
<td>(k = r\gamma/2)</td>
</tr>
</tbody>
</table>

\(V = \text{interaction volume}\)
\(N_A = \text{Avogadro's number}\)

Think \(\gamma = 1\)
i.e. \(V = 1/N_A\)

\(M = \text{mol} \cdot \text{L}^{-1}\)
molarity (concentration)
From Reactions to ODEs (Law of Mass Action)

Write the coefficients by columns

<table>
<thead>
<tr>
<th>reactions</th>
<th>stoichiometric matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>v₁: A+B → k₁ C+C</td>
<td>N</td>
</tr>
<tr>
<td>v₂: A+C → k₂ D</td>
<td>A</td>
</tr>
<tr>
<td>v₃: C → k₃ E+F</td>
<td>B</td>
</tr>
<tr>
<td>v₄: F+F → k₄ B</td>
<td>C</td>
</tr>
</tbody>
</table>

Set a rate law for each reaction
(Degradation/Hetero/Homeo)

E.g. d[A]/dt = -l₁ - l₂
      d[B]/dt = -l₁ + l₄
      d[C]/dt = 2l₁ - l₂ - l₃
      d[D]/dt = l₂
      d[E]/dt = l₃
      d[F]/dt = l₃ - 2l₄

Rate laws

Quantity changes

Stoichiometric matrix

Rate laws

Set a rate law for each reaction
(Degradation/Hetero/Homeo)

E.g. d[A]/dt = -k₁[A][B] - k₂[A][C]

X: chemical species
[-]: quantity of molecules
l: rate laws
k: kinetic parameters
N: stoichiometric matrix

d[X]/dt = N·l

d[A]/dt = -l₁ - l₂
d[B]/dt = -l₁ + l₄
d[C]/dt = 2l₁ - l₂ - l₃
d[D]/dt = l₂
d[E]/dt = l₃
d[F]/dt = l₃ - 2l₄

Write the coefficients by columns

<table>
<thead>
<tr>
<th>Stoichiometric Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

Read the concentration changes from the rows

<table>
<thead>
<tr>
<th>Process Algebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Chemistry</td>
</tr>
<tr>
<td>Discrete Chemistry</td>
</tr>
<tr>
<td>CTMC</td>
</tr>
<tr>
<td>=</td>
</tr>
<tr>
<td>CTMC</td>
</tr>
<tr>
<td>=</td>
</tr>
<tr>
<td>ODE</td>
</tr>
<tr>
<td>=</td>
</tr>
<tr>
<td>ODE</td>
</tr>
</tbody>
</table>

Luca Cardelli
2008-05-24
38
Process Rate Equation for Reagents $E$ in volume $\gamma$

\[
d[X]/dt = (\sum (Y \in E) \text{Accr}_E(Y,X)[Y]) - \text{Depl}_E(X)[X]
\]

for all $X \in E$

“The change in process concentration (!!) for $X$ at time $t$ is:

the sum over all possible (kinds of) processes $Y$ of:

the concentration at time $t$ of $Y$

times the accretion from $Y$ to $X$

minus the concentration at time $t$ of $X$

times the depletion of $X$ to some other $Y$”

\[
\text{Depl}_E(X) = \\
\sum (i: E.X.i=\tau(r);P) r + \\
\sum (i: E.X.i=?a(r);P) r \gamma \cdot \text{OutsOn}_E(a) + \\
\sum (i: E.X.i=!a(r);P) r \gamma \cdot \text{InsOn}_E(a)
\]

\[
\text{Accr}_E(Y, X) = \\
\sum (i: E.Y.i=\tau(r);P) \# X(P).r + \\
\sum (i: E.Y.i=?a(r);P) \# X(P).r \gamma \cdot \text{OutsOn}_E(a) + \\
\sum (i: E.Y.i=!a(r);P) \# X(P).r \gamma \cdot \text{InsOn}_E(a)
\]

\[
\text{InsOn}_E(a) = \sum (Y \in E) \# \{Y.i | E.Y.i=?a(r);P\}[Y] \\
\text{OutsOn}_E(a) = \sum (Y \in E) \# \{Y.i | E.Y.i=!a(r);P\}[Y]
\]
Continuous State Equivalence

- **Def:** $\approx$ is equivalence of polynomials over the field of reals.

- **Thm:** $E \approx \text{Cont}(\text{Ch}(E))$

- **Thm:** $\text{Cont}(C) \approx \text{Pi}(C)$

- For each $E$ there is an $E' \approx E$ that is detangled ($E' = \text{Pi}(\text{Ch}(E))$)

- For each $E$ in automata form there is an $E' \approx E$ that is detangled and in automata form ($E' = \text{Detangle}(E)$).
Exercise: Making Waves

Or: build me a population like this:
Nonlinear Transition (NLT)

\[ A = \text{?c}(s); B \]
\[ B = \text{!c}(s); B \]

\[ A + B \rightarrow^s B + B \]

\[[A]^\bullet = -s[A][B] \]
\[[B]^\bullet = s[A][B] \]

N.B.: needs at least 1 B to "get started".

Matlab

```matlab
interval/step [0.0:0.001:0.0]
(A) dx1/dt = -x1*x2 1000.0
(B) dx2/dt = x1*x2 1.0
```
Two NLTs: Bell Shape

\[
[B]^{\bullet} = [B][A] - [C]
\]

\[
A = ?b_{(1)}; B
\]
\[
B = !b_{(1)}; B \oplus ?c_{(1)}; C
\]
\[
C = !c_{(1)}; C
\]

\[
A + B \rightarrow^{1} B + B
\]
\[
B + C \rightarrow^{1} C + C
\]

\[
[A]^{\bullet} = -[A][B]
\]
\[
[B]^{\bullet} = [A][B] - [B][C]
\]
\[
[C]^{\bullet} = [B][C]
\]

directive sample 0.0025 1000
directive plot B(); A(); C()
new b@1.0:chan new c@1.0:chan
let A() = ?b; B()
and B() = do !b; B() or ?c; C()
and C() = !c; C()
run ((10000 of A()) | B() | C())
NLT in a Cycle: Oscillator (unstable)

A = !a(s); A ⊕ ?b(s); B
B = !b(s); B ⊕ ?c(s); C
C = !c(s); C ⊕ ?a(s); A

A+B →s B+B
B+C →s C+C
C+A →s A+A

\[
\begin{align*}
[A] &= -s[A][B] + s[C][A] \\
[B] &= -s[B][C] + s[A][B] \\
[C] &= -s[C][A] + s[B][C]
\end{align*}
\]

Matlab

\[
\begin{align*}
\text{interval/step} &\text{ [0:0.001:20.0]} \\
(A) &\quad dx_1/dt = -x_1*x_2 + x_3*x_1 \quad 0.9 \\
(B) &\quad dx_2/dt = -x_2*x_3 + x_1*x_2 \quad 0.5 \\
(C) &\quad dx_3/dt = -x_3*x_1 + x_2*x_3 \quad 0.1
\end{align*}
\]

directive sample 0.03 1000
directive plot A(); B(); C()
new a@1.0:chan new b@1.0:chan new c@1.0:chan
let A() = do !a; A() or ?b; B() and B() = do !b; B() or ?c; C() and C() = do !c; C() or ?a; A()
run (900 of A() | 500 of B() | 100 of C())
Oscillator (stable)

N.B. this does not deadlock!

A →ₙ B
A+B →₅ A'+B
A'+B →₅ B+B
B →ₙ C
B+C →₅ B'+C
B'+C →₅ C+C
C →ₙ A
C+A →₅ C'+A
C'+A →₅ A+A

[A]⁺ = -r[A]-s[A][B]+r[C]+s[C'][A]
[C]⁺ = -r[C]-s[C][A]+r[B]+s[B'][C]
[A']⁺ = -s[A'][B] + s[A][B]
[B']⁺ = -s[B'][C] + s[B][C]
[C']⁺ = -s[C'][A] + s[C][A]
GMA ≠ CME

Continuous ODE = Continuous Chemistry

Discrete Chemistry = Process Algebra

CTMC = CTMC

Semantics #1

Syntax

Semantics #2
\[ A+A \rightarrow^{2r} A \quad =? \quad A+A \rightarrow^{r} 0 \]

1*reaction rate \( \gamma \) because 1* \( A \) is lost in reaction.

2*reaction rate \( \gamma/2 \) because 2* \( A \) are lost in reaction.

- \( d[A]/dt = -r\gamma[A]^2 \)
- \( [A]_0 = 2/\gamma \)
- \( A+A \rightarrow^{2r} A \)
- \( A+A \rightarrow^{r} 0 \)

Law of Mass Action

Gillespie conversion

CTMC
Continuous vs. Discrete Groupies

(all with doping)

Matlab

SPiM

2000xA, 0xB, 1xA_d, 1xB_d, r = 1.0
Scientific Predictions

After a while, all 4 states are almost equally occupied.

The 4 states are almost never equally occupied.
Model Compactness

\[
\text{Continuous Chemistry} \quad \overset{=}{\longrightarrow} \quad \text{Discrete Chemistry}
\]

\[
\text{ODE} \quad \overset{=}{\longrightarrow} \quad \text{Process Algebra}
\]

\[
\text{CTMC} \quad \overset{=}{\longrightarrow} \quad \text{CTMC}
\]
Entangled vs detangled

Detangle($E_3$) (closely related to $\Pi(\text{Ch}(E_3))$)
n² Scaling Problems

- Eₙ has 2n variables (nodes) and 2n terms (arcs).
- Ch(Eₙ) has 2n species and n² reactions.
- The stoichiometric matrix has size 2n·n² = 2n³.
- The ODEs have 2n variables and 2n(n+n) = 4n² terms

E₃

| X₀ = ?aᵣ; X₁ |
| X₁ = ?aᵣ; X₂ |
| X₂ = ?aᵣ; X₀ |
| Y₀ = !aᵣ; Y₁ |
| Y₁ = !aᵣ; Y₂ |
| Y₂ = !aᵣ; Y₀ |

Ch(E₃)

| a₀₀: X₀+Y₀ → rX₁+Y₁ |
| a₀₁: X₀+Y₁ → rX₁+Y₂ |
| a₀₂: X₀+Y₂ → rX₁+Y₀ |
| a₁₀: X₁+Y₀ → rX₂+Y₁ |
| a₁₁: X₁+Y₁ → rX₂+Y₂ |
| a₁₂: X₁+Y₂ → rX₂+Y₀ |
| a₂₀: X₂+Y₀ → rX₀+Y₁ |
| a₂₁: X₂+Y₁ → rX₀+Y₂ |
| a₂₂: X₂+Y₂ → rX₀+Y₀ |

StoichiometricMatrix(Ch(E₃))

<table>
<thead>
<tr>
<th></th>
<th>a₀₀</th>
<th>a₀₁</th>
<th>a₀₂</th>
<th>a₁₀</th>
<th>a₁₁</th>
<th>a₁₂</th>
<th>a₂₀</th>
<th>a₂₁</th>
<th>a₂₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>X₀</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X₁</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X₂</td>
<td></td>
<td></td>
<td></td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Y₀</td>
<td>-1</td>
<td></td>
<td></td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td></td>
<td>+1</td>
</tr>
<tr>
<td>Y₁</td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Y₂</td>
<td></td>
<td>+1</td>
<td>-1</td>
<td></td>
<td></td>
<td>+1</td>
<td>-1</td>
<td></td>
<td>-1</td>
</tr>
</tbody>
</table>

ODE(E₃)

\[
\begin{align*}
\frac{d[X₀]}{dt} &= -r[X₀][Y₀] - r[X₀][Y₁] - r[X₀][Y₂] + r[X₂][Y₀] + r[X₂][Y₁] + r[X₂][Y₂] \\
\frac{d[X₁]}{dt} &= -r[X₁][Y₀] - r[X₁][Y₁] - r[X₁][Y₂] + r[X₀][Y₀] + r[X₀][Y₁] + r[X₀][Y₂] \\
\frac{d[X₂]}{dt} &= -r[X₂][Y₀] - r[X₂][Y₁] - r[X₂][Y₂] + r[X₁][Y₀] + r[X₁][Y₁] + r[X₁][Y₂] \\
\frac{d[Y₀]}{dt} &= -r[X₀][Y₀] - r[X₁][Y₀] - r[X₂][Y₀] + r[X₀][Y₂] + r[X₁][Y₂] + r[X₂][Y₂] \\
\frac{d[Y₁]}{dt} &= -r[X₀][Y₁] - r[X₁][Y₁] - r[X₂][Y₁] + r[X₀][Y₀] + r[X₁][Y₀] + r[X₂][Y₀] \\
\frac{d[Y₂]}{dt} &= -r[X₀][Y₂] - r[X₁][Y₂] - r[X₂][Y₂] + r[X₀][Y₁] + r[X₁][Y₁] + r[X₂][Y₁]
\end{align*}
\]
On the Computational Power of Biochemistry

joint work with Gianluigi Zavattaro

University of Bologna

in: Algebraic Biology '08
Does this program halt?

A+B → B+B
B+C → C+C
C+A → A+A
“Experimental evidence”

Continuous-State Semantics

Discrete-State Semantics
But in a longer simulation...
Is termination decidable in Chemistry?

- Three notations for “basic chemistry“:
  - FSRN: Finite Stochastic Reaction Networks
    (finite systems of stochastic chemical reactions)
  - CGF: our process algebra (CTCM-equivalent to FSRN).
  - Place-transition Petri nets.

- **Answer:** termination (reachability) in Chemistry is *decidable*!
  - Termination in CGF can be reduced to termination in place-transition Petri Nets, where it (reachability) is decidable.

- Hence, basic chemistry is **NOT** Turing-complete!
Biochemistry = Interaction + Complexation

- Complexation is what proteins “do”, in contrast to simpler chemicals.

- Leading to a process algebra that we call the Biochemical Ground Form (BGF).
RAM encoding in BGF

i: Inc($r_j$)  k: DecJump($r_j$, s)

register $r_j$:

$Z_j$
Expressiveness of Biochemistry

- Basic chemistry (FSRN, or CGF) is not Turing-complete

- Biochemistry (FSRN + complexation, or BGF) is Turing-complete.

- More powerful process algebras of course are Turing complete
  - They (e.g. \(\pi\)-calculus) include BGF, but they also have mechanisms that are not directly biologically justifiable.
  - In BGF we have in a sense the minimal biologically-inspired extension of FSRN, and it is already Turing-complete.

- Intrinsic to biochemistry (but not to simple chemistry) is at least one Turing-complete mechanism.
Conclusions

- **Compositional modeling languages**
  - Accurate (at the “appropriate” abstraction level).
  - Manageable (so we can scale them up by composition).
  - Executable (stochastic simulation).

- **Analysis techniques**
  - Mathematical techniques: Markov theory, Chemical Master Equation, and Rate Equation
  - Computing techniques: Abstraction and Refinement, Model Checking, Causality Analysis.

- **Many lines of extensions**
  - Parametric processes for model factorization
  - Ultimately, rich process-algebra based modeling languages.

- **Quantitative techniques**
  - Important in the “real sciences”.