Stochastic Concurrent Constraint Programming

Luca Bortolussi

Department of Mathematics and Computer Science
University of Udine, Italy.

PRIN meeting, Padova, 10\textsuperscript{nd} April 2006
Outline

1. Introduction
   - Concurrent Constraint Programming
   - Continuous Time Markov Chains

2. Syntax and Operational Semantic
   - Syntax and Rates
   - Operational Semantic

3. Examples
   - Random Walk
   - Modeling Biochemical Reactions
   - Modeling Gene Regulatory Networks
Outline

1 Introduction
   - Concurrent Constraint Programming
   - Continuous Time Markov Chains

2 Syntax and Operational Semantic
   - Syntax and Rates
   - Operational Semantic

3 Examples
   - Random Walk
   - Modeling Biochemical Reactions
   - Modeling Gene Regulatory Networks
Introduction

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 "pot", called the constraint store, but can never be removed.

Syntax of CCP

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)

Syntax of CCP

$$
\begin{align*}
\text{Program} &= \text{Decl}.A \\
D &= \varepsilon \mid \text{Decl}.\text{Decl} \mid p(x) : \neg A \\
A &= 0 \\
&\mid \text{tell}(c).A \\
&\mid \text{ask}(c_1).A_1 + \text{ask}(c_2).A_2 \\
&\mid A_1 \parallel A_2 \mid \exists x.A \mid p(x)
\end{align*}
$$
A **Continuous Time Markov Chain** (CTMC) is a directed 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 $S_0$ to $S_1$ with prob. $\frac{\lambda_1}{\lambda_1 + \lambda_2}$).

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 ($\lambda_1 + \lambda_2$).
Outline

1. Introduction
   - Concurrent Constraint Programming
   - Continuous Time Markov Chains

2. Syntax and Operational Semantic
   - Syntax and Rates
   - Operational Semantic

3. Examples
   - Random Walk
   - Modeling Biochemical Reactions
   - Modeling Gene Regulatory Networks
Syntax of Stochastic CCP

\[
\text{Program} = \text{Decl}.A \\
D = \varepsilon | \text{Decl}.\text{Decl} | p(x) : -A \\
\begin{align*}
A &= \mathbf{0} | \text{tell}_\lambda(c).A | \text{ask}_\lambda(c).A | [p(x)]_\lambda | \\
    &\quad \exists x A \mid (A_1 + A_2) \mid (A_1 \parallel A_2)
\end{align*}
\]

Each basic instruction (tell, ask, procedure call) has a rate attached to it. Rates are functions from the constraint store \( C \) to positive reals: \( \lambda : C \rightarrow \mathbb{R}^+ \).
SOS

\[ \langle \text{tell}_\lambda(c).A, d, V \rangle \longrightarrow_{(1, \lambda(d))} \langle A, d \sqcup c, V \rangle \]

\[ \langle \text{ask}_\lambda(c), d, V \rangle \longrightarrow_{(1, \lambda(d))} \langle A, d, V \rangle \quad \text{if } d \vdash c \]

\[ \langle [p(y)]_\lambda, d, V \rangle \longrightarrow_{(1, \lambda(d))} \langle A[y/x], d, V \rangle \quad \text{if } p(x) : -A \]

\[ \frac{\langle A_1, d, V \rangle \longrightarrow_{(p, \eta)} \langle A'_1, d', V \rangle}{\langle A_1 + A_2, d, V \rangle \longrightarrow_{(p', \eta')} \langle A'_1, d', V \rangle} \]

\[ \frac{\langle A_1, d, V \rangle \longrightarrow_{(p, \eta)} \langle A'_1, d', V \rangle}{\langle A_1 \parallel A_2, d, V \rangle \longrightarrow_{(p', \eta')} \langle A'_1 \parallel A_2, d', V \rangle} \]

with \( p' = \frac{p \eta}{\eta + \text{rate}(A_2, d)} \) and \( \eta' = \eta + \text{rate}(A_2, d) \)

rate returns the sum of rates of all active agents, evaluated w.r.t. the current configuration of the store.
**Operational Semantic**

**Examples**

**Example**

\[
\langle \text{tell}_1(c), \top, \emptyset \rangle \longrightarrow_{(1,1)} \langle 0, c, \emptyset \rangle
\]

\[
\langle \text{ask}_1(c).\text{tell}_{100}(d) \parallel \text{tell}_1(c), \top, \emptyset \rangle
\]

\[
\longrightarrow_{(1,1)} \langle \text{ask}_1(c).\text{tell}_{100}(d'), c, \emptyset \rangle
\]

\[
\longrightarrow_{(1,1)} \langle \text{tell}_{100}(d'), c, \emptyset \rangle
\]

\[
\longrightarrow_{(1,1,100)} \langle 0, c \sqcup d, \emptyset \rangle.
\]

**Example**

\[
\langle \text{tell}_1(c) + \text{tell}_1(d), \top, \emptyset \rangle \longrightarrow_{(0.5,2)} \langle 0, c, \emptyset \rangle
\]

\[
\langle \text{tell}_1(c) + \text{tell}_1(d), \top, \emptyset \rangle \longrightarrow_{(0.5,2)} \langle 0, d, \emptyset \rangle
\]
Outline

1. Introduction
   - Concurrent Constraint Programming
   - Continuous Time Markov Chains

2. Syntax and Operational Semantic
   - Syntax and Rates
   - Operational Semantic

3. Examples
   - Random Walk
   - Modeling Biochemical Reactions
   - Modeling Gene Regulatory Networks
The language has been implemented by writing a meta-interpreter in SICStus Prolog.

We can model random walk as a stochastic process increasing or diminishing of one unit the value of a variable \( X \).

\[
\text{random}(X) :- \\
\exists Y ( \text{tell}_1(Y = X + 1) \ \\
+ \text{tell}_1(Y = X - 1)) . \text{random}(Y)
\]
The stochastic extension of Concurrent Constraint Programming can be used to model biological systems, similarly to $\pi$-calculus.

$\pi$-calculus for system biology

<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>
The stochastic extension of Concurrent Constraint Programming can be used to model biological systems, similarly to $\pi$-calculus.

**CCP for Biological Simulation**

- Molecule (Type) / Reaction $\leftrightarrow$ Process
- Modification $\leftrightarrow$ State change
- Environment $\leftrightarrow$ Constraint Store
- Interaction with Environment $\leftrightarrow$ Asynchronous Communication
- Direct Interaction capability $\leftrightarrow$ Channel
- Interaction $\leftrightarrow$ Synchronous Communication

We need to extend the concept of rate: a rate needs to be a function $\lambda : C \rightarrow \mathbb{R}^+$. 
A simple reaction: $H + Cl \Leftrightarrow HCl$

**π-calculus**

We model atom H and atom Cl. Reaction happens by a synchronous communication of these two processes. We need several copy of these processes.

Covalent Bonding: $H + Cl \Leftrightarrow HCl$

- $H$ has a private electron $e$.
- $H$ can share its electron with $Cl$ to form $HCl$, with rate$(share) = 100\text{s}^{-1}$
- $HCl$ can break its private bond, with rate$(e) = 10\text{s}^{-1}$

A. Phillips - Feb. 2005
A simple reaction: $H + Cl \rightleftharpoons HCl$

We write a reaction agent, while the reagents and the product are modeled in the constraint store (put down in the environment). Independent on the number of agents.

```
reaction(H, CL, HCL) :-
    ( tell_{shareRate}(H, Cl) (share(H, CL, HCL)) +
    tell_{relRate}(H, Cl) (rel(H, CL, HCL))
    ).reaction(H, CL, HCL)
```
A simple reaction: $\text{H} + \text{Cl} \rightleftharpoons \text{HCl}$

**sCCP**

We write a reaction agent, while the reagents and the product are modeled in the constraint store (put down in the environment). Independent on the number of agents.
Another reaction: $\text{Na} + \text{Cl} \leftrightarrow \text{Na}^+ + \text{Cl}^-$

$$\text{ionization}(\text{Na}, \text{Cl}, \text{Na}^+, \text{Cl}^-) :-$$

(  
  $\text{tell}_{\text{ionizeRate}}(\text{Na}^+, \text{Cl}^-) (\text{ionize}(\text{Na}, \text{Cl}, \text{Na}^+, \text{Cl}^-)) +$
  $\text{tell}_{\text{deionizeRate}}(\text{Na}^+, \text{Cl}^-) (\text{deionize}(\text{Na}, \text{Cl}, \text{Na}^+, \text{Cl}^-))$
).

$\text{ionization}(\text{Na}, \text{Cl}, \text{Na}^+, \text{Cl}^-)$
The gene machine

Modeling Gene Regulatory Networks

Introduction
Syntax and Operational Semantic
Examples

Luca Bortolussi
Stochastic CCP
The instruction set

degradator(X) :- \text{tell}
\text{degRate}(X)(\text{degrade}(X)).\text{degradator}(X)

null(X) :- \text{tell}
\text{prodRate}(X)(\text{produce}(X)).\text{null}(X)

pos(X, Y) :- ( \text{tell}
\text{prodRate}(X)(\text{produce}(X))
+ \text{ask}
\text{enhanceRate}(Y)(Y > 0).\text{tell}
\text{enhProdRate}(X)(\text{produce}(X))
).\text{pos}(X, Y)

neg(X, Y) :- ( \text{tell}
\text{prodRate}(X)(\text{produce}(X))
+ \text{ask}
\text{inhibitRate}(Y)(Y > 0).\text{ask}
\text{delayRate}(X)(\text{true})
).\text{neg}(X, Y)

null_gate(X) :- \text{null}(X) \parallel \text{degradator}(X)

pos_gate(X, Y) :- \text{pos}(X, Y) \parallel \text{degradator}(X)

neg_gate(X, Y) :- \text{neg}(X, Y) \parallel \text{degradator}(X)

Null Gate

null \rightarrow b

neg\_gate(A)

null(a) time evolution in sCCP
Modeling Gene Regulatory Networks

Pos Gate

\[
\text{pos\_gate}(A, B)
\]

\begin{figure}
\centering
\includegraphics[width=\textwidth]{pos_gate.png}
\caption{Pos gate time evolution in sCCP}
\end{figure}
Neg Gate

\[ \text{neg}_\text{gate}(A, B) \]

Modeling Gene Regulatory Networks

Luca Bortolussi

Stochastic CCP
Self Repression

\[ \text{neg\_gate}(A, A) \]
Mutual Repression

\[ \text{neg\_gate}(A, B) \parallel \text{neg\_gate}(B, A) \]
Towards verification of models

PRISM

We have defined a mapping from a sublanguage of sCCP (restriction on parallel operators) to the modeling language of PRIMS, a symbolic probabilistic model checker.

Expected value of protein $b$ at time $T$. 

Luca Bortolussi

Stochastic CCP
We have introduced a stochastic version of CCP.
We showed that sCCP may be used for modeling biological systems, via examples.
We showed first examples of verifying properties of these systems, using PRISM.
THANKS FOR THE ATTENTION!

QUESTIONS?