An Introduction to Stochastic π-Calculus

Linguaggi e Modelli Computazionali L-S
Prof. Mirko Viroli

Dott. Ing. Luca Gardelli
{luca.gardelli}@unibo.it

II Facoltà di Ingegneria
Via Venezia, 52 - 47023 Cesena (FC), Italy
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2007/2008

Outline

Quantitative Aspects in Process Algebras

- Process Algebras (PA) allow to model qualitative aspects of systems, i.e. behavioural descriptions
- Quantitative aspects are very important, particularly when dealing with concurrent or distributed systems
- Furthermore quantitative aspects allow to reason about performance

Probabilistic vs. Temporal Process Algebra

- An attempt to model quantitative aspects is to consider probability or time
- Probabilistic PA attach probabilities to branching point, ruling out nondeterminism
- Temporal PA associate a fixed duration to each transition to model execution time

Stochastic Process Algebra

- The approach followed by Stochastic PA is to add probabilistic distributions to prefixes
- The firing of a prefix occurs after a delay of Δt defined by the respective probabilistic distribution
- Such delay can model the actual duration of the action associated with the prefix

Exponential Distribution

- Most PAs in the literature are defined according to exponential distribution $P = 1 - e^{-rt}$, where the mean value is $1/r$
- Exponential distributions are completely characterised by a single parameter r
- Exponential distributions enjoy the memoryless property, that is the duration of a transition is independent from the history of transitions

Action Duration

- In stochastic PA a prefix is defined by a couple (a,r) where a is the action and r is the *activity rate*
- The activity rate denote the duration of an action according to the exponential distribution
- When r is specified the activity is termed *active* otherwise it is *passive*

Stochastic PA and Markov Chains

- Exponential Distribution allow to derive from a PA specification a continuous time Markov Chain (CTMC)
- A CTMC is a memoryless stochastic process, i.e. a collection of time dependent random variables following an exponential distribution [Brinksma and Hermanns, 2001]
- Markov Chains are commonly used in numerical techniques to obtain performance measures
- Particularly interesting for the performance evaluation task is the stochastic process algebra PEPA

More on Markov Chains

- A CTMC can be represented by a labelled transition system where nodes represent the states and transitions are labelled with rates [Brinksma and Hermanns, 2001]
- The probability for a specific transition to happen is given by the ratio between the rate and the sum of the exit rates

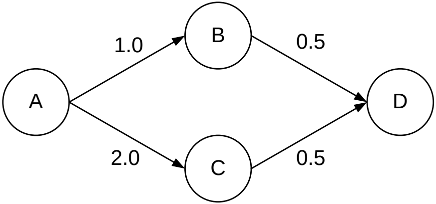
$$P_j = \frac{r_j}{\sum_i r_i}$$

- Given the selected transition, the duration of the actions is distributed according to the sum of rates

$$P = 1 - e^{-\sum_i r_i t}$$

CTMC Example

- Consider the CTMC in the Figure below: from the initial state A the probability to move to B is $P(A \rightarrow B) = 1/(1+2) = 0.\overline{3}$, while the probability to move to C is $P(A \rightarrow C) = 2/(1+2) = 0.\overline{6}$
- The duration of both transitions is distributed exponentially according to the sum of exit rates $r_t = 1.0 + 2.0 = 3.0$



Outline

π-Calculus Syntax

- A process in π-calculus is defined by Milner [Milner et al., 1992] [Milner, 1999] accordingly to the following syntax

$$P ::= 0 \mid P_1 + P_2 \mid \bar{y}x.P \mid y(x).P \mid \tau.P \mid P_1|P_2 \mid (\nu x)P \mid [x = y]P \mid !P$$

- 0 is the empty process, and it's called *inaction*, often omitted
- the silent prefix τ means that a silent action is performed
- the replication $!P$ means that you can have as many copies – but a finite number – as you wish, i.e. $P|P|P|..$

π-Calculus Syntax

- summation $P_1 + P_2$ means that the process can perform P_1 or P_2
- the prefix $\bar{y}x$ is a sort of output port, so $\bar{y}x.P$ means send x across y channel and then behave like P
- the prefix $y(x)$ is a sort of input port, so $y(x).P$ means receive a value across y channel, name it x and then behave like P
- the composition $P_1|P_2$ means that the two processes are executed in parallel
- the restriction $(\nu x)P$ means that the process behaves like P except for the fact that any action across x channel is prohibited
- $[x = y]P$ means that the process behaves like P if y matches x, otherwise 0

Stochastic π-Calculus Syntax

- The following description of the stochastic π-Calculus is based on the proposal in [Priami, 1995]
- Prefixes are annotated by the activity rate, i.e. prefixes become a pair (a,r) where a is the action and r is the activity rate
- The time to complete the activity Δt is drawn according to the exponential distribution defined by r

Race Condition

- In summation the first activity that completes is executed while the others are discarded
- All the activities enabled attempt to proceed, although only the fastest one is executed: this criteria is called *race condition*
- Each time the fastest activity may differ since execution time is a random variable, hence it involves a probabilistic choice

Probabilistic Choice

- The probability for an action to be executed is given by the ratio of its rate and the *exit rate*, i.e. the sum of the rates enabled
- For example in (a,2)+(b,6) a has a probability of $2/(2+6) = 0.25$ of being executed
- In general, the probability for an action i of being chosen among the j enabled actions is

$$p_i = \frac{r_i}{\sum_{j=1}^{n} r_j}, \quad 1 \leq i \leq n \ . \tag{1}$$

Growing Interest

- Interest in Stochastic π-Calculus has grown considerably in the last decade
- In particular it has been exploited for
  - modelling concurrent and distributed systems
  - evaluate performance of distributed systems
  - simulate and analyse systems, especially with applications in chemistry and biology

Outline

SPiM

- SPiM [Phillips and Cardelli, 2004][Phillips, 2007b] is a simulator for stochastic π-Calculus specifications, initially developed for investigating biological systems
- The correctness of the machine has been formally proven with respect to the calculus
- The machine defines a variant of $\pi - \mathit{calculus}$
- Notice also that for programming convenience SPiM language use a different syntax from standard π-Calculus: for details please refer to the SPiM Language Manual [Phillips, 2007a]

Stochastic Selection Algorithm

- The stochastic selection algorithm is based on [Gillespie, 1977] and exploits the notion of channel activity, which is defined by

$$Act_x(A) = (In_x(A) \times Out_x(A)) - Mix_x(A) \qquad (2)$$

- where $In_x(A)$ and $Out_x(A)$ are the number of unguarded inputs and outputs on channel x in A
- and $MiX_x(A)$ is the sum of $In_x(\Sigma_i) \times Out_x(\Sigma_i)$ for each Σ_i in A

Stochastic Selection Algorithm

- For all $x \epsilon fn(A)$ calculate $a_x = Act_x(A) \times rate(x)$
- Store non-zero values of a_x in a list (x_μ, a_μ) where $\mu \epsilon 1..M$
- Calculate $a_0 = \sum_{\nu=0}^{M} a_\nu$
- Generate two random numbers $n_1, n_2 \epsilon [0, 1]$ and calculate τ, μ such that:

$$\tau = (1/a_0) ln(1/n_1)$$

$$\sum_{\nu=1}^{\mu-1} a_\nu < n_2 a_0 \leq \sum_{\nu=1}^{\mu} a_\nu$$

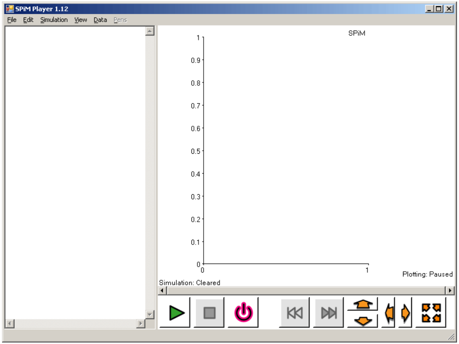
- $Next(A) = x_\mu$ and $Delay(A) = \tau$

SPiM Command Syntax

- In order to use SPiM you have to switch to shell mode: the command syntax is *spim filename.spi*
- As a convention SPiM specifications have the extension *.spi*
- By default, the results are written into a file having the same name of the specification appended with the extension *.csv*, i.e. comma separated values: for example, in the previous example, the output file would be *filename.spi.csv*
- Furthermore SPiM allows to export the specification to a graphical notation: the file has *.dot* extension and can be read by GraphViz [GraphViz, 2007]

SPiM Player

- As an alternative it is possible to use the SPiM player, a Graphical User Interface to SPiM command
- Since it has plotting capabilities it allows previewing the system dynamics without switching to another software



NaCl Example

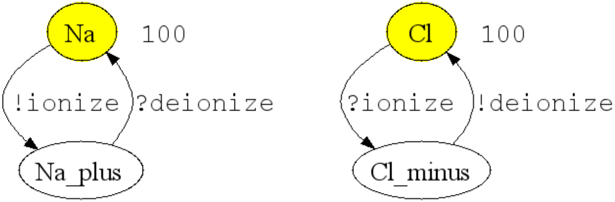
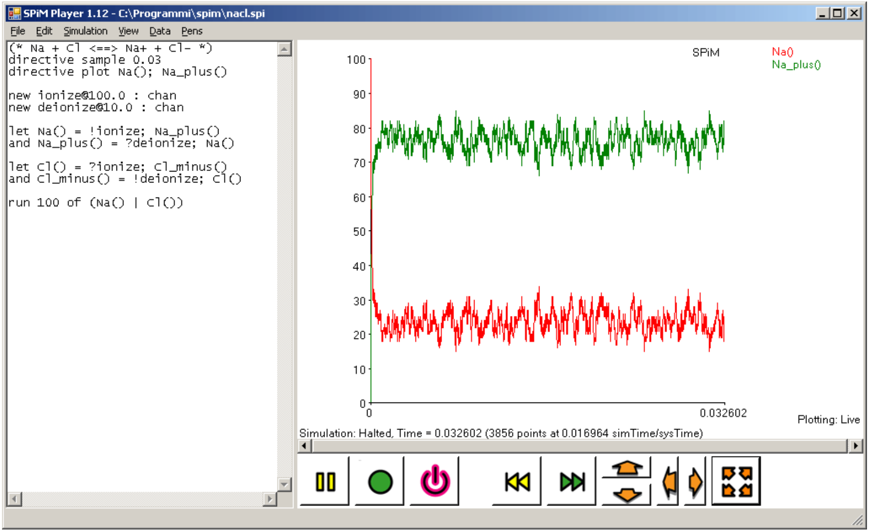


Figure: The representation of the NaCl specification using GraphViz [GraphViz, 2007].

NaCl Example



Dissecting the NaCl Specification

Directives

- directive sample 0.03
- directive plot Na(); Naplus()

Channel Declaration

- new ionize@100.0 : chan
- new deionize@10.0 : chan

Processes Definition

- let Na() = !ionize; Naplus() and Naplus() = ?deionize; Na()
- let Cl() = ?ionize; Clminus() and Clminus() = !deionize; Cl()

Run command

- run 100 of (Na() — Cl())

Dissecting the NaCl Specification

Directives

- directive sample 0.03
- directive plot Na(); Naplus()

Channel Declaration

- new ionize@100.0 : chan
- new deionize@10.0 : chan

Processes Definition

- let Na() = !ionize; Naplus() and Naplus() = ?deionize; Na()
- let Cl() = ?ionize; Clminus() and Clminus() = !deionize; Cl()

Run command

- run 100 of (Na() — Cl())

Dissecting the NaCl Specification

Directives

- directive sample 0.03
- directive plot Na(); Naplus()

Channel Declaration

- new ionize@100.0 : chan
- new deionize@10.0 : chan

Processes Definition

- let Na() = !ionize; Naplus() and Naplus() = ?deionize; Na()
- let Cl() = ?ionize; Clminus() and Clminus() = !deionize; Cl()

Run command

- run 100 of (Na() — Cl())

Dissecting the NaCl Specification

Directives

- directive sample 0.03
- directive plot Na(); Naplus()

Channel Declaration

- new ionize@100.0 : chan
- new deionize@10.0 : chan

Processes Definition

- let Na() = !ionize; Naplus() and Naplus() = ?deionize; Na()
- let Cl() = ?ionize; Clminus() and Clminus() = !deionize; Cl()

Run command

- run 100 of (Na() — Cl())

Outline

Summing up

- Stochastic π-Calculus labels π-Calculus prefixes with rates: rates completely characterise exponential distributions
- This allows the definition of a stochastic selection algorithm and the mapping with Markov Chains
- The modelling bricks in Stochastic π-Calculus are Processes and Channels
- SPiM implements a variant of the Stochastic π-Calculus and allows to run simulations directly from the specifications for quantitative analysis

Brinksma, E. and Hermanns, H. (2001).
Process algebra and markov chains.
In Brinksma, E., Hermanns, H., and Katoen, J.-P., editors, *Lectures on Formal Methods and Performance Analysis : First EEF/Euro Summer School on Trends in Computer Science Berg en Dal, The Netherlands, July 3-7, 2000, Revised Lectures*, volume 2090 of *LNCS*, pages 183–231. Springer.

Gillespie, D. T. (1977).
Exact stochastic simulation of coupled chemical reactions.
The Journal of Physical Chemistry, 81(25):2340–2361.

GraphViz (2007).
Graphviz: Graph visualization software.
Version 2.14.1 available online at http://www.graphviz.org.

Milner, R. (1999).
Communicating and Mobile Systems: the π-Calculus.
Cambridge University Press.

Milner, R., Parrow, J., and Walker, D. (1992).
A calculus of mobile processes, part I/II.
Information and Computation, 100(1).

Phillips, A. (2007a).
The SPiM language.
Version 0.044 available online at http://research.microsoft.com/~aphillip/spim/.

Phillips, A. (2007b).
The stochastic pi machine (SPiM).
Version 0.044 available online at http://research.microsoft.com/~aphillip/spim/.

Phillips, A. and Cardelli, L. (2004).
A correct abstract machine for the stochastic pi-calculus.
In *Concurrent Models in Molecular Biology (Bioconcur'04)*, London.

Priami, C. (1995).
Stochastic π-calculus.
The Computer Journal, 38(7):578–589.