PERSPECTIVES ON CONSTRAINTS, PROCESS ALGEBRAS, AND HYBRID SYSTEMS

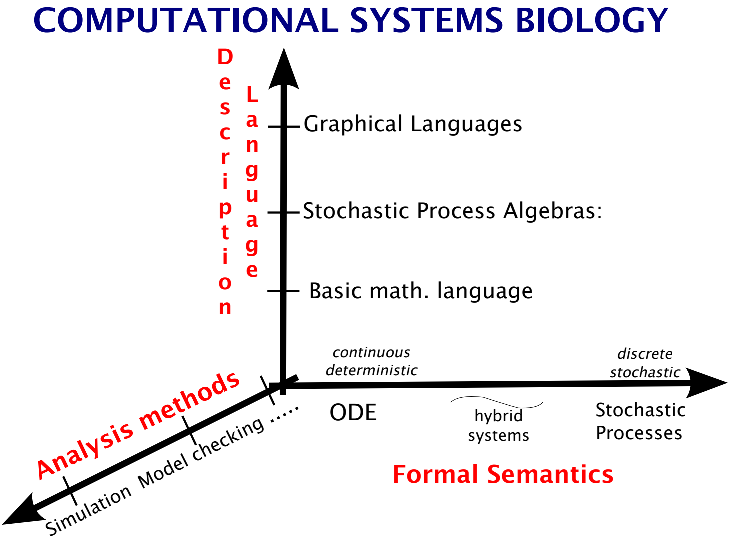
Luca Bortolussi[1] Alberto Policriti[2]

[1]Dipartimento di Matematica ed Informatica
Università degli studi di Trieste
luca@dmi.units.it

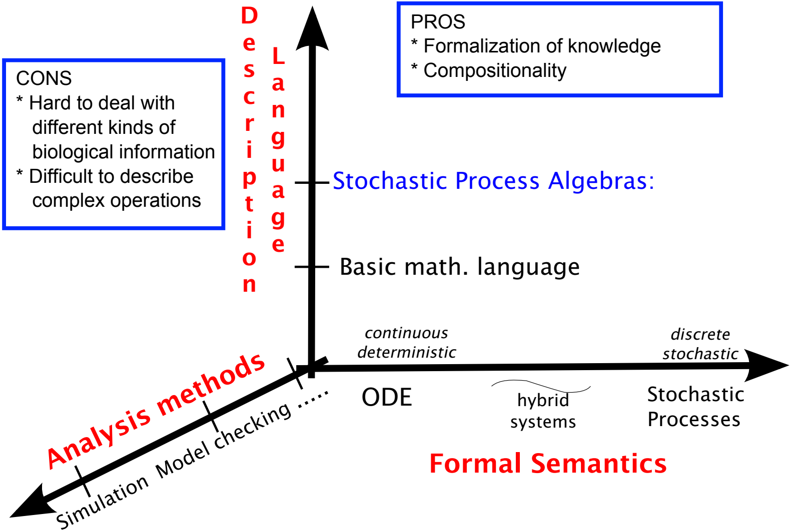
[2]Dipartimento di Matematica ed Informatica
Università degli studi di Udine
policriti@dimi.uniud.it

WCB, Edinburgh, 21th July 2010

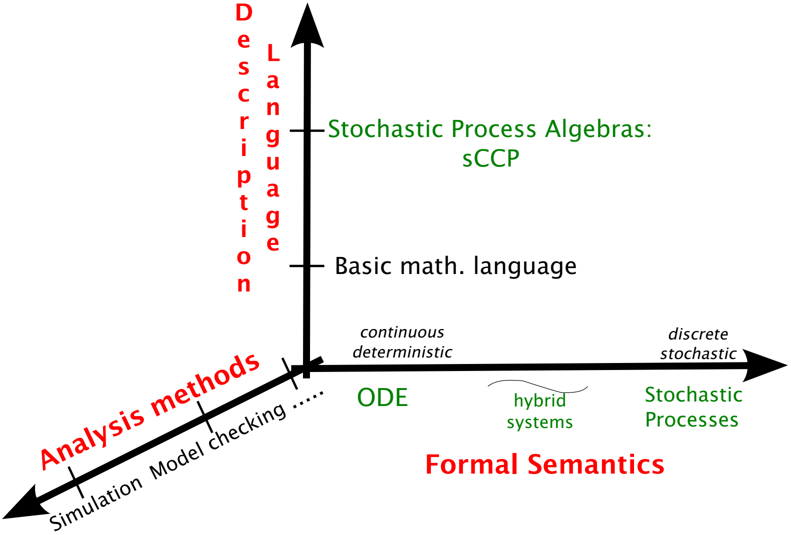
THE BEGINNING OF THE STORY



THE BEGINNING OF THE STORY



THE BEGINNING OF THE STORY



OUTLINE

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 λ from the constraint store $\mathcal{C}$ to positive reals*.

SYNTAX OF CCP

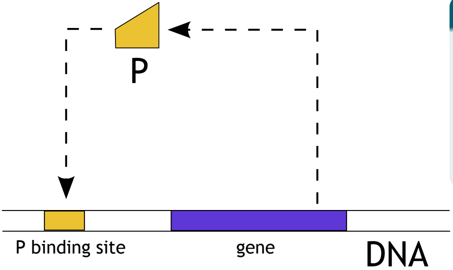
$$\begin{aligned}
&Program = D.A \\
&D = \varepsilon \mid D.D \mid p(\mathbf{x}) : -A \\
&\pi = \mathrm{tell}_\lambda(c) \mid \mathrm{ask}_\lambda(c) \\
&M = \pi.A \mid \pi.p(\mathbf{y}) \mid M + M \\
&A = \mathbf{0} \mid \exists_x A \mid M \mid (A \parallel A)
\end{aligned}$$

MODELING BIOLOGICAL SYSTEMS IN sCCP

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.

- L. Bortolussi, A. Policriti. Modeling Biological systems in sCCP, *Constraints*, 13(1), 2008.

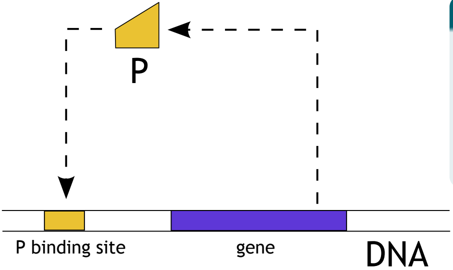
$$\mathtt{gene}_{\mathrm{on}}(X) \parallel \mathtt{degrade}(X)$$

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

$$\mathtt{gene}_{\mathrm{off}}(X) \text{ :- } \mathrm{ask}(\mathit{true})_{k_u}.\mathtt{gene}_{\mathrm{on}}(X)$$

$$\mathtt{degrade}(X) \text{ :- } \mathrm{ask}(X > 0)_{k_d X}.\mathrm{tell}(\mathrm{decrease}(X))_{\infty}.\mathtt{degrade}(X)$$

Stochastic Concurrent Constraint Programming



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

Model in sCCP

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

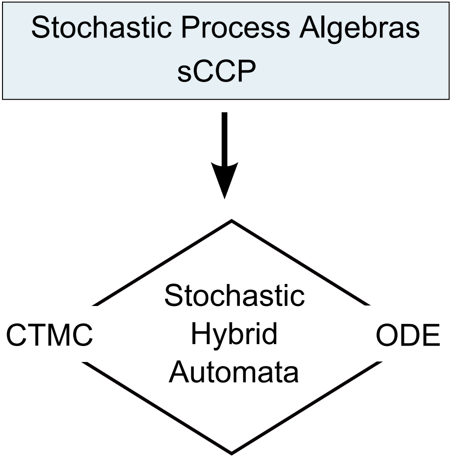
- L. Bortolussi, A. Policriti. Modeling Biological systems in sCCP, *Constraints*, 13(1), 2008.

$$\mathtt{gene}_{\mathrm{on}}(X) :\!-\ \mathrm{tell}(\mathrm{increase}(X))_{k_p}.\mathtt{gene}_{\mathrm{on}}(X) + \mathrm{ask}(X > 0)_{k_b X}.\mathtt{gene}_{\mathrm{off}}(X)$$

$$\mathtt{gene}_{\mathrm{off}}(X) :\!-\ \mathrm{ask}(\mathit{true})_{k_u}.\mathtt{gene}_{\mathrm{on}}(X)$$

$$\mathtt{degrade}(X) :\!-\ \mathrm{ask}(X > 0)_{k_d X}.\mathrm{tell}(\mathrm{decrease}(X))_{\infty}.\mathtt{degrade}(X)$$

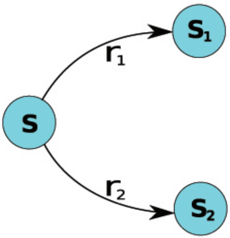
SEMANTICS FOR sCCP



CONTINUOUS TIME MARKOV CHAINS

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).



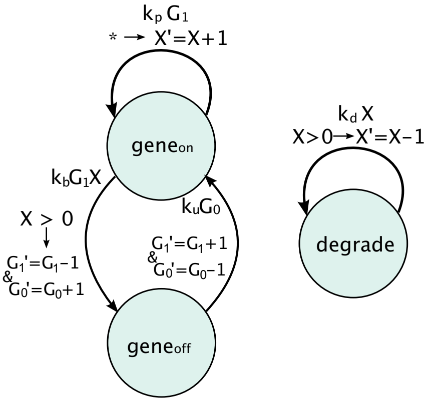
- In each state, we select the next state according to a *probability distribution* obtained normalizing rates (from S to S_1 with prob. $\frac{r_1}{r_1+r_2}$).
- The time spent in a state is given by an exponentially distributed random variable, with rate given by the *sum of outgoing transitions* from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

FLUID SEMANTICS

$$\mathtt{gene_{on}}(X) \text{ :- } \mathsf{tell}(X' = X + 1)_{k_p}.\mathtt{gene_{on}}(X) + \mathsf{ask}(X > 0)_{k_b X}.\mathtt{gene_{off}}(X)$$

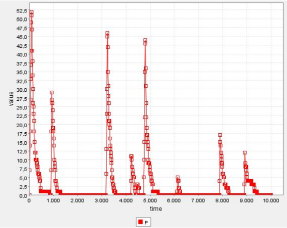
$$\mathtt{gene_{off}}(X) \text{ :- } \mathsf{ask}(\mathit{true})_{k_u}.\mathtt{gene_{on}}(X)$$

$$\mathtt{degrade}(X) \text{ :- } \mathsf{ask}(X > 0)_{k_d X}.\mathsf{tell}(X' = X - 1)_{\infty}.\mathtt{degrade}(X)$$

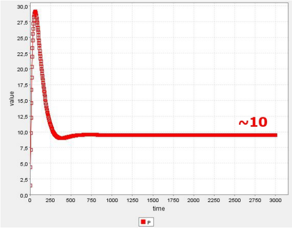


$$\begin{cases} \dot{X} = k_p G_1 - k_d X \\ \dot{G}_1 = k_u G_0 - k_b X G_1 \\ \dot{G}_0 = k_b X G_1 - k_u G_0 \end{cases}$$

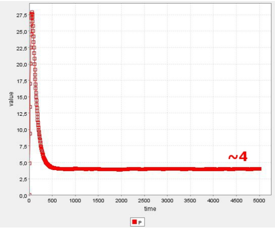
CTMC vs ODE



stochastic



ODE



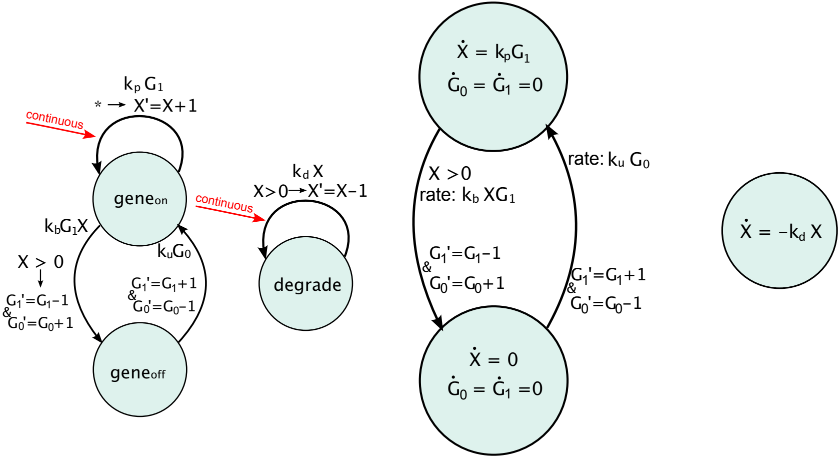
average

Issues

- Fluid semantics has just (approximate) information about average
- The CTMC and ODE systems are not “behaviorally equivalent” (in terms of CSL/CTL formulae)

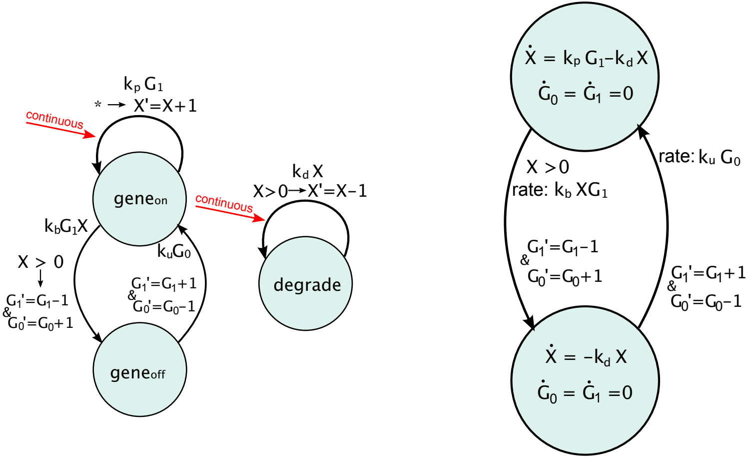
HYBRID SEMANTICS

Associate a Stochastic Hybrid Automaton to each **sCCP** agent in parallel. Then, compose them by a product construction.



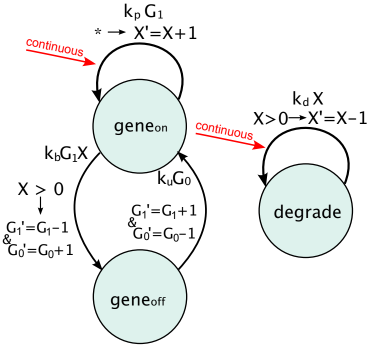
HYBRID SEMANTICS

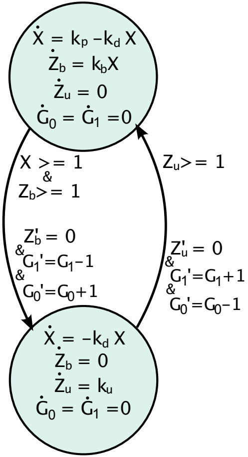
Associate a Stochastic Hybrid Automaton to each **sCCP** agent in parallel. Then, compose them by a product construction.



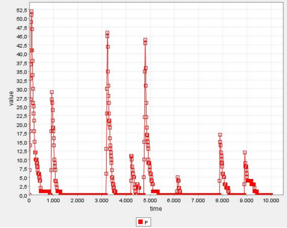
HYBRID SEMANTICS

Associate a Hybrid Automaton to each **sCCP** agent in parallel.
Then, compose them by a product construction.

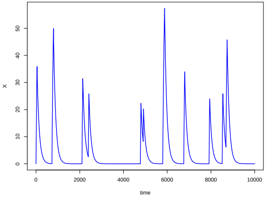




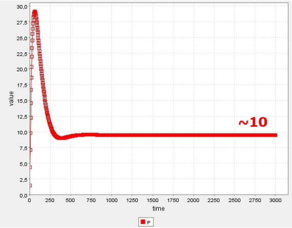
DYNAMICS



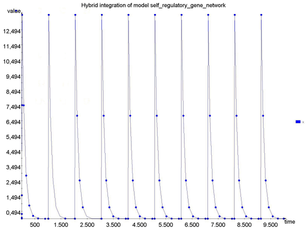
stochastic



SHA, bind/unbind discrete



ODE



non-stochastic HA

LATTICE OF TDSHA

- 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**.

Fluid ODE

⋮

SHA

⋮

CTMC

OPEN ISSUES: CONSTRAINT STORE

EXPLOITING 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/cathegory 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 prohibitivly 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

THANKS FOR THE ATTENTION

This is the end!

Well, almost the end...

THANKS FOR THE ATTENTION

This is the end!

Well, almost the end...

ANNOUNCEMENT: BCI 2010

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/