A Sound and Complete Proof System for Probabilistic Processes

F. Cuartero Gómez
Dpto. Informática
E.U. Politécnica
U. Castilla-La Mancha
02071 Albacete, SPAIN
fernando@info-ab.uclm.es

D. de Frutos Escrig
D. Informática y Automática
Fac. Matemáticas
Univ. Complutense
28040 Madrid, SPAIN
defrutos@eucmax.sim.ucm.es

V. Valero Ruiz
Dpto. Informática
E.U. Politécnica
U. Castilla-La Mancha
02071 Albacete, SPAIN
valentin@info-ab.uclm.es

Abstract. In this paper we present a process algebra model of probabilistic communicating processes based on classical CSP. To define our model we have replaced internal non-determinism by generative probabilistic choices, and external non-determinism by reactive probabilistic choices, with the purpose of maintaining the meaning of the classical CSP operators, once generalized in a probabilistic way. Thus we try to keep valid, as far as possible, the laws of CSP. This combination of both internal and external choice makes strongly difficult the definition of a probabilistic version of CSP. In fact, 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.

Starting with a denotational semantics where the corresponding domain is a set of probabilistic trees with two kinds of nodes, representing the internal and external choices, we define a sound and complete proof system, with very similar laws to those of the corresponding CSP.

1 Introduction

During the last years there has been a great activity devoted to the study of time and probabilistic extensions of concurrent processes. These extensions are very adequate for the specification of *real* systems which strongly depend on stochastic behaviors or on time constraints, and have been proved useful for the specification of communication protocols, real-time systems, and fault-tolerant systems. Next we summarize some works on probabilistic processes which are related in some way with this paper.

Some of them just studied probabilistic transition systems [2]; others [7, 8] focus on probabilistic versions of the SCCS calculus, which is a synchronous version of the more popular CCS. Some others have concentrated on the study of probabilistic versions of asynchronous process algebras.

Among them, Hansson and Jonsson [10] present an asynchronous CCS maintaining the non-determinism mixed with a random behaviour of the environment, which H. Hansson studies in depth in his Ph.D. Thesis [9], where an operational semantics and a computation tree logic are presented. The semantical domain there presented is very similar to the one that we have obtained, which strengthens the interest of our model. Also, in [15] it is presented a proposal for a testing semantics of a probabilistic extension of CCS, which also includes both non-deterministic and probabilistic choices.

Some people in Oxford University have spent a considerable effort on the subject. Karen Seidel [14] has developed 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 has some problems when defining the semantics for the external choice operator, she develops a second semantics, using conditional probability measures, but in this case it is not possible to define the hiding operator in a satisfactory way. On the other hand, Gavin Lowe [12] 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.

Cleaveland et al [3] have studied a testing semantics for probabilistic processes whose starting point is the transition system defining the operational semantics of both processes and tests, however, they do not consider any concrete syntax for processes. Also, [16] has published a *fully abstract* characterization of a testing semantics. This semantics is based on finding a *complete* subset of tests, with the same strength that the full set of tests.

These last papers are closely related with the work we have developed, but the elements of the corresponding domain of processes there used are extensionally defined, and in our opinion, a more denotational characterization would be interesting.

Essentially, these works mainly follow two trends when they 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 [8]. On the other hand, [13] study probabilistic processes in a very close way to ours, considering a more *generative* interpretation of probabilities for the language, with two choice operators and defining a testing semantics, giving for it an alternative characterization, based on the idea of *acceptance sets*, and proving that this equivalence is the same as the testing equivalence. Finally, it is presented a fully abstract denotational semantics based on acceptance trees.

The model we study is a probabilistic version of CSP, which intends to generalize the classical operators of CSP, keeping as far as possible their classical meaning. Concretely, we have probabilistic choice operators, where the behaviour of the internal choice is now completely probabilistic. Thus, we do not have a *pure* internal choice. This could be useful in some cases, but we have preferred to start our study with the full probabilistic version of the language, and afterwards, we will extend our language with new operators, including the classical ones, and some new others, related with real-time constraints and so on.

The denotational model we have used has been presented in [5], although a summary is shown in section 3, and it is a fully abstract characterization of a testing semantics defined in [4]. 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 [6], including both aforementioned semantics as well as an equivalent operational semantics, and finally the sound and complete axiomatization here presented. In our opinion, the fact that we have been able to define this set of semantics, whose equivalence has been proved, is a good hint for the value of the proposed model.

The paper is structured as follows. Section 2 presents our language, in Section 3 we briefly describe the denotational semantics for it (a more complete study may be found in [5]). In Section 4 the full set of axioms and inference rules is presented. Section 5 shows the soundness and completeness of this proof system, and finally in Section 6 we give our conclusions and some outlines for future work.

2 Syntax of PCSP

Definition 1. Given a finite alphabet of actions Σ, 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 $X \in Id$. □

As usual STOP represents deadlock, $a \rightarrow P$ a prefix process, and $\mu X.P$ the recursion operator. DIV is a *divergent* process, unable to execute any action, but also unable to stop. $P \sqcap_p Q$ is a process that behaves as P with probability p and as Q with probability $1-p$. $P \Box_p Q$ is a process that behaves as $P \sqcap_p Q$ when both processes may execute a given action selected by the environment, but if only one of them may execute it, $P \Box_p Q$ will behave as this process, and the probability is meaningless in this case. With this intuition behind, there is a reactive interpretation of the external choice. Finally, let us observe that the parallel operator is also quantified with a probability, because our non-deterministic operators are both probabilistically quantified. Thus, $P \parallel_A^p Q$ is a process representing the parallel execution of P and Q synchronizing on the actions in A, the parameter p is again used when a given action, no belonging to A, may be executed by both processes P and Q.

3 Denotational Semantics

In order to obtain a denotational model of PCSP, the kind of mathematical objects representing the semantics of processes must be defined. For that, let us have a look at the behaviour of a process.

Along the execution of a process, we may see two different stages at each step of its evolution: firstly, the process reaches a *stable state* after the resolution of several internal choices, but without executing any action at all. After that, one action belonging to this state is executed. Thus, the main idea to define the behaviour of a process is the description of the reached states along its execution.

Then, the denotational semantics here presented is based on a domain whose mathematical objects are semantical trees with two kinds of nodes. These nodes represent the internal and external choices, and the root must be 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 pair consisting of a set and a probability; while for the external nodes arcs are 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. These kind of objetcs generalize in a very natural way the corresponding trees for nonprobabilistic process algebras.

3.1 Domain of probabilistic processes

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, each one labelled with a different action of that state (these arcs reach again internal nodes).

Definition 2. (Probabilistic Processes)
We define the *semantical probabilistic processes* by the following 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 $\forall A \in \mathcal{A} \ : \ p_A \neq 0 \wedge \sum\limits_{A\subseteq\mathcal{A}} p_A \leq 1$.

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

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

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

□

We will also denote by $P/(a,A)$ the semantical process obtained after executing the action a at the state 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:

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:

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

□

Next, we introduce the partial order relating semantical probabilistic processes.

Definition 5. (Order relation)
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). □

3.2 Semantics of the operators

We define the semantics of PCSP operators in a denotational style, by associating a semantical probabilistic process $[\![P]\!]$ (i. e. a tree) to each process P.

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

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

$[\![STOP]\!]$ is a tree with two nodes: one internal, the root, and the other 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 (a new root) and one external. The arc connecting them is labelled by $(\{a\}, 1)$, and the arc connecting this new external node with 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:

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

Finally, to obtain the tree $[\![P \,\square_p\, Q]\!]$ firstly we must consider all the possible states of $[\![P]\!]$ and $[\![Q]\!]$ at the first level. Then, the arcs of $[\![P \,\square_p\, Q]\!]$ 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:

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

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 $[\![P \square_p Q]\!]$ are obtained from the branches of the process executing the action. On the other hand, when the action belongs to both sets, branches of $[\![P \,\square_p\, Q]\!]$ are obtained by combining those of both processes in a very similar way to that used for the internal choice.

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

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

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

For the parallel operator $[\![P \|_A^p Q]\!]$ we only present here a short sketch (a full definition may be found in [5]). The idea in this case is very similar to the external choice operator, considering all the possible states of $[\![P]\!]$, $[\![Q]\!]$. However, in this case, 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 only to one of the sets B or C, because these actions should be executed by both processes P and Q simultaneously. Thus, every reached state will be $D = ((B \cup C) - A) \cup (B \cap C)$.

4 Proof System

We present in this section a proof system, which is sound and complete with respect to the denotational semantics.

The first set of axioms, related with the operators STOP, prefix, external and internal choices, is presented in table 1.

A0] $P \equiv P$

A1] $P \sqcap_p P \equiv P$

A2] $P \sqcap_p Q \equiv Q \sqcap_{1-p} P$

A3] $P \sqcap_p (Q \sqcap_{\frac{q}{q+r}} R) \equiv (P \sqcap_{\frac{p}{p+q}} Q) \sqcap_{p+q} R \qquad (p, q, r > 0)$

A4] $P \Box_p Q \equiv Q \Box_{1-p} P$

A5] $P \Box_p STOP \equiv P$

A6] $P \Box_p (Q \sqcap_q R) \equiv (P \Box_p Q) \sqcap_q (P \Box_p R)$

Table 1. First set of axioms for probabilistic processes

We can see that the internal choice operator is idempotent and commutative (A1, A2), and associative in a probabilistic sense (A3), whenever there are no null probabilities, so it can be generalized to an arbitrary number of arguments. Thus, we can write $\overset{n}{\underset{i=1}{\sqcap}}[p_i]P_i$ to denote an internal choice among n processes, each one with probability $p_i > 0$.

The external choice has a zero (A5), the STOP process, it is commutative(A4) and distributes over the internal choice (A6); however it is not associative, so we cannot generalize in the same way this operator. This problem could make it very difficult the task of finding *normal forms* of processes, if it would not be the case that associativity is maintained when the sets of actions offered by the composed processes are disjoint. To formalize this idea we introduce a generalization of the external choice, with an arbitrary number of *prefixed* arguments.

Definition 7. Let $\{a_i\}_{i=1,\ldots,n} \subseteq \Sigma$ be a set of actions, and let $\{P_i\}_{i=1,\ldots,n}$ be a set of processes; then, we define the process $\overset{n}{\underset{i=1}{\Box}} a_i \to P_i$, as follows

A7] $\overset{1}{\underset{i=1}{\Box}} a_i \to P_i = a_1 \to P_1$

A8] $\overset{n}{\underset{i=1}{\Box}} a_i \to P_i = a_1 \to P_1 \Box_{\frac{1}{n}} (\overset{n-1}{\underset{i=1}{\Box}} a_{i+1} \to P_{i+1})$ When $n > 1$. $\Box$

Obviously, this operator is associative, since the actions prefixing the external choice are all of them different, and in that case the probabilities are useless.

Sometimes, we will denote this operator using a *set notation* by $\underset{a \in A}{\Box} a \to P_a$, and using this operator we may introduce the following axioms:

A9] $\quad (\mathop{\Box}_{a\in A} a \to P_a) \sqcap_p (\mathop{\Box}_{a\in A} a \to Q_a) \;\equiv\; \mathop{\Box}_{a\in A} a \to (P_a \sqcap_p Q_a)$

A10] $\quad \mathop{\Box}_{a\in A} a \to P_a \Box_p \mathop{\Box}_{b\in B} b \to Q_b \;\equiv\; \mathop{\Box}_{c\in A\cup B} c \to \begin{cases} P_c & \text{If } c \in A - B \\ Q_c & \text{If } c \in B - A \\ P_c \sqcap_p Q_c & \text{If } c \in A \cap B \end{cases}$

A9 establishes the distributivity of the generalized external choice over the internal one, while A10 relates both external choice operators, showing that the combination of both operators is coherent.

The following set of *expansion* axioms define the behaviour of the parallel operator, which is *derived* from the previous ones:

P1] $\quad P\|_A^p(Q \sqcap_q R) \;\equiv\; (P\|_A^p Q) \sqcap_q (P\|_A^p R)$

P2] $\quad \mathop{\Box}_{b\in B} b \to P_b \|_A^p STOP \;\equiv\; \mathop{\Box}_{b\in B-A} b \to P_b$

P3] $\quad P\|_A^p Q \;\equiv\; Q\|_A^p P$

P4] $\quad (\mathop{\Box}_{b\in B} b \to P_b \|_A^p \mathop{\Box}_{c\in C} c \to Q_c) \;\equiv$

$$\mathop{\Box}_{a\in D} a \to \begin{cases} P_a\|_A^p Q_a & \text{If } a \in A\cap B\cap C \\ P_a\|_A^p \mathop{\Box}_{c\in C} c \to Q_c & \text{If } a \in B - C - A \\ \mathop{\Box}_{b\in B} b \to P_b\|_A^p Q_c & \text{If } a \in C - B - A \\ (P_a\|_A^p \mathop{\Box}_{c\in C} c \to Q_c)\sqcap_p & \\ (\mathop{\Box}_{b\in B} b \to P_b\|_A^p Q_c) & \text{If } a \in (C\cap B) - A \end{cases}$$

where $D = ((B\cup C) - A) \cup (A\cap B\cap C)$

The set of inference rules for the recursion is presented below. As usual it is based on finite approximations, using an order relation also defined the rules (O1, O2, O3 establish that the relation $\sqsubseteq$ is an order and C1, C2, C3 that it is a congruence). Finite approximations begin with the process DIV (whose behaviour is defined by axioms D1 to D5). Rules R1 and R2 are the classical ones for the recursion operator, while R3 has been introduced due to a technical reason: real numbers with its natural order do not constitute an ω-algebraic domain (a more detailed explanation will be given in next section, theorem 11).

O1] $\dfrac{P \sqsubseteq Q \sqsubseteq P}{P \equiv Q}$ **O2]** $\dfrac{P \equiv Q}{P \sqsubseteq Q \sqsubseteq P}$ **O3]** $\dfrac{P \sqsubseteq Q \sqsubseteq R}{P \sqsubseteq R}$

C1] $\dfrac{P \sqsubseteq Q}{a \to P \;\sqsubseteq\; a \to Q}$ **C2]** $\dfrac{P \sqsubseteq Q \;\wedge\; P' \sqsubseteq Q'}{(P\Box P') \;\sqsubseteq\; (Q\Box Q')}$

C3] $\dfrac{P \sqsubseteq Q \;\wedge\; P' \sqsubseteq Q'}{(P \sqcap_p P') \;\sqsubseteq\; (Q \sqcap_p Q')}$

D1] $DIV \sqsubseteq P$

D2] $P \sqcap_p DIV \sqsubseteq P$

D3] $P \Box_p DIV \equiv DIV$

D4] $P \|_A^p DIV \equiv DIV$

D5] $DIV \backslash (a, q) \equiv DIV$

R1] $$\frac{}{P[\mu\xi.P|\xi] \sqsubseteq \mu\xi.P}$$

R2] $$\frac{\forall Q \in APX(P) \,:\, Q \sqsubseteq R}{P \sqsubseteq R}$$

R3] $$\frac{\forall n \in \mathbb{N} \,:\, P \sqcap_{\frac{n-1}{n}} DIV \sqsubseteq R}{P \sqsubseteq R}$$

5 Soundness and Completeness

We present in this section some technical results, which prove that the proof system here introduced is sound and complete with respect to the denotational semantics. As usual, completeness is proved by means of a set of normal forms, which are very similar to those of plain CSP.

Theorem 8. Axioms and Rules are sound with respect to the denotational semantics.

Proof. For A0 and A1, the proof is trivial. For the remaining, let us consider a sequence $s \in SEQ$. Then we have for A2

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

The proof of the soundness of axioms A3 and A4 is very similar, so we omit it. With respect to axiom A5, we have

$$\begin{aligned} p([\![P \Box_p STOP]\!], s) &= \sum_{A \subseteq \Sigma} \sum_{a \in A} p([\![P]\!], A) \cdot p(STOP, \emptyset) \cdot p(P/(a, A), \langle A, a \rangle s') \\ &= \sum_{A \subseteq \Sigma} \sum_{a \in A} p([\![P]\!], A) \cdot p(P/(a, A), \langle A, a \rangle s') \;=\; p([\![P]\!], s) \end{aligned}$$

We finish with the soundness of axiom A6, taking $S = Q \sqcap_q R$.

$$
\begin{aligned}
& p(P \Box_p S, s) = \\
& \sum_{A,B} \Big[\sum_{a \in A-B} p(P, s_A) \cdot p(S,B) \; + \sum_{a \in B-A} p(S, s_B) \cdot p(P,A) \; + \\
& \qquad \sum_{a \in A \cap B} (p \cdot p(P, s_A) \cdot p(S,B) + (1-p) \cdot p(S, s_B) \cdot p(P,A)) \Big] \;\; = \\
& \sum_{A,B} \sum_{a \in A-B} p(P, s_A) \cdot (q \cdot p(Q,B) + (1-q) \cdot p(R,B)) \; + \\
& \qquad \sum_{a \in B-A} (q \cdot p(Q, s_B) + (1-q) \cdot p(R, s_B)) \cdot p(P,A) \; + \\
& \qquad \sum_{a \in A \cap B} (p \cdot p(P, s_A) \cdot (q \cdot p(Q,B) + (1-q) \cdot p(R,B)) + \\
& \qquad\qquad (1-p) \cdot (q \cdot p(Q, s_B) \cdot p(P,A) + (1-q) \cdot p(R, s_B) \cdot p(P,A)) \;\; = \\
& \sum_{A,B} \Big[\sum_{a \in A-B} p(P, s_A) \cdot q \cdot p(Q,B) \; + \sum_{a \in B-A} q \cdot p(Q, s_B) \cdot p(P,A) \; + \\
& \qquad \sum_{a \in A \cap B} (p \cdot p(P, s_A) \cdot q \cdot p(Q,B) + (1-p) \cdot q \cdot p(Q, s_B) \cdot p(P,A) \Big] + \\
& \sum_{A,B} \Big[\sum_{a \in A-B} p(P, s_A) \cdot (1-q) \cdot p(R,B) \; + \sum_{a \in B-A} (1-q) \cdot p(R, s_B) \cdot p(P,A) \; + \\
& \qquad \sum_{a \in A \cap B} (p \cdot p(P, s_A) \cdot (1-q) \cdot p(R,B) + (1-p) \cdot (1-q) \cdot p(R, s_B) \cdot p(P,A) \Big] \; = \\
& \qquad q \cdot p(P \Box_p Q, s) + (1-q) \cdot p(P \Box_p R, s) \; = \; p((P \Box_p Q) \sqcap_q (P \Box_p R), s)
\end{aligned}
$$

Very similar ideas can be applied to prove the soundness of the remaining axioms and rules, thus these proofs are omitted (See [6]). □

As usual, to prove the completeness, we look for the corresponding normal forms of processes. Essentially, these normal forms represent the different ways a process has to complete its execution. This leads us to a generalized internal choice among a set of states, followed by a generalized prefixed external choice among the actions in that set, whose continuations are also in normal form.

Definition 9. We define the kind of processes in normal forms as follows

$$
P \;=\; \prod_{i=1}^{n} [p_i] \mathop{\Box}_{j=1}^{m_i} a_{ij} \to P_{ij}
$$

where each P_{ij} is also in normal form and

$$
n \geq 0\,, \quad \forall i \in \{1, \ldots, n\}\; m_i \geq 0\,, \quad p_i > 0\,, \quad \sum_{[} Ai = 1^n p_i \leq 1
$$

$$
j \neq k \Rightarrow a_{ij} \neq a_{ik} \;, \quad i \neq k \Rightarrow \{a_{ij}\}_{j \in 1, m_i} \neq \{a_{kj}\}_{j \in 1, m_k}
$$

□

These normal forms are quite similar to those of classical CSP processes, but with two important differences:

- The first one is that we have not convexity requirements. For instance, the process $a \to STOP \sqcap (a \to STOP \Box b \to STOP \Box c \to STOP)$ is not in CSP normal form but its probabilistic version is in PCSP normal form.

- The other important difference is that processes P_{ij} do not need to be equal for coincident actions a_{ij}. For instance, the process $a \to b \to STOP \sqcap (a \to STOP \Box b \to STOP)$ is not again in normal form in plain CSP, because the continuation of action a is different in both sides of the internal choice; on the contrary, in our probabilistic model the probabilistic version is now in normal form.

Another important result about the normal forms obtained in our model is that they are very similar of probabilistic processes studied by Hansson and Jonsson (see [10]), with the difference that their processes are defined in that alternating way as the starting point, and we begin with a syntax similar to plain CSP, and we have as conclusion that the normal forms follow this alternating way, probabilistic choices followed by deterministic ones.

With these normal forms, we have the following results:

Lemma 10. Every finite PCSP term can be transformed, using the given proof system above, into another equivalent term in normal form.

Proof. We proceed by induction on the depth of the term. The base case is immediate (both DIV and $STOP$ are already in normal form). For the induction step let us consider the different operators:

- For $P = P_1 \sqcap_p P_2$, we have:

  $$P_1 \equiv \bigsqcap_{A \in \mathcal{A}} [p_A] \mathop{\Box}_{a \in A} a \to P_{a,A} \quad \text{and} \quad P_2 \equiv \bigsqcap_{B \in \mathcal{B}} [p_B] \mathop{\Box}_{b \in B} b \to Q_{b,B}$$

  Then, we have

  $$P \equiv \bigsqcap_{C \in \mathcal{C}} [p_C] \mathop{\Box}_{c \in C} c \to R_{c,C}$$

  Where $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ and

  $$\begin{array}{ll} C = A \in \mathcal{A} \wedge C \notin \mathcal{B} & \Rightarrow\ p_C = p \cdot p_A \ \wedge\ \forall c \in C:\ R_{c,C} = P_{a,A} \\ C = B \in \mathcal{B} \wedge C \notin \mathcal{A} & \Rightarrow\ p_C = (1-p) \cdot p_B \ \wedge\ \forall c \in C:\ R_{c,C} = Q_{b,B} \\ C = A \in \mathcal{A} \wedge C = B \in \mathcal{B} & \Rightarrow\ p_C = p \cdot p_A + (1-p) \cdot p_b \\ & \quad\ \wedge\ \forall c \in C:\ R_{c,C} = P_{c,A} \sqcap_p P_{c,B} \end{array}$$

  Then, we only need to write each $R_{c,C}$ in normal form, which is possible by the induction hypothesis.
- For $P = P_1 \Box_p P_2$, we have again, by the induction hypothesis

  $$P_1 \equiv \bigsqcap_{A \in \mathcal{A}} [p_A] \mathop{\Box}_{a \in A} a \to P_{a,A} \quad \text{and} \quad P_2 \equiv \bigsqcap_{B \in \mathcal{B}} [p_B] \mathop{\Box}_{b \in B} b \to Q_{b,B}$$

  Thus, we have

  $$P \equiv \bigsqcap_{\substack{A \in \mathcal{A} \\ B \in \mathcal{B}}} [p_A \cdot p_B] \mathop{\Box}_{c \in A \cup B} c \to R_{c,A \cup B}$$

  where $R_{c,A \cup B}$ are normal form terms.

  This is not a normal form yet, because several pairs A, B may exist such that their union is the same, say C. But in this case, we only need to apply axiom A6$'$ to obtain the normal form.

– The parallel operator is similar to the external choice.

□

Theorem 11. (Completeness)
Let $P, Q \in PCSP$ be two processes, then $[\![P]\!] \sqsubseteq_D [\![Q]\!]$ if and only if $P \sqsubseteq Q$.

Proof. Let us firstly suppose that both processes are finite. When $P = DIV$ the proof is immediate, so let us suppose that $P \neq DIV$. Then, by Lemma 10, we can restrict the proof to normal forms. Then, let us take N_1 and N_2 two not equivalent normal forms:

$$N_1 \equiv \bigsqcap_{A\in\mathcal{A}}[p_A] \mathop{\Box}_{a\in A} a \to P_{a,A} \quad \text{and} \quad N_2 \equiv \bigsqcap_{B\in\mathcal{B}}[p_B] \mathop{\Box}_{b\in B} b \to Q_{b,B}$$

Three cases may occur:

– The sets $\mathcal{A}$ and $\mathcal{B}$ are different. Then, let us suppose there is a set A such that $A \notin \mathcal{A}$ and $A \in \mathcal{B}$. Then, we have in the denotational semantics $p(N_1, A) = 0$, and $p(N_2, A) = q_A \neq 0$. Thus, applying axiom A3, we can write

$$\begin{aligned} P &= P' \sqcap_{1-q_A} DIV \\ Q &= Q' \sqcap_{1-q_A} \mathop{\Box}_{a\in A} a \to Q_{a,A} \end{aligned}$$

where P' is like P. Then, we can apply axioms D1 and C3 in order to conclude the result $P \sqsubseteq Q$.
– The sets $\mathcal{A}$ and $\mathcal{B}$ are equal, but there is a state A with $p(N_1, A) \neq p(N_2, A)$. We can reason in the same way as the previous case, using now axioms C2 and C3.
– The sets $\mathcal{A}$ and $\mathcal{B}$ are equal, and for every state A we have $p(N_1, A) = p(N_2, A)$. Then, we must have a state A, and an action a such that $[\![P_{a,A}]\!] \sqsubseteq_D [\![Q_{a,A}]\!]$. But in this case, we can reason by induction, assuming that $P_{a,A} \sqsubseteq Q_{a,A}$. Again, applying C2 and C3 we conclude the desired result.

Let us now suppose that P is infinite and Q is finite. Then, the sequence of finite approximations to P satisfies $[\![P^0]\!] \sqsubseteq_D [\![P^1]\!] \sqsubseteq_D \ldots [\![P^n]\!] \sqsubseteq_D \ldots \sqsubseteq_D [\![P]\!] \sqsubseteq_D [\![Q]\!]$. Then, since every P^n is finite, we can prove for every n that $P^n \sqsubseteq Q$, and applying the rule R2, we can conclude $P \sqsubseteq Q$.

On the other hand, if P is finite and Q is infinite, a possible way to prove the desired result would be to show the existence of one n such that $[\![P]\!] \sqsubseteq_D [\![Q^n]\!]$, as occurs in ω-algebraic domains. But this is not true in our model, as we have proved in [6]. Therefore, we need a different approach; in particular we require axiom R3.

Then, if there is an n such that $[\![P]\!] \sqsubseteq_D [\![Q^n]\!]$, just applying R2 we conclude the proof. If there is not such an n, we have for every natural n that $[\![P \sqcap_{\frac{k-1}{k}} DIV]\!] \sqsubseteq_D [\![Q^n]\!]$, and since these two processes are finite, we can prove the same relation by using the axioms, and finally just applying rule R3 we conclude the proof.

Finally, if both processes are infinite, then for every approximation n we have $P^n \sqsubseteq Q$, hence applying R2 we finish the proof.

□

Example 1. Let us take $P = a \to STOP$ and $Q = \mu X.(a \to STOP \sqcap_{\frac{1}{2}} X)$, where the first one is finite, and the second one is infinite, with the following sequence of finite approximations:

$$\begin{array}{ll} Q^0 & = DIV \\ Q^1 & = a \to STOP \sqcap_{\frac{1}{2}} DIV \\ Q^2 & = a \to STOP \sqcap_{\frac{3}{4}} DIV \\ \cdots & \cdots \\ Q^n & = a \to STOP \sqcap_{1-\frac{1}{2^n}} DIV \\ \cdots & \cdots \end{array}$$

Let us see that $P \equiv Q$: the case $Q \sqsubseteq P$ is quite easy, because we have $Q^n \sqsubseteq P$ (axiom D2), and thus, we only need to apply rule R2 to conclude the result.

The proof of the case $P \sqsubseteq Q$ justifies the existence of rule R3. It would not be necessary if we could find a number n such that $P \sqsubseteq Q^n$; however, there is not such a number n, because every finite approximation has a component DIV with a nonzero probability. Thus, the sequence of processes $P_n = P \sqcap_{\frac{n-1}{n}} DIV$ is smaller than Q, because for every P_i there exists a finite approximation Q^j such that for all $n > j$ we have $P_i \sqsubseteq Q^n$, then we can conclude $P \sqsubseteq Q$. □

6 Conclusion and Future Work

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 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 have also defined a system proof, consisting of a set of axioms and inference rules for algebraicaly reasoning about the behaviour of processes. We have proved that this proof system is sound and complete with respect to the denotational semantics. The main interesting points here are that our normal forms are quite similar to those of the classical CSP, but we have no need to introduce convexity requirements and processes after the execution of the same action do not need to be equal. Furthermore, our normal forms are very similar of probabilistic processes studied in [10].

Thus, we consider that the main contribution of the paper is the definition of a very simple proof system for the language, which is equivalent to the denotational semantics presented in a previous work [5].

A natural extension of our language would consist on the introduction of a hiding operator. We have also studied this concrete extension in [6], concluding that for defining this hiding operator we must add some probabilistic information to it if we want to maintain the classical interpretation of this operator. A sketch of the meaning of our hiding operator is showed by means of the following equation valid in CSP:

$$\mathop{\Box}_{a\in A} a \to P_a \backslash c \equiv \begin{cases} \mathop{\Box}_{a\in A} a \to (P_a \backslash c) & \text{if } c \notin A \\ P_c \backslash c \sqcap (P_c \backslash c \,\Box \mathop{\Box}_{a\in A} a \to (P_a \backslash c)) & \text{if } c \in A \end{cases}$$

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 appearing in the right hand side. A first idea could be to consider the same probability associated to the external choice, but this is not possible, because we have that probabilities are meaningless in the context of an external choice among processes prefixed (axioms A8 and A10).

Then, it is necessary to introduce a new probability for this new internal choice, taken it 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 the generalized external choice is not probabilistic; thus, we have chosen 0 as value for the probability of this external choice, on the base of the classical interpretation. Thus, with our model we have the following equation:

$$\mathop{\Box}_{a\in A} a \to P_a \backslash (c,q) \equiv \begin{cases} \mathop{\Box}_{a\in A} a \to (P_a \backslash (c,q)) & \text{if } c \notin A \\ P_c \backslash (c,q) \sqcap_q (P_c \backslash (c,q) \,\Box_0 \mathop{\Box}_{a\in A} a \to (P_a \backslash (c,q))) & \text{if } c \in A \end{cases}$$

In addition to our denotational model, we have also developed some additional equivalent semantics. In particular, we have a testing semantics and an operational semantics (see [4, 6]).

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. 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 $p(P, \langle A{\cdot}a \rangle s) = p \cdot p(Q,s)$.

Finally, the testing semantics is based on the idea of offering tests to processes, being a test a probabilistic tree, where each level represents a set of different actions offered by the observer at that step, from which one will be selected to be executed. The arcs of these probabilistic trees are labelled both by an action and a probability, thus the accepted subset of actions at each step of the process evolution will have associated a certain probability distribution, so that the action to be executed is selected from this set according to it. In this way, the semantics of each process is defined by the probabilities to pass each possible test.

References

1. I. Christoff. *Distinguishing probabilistic processes through testing.* Proceedings Nordic Workshop on Program Correctness, 1989.
2. I. Christoff. *Testing equivalences and fully abstract models for probabilistic processes*, CONCUR-90, LNCS 458, 1990.
3. R. Cleaveland, S.A. Smolka and A. Zwarico. *Testing preorders for probabilistic processes.* ICALP-92, LNCS 624, 1992.
4. F. Cuartero, D. de Frutos and V. Valero. *A Testing Semantics for probabilistic CSP (PCSP)*, Int. Report Dept. Informática y Automática. Univ. Complutense de Madrid (Spain), 1994.
5. F. Cuartero, D. de Frutos and V. Valero. *PCSP: A Denotational Semantics for probabilistic CSP*, Proc. of III AMAST Whorkshop on Real Time Systems. Utah, 1996.
6. F. Cuartero. *CSP probabilístico (PCSP). Un Modelo Probabilístico de Procesos Concurrentes*, Ph.D. Thesis, Univ. Complutense Madrid (Spain), 1993.
7. A. Giacalone, C.-C. Jou and S.A. Smolka. *Algebraic reasoning for probabilistic concurrent systems.* Proc. IFIP TC 2, Sea of Galilee, Israel, 1990.
8. 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.
9. H.A. Hansson. *Time and Probability in Formal Design of Distributed Systems.* Ph.D. Thesis. Dep. of Computer Systems, Uppsala University, 1991.
10. H. Hansson and B. Jonsson. *A calculus for communicating systems with time and probabilities.* Proc. Real-Time Systems Symposium. Florida, 1990.
11. C.-C. Jou and S.A. Smolka. *Equivalences, congruences and complete axiomatizations for probabilistic processes.* CONCUR-90, LNCS 458, 1990.
12. Gavin Lowe. *Prioritized and Probabilistic Models of Timed CSP.* PRG-TR-24-91 Report. Oxford University Computing Laboratory, 1991.
13. M. Nunẽz, D. De Frutos, L. Llana. *Acceptance Trees for Probabilistic Processes.* CONCUR-95, LNCS 962, 1995.
14. K. Seidel. *Probabilistic CSP*, Ph.D. Thesis, Oxford University, 1992.
15. W. Yi and K. Larsen. *Testing probabilistic and nondeterministic processes.* Proc. XII Protocol Specification, Testing and Verification, North Holland 1992.
16. S. Yuen, R. Cleaveland, Z. Dayar, and S.A. Smolka. *Fully Abstract Characterizations of Testing Preorders for Probabilistic Processes*, CONCUR-94, LNCS 836, 1994.