# PCSP: A Denotational Model of Probabilistic Processes


| F. Cuartero Gómez | D. de Frutos Escrig | V. Valero Ruiz |
|---|---|---|
| Dpto. Informática | D. Informática y Automática | Dpto. Informática |
| E.U. Politécnica | Fac. Matemáticas | E.U. Politécnica |
| U. Castilla-La Mancha | Univ. Complutense | U. Castilla-La Mancha |
| 02071 Albacete, SPAIN | 28040 Madrid, SPAIN | 02071 Albacete, SPAIN |
| fernando@info-ab.uclm.es | defrutos@eucmax.sim.ucm.es | valentin@info-ab.uclm.es |



**Abstract.** We present a model of probabilistic processes based on an asynchronous algebra, starting from the classical CSP; replacing internal non-determinism by generative probabilistic choices, and external non-determinism by reactive probabilistic choices. Our purpose when defining the model has been to maintain, as far as possible, the meaning of all the operators in classical CSP, generalizing their meaning in a probabilistic way. Thus we try to keep valid (once probabilistically generalized), as far as possible, the laws of CSP. It is the combination of both internal and external choice that makes strongly difficult the definition of a probabilistic version of CSP. We can find in the current literature quite a number of papers on probabilistic processes, but only in a few of them internal and external choices are combined, trying to preserve their original meaning. The denotational semantics here presented is based on a domain of probabilistic trees with two kinds of nodes, representing the internal and external choices, the root being an internal one, which generalize in a very natural way the corresponding trees for nonprobabilistic process algebras.



**Topics:** algebraic models, process algebra, formal languages, probabilistic models.


## 1 Introduction

There are mainly two trends when trying to define the semantics of probabilistic processes. The first is the so called *generative*, which distributes the probabilities among all the possible computations; this clearly corresponds to the natural probabilistic generalization of the internal choice of CSP. On the other hand, the *reactive* approach distributes the probabilities among all the computations beginning with the same action; this is, in our opinion, the reasonable way to cope with the external choice of CSP. These models are separately studied in [GSST90].

Some papers studying the semantics of probabilistic processes have appeared in the last years. First of them just studied probabilistic transition systems [LaSk89, Chr90]; others [GJS90, GSST90] focus on probabilistic versions of the

SCCS calculus [Mil83], which is a synchronous version of the more popular CCS [Mil89], and [HaJo90] presents an asynchronous CCS, but maintaining the non-determinism mixed with a random behaviour of the environment. This model is studied in depth by H. Hansson in his Ph.D. Thesis [Han91]. The semantical domain there presented is very similar to the one that we have obtained, which strengthens the interest of our model.

[YiLa92] presents a proposal for a testing semantics of a probabilistic extension of CCS, also including both non-deterministic and probabilistic choices.

More recently, some people in Oxford University have spent a considerable effort on the subject, and two Ph.D. Thesis have been produced. Karen Seidel [Sei92] has developped two different probabilistic models of CSP. In the first one, she gives a semantics in terms of probability measures on the space of infinite traces, but as this first model raises some problems when defining the semantics for the external choice operator, she develops a second semantics, using conditional probability measures. On the other hand, Gavin Lowe [Low93a, Low91, Low93b] has also defined a denotational model covering both internal and external probabilistic behaviour in Timed CSP, but there are several important differences with our approach, both at the intuitive and at the technical level. The most important difference is that he maintains a pure (without any probabilistic information) non-deterministic choice operator.

Recently, a new contribution has appeared [YCDS94], which presents a fully abstract characterization of the testing semantics defined in [CSZ92]. That characterization is based on finding a complete subset of tests, with the same strength that the full set of tests. Finally, in [**?**] is studied a *generative* interpretation of probabilities for a language with two choice operators very close to our, defining a testing semantics for that model, and a fully abstract denotational semantics based on *acceptance trees* following similar ideas to that here presented.

The denotational model we present in this paper is a fully abstract characterization of a testing semantics defined in [CFV92]. In fact, both semantics were developed in parallel, trying to maintain both of them as close as possible to the corresponding semantics for plain CSP. A fully study of the subject is presented in [Cua93], including both aforementioned semantics as well as an equivalent operational semantics, and finally a sound and complete axiomatization.

The paper is structured as follows: In Section 2 we present the syntax of the language. In Section 3 we introduce the mathematical objects representing semantics processes, that is the domain of processes. Semantics of the operators introduced in Section 2 is defined, by structural induction, in Section 4. Afterwards, a simple protocol is presented as example in Section 5. Finally, in Section 6 an outline of the denotational semantics of the hiding operator is presented, as well as an sketch of the operational semantics for our language.

## 2 Syntax of PCSP

**Definition 1.** Given a finite alphabet of actions $\Sigma$, and a set of identifiers $Id$, the set of PCSP processes is defined by the following BNF-expression:

$$P \ ::= \ STOP \mid DIV \ \mid X \mid a \rightarrow P \mid P \sqcap_p P \mid P \,\Box_p\, P \mid P \parallel_A^p P \mid \mu X.P$$

where $p \in [0,1]$, $a \in \Sigma$, $A \subseteq \Sigma$ and $X \in Id$. □

We have omitted the study of the hiding operator, due to its special difficulties, although a discussion on the subject can be found in the final section.

The intuitive meaning of each operator is as follows:

- $STOP$, $a \rightarrow P$ and $\mu X.P$: They are similar to the corresponding CSP operators.
- $DIV$ is a *divergent* process, unable to execute any action, but also unable to stop.
- Internal Choice: $P \sqcap_p Q$ is a process that behaves as $P$ with probability $p$ and as $Q$ with probability $1-p$.
- External Choice: $P \,\Box_p\, Q$ is a process whose behaviour is mainly deterministic. If the environment offers an action which can be executed either by $P$ or by $Q$, but not for both of them, then it will be executed for the corresponding process with the probability with which this process can execute it. If both processes can execute the offered action, then the choice is made randomly according to the parameter $p$ (as in the internal choice). With this intuition behind, there is a reactive interpretation of the external choice.
- Parallel Composition: $P \parallel_A^p Q$ is a process which represents the parallel execution of $P$ and $Q$ syncronizing on the actions in $A$, and the meaning of the parameter $p$ is similar to that in the external choice when an action not belonging to $A$ is offered by the environment.

## 3 Domain of probabilistic processes

In order to obtain a denotational model of PCSP, we have to define the kind of mathematical objects representing semantics processes. For that, let us have a look at the behaviour of a process in our model. Along the execution of a process we find two different stages at each step of its evolution. Firstly, the process reaches a *stable state* after the execution of several internal choices. After that, the environment selects, among all the actions of that state, which is the concrete action to be executed.

Thus, the main idea to define the behaviour of a process will be the description of the reached states along its execution. Naturally, due to the probabilistic characteristics of our model, we need to add, in the appropriate way, the adequate probabilistic information.

We define a probabilistic process by means of a tree with two kinds of alternating nodes, which we call internal and external nodes, the root being an internal one. Arcs leaving internal nodes are labelled with a pair $(A, p)$, where $A$ is a state (i.e. a set of actions) and $p$ is a probability. These arcs reach external nodes, from which as many arcs as actions of $A$ leave, labelled each one with a different action of that state; and reaching again another internal node.

Graphically, a partial representation can be seen in Fig. 1.

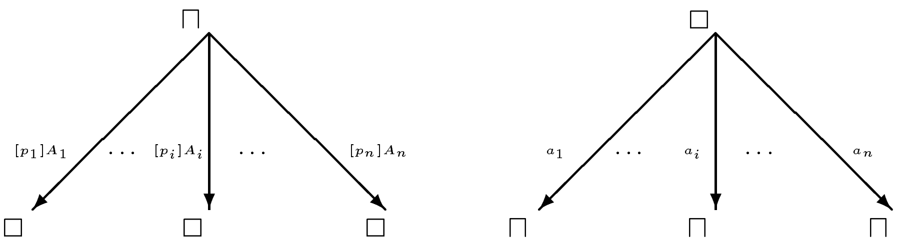


**Fig. 1.** Internal and External nodes of probabilistic processes.

**Definition 2.** (Probabilistic Processes)
We define *semantical probabilistic processes* by the following regular expression:

$$P := \bigsqcap_{A \in \mathcal{A}} [p_A] \mathop{\Box}_{a \in A} a.P$$

where $\mathcal{A} \subseteq \mathcal{P}(\Sigma)$, $\mathcal{A} \neq \emptyset$ and $\forall A \in \mathcal{A} \, : \, p_A \neq 0 \, \wedge \, \sum_{A \subseteq \mathcal{A}} p_A \leq 1$.

We will denote by $\mathcal{P}$ the set of semantical probabilistic processes. □

**Definition 3.** (Probability to reach a state)
Let $P = \bigsqcap_{A \in \mathcal{A}} [p_A] \mathop{\Box}_{a \in A} a.P_{a,A}$ a semantical probabilistic process and $A \subseteq \Sigma$. We define the probability with which $P$ reaches the state $A$, which we denote by $p(P, A)$, by

$$p(P, A) \; = \; \begin{cases} p_A & \text{if } A \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}$$

We will also denote by $P/(a, A)$ the process obtained after executing the action $a$ at the state $A$, i.e. $P_{a,A}$. □

External nodes of a semantical probabilistic process can be characterized by means of a unique sequence of alternating states and actions. We will denote by $SEQ$ the set of these sequences characterizing all possible external nodes:

$$SEQ \; = \; \{ \, \langle \, A_1 \cdot a_1 \dots A_n \, \rangle \mid a_i \in A_i, \, A_i \in \mathcal{P}(\Sigma), \, n \geq 1 \, \}$$

We can associate to each such a sequence the probability with which a corresponding computation will be executed, as follows:

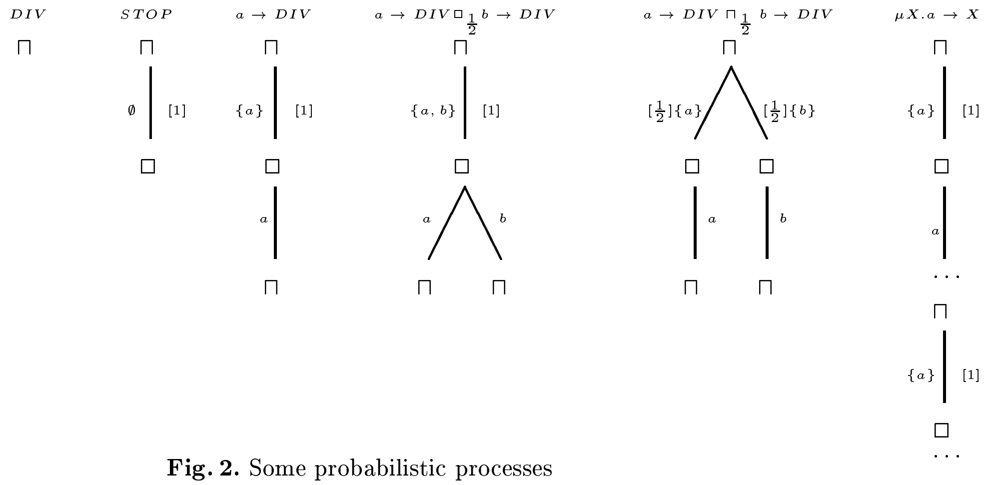


**Fig. 2.** Some probabilistic processes

**Definition 4.** (Probability to reach an external node)
Let $P \in \mathcal{P}$, $s = \langle A_1 \cdot a_1 \dots A_n \rangle \in SEQ$. We define the probability with which $P$ reaches the external node represented by the sequence $s$, denoted by $p(P, s)$, as follows:

$$\begin{aligned} p(P, \langle A_1 \rangle) &= p(P, A_1) \\ p(P, \langle A_1 \cdot a_1 \rangle \cdot s) &= p(P, A_1) \cdot p(P/(a_1, A_1), s) \end{aligned}$$

□

Once we have the objects of our domain of processes, we introduce the partial order between them.

**Definition 5.** (Order between semantical probabilistic processes)
Let $P, Q \in \mathcal{P}$. We say that $P \sqsubseteq Q$ if and only if for any sequence $s \in SEQ$, we have $p(P, s) \leq p(Q, s)$.

□

The following result (whose proof is immediate) allows us to state that with this ordering we can define a denotational semantics using the *fixed point* approach.

**Theorem 6.** $(\mathcal{P}, \sqsubseteq)$ is a complete partial order (cpo).

□

## 4 Operator Semantics

The semantics of the considered operators is defined in a denotational style, by associating a function between semantical probabilistic processes to each operator. As usual, in the sequel $[\![P]\!]$ will denote the semantics of $P$.

### 4.1 Simple operators

- $[\![DIV]\!]$ is defined by a tree with a single node (the root), and no arcs at all. Thus

$$p([\![DIV]\!], s) \;=\; 0 \;\; \forall s \in SEQ$$

- $[\![STOP]\!]$ is defined by a tree with two nodes: one internal and one external. The arc connecting them is labelled by $(\emptyset, 1)$.

$$p([\![STOP]\!],\, s) \;=\; \begin{cases} 1 & \text{if } s = \langle\, \emptyset \,\rangle \\ 0 & \text{otherwise} \end{cases}$$

- $[\![a \to P]\!]$ is defined from $[\![P]\!]$, adding two new nodes to it: one internal (new root) and one external. The arc connecting them is labelled by $(\{a\}, 1)$, and the arc connecting this new external node and the root of $[\![P]\!]$ is labelled by $a$. Thus, we have:

$$p([\![a \to P]\!],\, s) \;=\; \begin{cases} p([\![P]\!], s') & \text{if } s = \langle\, \{a\} \cdot a \,\rangle s' \\ 0 & \text{otherwise} \end{cases}$$

- Branches of $[\![P \sqcap_p Q]\!]$ are obtained from the branches of $[\![P]\!]$ and $[\![Q]\!]$, weighted by $p$ and $1-p$ respectively. When $[\![P]\!]$ and $[\![Q]\!]$ have two branches with identical sequences, only one of them appears in $[\![P \sqcap_p Q]\!]$, taking as probability the weighted addition of the probabilities associated to them. Thus, we have:

$$p([\![P \sqcap_p Q]\!],\, s) \;=\; p \cdot p([\![P]\!], s) + (1-p) \cdot p([\![Q]\!], s)$$

### 4.2 External choice

To obtain the tree $[\![P \Box_p Q]\!]$ firstly we must consider all the possible states of $[\![P]\!]$ and $[\![Q]\!]$ at the first level. Then, the arcs of $[\![P \Box_p Q]\!]$ leaving the root are obtained considering all the union sets of them. For each state $C$ so obtained, its associated probability is calculated adding up the products of the probabilities corresponding to all the pairs $A$, $B$ such that $C = A \cup B$.

Thus, at the first level we have:

$$p([\![P \Box_p Q]\!],\, \langle\, C \,\rangle) \;=\; \sum_{\substack{A,B \\ A \cup B = C}} p([\![P]\!],\, \langle\, A \,\rangle) \cdot p([\![Q]\!],\, \langle\, B \,\rangle)$$

This way, we have solved the internal choices at the first level of $P$ and $Q$, and then we have to face with the external ones under those. For that, let us consider a pair $A$, $B$,such that $C = A \cup B$, and let $a \in C$. Then, either this action just belongs to one of these sets, or it does to both of them. In the first case, the corresponding process executes the action and continues its execution. Thus, in this case the corresponding branches of $[\![P \Box_p Q]\!]$ are obtained from the branches of the process executing the action. On the other hand, when the action belongs to both sets, branches of $[\![P\Box_p Q]\!]$ are obtained by combining those of both processes in a very similar way to that used for the internal choice.

$$p([\![P\Box_p Q]\!], \langle C \cdot a \rangle s) =$$

$$\sum_{\substack{a\in A-B\\ C=A\cup B}} p([\![P]\!], \langle A \cdot a \rangle s) \cdot p([\![Q]\!], \langle B \rangle) + \sum_{\substack{a\in B-A\\ C=A\cup B}} p([\![Q]\!], \langle B \cdot a \rangle s) \cdot p([\![P]\!], \langle A \rangle) +$$

$$\sum_{\substack{a\in A\cap B\\ C=A\cup B}} [p \cdot p([\![P]\!], \langle A \cdot a \rangle s) \cdot p([\![Q]\!], \langle B \rangle) + (1-p) \cdot p([\![Q]\!], \langle B \cdot a \rangle s) \cdot p([\![P]\!], \langle A \rangle)]$$

*Example 1.* Let $P, Q$ be the semantical probabilistic processes represented in Fig. 3, the corresponding process $R = P\Box_{0.2}Q$ is also partially represented in that Figure. Computations of the states and probabilities reached at the first level have been obtained applying the corresponding definitions. For instance, we have

$$\begin{aligned} p([\![R]\!], \{a,b\}) = {} & p([\![P]\!], \{a,b\}) \cdot p([\![Q]\!], \{a\}) + p([\![P]\!], \{a,b\}) \cdot p([\![Q]\!], \{b\}) + \\ & p([\![P]\!], \{a\}) \cdot p([\![Q]\!], \{b\}) = 0.55 \end{aligned}$$

We can also apply the previous definition to obtain the processes $R_i$ in that Figure, althoug calculations are a bit more involved. For instance, we present here the computation of the behaviour of $R_1$. For that, let $s$ be a sequence of $R_1$, then $\langle \{a,b\} \cdot a \rangle s$ is a sequence of $R$; thus we have

$$\begin{aligned} p([\![R]\!], \langle \{a,b\} \cdot a \rangle s) &= p([\![P]\!], \langle \{a,b\} \cdot a \rangle s) \cdot p([\![Q]\!], \{b\}) + \\ & \quad p([\![P]\!], \langle \{a\} \cdot a \rangle s) \cdot p([\![Q]\!], \{b\}) + \\ & \quad 0.2 \cdot p([\![P]\!], \langle \{a,b\} \cdot a \rangle s) \cdot p([\![Q]\!], \{a\}) + \\ & \quad 0.8 \cdot p([\![P]\!], \{a,b\}) \cdot p([\![Q]\!], \langle \{a\} \cdot a \rangle s) \\ &= 0.7 \cdot p([\![P_1]\!], s) \cdot 0.3 + 0.3 \cdot p([\![P_3]\!], s) \cdot 0.3 + \\ & \quad 0.2 \cdot 0.7 \cdot p([\![P_1]\!], s) \cdot 0.5 + 0.8 \cdot 0.7 \cdot 0.5 \cdot p([\![Q_1]\!], s) \end{aligned}$$

$\Box$

### 4.3 Parallel Composition

The semantical probabilistic process $[\![P||^p_A Q]\!]$ is defined in a similar way as the external choice, considering all the possible states of $[\![P]\!], [\![Q]\!]$. However, now we cannot only consider simple unions of states, due to the presence of the synchronization set $A$, and we must remove from each union set $B \cup C$ the actions in $A$ belonging just to one of the sets $B$ or $C$, because these actions should be executed by both processes $P$ and $Q$ simultaneously. Thus, the reached state will be $D = ((B \cup C) - A) \cup (B \cap C)$.

To simplify our notation we will split the states reached by the process $P$ into two components, $B$ and $X_1$ (resp. for $Q$, $C$ and $X_2$), where sets $B$ and $C$ do not hold any actions belonging to both sets $A$ and $B \cap C$. However, all these kind of actions will be taken into account in two sets, $X_1$, $X_2$, where $X_1$ holds the actions belonging to the synchronization set which can be executed only by process $P$, and respectively $X_2$.

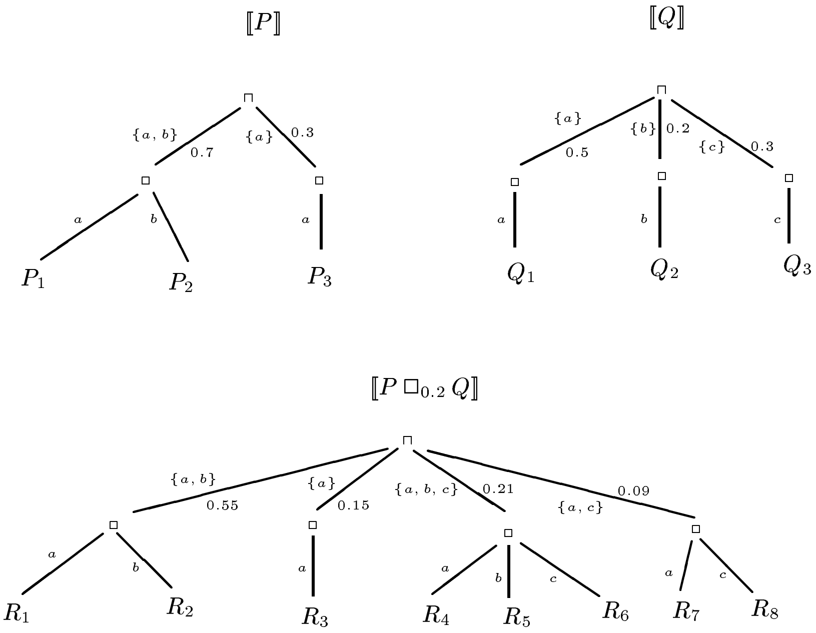


**Fig. 3.** Semantical Probabilistic Process for an External Choice

Then, at the first level we have to consider all the possible states $B \cup X_1$ of $[\![P]\!]$ and $C \cup X_2$ of $[\![Q]\!]$, for $X_1 \cap X_2 = \emptyset$, $X_1, X_2 \subseteq A$, and $B \cap C \subseteq A$.

$$p([\![P\|_A^p Q]\!], \langle\, D \,\rangle) \;=\; \sum_{B \cup C = D} \; \sum_{\substack{X_1, X_2 \subseteq A \\ X_1 \cap X_2 = \emptyset}} p([\![P]\!], \langle\, B \cup X_1 \,\rangle) \cdot p([\![Q]\!], \langle\, C \cup X_2 \,\rangle)$$

Let $\langle\, D \cdot a \,\rangle s \in \mathit{SEQ}$, and $B$, $C$, $X_1$, $X_2$ as previously. We have to distinguish four cases to define $p([\![P\|_A^p Q]\!], \langle\, D \cdot a \,\rangle s)$:

- If $a \in B - C$, then $P$ executes the action $a$, and the parallel process $P/(a, B \cup X_1)\|_A^p Q$ must execute the sequence $s$.
- If $a \in C - B$, then $Q$ executes the action $a$, and the parallel process $P\|_A^p Q/(a, C \cup X_2)$ must execute the sequence $s$.
- When $a \in (B \cap C) - A$, this action is executed randomly either by $P$ or $Q$ with the associated probability. Then, the process $P/(a, B \cup X_1)\|_A^p Q$ must execute the sequence $s$ with probability $p$, and $P\|_A^p Q/(a, C \cup X_2)$ with probability $1 - p$.
- When $a \in A \cap B \cap C$, both processes must cooperate to execute the action, and the process $P/(a, B \cup X_1)\|_A^p Q/(a, C \cup X_2)$ must execute the sequence $s$.

$$p(\llbracket P \|_A^p Q \rrbracket, \langle D \cdot a \rangle s) \;=\;$$
$$\sum_{\substack{B \cup C = D \\ A \cap B = A \cap C}} \; \sum_{\substack{X_1, X_2 \subseteq A \\ X_1 \cap X_2 = \emptyset}} p(\llbracket P \rrbracket, \langle B \cup X_1 \rangle) \;\cdot p(\llbracket Q \rrbracket, \langle C \cup X_2 \rangle) \;\cdot$$
$$\begin{cases} p(\llbracket P/(a, B \cup X_1) \|_A^p Q \rrbracket, s) & \text{If } a \in B - C \\ p(\llbracket P \|_A^p Q/(a, C \cup X_2) \rrbracket, s) & \text{If } a \in C - B \\ p \cdot p(\llbracket P/(a, B \cup X_1) \|_A^p Q \rrbracket, s) + & \\ (1-p) \cdot p(\llbracket P \|_A^p Q/(a, C \cup X_2) \rrbracket, s) & \text{If } a \in B \cap C - A \\ p(\llbracket P/(a, B \cup X_1) \|_A^p Q/(a, C \cup X_2) \rrbracket, s) & \text{If } a \in B \cap C \cap A \end{cases}$$

### 4.4 Recursion

To define the semantics of $\mu X.P(X)$, where $P(X)$ is a process where $X$ appears as free variable $X$, we use the fix point technique.

**Proposition 7.** Semantical definitions of the previous operators are monotonous and continuous.

*Proof.* We only present here the proof corresponding to the external choice, which we consider the most significant case. However, the whole proof can be found in [Cua93].

MONOTONICITY: Let $P, P'$ be processes, such that $\langle P \rangle \sqsubseteq \langle P' \rangle$.

Let $s \in SEQ$. If $s = \langle A \rangle$ the result is immediate. Let us then assume that $s = \langle A \cdot a \rangle s'$. For any $B \subseteq A$ with $a \in B$ we will write by $s_B$ the sequence $\langle B \cdot a \rangle s'$. Then

$$\begin{aligned}
& p(\llbracket P \Box_p Q \rrbracket, s) \\
&= \sum_{\substack{A = B \cup C \\ a \in B - C}} p(\llbracket P \rrbracket, s_B) \cdot p(\llbracket Q \rrbracket, \langle B \rangle) \;+\; \sum_{\substack{A \cup B = C \\ a \in C - B}} p(\llbracket Q \rrbracket, s_C) \cdot p(\llbracket P \rrbracket, \langle C \rangle) \;+ \\
&\quad \sum_{\substack{A = B \cup C \\ a \in B \cap C}} p \cdot p(\llbracket P \rrbracket, s_B) \cdot p(\llbracket Q \rrbracket, \langle B \rangle) \;+ (1-p) \cdot p(\llbracket Q \rrbracket, s_C) \cdot p(\llbracket P \rrbracket, \langle C \rangle) \\
&\leq \sum_{\substack{A = B \cup C \\ a \in B - C}} p(\llbracket P' \rrbracket, s_B) \cdot p(\llbracket Q \rrbracket, \langle B \rangle) \;+\; \sum_{\substack{A \cup B = C \\ a \in C - B}} p(\llbracket Q \rrbracket, s_C) \cdot p(\llbracket P' \rrbracket, \langle C \rangle) \;+ \\
&\quad \sum_{\substack{A = B \cup C \\ a \in B \cap C}} p \cdot p(\llbracket P' \rrbracket, s_B) \cdot p(\llbracket Q \rrbracket, \langle B \rangle) \;+\; (1-p) \cdot p(\llbracket Q \rrbracket, s_C) \cdot p(\llbracket P' \rrbracket, \langle C \rangle) \\
&= p(\llbracket P' \Box_p Q \rrbracket, s)
\end{aligned}$$

CONTINUITY: Let $\{P_i\}_{i \in I}$ be a chain of processes, and $Q$ another process, then $\{P_i \Box_p Q\}_{i \in I}$ is a chain, and the probability to execute a sequence $s$ verifies:

$$p(\sqcup \llbracket P_i \Box_p Q \rrbracket, s) \;=\; \lim_i p(\llbracket P_i \Box_p Q \rrbracket, s)$$

$$= \lim_i \sum_{\substack{A=B\cup C\\ a\in B-C}} p([\![P_i]\!], s_B)\cdot p([\![Q]\!], \langle\, B\,\rangle) \;+\; \sum_{\substack{A=B\cup C\\ a\in C-B}} p([\![Q]\!], s_C)\cdot p([\![P_i]\!], \langle\, C\,\rangle) \;+$$

$$\sum_{\substack{A=B\cup C\\ a\in B\cap C}} p\cdot p([\![P_i]\!], s_B)\cdot p([\![Q]\!], \langle\, B\,\rangle) \;+\; (1-p)\cdot p([\![Q]\!], s_C)\cdot p([\![P_i]\!], \langle\, C\,\rangle)$$

$$= \sum_{\substack{A=B\cup C\\ a\in B-C}} \lim_i p([\![P_i]\!], s_B)\cdot p([\![Q]\!], \langle\, B\,\rangle) \;+\; \sum_{\substack{A=B\cup C\\ a\in C-B}} p([\![Q]\!], s_C)\cdot \lim_i p([\![P_i]\!], \langle\, C\,\rangle) \;+$$

$$\sum_{\substack{A\cup B=C\\ a\in B\cap C}} p\cdot \lim_i p([\![P_i]\!], s_B)\cdot p([\![Q]\!], \langle\, B\,\rangle) \;+\; (1-p)\cdot p([\![Q]\!], s_C)\cdot \lim_i p([\![P_i]\!], \langle\, C\,\rangle)$$

$$= \sum_{\substack{A=B\cup C\\ a\in B-C}} p(\sqcup[\![P_i]\!], s_B)\cdot p([\![Q]\!], \langle\, B\,\rangle) \;+\; \sum_{\substack{A\cup B=C\\ a\in C-B}} p([\![Q]\!], s_C)\cdot p(\sqcup[\![P_i]\!], \langle\, C\,\rangle) \;+$$

$$\sum_{\substack{A=B\cup C\\ a\in B\cap C}} p\cdot p(\sqcup[\![P_i]\!], s_B)\cdot p([\![Q]\!], \langle\, B\,\rangle) \;+\; (1-p)\cdot p([\![Q]\!], s_C)\cdot p(\sqcup[\![P_i]\!], \langle\, C\,\rangle)$$

$$= p(\sqcup[\![P_i]\!]\Box_p[\![Q]\!], s)$$

$\square$

As a consequence of theorem 6 and proposition 7 we may define the recursion operator semantics as the least fix point of the corresponding functional, which coincides with the limit of the finite approximations:

$$[\![\mu X.P(X)]\!] \;=\; \sqcup[\![P^n]\!]$$

where $P^0 = DIV,\ \ldots\ ,\ P^n = P(P^{n-1}),\ \ldots\ .$

## 5 Example

We present a modeling of the AUY-protocol for data transmission. This protocol ensures a reliable transmission in a system in which the channels can fail.

The system consists of two agents, a transmitter and a receiver, which communicate messages. The transmitter sends a message to the receiver using an unreliable channel, which can lose it with some known probability $p$. We suppose that if the original message is lost, the channel will transmit a null message $\lambda$ to the receiver. The receiver takes a message from its input channel, and if it is not $\lambda$, it sends the message to the outside world, and an acknowledgement message to the transmitter using a second channel. Again, this channel can lose the message (we suppose that with the same probability $p$ although this is not necessary at all), in which case it sends a null message $\lambda'$ to the transmitter. If the transmitter receives through this second channel a $\lambda'$, it will repeat the emission, sending again the original message. Finally, when it receives the acknowledgement it accepts a new incoming message, and a new cycle begins.

The PCSP process modeling this protocol is just the $P$ process defined as follows:

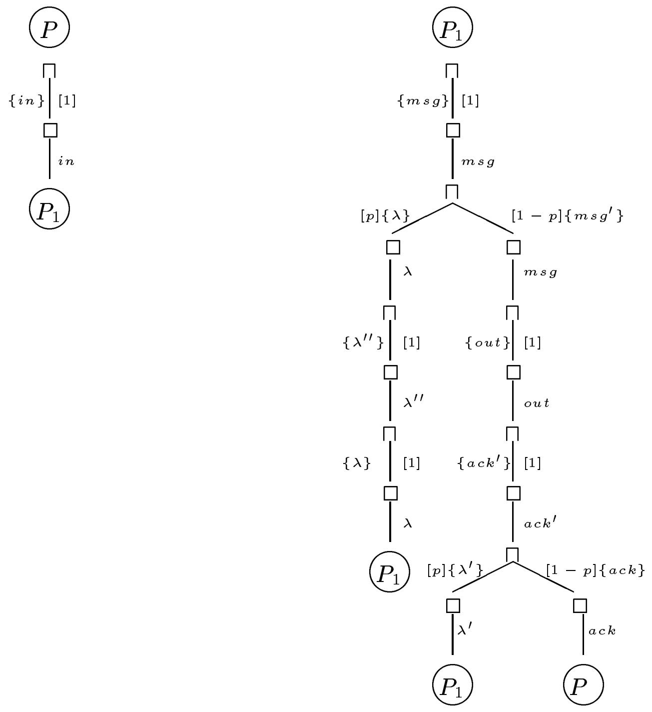


**Fig. 4.** Semantics of AUY-protocol

Transmitter:

$T = in \rightarrow T_1$

$T_1 = msg \rightarrow (ack \rightarrow T \Box \lambda' \rightarrow T_1)$

First channel:

$C_1 = msg \rightarrow ([p]\lambda \rightarrow C_1 \sqcap [1-p]msg' \rightarrow C_1)$

Second channel:

$C_2 = (ack' \rightarrow ([p]\lambda' \rightarrow C_2 \sqcap [1-p]ack \rightarrow C_2)) \Box (\lambda'' \rightarrow \lambda' \rightarrow C_2)$

Receiver:

$R = (msg' \rightarrow out \rightarrow ack' \rightarrow R) \Box (\lambda \rightarrow \lambda'' \rightarrow R)$

AUY-protocol:

$$P = T \|_{\{msg,ack,\lambda'\}} ((C_1 \|_{\emptyset} C_2) \|_{\{msg',ack',\lambda,\lambda''\}} R)$$

We have omitted the probabilities in the appearances of the external choice and parallel operators all along the example, since in fact they are not necessary in this particular case, because the first actions of the components of those operators are different, as we have proved in [Cua93]).

The application of the definitions to obtain the corresponding semantic tree for this process is rather boring, so we only present here the final result (Fig. 4). From this tree we can see that this system is essentially deterministic. There are only two internal nodes representing real random choices, and in all the external nodes there is only one action to choice among. Both internal nodes represent the possibility of losing information (messages or acknowledgements).

## 6 Discussion

In this paper we have presented a denotational semantics for a probabilistic version of CSP, whose mathematical objects are semantical trees with two kinds of nodes, representing the internal and external choices, the root being an internal one. Arcs of these trees are differently labelled, depending on the kind of the starting node: if this is an internal one, the label is a set and a probability; while for the external nodes arcs are only labelled with an action. Thus, only the arcs leaving internal nodes have associated a probabilistic information, which are points where the system makes probabilistic (internal) decisions. Then, the external nodes represent the deterministic participation of the environment.

We consider that the main contribution of the paper is the definition of a very simple denotational semantics, which is a first step to construct a full model, considering some additional operators.

In particular, a natural extension of our language would consist on the introduction of a hiding operator. We have studied this concrete extension, concluding that this hiding operator requires some probabilistic information if we want to maintain the classical interpretation of this operator. A sketch of the meaning of our hiding operator is briefly presented below.

Being $P$ a process, $a$ an action, and $q$ a probability, we have defined $P\backslash(a, q)$ as a process behaving as $P$, but hiding the executions of $a$. The main difference with the CSP hiding operator is the introduction of the new probabilistic argument $q$, justified next, by means of an example.

Consider the following CSP process: $a \to P \Box b \to Q$. For it, the environment has a complete control on its behaviour choosing the action to be executed. However, if we hide the action $a$ we have the following semantical equality:

$$(a \to P \Box b \to Q)\backslash a \equiv P\backslash a \sqcap (P \Box b \to Q)\backslash a$$

In our model, we were looking for a probabilistic interpretation of this behaviour, but we faced with the problem of associating a probability to the new internal choice that appears with this interpretation. A first idea could be to consider the same probability associated to the external choice, but this is not possible, because we have proved that probabilities are meaningless in the context of an external choice among processes prefixed by different actions, i.e. $a \to P \Box_p b \to Q \equiv a \to P \Box_q b \to Q$.

Then, it is necessary to introduce a new probability to this new internal choice, taken from the syntax of our hiding operator. With respect to the external choice of the right hand side, we cannot use the probability associated to the external choice of the left hand side, because this probability is meaningless in some cases; nevertheless it is meaningfull in the right hand side. We have chosen 0 as this probability, based on the classical interpretation. Thus, $q$ represents the probability with which the system decides to execute the hidden action $a$, and after that, if the system does not decide to execute it, the environment can choose the next action to be executed. Let us suppose it is $b$, then $b \to Q$ should have priority over $P$ if both processes may execute it.

Thus, with our model we have the following equation, instead of the corresponding to the classical case:

$$(a \to P\Box_p b \to Q)\backslash(a,q) \equiv P\backslash(a,q) \sqcap_q (P\Box_0 b \to Q)\backslash(a,q)$$

Unfortunately, with this interpretation we have found some troubles in order to define the semantics, because now we have a recursive definition, which implies we have to guarantee the soundness of that recursion. This introduces some technical problems, solved in [Cua93].

In addition to our denotational model, we have also developed some additional equivalent semantics. In particular, we have a testing semantics, an operational semantics, and a sound and complete system proof (see [CFV92, Cua93]).

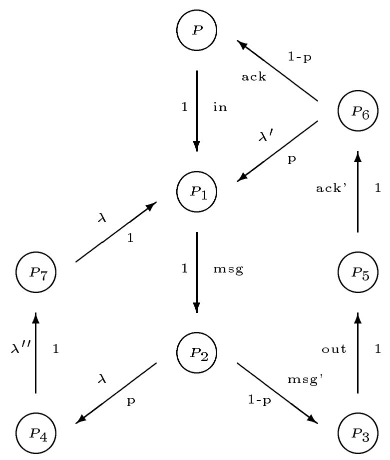


**Fig. 5.** Probabilistic Transition System for AUY protocol

The operational semantics is quite easy to understand, once we have presented the denotational semantics, because the transition system is based on the semantical trees of our denotational domain, but a transition summarizes the evolution along two arcs in the corresponding tree defining the process. Thus, a transition has the following form:

$$P \xrightarrow{A,a}_p Q$$

which means that the process $P$ reaches the state $A$, and after executing the action $a$ behaves as process $Q$ with probability $p$. In our presented denotational model, this is equivalent to say that for every sequence $s$, we have $p(P, \langle A{\cdot}a \rangle s) = p \cdot p(Q, s)$ .

For example, the behaviour of the process defining the AUY protocol is given by the probabilistic transition system presented in figure 5. To simplify the figure, since all the states have only one action, each transition $P \xrightarrow{a,\{a\}}_p Q$ is drawn as $P \xrightarrow{a}_p Q$.

**Acknowledgements:** This paper has been partially suported by CICYT projects TIC-94-0851-c02-02 and PB-092-511.

## References


[Chr89] I. Christoff. *Distinguishing probabilistic processes through testing.* Proceedings Nordic Workshop on Program Correctness, 1989.

[Chr90] I. Christoff. *Testing equivalences and fully abstract models for probabilistic processes*, CONCUR-90, LNCS 458, 1990.

[CSZ92] R. Cleaveland, S.A. Smolka and A. Zwarico. *Testing preorders for probabilistic processes.* ICALP-92, LNCS 624, 1992.

[CFV92] F. Cuartero, D. de Frutos and V. Valero. *CSP probabilístico (PCSP). Un modelo operacional*, Proc. PANEL-92, Las Palmas de Gran Canaria (Spain), 1992.

[CFV93] F. Cuartero, D. de Frutos and V. Valero. *CSP probabilístico (PCSP)*, Proc. III Jornadas de Concurrencia, Gandía (Spain), 1993.

[Cua93] F. Cuartero. *CSP probabilístico (PCSP). Un Modelo Probabilístico de Procesos Concurrentes*, Ph.D. Thesis, Univ. Complutense Madrid (Spain), 1993.

[GJS90] A. Giacalone, C.-C. Jou and S.A. Smolka. *Algebraic reasoning for probabilistic concurrent systems.* Proc. IFIP TC 2, Sea of Galilee, Israel, 1990.

[GSST90] R. van Glabbek, S.A. Smolka, B.U. Steffen and C.M.N. Tofts. *Reactive, generative and stratified models of probabilistic processes.* Proceedings 5th Annual IEEE Symposium on Logic in Computer Science, 130-141, Philadelphia, PA, 1990.

[Han91] H.A. Hansson. *Time and Probability in Formal Design of Distributed Systems.* Ph.D. Thesis. Dep. of Computer Systems, Uppsala University, 1991.

[HaJo90] H. Hansson and B. Jonsson. *A calculus for communicating systems with time and probabilities.* Proc. Real-Time Systems Symposium. Florida, 1990.

[Hen88] M. Hennessy. *Algebraic Theory of Processes.* MIT Press, Cambridge Mass., 1988.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, Englewood Cliffs, NJ, 1985.

[JoSm90] C.-C. Jou and S.A. Smolka. *Equivalences, congruences and complete axiomatizations for probabilistic processes.* CONCUR-90, LNCS 458, 1990.

[LaSk89] K.G. Larsen, A. Skou. *Bisimulation through probabilistic testing.* Proceedings 16th Annual ACM Symposium on Principles of Programming Languages, 1989.

[Low91] Gavin Lowe. *Prioritized and Probabilistic Models of Timed CSP.* PRG-TR-24-91 Report. Oxford University Computing Laboratory, 1991.

[Low93a] Gavin Lowe. *Probabilities and Priorities in Timed CSP.* Ph. D. Thesis. Oxford, 1993.

[Low93b] Gavin Lowe. *Representing Nondeterminism and Probabilistic Behaviour in Reactive Processes*. PRG-TR-11-93 Report. Oxford University Computing Laboratory, 1993.

[Mil83] R. Milner. *Calculi for synchrony and asynchrony*. Theoretical Computer Science, 25:267-310, 1983.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall International, 1985.

[Sei92] K. Seidel. *Probabilistic CSP*, Ph.D. Thesis, Oxford University, 1992.

[SmSt90] S.A. Smolka and B.U. Steffen. *Priority as extremal probability*, CONCUR-90, LNCS 458, 1990.

[YiLa92] W. Yi and K. Larsen. *Testing probabilistic and nondeterministic processes*. Proc. XII Protocol Specification, Testing and Verification, North Holland 1992.

[YCDS94] S. Yuen, R. Cleaveland, Z. Dayar, and S.A. Smolka. *Fully Abstract Characterizations of Testing Preorders for Probabilistic Processes*, CONCUR-94, LNCS 836, 1994.

[Zic90] J.J. Zic. *Some thoughts on communication system performance specification*. Proc. Open Distributed Processing Workshop, Sydney 1990.

This article was processed using the LaTeX macro package with LLNCS style