Stochastic Process Algebras and Ordinary Differential Equations Salad

Luca Bortolussi\textsuperscript{1} \hspace{1cm} Alberto Policriti\textsuperscript{1,2}

\textsuperscript{1}Department of Mathematics and Computer Science
University of Udine, Italy.
\textsuperscript{2}Center for Biomolecular Medicine
Trieste, Italy.

PASTA, London, 29\textsuperscript{th} June 2006
The picture

Stochastic process algebras
(\(\pi\)-calculus)
(sCCP)

stochastic discrete “internal” modeling

(Ordinary) Differential Equations
(S-Systems)
(RRE)

deterministic continuous “external” modeling
A more detailed picture

**Stochastic process algebras**

**(Ordinary) Differential Equations**

Behavioral expressivity
Interchangeability
Hybrid modeling

Phenomenon to be modeled
*(expected/observed behaviour)*
Outline

1. Preliminaries
   - ODEs
   - SPAs

2. From ODEs to SPAs
   - Translation into sCCP

3. From SPAs to ODEs
   - Hillston’s method
## Preliminaries
- ODEs
- SPAs

## From ODEs to SPAs
- Translation into sCCP

## From SPAs to ODEs
- Hillston’s method
Chemical Equations

Chemical reactions can be represented by a set of chemical equations of the form:

\[ R_1 + \ldots + R_n \rightarrow_k P_1 + \ldots + P_m, \]

where
- \( R_i \)'s are the reactants;
- \( P_j \)'s are the products;
- \( k \) is the basic rate (speed).

Example

\[ A + B \rightarrow_k 2C \]

\[
\begin{align*}
\dot{A} &= \dot{B} = -k[A][B] \\
\dot{C} &= 2k[A][B]
\end{align*}
\]
**S-Systems**

S-systems describe the dynamical behavior of a biological system by a set of differential equations over reactants:

- non-linear, time-invariant, DAE systems;
- biologically plausible and expressive;
- analytical approximation power.

**Definition**

An S-system’s equation has the form:

\[
\dot{X}_i = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}}
\]

with \( \alpha_i, \beta_i \geq 0 \) called rate constants and \( g_{ij}, h_{ij} \) called kinetic orders.

---

A paradigmatic example — the Repressilator

The Repressilator:
A cyclic, three-repressor, transcriptional network

[Diagram of the Repressilator network]

- TetR
- λ cI
- LacI

- PC: gene A
- PA: gene B
- PB: gene C

- mRNA A
- mRNA B
- mRNA C

- Protein A
- Protein B
- Protein C
S-Systems — the Repressilator equations

\[
\begin{align*}
\dot{X}_1 &= \alpha_1 X_3^{-1} - X_1^{0.5}, & \alpha_1 &= 0.2, \\
\dot{X}_2 &= \alpha_2 X_1^{-1} - X_2^{0.578151}, & \alpha_2 &= 0.2, \\
\dot{X}_3 &= \alpha_3 X_2^{-1} - X_3^{0.5}, & \alpha_3 &= 0.2.
\end{align*}
\]
Biology and π-calculus

A (restricted) syntax of π-calculus

\[
P, Q ::= \Sigma \quad \text{Summation} \quad \Sigma ::= 0 \quad \text{Null} \\
| P|Q \quad \text{Parallel} \quad | \pi.P + \Sigma \quad \text{Action} \\
| !\pi.P \quad \text{Replication} \quad \pi ::= x\langle n \rangle \quad \text{Output} \\
| x(m) \quad \text{Input, } x \neq m
\]

Quantitative aspect: interaction “rates” assigned to channels.


Can we use π-calculus for (biological) simulation?

<table>
<thead>
<tr>
<th>Molecule</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction capability</td>
<td>Channel</td>
</tr>
<tr>
<td>Interaction</td>
<td>Communication</td>
</tr>
<tr>
<td>Modification</td>
<td>State change</td>
</tr>
<tr>
<td>(of cellular components)</td>
<td>(state transition systems)</td>
</tr>
</tbody>
</table>

Repressilator in $\pi$

Protein(p) → degrade → !p
Gate(a,b) → activate → Protein(b)
Gate(b,c) → Blocked → Protein(b)
Gate(c,a)

L. Cardelli - Sept. 2005
Concurrent Constraint Programming

Constraint Store

- In this process algebra, the main object are constraints, which are formulae over an interpreted first order language (i.e. $X = 10$, $Y > X - 3$).
- Constraints can be added to a "container", the constraint store, but can never be removed.

Agents

Agents can perform two basic operations on this store (asynchronously):

- Add a constraint (tell ask)
- Ask if a certain relation is entailed by the current configuration (ask instruction)

Syntax of CCP

$$
Program = Decl.A
\begin{align*}
D &= \varepsilon \mid Decl.Decl \mid p(x) : \neg A \\
A &= \begin{array}{c}
0 \\
tell(c).A \\
ask(c_1).A_1 + ask(c_2).A_2 \\
A_1 \parallel A_2 \mid \exists A \mid p(x)
\end{array}
\end{align*}
$$

V. Saraswat, Concurrent Constraint Programming, MIT press, 1993
Stochastic CCP

Syntax of Stochastic CCP

\[\text{Program} = D \cdot A\]
\[D = \varepsilon \mid D \cdot D \mid p(x) : -A\]
\[\pi = \text{tell}_\lambda(c) \mid \text{ask}_\lambda(c)\]
\[M = \pi \cdot A \mid M + M'\]
\[A = 0 \mid [p(x)]_\lambda \mid M \mid \exists x A \mid (A_1 \parallel A_2)\]


Stochastic information

Each basic instruction (tell, ask, procedure call) has a rate attached to it.

*Rates are functions from the constraint store \(C\) to positive reals:*

\[\lambda : C \rightarrow \mathbb{R}^+\]

Why another Process Algebra?

- Constraints are powerful and easy to program.
- Variables allow to store numerical information.
- We can use “clever” stochastic rates.
Repressilator in sCCP

degradator(X) :- tell[λ_D]·(X = X - 1).degradator(X)

neg(X, Y) :- ( tell[λ_P]·(X = X + 1)
              + ask[λ_I]·(Y > 0).ask[λ_U](true)).neg(X, Y)
          )·neg(X, Y)

neg_gate(X, Y) :- neg(X, Y) || degradator(X)
Using different kinetic laws: enzymatic reaction

Non-constant rates allow to describe more complicated kinetic dynamics than Mass Action’s one.

**Mass Action dynamics**

\[
S + E \rightleftharpoons^{k_1}_{k_2} ES \rightarrow^{k_3} P + E \\
P \rightarrow^{k_4} \rightarrow^{k_5} S
\]

The rate of \( S + E \rightarrow^{k_1} ES \) is \( k_1 [S][E] \).

**Michaelis-Menten dynamics**

\[
S \rightleftharpoons_{E, V_0} K P \\
P \rightarrow^{k_4} \rightarrow^{k_5} S
\]

The rate of \( S \rightleftharpoons_{E, V_0} K P \) is \( \frac{V_0[S]}{[S]+K} \).
Outline

1. Preliminaries
   - ODEs
   - SPAs

2. From ODEs to SPAs
   - Translation into sCCP

3. From SPAs to ODEs
   - Hillston’s method
Simulating ODEs with SPAs

Motivations

- Study **expressivity** in terms of representable behaviors of SPA
- From “external” to “internal” descriptions: identify **logical patterns of interactions**.
Using S-Systems to determine rate functions

A generic form for S-System equations is

$$\dot{X}_i = V^+(X_1, \ldots, X_{m+n}) - V^-(X_1, \ldots, X_{m+n}).$$

production speed degradation speed

A generic S-System equation has *non-linear dependencies* on variables.

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}}$$

We can use these expressions as rates.

Using sCCP, we can associate to each dependent variable an agent \( \text{subs}(X_i) \):

$$\left( \text{tell}(X_i = X_i + \sigma) \left[ \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} \right] + \text{tell}(X_i = X_i - \sigma) \left[ \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}} \right] \right).\text{subs}(X_i).$$
Encoding S-System’s Repressilator

\[
\begin{align*}
\dot{X}_1 &= \alpha_1 X_3^{-1} - \beta_1 X_1^{0.5}, \\
\dot{X}_2 &= \alpha_2 X_1^{-1} - \beta_2 X_2^{0.5}, \\
\dot{X}_3 &= \alpha_3 X_2^{-1} - \beta_3 X_3^{0.5}, \\
\alpha_i &= 0.2, \quad \beta_i = 1.
\end{align*}
\]

\[
\begin{align*}
\text{subs}(X_1) &::= (\text{tell}(X_1 = X_1 + \sigma)[\alpha_1 X_3^{-1}]
+ \text{tell}(X_1 = X_1 - \sigma)[\beta_1 X_1^{0.5}]).\text{subs}(X_1) \\
\text{subs}(X_2) &::= (\text{tell}(X_2 = X_2 + \sigma)[\alpha_2 X_1^{-1}]
+ \text{tell}(X_2 = X_2 - \sigma)[\beta_2 X_2^{0.5}]).\text{subs}(X_2) \\
\text{subs}(X_3) &::= (\text{tell}(X_3 = X_3 + \sigma)[\alpha_3 X_2^{-1}]
+ \text{tell}(X_3 = X_3 - \sigma)[\beta_3 X_3^{0.5}]).\text{subs}(X_3) \\
\sigma &= 1
\end{align*}
\]
S-System’s model of repressilator suffers from an high sensitivity from parameters, differently from the usual PA models. sCCCP model with variable rates has the same “wild” behaviour!

\[
\beta_i = 0.01
\]

\[
\beta_i = 0.001
\]
There's a trick...

The magnitude of fluctuations is small. We used $\sigma = 0.01$.

In this way we can reduce the perturbation effect of stochastic fluctuations.
Dependency on the step size $\sigma$

- $\sigma = 0.001$
- $\sigma = 0.01$
- $\sigma = 0.1$
- $\sigma = 1$
Outline

1. Preliminaries
   - ODEs
   - SPAs

2. From ODEs to SPAs
   - Translation into sCCP

3. From SPAs to ODEs
   - Hillston’s method
Hillston’s method

Jane has developed a method to associate a set of ODEs to a SPA (precisely PEPA) program.

The method can be easily adapted to the (restricted version of) stochastic $\pi$-calculus.

The translation produces always a set of Mass Action Equations, due to the definition of the SOS.

Question

What does these ODEs tell us? Something about the average behavior?

ODEs for the repressilator in $\pi$-calculus

\[
\begin{align*}
\dot{X}_1 &= \lambda_P Y_1 - \lambda_D X_1 \\
\dot{X}_2 &= \lambda_P Y_2 - \lambda_D X_2 \\
\dot{X}_3 &= \lambda_P Y_3 - \lambda_D X_3 \\
\dot{Y}_1 &= \lambda_U Z_1 - \lambda_I Y_1 X_3 \\
\dot{Y}_2 &= \lambda_U Z_2 - \lambda_I Y_2 X_1 \\
\dot{Y}_3 &= \lambda_U Z_3 - \lambda_I Y_3 X_2 \\
\dot{Z}_1 &= \lambda_I Y_1 X_3 - \lambda_U Z_1 \\
\dot{Z}_2 &= \lambda_I Y_2 X_1 - \lambda_U Z_2 \\
\dot{Z}_3 &= \lambda_I Y_3 X_2 - \lambda_U Z_3
\end{align*}
\]
What’s the average?

What’s the relationship between the solution of the ODEs and the average value of proteins in the $\pi$-repressilator? *Neither the ODEs nor the average oscillates*, but they stabilize at values different two orders of magnitude.
“non-constant rates” are a powerful addition to SPAs (simulating ODEs, complex kinetic laws, etc.)

stochastic fluctuations sometimes dominate, and they cannot be safely neglected in a translation process.

Can we find translation techniques invariant w.r.t. the observed behaviour of the “real” system?