Models and Languages for Computational Systems Biology

Lecture 15: Process Algebras for Systems Biology

Ian Stark

School of Informatics
The University of Edinburgh

Thursday 11 March 2010
Semester 2 Week 9

Scope of Study

Processes

- Metabolic networks
- Regulatory systems: promotion, inhibition
- Signalling pathways
- Gene expression: translation, transcription

Models

- Discrete time, continuous time
- Discrete space, continuous space
- Deterministic, nondeterministic, probabilistic
- Qualitative, quantitative

Biochemical Simulation

Biologists routinely use one of two alternative approaches to computational modelling of biochemical systems:

- Stochastic simulation
  - Discrete behaviour: tracking individual molecules
  - Randomized: Gillespie's algorithm
- Ordinary Differential Equations
  - Continuous behaviour: chemical concentrations
  - Deterministic: Numerical ODE solutions

The classical approach is to use the mathematics directly as the target formal system. However, experience in Computer Science suggests the value of an intermediate *language* to describe a system. An expression in this language can then be analysed as it stands, or further mapped into (one or more) mathematical representations.

Process AIgebras

A *process algebra* or *process calculus* is any one of a large range of formal languages for describing the behaviour of concurrent systems.

The "concurrent systems" may be drawn from a variety of domains:

- Distributed software
- Embedded computational devices
- Business processes
- Manufacturing and industrial processes
- Biological systems
- ...

Process Algebras

A *process algebra* or *process calculus* is any one of a large range of formal languages for describing the behaviour of concurrent systems.

Terms in the process algebra may play different roles in relation to the concurrent system:

- **Description** — Expressing observed behaviour of an existing system.
- **Specification** — Stating behaviour required of a system to be built.
- **Modelling** — Setting out a mechanism for some observed behaviour.
- **Implementation** — Prescribing how a system should be constructed.

Process Algebras

A *process algebra* or *process calculus* is any one of a large range of formal languages for describing the behaviour of concurrent systems.

There are several reasons why using a formal language may be thought desirable.

- Modularity / Compositionality: Building the whole from its parts.
- Precision: Using formal mathematical semantics to make statements exact.
- Abstraction: Extracting and focusing on relevant computational content.
- Separation: Distinguish between modelling and analysis.
- Succinctness: Brief descriptions for communicating between people, between machines, and from one to the other.

Process AIgebras

A *process algebra* or *process calculus* is any one of a large range of formal languages for describing the behaviour of concurrent systems.

Although some process calculi are very close to concurrent programming languages, they are traditionally expressed in mathematical rather than programming notation. This leads to syntax that looks like this:

$$\overline{a}\langle b\rangle.P \mid a(x).Q(x)$$

instead of this:

```
spawn { put(a,b); P() }; spawn { let x=get(a) in Q(x) }
```

although in practice the two are interconvertible.

Process Algebras in Systems Biology

Regev observed that the properties of name-passing and scope extrusion in Milner's π-calculus could be used to represent interaction and complexation in biochemical systems.

Aviv Regev, William Silverman, and Ehud Y. Shapiro.
Representation and Simulation of Biochemical Processes Using the π-Calculus Process Algebra. In *Proceedings of the 6th Pacific Symposium on Biocomputing PSB 2001)*, pages 459–470.

Some Process Algebras in Systems Biology

- π-calculus; stochastic π
- BioSPI; SPiM
- Beta binders; BlenX
- Ambients, bioAmbients
- Brane calculi; Bitonal systems
- PEPA, bioPEPA
- Kappa
- PRISM
- Pathway Logic
- Continuous π-calculus
- ...

Performance Evaluation Process Algebra

PEPA: Performance Evaluation Process Algebra was developed by Hillston to bring process algebra techniques to modelling and analysing the performance of concurrent systems.

Jane Hillston.
A Compositional Approach to Performance Modelling. Cambridge University Press, 1996.

Stephen Gilmore and Jane Hillston.
The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. In *Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Lecture Notes in Computer Science 794, pp. 353–368. Springer-Verlag, 1994.

BioPEPA

BioPEPA modifies PEPA to better handle particular features of biological models: in particular stoichiometry and the use of general kinetic laws.

BioPEPA embodies a specific mapping from biochemical systems, where each species is represented by a process. This is in some ways restrictive, compared to raw PEPA; but it also makes the model more direct, and means that BioPEPA and its tools can be more effectively tuned to fit the requirements of systems biology.

Federica Ciocchetta and Jane Hillston
Bio-PEPA: A framework for the modelling and analysis of biological systems
Theoretical Computer Science 410(33–34):3065–3084, August 2009.

Minimal PEPA and BioPEPA

PEPA

$$P ::= (\alpha, r).P \quad | \quad (\alpha, \top).P \quad | \quad P + Q \quad | \quad P \underset{L}{\bowtie} Q \quad | \quad P/L$$

BioPEPA

$$S ::= (\alpha, \kappa)\ op\ S \quad | \quad S + S' \qquad op ::= \uparrow \mid \downarrow \mid \oplus \mid \ominus \mid \odot$$

$$P ::= P \underset{L}{\bowtie} P' \quad | \quad S(x)$$

Advertisement

Complementary Approaches to Understanding the Plant Circadian Clock

Maria Luisa Guerriero
Centre for Systems Biology at Edinburgh

PEPA club talk
Room 3.02, Informatics Forum
11am Friday 12 March 2010