Probabilistic Process Algebra

Lecture #17 of Modeling Concurrent and Probabilistic Systems

Joost-Pieter Katoen and Thomas Noll

Lehrstuhl 2: Software Modeling and Verification

E-mail: `katoen@cs.rwth-aachen.de`

July 3, 2009

Overview Lecture #17

⇒ *Probabilistic process algebra*

- Probabilistic transition systems
- Syntax (for sequential processes)
- Semantics

Motivation

- Realistic systems are complex and consist of many components

$\Rightarrow$ A monolithic modeling approach is insufficient

  - it will yield complicated and incomprehensible models

- Proposal: use a compositional approach
  - we will adopt process algebra as a framework

- Advantages:
  - models of components can be glued together to obtain complete system models
  - using (bi)simulation relations, models can be compared
  - if these notions are congruences, this comparison can be done component-wise
  - axioms can be used to simplify models at a syntactic level

first step: equip FPS with potential for interaction (= actions)

Fully probabilistic system

A *fully probabilistic system* (FPS) is a pair $\mathcal{D} = (S, \mathbf{P})$ where:

- S is a countable set of states
- $\mathbf{P} : S \times S \to [0, 1]$ is a probability transition function satisfying

$$\sum_{s' \in S} \mathbf{P}(s, s') \in [0, 1] \quad \text{for all} \quad s \in S$$

Probabilistic transition system

A *probabilistic transition system* is a quadruple $(S, \mathit{Act}, \mathbf{P}, s_0)$ where

- S is a countable set of states and $s_0 \in S$ is the initial state
- *Act* is a countable set of actions, and
- $\mathbf{P} \in S \times \mathit{Act} \times S \to [0,1]$ a transition probability function satisfying:

$$\sum_{\alpha} \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in [0,1] \quad \text{for each} \quad \alpha \in \mathit{Act} \text{ and } s \in S$$

ignoring actions yields a fully probabilistic system (FPS)

Probabilistic transition system

- $\mathbf{P}(s, \alpha, s')$ = probability to move from state s to s' by performing α
- For $C \subseteq S$, let $\mathbf{P}(s, \alpha, C) = \sum_{s' \in C} \mathbf{P}(s, \alpha, s')$
  - $\mathbf{P}(s, \alpha, C)$ is the probability to move from s to C by performing action α
- The deadlock probability of state s on action α is:

$$\mathbf{P}(s, \alpha, \perp) = 1 - \mathbf{P}(s, \alpha, S)$$

Example PTS

Button-pushing experiments on LTS

- Consider a *labelled* transition system
  - "buttons" of system = interface to outside world = observable actions
- Observer tries to depress one *of* several buttons
  - if button is unlocked and goes down $\Rightarrow$ experiment is successful
  - otherwise, experiment fails
- In response to successful experiment, transition system evolves
  - probably nondeterministically
  - and is afterwards ready for further experimentation

$\Rightarrow$ transition system *reacts* to observer's stimuli

Button-pushing experiments on PTS

- Consider a (generative) probabilistic transition system
- Observer may attempt to depress *several* buttons simultaneously
  - the process decides *probabilistically* which (if any) button will go down
- In response to successful experiment, transition system evolves
  - according to the distribution with which it selected the button, but
  - *conditioned* by button choice of observer

Example revisited

What is a probabilistic process algebra?

- It is a theory about *probabilistic processes*
- It is a theory about *concurrent* probabilistic processes
- It is an *algebra*
  - with probabilistic processes and actions as domain
  - with operators to combine processes (and actions)
  - with laws to rewrite processes into equivalent (= bisimilar) ones
- It supports *compositionality* and *abstraction*

how can such algebra be constructed for PTSs?

Preliminaries

- Let *Act* be a countable set of actions with $\tau \notin$ *Act*
  - ranged over by α, β, and so on
  - there is no need here to distinguish names (like a) and co-names (e.g., $\overline{a}$)
- Let *Pid* be a set of process identifiers
  - ranged over by X, Y and so on
- For simplicity, we first do not consider parallel composition

A process algebra for sequential processes

The set Prc_p of probabilistic process expressions is defined by the syntax:

- nil (inaction)
- $\alpha.P$ (prefixing)
- $\sum_{j \in J} [p_j] \, P_j$ (probabilistic choice)
  - where J is a finite index set and probability $p_j \in (0,1)$ with $\sum_{j \in J} p_j = 1$
- $A(\alpha_1, \ldots, \alpha_n)$ (process instantiation)
  - where $A \in \mathit{Pid}$ and $\alpha_i \in \mathit{Act}$ $(0 < i \leqslant n)$

there is no nondeterministic choice!

Recursive process definitions

A (recursive) process definition is an equation system of the form:

$$\{ A_i(\alpha_{i1}, \ldots, \alpha_{in_i}) \ = \ P_i \mid 0 < i \leqslant k \}$$

where $k > 0$, $A_i \in Pid$ (pairwise different), $\alpha_{ij} \in$ *Act*, and $P_i \in Prc_p$

(with process identifiers from $\{ A_1, \ldots, A_k \}$)

Meaning of process algebra constructs

- nil is an inactive process that cannot do anything
- $\alpha.P$ may execute action α and then behaves like P
- $\sum_{j \in J} [p_j] \, P_j$ denotes a probabilistic choice:
  - process P_j is selected with probability p_j
- The behavior of a process call $A(\alpha_1, \ldots, \alpha_n)$ is defined by the right-hand side of the equation $A = P$ where $\alpha_1, \ldots, \alpha_n$ replace the formal parameters

Semantics

- The semantics of term $P \in Prc_p$ is a PTS
  - recall that the semantics of term $P \in Prc$ (non-probabilistic) is an LTS

- The transition probability relation is defined using derivation rules of the form:

$$\frac{\text{premise(s)}}{\text{conclusion}} \ \text{(rule name)}$$

- The initial state of the PTS is the term P

- The state space is the set of derivatives of P

for simplicity, let us first consider the derivation rules while ignoring the probabilities

Non-probabilistic semantics

A process definition $A_i(\alpha_{i1}, \ldots, \alpha_{in_i})$ determines the PTS

$$\big(Prc_p, \textit{Act}, \mathbf{P}, A_i(\alpha_{i1}, \ldots, \alpha_{in_i})\big)$$

whose transitions can be derived by the following rules:

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \text{ (Act)}$$

$$\frac{A(\vec{\alpha}) = P \quad P[\vec{\alpha} \mapsto \vec{\beta}] \xrightarrow{\alpha} P'}{A(\vec{\beta}) \xrightarrow{\alpha} P'} \text{ (Call)}$$

$$\frac{P_k \xrightarrow{\alpha} P' \quad k \in J}{\sum_{j \in J} [p_j] P_j \xrightarrow{\alpha} P'} \text{ (Psum)}$$

let us now consider the transition probabilities

Naive, incorrect semantics

$$\frac{}{\alpha.P \xrightarrow{\alpha,1} P} \text{(Act)}$$

$$\frac{A(\vec{\alpha}) = P \quad P[\vec{\alpha} \mapsto \vec{\beta}] \xrightarrow{\alpha,p} P'}{A(\vec{\beta}) \xrightarrow{\alpha,p} P'} \text{(Call)}$$

$$\frac{P_k \xrightarrow{\alpha,p} P' \quad k \in J}{\sum_{j \in J} [p_j] P_j \xrightarrow{\alpha, p_k \cdot p} P'} \text{(Psum)}$$

Naive, incorrect semantics

$$\frac{}{\alpha.P \xrightarrow{\alpha,1} P} \text{ (Act)}$$

$$\frac{A(\vec{\alpha}) = P \quad P[\vec{\alpha} \mapsto \vec{\beta}] \xrightarrow{\alpha,p} P'}{A(\vec{\beta}) \xrightarrow{\alpha,p} P'} \text{ (Call)}$$

$$\frac{P_k \xrightarrow{\alpha,p} P' \quad k \in J}{\sum_{j \in J} [p_j] P_j \xrightarrow{\alpha,p_k \cdot p} P'} \text{ (Psum)}$$

- process α.nil $\oplus_{\frac{1}{2}}$ α.nil should terminate with probability one
- . . . . . . but the *only* transition that can be derived is: α.nil $\oplus_{\frac{1}{2}}$ α.nil $\xrightarrow{\alpha,\frac{1}{2}}$ nil!

$\Rightarrow$ need to distinguish between different applications of same derivation rule

Solution I: use indexed transitions

An *indexed* PTS is a quintuple $(S, \mathit{Act}, \mathbf{P}, J, s_0)$ where

- $S, s_0 \in S$, and *Act* are as before, and
- J is a set of *indices*, and
- $\mathbf{P} \in S \times \mathit{Act} \times J \times S \to [0,1]$ a transition probability function satisfying:
  1. $s \xrightarrow{\alpha,p}_i s'$ and $s \xrightarrow{\beta,q}_i t' \quad \Rightarrow \quad \alpha = \beta \;\wedge\; p = q \;\wedge\; s' = t'$
  2. $\sum_\alpha \sum_{j \in J} \mathbf{P}(s, \alpha, j, S) \leqslant 1$ for each $s \in S$ and $\alpha \in \mathit{Act}$

notation: $\mathbf{P}(s, \alpha, j, s') = p$ is written as $s \xrightarrow{\alpha,p}_j s'$

Indexed probabilistic semantics

$$\frac{}{\alpha.P \xrightarrow{\alpha,1}_0 P} \text{(Act)}$$

$$\frac{A(\vec{\alpha}) = P \quad P[\vec{\alpha} \mapsto \vec{\beta}] \xrightarrow{\alpha,p}_j P'}{A(\vec{\beta}) \xrightarrow{\alpha,p}_j P'} \text{(Call)}$$

$$\frac{P_k \xrightarrow{\alpha,p}_n P' \quad k \in J}{\sum_{j \in J} [p_j] P_j \xrightarrow{\alpha, p_k \cdot p}_{k.n} P'} \text{(Psum)}$$

Indexed probabilistic semantics

$$\frac{}{\alpha.P \xrightarrow{\alpha,1}_{0} P} \text{ (Act)}$$

$$\frac{A(\vec{\alpha}) = P \quad P[\vec{\alpha} \mapsto \vec{\beta}] \xrightarrow{\alpha,p}_{j} P'}{A(\vec{\beta}) \xrightarrow{\alpha,p}_{j} P'} \text{ (Call)}$$

$$\frac{P_k \xrightarrow{\alpha,p}_{n} P' \quad k \in J}{\sum_{j \in J} [p_j] P_j \xrightarrow{\alpha,p_k \cdot p}_{k.n} P'} \text{ (Psum)}$$

we now obtain $\alpha.\text{nil} \oplus_{\frac{1}{2}} \alpha.\text{nil} \xrightarrow{\alpha,\frac{1}{2}}_{1.0}$ nil and $\alpha.\text{nil} \oplus_{\frac{1}{2}} \alpha.\text{nil} \xrightarrow{\alpha,\frac{1}{2}}_{2.0}$ nil, as desired

Example

What is the (indexed) PTS for the following process terms:

- $X = \alpha.\left(\beta.\mathsf{nil} \oplus_{\frac{1}{3}} \gamma.\mathsf{nil}\right) \oplus_{\frac{3}{4}} \gamma.\mathsf{nil}$
- $X = \alpha.X \oplus_{\frac{1}{4}} Y$ and $Y = \alpha.Y \oplus_{\frac{1}{3}} X$

solution on the black board

Solution II: separate transitions and probabilities

Do not use transition indexes, but let $\rightarrow$ be defined as for CCS:

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \frac{P \xrightarrow{\alpha} P' \quad A = P}{A \xrightarrow{\alpha} P'} \qquad \frac{P_k \xrightarrow{\alpha} P'}{\sum_{j\in J} [p_k] P_k \xrightarrow{\alpha} P'} \ (k \in J)$$

and define $\mathbf{P}$ as *the least solution* satisfying the recursive equations:

$$\begin{aligned} \mathbf{P}(\alpha.P, \alpha, P) &= 1 \\ \mathbf{P}(\textstyle\sum_{j\in J} [p_j]\, P_j, \alpha, P) &= \textstyle\sum_{j\in J}\, p_j \cdot \mathbf{P}(P_j, \alpha, P) \\ \mathbf{P}(A, \alpha, P') &= \mathbf{P}(P, \alpha, P') \quad \text{provided } A = P \text{ and } A \in Pid \end{aligned}$$

Example

Deadlock

Consider the sub-stochastic process:

$$P = \alpha.\mathsf{nil} \oplus_{\frac{1}{3}} \mathsf{nil}$$

- this process can perform action α with probability $\frac{1}{3}$, but
- deadlocks with probability $1 - \frac{1}{3}$

A (fully) stochastic semantics can be easily obtained:

- by adding a deadlock state $\bot \notin S$
- . . . and a special action $0 \notin$ *Act* such that
- for each state s with a “rest” probability $p > 0$,
- we add $s \xrightarrow{0,p} \bot$ (and thus nil $\xrightarrow{0,1} \bot$)

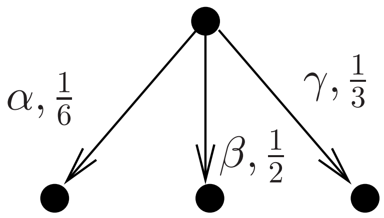
Restriction

- Recall the restriction operator of CCS:
  - new β P declares β as a local name to P
- Formal semantics

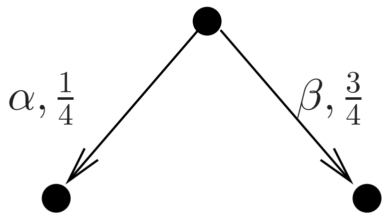
$$\frac{P \xrightarrow{\alpha} P' \quad \alpha \neq \beta}{\text{new } \beta \; P \xrightarrow{\alpha} \text{new } \beta \; P'} \text{ (New)}$$

- What does it mean probabilistically that action β is prohibited?

Restriction: an example



$$P = [\tfrac{1}{6}]\alpha.\mathsf{nil} + [\tfrac{1}{2}]\beta.\mathsf{nil} + [\tfrac{1}{3}]\gamma.\mathsf{nil}$$



$$P\backslash\backslash\{\,\gamma\,\}$$

How can the result of restriction be justified?

Justification

- The probabilities in new $\beta\ P$ are conditioned to not performing β
- These probabilities are normalised
  - the normalisation factor = probability that P does not perform β
- Normalisation can be seen as a repeated experiment:
  - probabilistically select one of the alternative transitions
  - in case a prohibited transition (i.e., β) has been selected, start over
  - continue this process until a possible transition (i.e., non-β) has been selected

Semantics of restriction

For $\beta \in$ *Act*, the derivation rule for restriction new β P is:

$$\frac{P \xrightarrow{\alpha, p}_j P' \quad \alpha \neq \beta}{\text{new } \beta \; P \xrightarrow{\alpha, \frac{p}{\nu(P,\beta)}}_j \text{new } \beta \; P'} \; \text{(New)}$$

where

$$\nu(P, \beta) = 1 - \sum_j \{\!| \, p \mid P \xrightarrow{\beta, p}_j P' \, |\!\}$$

is the probability that P does not perform a β-transition

$\{\!| \ldots |\!\}$ denotes a bag, or a multiset