PERSPECTIVES ON CONSTRAINTS, PROCESS ALGEBRAS, AND HYBRID SYSTEMS

Luca Bortolussi\textsuperscript{1}   Alberto Policriti\textsuperscript{2}

\textsuperscript{1}Dipartimento di Matematica ed Informatica
Università degli studi di Trieste
luca@dmi.units.it

\textsuperscript{2}Dipartimento di Matematica ed Informatica
Università degli studi di Udine
policriti@dimi.uniud.it

WCB, Edinburgh, 21\textsuperscript{th} July 2010
THE BEGINNING OF THE STORY

COMPUTATIONAL SYSTEMS BIOLOGY

Graphical Languages

Stochastic Process Algebras:

Basic math. language

Analysis methods

Simulation Model checking

Analysis methods

Simulation Model checking

ODE

continuous

deterministic

hybrid systems

Stochastic Processes

discrete

stochastic

Formal Semantics
The beginning of the story

**PROS**
- Formalization of knowledge
- Compositionality

**CONS**
- Hard to deal with different kinds of biological information
- Difficult to describe complex operations

- **Analysis methods**
  - Simulation
  - Model checking

- **Description Language**
  - Basic math. language
  - Stochastic Process Algebras:
    - Continuous deterministic
    - Discrete stochastic
    - Hybrid systems

- **ODE**
- **Stochastic Processes**

- **Formal Semantics**
THE BEGINNING OF THE STORY

Stochastic Process Algebras: sCCP

Basic math. language

continuous deterministic

ODE

hybrid systems

Stochastic Processes

discrete stochastic

Analysis methods
Simulation  Model checking

Model checking

Formal Semantics
OUTLINE

1. INTRODUCTION

2. STOCHASTIC CONCURRENT CONSTRAINT PROGRAMMING

3. HYBRID SEMANTICS FOR sCCP

4. PERSPECTIVES
**Stochastic Concurrent Constraint Programming**

**Constraint Store**
- The informational unit are **constraints**, which are *formulae over an interpreted first order language* (i.e. $X = 10$, $Y > X - 3$).
- Constraints are stored in a shared memory, the **constraint store**.

**Agents**
Agents can perform two basic operations on this store:
- Add a constraint (**tell** ask)
- Ask if a certain relation is entailed by the current configuration (**ask** instruction)

Each basic instruction has a **rate** attached to it, i.e. **a function $\lambda$ from the constraint store $C$ to positive reals**.

**Syntax of CCP**

```
Program = D.A
D = \varepsilon \mid D.D \mid p(x) : \neg A
\pi = \text{tell}_\lambda(c) \mid \text{ask}_\lambda(c)
M = \pi.A \mid \pi.p(y) \mid M + M
A = 0 \mid \exists x A \mid M \mid (A \parallel A)
```
The distinction between agents and constraints in sCCP allows the separation of the description of interaction patterns and the effect of each interaction.

- The state of the system is represented in the constraint store $\mathcal{C}$, and described by a set of (stream) variables and constraints.
- Each agent can perform one or more actions that alter the state of $\mathcal{C}$; their internal states can control its interaction capabilities.
- Constraints allow to define arbitrary complex descriptions of systems and arbitrary complex modifications.
STOCHASTIC CONCURRENT CONSTRAINT PROGRAMMING

**MODEL IN sCCP**

- Protein $P$: store variable $X$
- Gene: agent with two internal states: on and off.
- Degradation: single-state agent.


$$\text{gene}_{\text{on}}(X) \parallel \text{degrade}(X)$$

$$\text{gene}_{\text{on}}(X) :\text{ tell}(\text{increase}(X))_{k_p} \cdot \text{gene}_{\text{on}}(X) + \text{ask}(X > 0)_{k_b}X \cdot \text{gene}_{\text{off}}(X)$$

$$\text{gene}_{\text{off}}(X) :\text{ ask}(\text{true})_{k_u} \cdot \text{gene}_{\text{on}}(X)$$

$$\text{degrade}(X) :\text{ ask}(X > 0)_{k_d}X \cdot \text{tell}(\text{decrease}(X))_{\infty} \cdot \text{degrade}(X)$$
**Stochastic Concurrent Constraint Programming**

Model in sCCP:
- **Protein** $P$: store variable $X$
- **Gene**: agent with two internal states: on and off.
- **Degradation**: single-state agent.


\[
\begin{align*}
\text{gene}_{on}(X) \parallel \text{degrade}(X) \\
\text{gene}_{on}(X) & \leftarrow \text{tell}(\text{increase}(X))_{k_p}.\text{gene}_{on}(X) + \text{ask}(X > 0)_{k_b}X.\text{gene}_{off}(X) \\
\text{gene}_{off}(X) & \leftarrow \text{ask}(\text{true})_{k_u}.\text{gene}_{on}(X) \\
\text{degrade}(X) & \leftarrow \text{ask}(X > 0)_{k_d}X.\text{tell}(\text{decrease}(X))_{\infty}.\text{degrade}(X)
\end{align*}
\]
SEMANTICS FOR sCCP

Stochastic Process Algebras

sCCP

CTMC

Stochastic

Hybrid

Automata

ODE
The standard sCCP semantics is given in terms of Continuous Time Markov Chains (CTMC), constructed from the labeled transition system derived by operational rules.

A CTMC is a direct graph with edges labeled by a real number, called the rate of the transition (representing the speed or the frequency at which the transition occurs).

- In each state, we select the next state according to a probability distribution obtained normalizing rates (from $S$ to $S_1$ with prob. $\frac{r_1}{r_1+r_2}$).
- The time spent in a state is given by an exponentially distributed random variable, with rate given by the sum of outgoing transitions from the actual node ($r_1 + r_2$).
Fluid Semantics

\[
\begin{align*}
gene_{on}(X) & :\ - \ tell(X' = X + 1)_{k_p} \cdot gene_{on}(X) + ask(X > 0)_{k_b} X \cdot gene_{off}(X) \\
gene_{off}(X) & :\ - \ ask(\text{true})_{k_u} \cdot gene_{on}(X) \\
degraded(X) & :\ - \ ask(X > 0)_{k_d} X \cdot tell(X' = X - 1)_{\infty} \cdot degraded(X)
\end{align*}
\]

\[
\begin{align*}
\dot{X} &= k_p G_1 - k_d X \\
\dot{G}_1 &= k_u G_0 - k_b X G_1 \\
\dot{G}_0 &= k_b X G_1 - k_u G_0
\end{align*}
\]
CTMC vs ODE

**Issues**

- Fluid semantics has just (approximate) information about average
- The CTMC and ODE systems are not “behaviorally equivalent” (in terms of CSL/CTL formulae)
Associate a Stochastic Hybrid Automaton to each sCCP agent in parallel. Then, compose them by a product construction.
Associate a Stochastic Hybrid Automaton to each sCCP agent in parallel. Then, compose them by a product construction.
Associate a Hybrid Automaton to each sCCP agent in parallel. Then, compose them by a product construction.
**Dynamics**

- **stochastic**
  - SHA, bind/unbind discrete

- **ODE**

- **non-stochastic HA**
Different partitions of edges into continuous and discrete correspond to different TDSHA.

They can be arranged into a lattice of hybrid automata w.r.t. increasing degree of continuity.

Bottom element: CTMC associated to sCCP by its standard semantics.

Top element: fluid approximation of sCCP.
OPEN ISSUES: CONSTRAINT STORE

EXPLOTING CONSTRAINTS
Programming the constraint store, we can model systems at different complexity, like:

- dynamics of compartments (WCB09)
- formation of biochemical complexes (BIBM07)

THE NEED TO KEEP THEM SIMPLE
The hybrid semantics is defined for simple constraints, i.e. updates of “stream” variables by constant quantities.

How can we reconcile these two features?

“Ground” store: only stream vars and simple updates.
Notion of equivalent constraint stores (w.r.t. operations on stream variables).
Equivalence preserving transformations.
Open Issues: Comparison of Semantics

Given two different semantics of the same sCCP program, how can we compare them?

Limit Behavior
Convergence of CTMC to ODE and to SHA in the limit of infinite populations (ASMTA 2010).

Logic Based Comparison
Two semantics are equivalent w.r.t. (a subset of) the formulae of a suitable logic. How can we compare stochastic and non-stochastic models (measure theory/category theory)?
**Open Issues: Choosing the Right Partitioning**

When constructing the hybrid semantics, we can choose how to partition actions into discrete and continuous.

**What is the “right” partitioning?**

Criteria to evaluate a partitioning: exhibited behavior, efficiency, accuracy.

**Stochastic Models**

Accurate, but not efficient (simulation is costly, model checking prohibitively costly).

**Non-stochastic Hybrid Automata**

Simulation is cheap, model checking can be **undecidable**! The fully continuous model is decidable, the fully discrete (timed automaton) is not.

Issue: find the best decidable model
This is the end!

Well, almost the end...
This is the end!

Well, almost the end...
Sixth International School on Biology, Computation and Information
September 20–24, Dobbiaco, Italy.

Key lecturers: Eugene Myers, Jasmin Fisher, Bud Mishra

http://www.dmi.units.it/bci2010/