An Axiomatization of Probabilistic Testing*

Manuel Núñez

Dept. de Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain
e-mail: manuelnu@eucmax.sim.ucm.es

Abstract. In this paper we present a sound and complete axiom system for a probabilistic process algebra with recursion. Soundness and completeness of the axiomatization is given with respect to the testing semantics defined in [19].

1 Introduction

During this decade researchers in process algebras have tried to close the gap between formal models and real systems. In particular, features which were abstracted before have been introduced in these models. This is the case of probabilistic information. Several models have introduced probabilities into process algebras, and in [22] models are classified with respect to the interpretation of probabilities in three groups: *reactive, generative,* and *stratified.* In the reactive model there is a different probability distribution for every action, that is, there is no probabilistic relation between different actions. In the generative model there is one probability distribution for all the actions. The stratified model is similar to the generative model but taking into account the probabilistic branching. We will try to explain the differences among these models by means of a few simple examples. Consider the (reactive) process $P = (a; P_1 +_{\frac{1}{3}} a; P_2) + (b; Q_1 +_{\frac{1}{4}} b; Q_2)$. If the environment offers a then P will execute a and then it will behave as either P_1 or P_2 with probabilities $\frac{1}{3}$ and $\frac{2}{3}$, respectively. Something similar happens if the environment offers the action b. Nevertheless, it is not specified how this process would behave if both actions were offered simultaneously. Consider the (generative) process $P' = (a; P_1 +_{\frac{1}{3}} b; P_2)$. If the environment offers a then P' will execute a with a probability 1 and then it will behave as P_1; if the environment offers b then P' will execute b with a probability 1 and then it will behave as P_2; if the environment offers both a and b then P' will execute a, with a probability $\frac{1}{3}$, or it will execute b, with a probability $\frac{2}{3}$. Finally, the processes $(a +_{\frac{1}{2}} b) +_{\frac{2}{3}} c$ and $a +_{\frac{1}{3}} (b +_{\frac{1}{2}} c)$ are equivalent in the generative model but they are not in the stratified one. In this paper we consider a generative interpretation of probabilities based on the following approach: it allows to specify probabilistic systems more precisely that the reactive interpretation, while the (semantic) models are not so complicated as the ones based on the stratified interpretation.

* Research partially supported by the CICYT projects TIC 94-0851-C02-02 and TIC 97-0669-C03-01.

Regarding the testing framework, there have been several proposals for probabilistic extensions (e.g. [3, 5, 23, 19, 18, 4, 10, 11]). In this paper we will consider an extension of the language PPA described in [19]. PPA is a probabilistic process algebra featuring two (probabilistic) choice operators: external and internal. Sometimes it has been argued that the external choice operator should not be extended with a probability, and in fact there are proposals featuring a probabilistic internal choice while the external choice operator does not have a probability parameter (e.g. [23]). Instead, we consider useful to have probabilities in both operators for a number of reasons. First, in order to have the same expressive power with our language as with a CCS-like language[1] we need to include probabilities in both operators. For example, we could not simulate the simple (generative) process $a +_p b$ which relate a and b probabilistically, with a nonprobabilistic external choice. Second, by working with two choice operators we automatically get that the testing equivalence is a congruence, and so we can axiomatize it (working with a CCS-like operator we need, as usual, to consider the largest congruence contained in the testing equivalence). Finally, there are behaviors which can be specified more precisely by using a probabilistic external choice operator. We will illustrate this by means of a simple example. Suppose that we want to specify the behavior of a library where two users can request books. If only one user requests a book then the book is given to him, but if both users request the same book the library must give *priority* to one of the users. On the other hand, the system must be somehow *fair*, avoiding the possibility that if the two users request the same book, this book is always given to the same person. A simplified version of the system can be specified as $P = a; P +_{\frac{1}{4}} b; P$, indicating that if both users request the same book, it will be given with a probability $\frac{1}{4}$ to the user a and with a probability $\frac{3}{4}$ to the user b. Note that if we use a probabilistic internal choice then there is no guarantee that if only one user requests the book, it is given to him, while if we use a nonprobabilistic external choice then we cannot specify the notion of *priority*.

An interesting alternative appears in [16] where the probabilistic external choice operator is considered as an operator derived from the probabilistic internal choice and the *priority* operators. Nevertheless it is also necessary to include some kind of *probability* (the *extremal* value 1) in the external choice operator.

Besides, PPA allows the definition of recursive processes. Moreover, in this paper we extend the language described in [19] with a parallel operator. Our parallel operator is parameterized by a set of actions (the synchronization set). Regarding probability parameters, no agreement has been reached about the parallel operator (see [7] for a discussion on the different possibilities). There are proposals with a probability parameter which assigns weights to the *interleaving* actions of both components (e.g. [6, 18]); alternatively, there are proposals adding two probability parameters assigning weights to the interleaving actions with respect to the synchronization actions, and assigning weights to the interleaving actions of both components, respectively (e.g. [1]). In any case, we claim that

[1] Actually, most of probabilistic models are based on CCS, or in labeled transition systems (which can be easily interpreted as CCS processes).

these parameters only change the probability with which actions are executed, while the *operational* behavior remains the same. Taking this into account, and for the sake of simplicity, in this paper we consider a parallel operator without any probability parameter, but other alternatives can be easily included in our framework (they are discussed in [17]).

The main goal of this paper is to provide a complete and sound axiomatization of testing equivalence for PPA. In [19] a probabilistic extension of the classical testing semantics [8, 12] was defined. Besides, an alternative characterization of the testing semantics (based on an extension of acceptance sets) as well as a fully abstract denotational semantics (based on acceptance trees) were given. So in order to conclude the *semantic trilogy* (alternative characterization, denotational semantics, and axiomatic semantics), a suitable axiomatization of the probabilistic testing semantics should be defined. The starting point for the definition of this axiomatization is (as it was for the other semantics) [12]. As it will be shown in this paper, some of the axioms are (more or less complicated) probabilistic versions of the axioms corresponding to the nonprobabilistic case, while we must add new axioms in order to cope with the specific problems introduced by probabilities.

There have been previous proposals for probabilistic axiom systems. For example, using Synchronous PCCS and *generative* probabilities [9, 13, 21], or with *reactive* probabilities [15]. An axiomatization for a subset of PCCS is presented in [20], and in [1] an axiomatization for ACP finite processes is given. These two proposals also use generative probabilities. Nevertheless, there exists an important difference between all these previous axiomatizations and ours: all of them axiomatize (strong) probabilistic bisimulation, in which there is no *abstraction* of internal movements (i.e. τ actions, or equivalently internal transitions). In fact, observational semantics cannot be directly translated from the nonprobabilistic setting, and a suitable definition of probabilistic *weak* bisimulation for general probabilistic systems was an open problem until [2]. The problem is that there exists some kind of *fairness* in these semantics. Consider $P = recX.\ (a; Nil) \oplus_p X$ (or $P = fix\ X.(a; Nil) +_p (\tau; X)$ using a CCS-like notation). If we *forget* probabilities, P is must equivalent to *divergence* (because of the τ loop), but in a probabilistic setting we would expect that if the environment offers a then P would execute it with probability 1, and so, P should be (probabilistic) testing equivalent to $a; Nil$. This example illustrates why the axiomatization of our testing equivalence cannot be a simple adaptation of the one for nonprobabilistic processes. We will need a rule to express that this kind of recursively defined processes have the same *meaning* as a finite one (in the previous case, $a; Nil$).

The work presented in [6] is the most similar to ours. The main differences between their work and ours are that while our testing semantics is defined following the classical approach (i.e. parallel composition of tested process and test), theirs is defined in an unusual (*ad hoc*) way; besides, they use a reactive interpretation of probabilities (within a probabilistic external choice) which leads to a simplification of the external choice treatment, but complicates the intuitive interpretation of some processes.

As far as we know, the axiomatization presented in this paper is the first one for a semantics abstracting internal movements (in this case a testing semantics), where recursive processes are allowed and probabilites are interpreted using the *generative* model.

The rest of the paper is structured as follows. In Section 2 we recall previous results for our calculus. In Section 3 we present a sound and complete axiomatization for finite processes without parallel composition. In Section 4 we consider recursion, and the previous axiomatization is extended to deal with recursive processes. In Section 5 we give axioms for the parallel operator, showing that this operator can be considered as a derived one. Finally, in Section 6 we present our conclusions and a discussion about the inclusion of hiding in our language.

The full proofs of the results in this paper can be found in [17].

2 Preliminaries

In this section we review our previous results for PPA. The only difference between the language described in this paper and the one presented in [19] is that here we have included a parallel operator. The composition of a process and a test will be defined using the parallel operator of the language. Besides, negative premises in our former operational semantics have been replaced by a syntactic predicate *stable*, and a new function *live*. Anyway, the induced labeled transition systems remains the same as previously.

Definition 1 Given a set of actions Act and a set of identifiers Id, the set of PPA processes is defined by the BNF expression:

$$P ::= \mathit{Nil} \mid \Omega \mid X \mid a;P \mid P \oplus_p P \mid P +_p P \mid P \parallel_A P \mid \mathit{rec}X.P$$

where $p \in (0,1)$, $a \in Act$, $A \subseteq Act$, and $X \in Id$. □

From now on, except if noted, we only consider closed processes, that is processes without free occurrences of variables, and we will omit trailing occurrences of *Nil*. In this process algebra *Nil* is a deadlocked process, Ω is a divergent process, $a;P$ denotes the action a prefixing the process P, $P \oplus_p Q$ denotes an internal choice between P and Q with associated probability p, $P +_p Q$ is an external choice between P and Q with associated probability p, $P \parallel_A Q$ is the parallel composition of P and Q with synchronization alphabet A, and finally $\mathit{rec}X.P$ is used to define recursive processes.

Next, we give a syntactic definition for the *stability* of a process. It expresses that a process has not unguarded internal choices, or equivalently that a process will not be able to execute an internal transition. We also define a function *live* computing whether a stable process is operationally equivalent to *Nil*.

Definition 2 We define the predicate $\mathit{stable}(P)$ over PPA processes as:

- $\mathit{stable}(\mathit{Nil}) = \mathit{stable}(a;P) = \mathit{True}$
- $\mathit{stable}(\Omega) = \mathit{stable}(X) = \mathit{stable}(P_1 \oplus_p P_2) = \mathit{stable}(\mathit{rec}X.P) = \mathit{False}$
- $\mathit{stable}(P_1 +_p P_2) = \mathit{stable}(P_1 \parallel_A P_2) = \mathit{stable}(P_1) \wedge \mathit{stable}(P_2)$

We define the function $\mathit{live}(P)$ over PPA processes as:

- $\mathit{live}(\mathit{Nil}) = 0$

$$(PRE)\frac{}{a;P \xrightarrow{a}_1 P} \qquad (INT1)\frac{}{P \oplus_p Q \succ\!\!\rightarrow_p P} \qquad (INT2)\frac{}{P \oplus_p Q \succ\!\!\rightarrow_{1-p} Q}$$

$$(EXT1)\frac{P \succ\!\!\rightarrow_q P' \ \wedge\ stable(Q)}{P +_p Q \succ\!\!\rightarrow_q P' +_p Q} \qquad (EXT2)\frac{Q \succ\!\!\rightarrow_q Q' \ \wedge\ stable(P)}{P +_p Q \succ\!\!\rightarrow_q P +_p Q'}$$

$$(EXT3)\frac{P \succ\!\!\rightarrow_{q_1} P' \ \wedge\ Q \succ\!\!\rightarrow_{q_2} Q'}{P +_p Q \succ\!\!\rightarrow_{q_1 \cdot q_2} P' +_p Q'}$$

$$(EXT4)\frac{P \xrightarrow{a}_q P' \ \wedge\ stable(Q)}{P +_p Q \xrightarrow{a}_{p \cdot \hat{q}} P'} \qquad (EXT5)\frac{Q \xrightarrow{a}_q Q' \ \wedge\ stable(P)}{P +_p Q \xrightarrow{a}_{(1-p) \cdot \hat{q}} Q'}$$

$$(PAR1)\frac{P \succ\!\!\rightarrow_p P' \ \wedge\ stable(Q)}{P \parallel_A Q \succ\!\!\rightarrow_p P' \parallel_A Q} \qquad (PAR2)\frac{Q \succ\!\!\rightarrow_p Q' \ \wedge\ stable(P)}{P \parallel_A Q \succ\!\!\rightarrow_p P \parallel_A Q'}$$

$$(PAR3)\frac{P \succ\!\!\rightarrow_p P' \ \wedge\ Q \succ\!\!\rightarrow_q Q'}{P \parallel_A Q \succ\!\!\rightarrow_{p \cdot q} P' \parallel_A Q'} \qquad (PAR4)\frac{P \xrightarrow{b}_p P' \ \wedge\ stable(Q) \ \wedge\ b \notin A}{P \parallel_A Q \xrightarrow{b}_{r_1} P' \parallel_A Q}$$

$$(PAR5)\frac{Q \xrightarrow{b}_p Q' \ \wedge\ stable(P) \ \wedge\ b \notin A}{P \parallel_A Q \xrightarrow{b}_{r_1} P \parallel_A Q'} \qquad (PAR6)\frac{P \xrightarrow{a}_p P' \ \wedge\ Q \xrightarrow{a}_q Q' \ \wedge\ a \in A}{P \parallel_A Q \xrightarrow{a}_{r_2} P' \parallel_A Q'}$$

$$(REC)\frac{}{rec\ X.P \succ\!\!\rightarrow_1 P\{rec\ X.P/X\}} \qquad (DIV)\frac{}{\Omega \succ\!\!\rightarrow_1 \Omega}$$

where $\hat{q} = \frac{q}{p \cdot live(P) + (1-p) \cdot live(Q)}$, $r_1 = \frac{p}{\mu(P,Q,A)}$, and $r_2 = \frac{p \cdot q}{\mu(P,Q,A)}$.

Fig. 1. Operational Semantics of PPA.

- $live(a;P) = 1$
- $live(P_1 +_p P_2) = live(P_1 \parallel_A P_2) = \max(live(P_1), live(P_2))$ □

Note that $live(\_)$ is not defined for non stable processes. The set of rules defining the operational semantics is given in Figure 1. There are two types of transitions. The intuitive meaning of an *external* transition $P \xrightarrow{a}_p Q$ is that if the environment offers all the actions in Act, then the probability with which P executes a and then behaves as Q equals p; the meaning of an internal transition $P \succ\!\!\rightarrow_p Q$ is that the process P evolves to Q with probability p, without interaction with the environment.

In order to avoid the problem of deriving the same transition in different ways, we use multisets of transitions. For example, consider $P = a +_{\frac{1}{2}} a$. If we are not careful, we will have the transition $P \xrightarrow{a}_{\frac{1}{2}} Nil$ only once, while we should have this transition twice (that is why we use multisets). This problem is similar for the $\oplus_p$ and $\parallel_A$ operators. So, in our model, if a transition can be derived in several ways, we consider that each derivation generates a different instance. In particular, when we define the testing semantics we will consider multisets of computations as well. Other approaches to solve this problem are to index transitions (e.g. [9]), to increase the number of rules (e.g. [14]), to define a transition probability function (e.g. [4, 20]), or to add the probabilities associated with the same transition (e.g. [23]).

While the rules for prefix, internal choice, divergence and recursion do not need any explanation, we will briefly explain the rest of the rules. $(EXT1-3)$ indicate that whenever any of the arguments of an external choice can evolve

via an internal transition, these transitions are performed until both arguments become *stable*. $(EXT4-5)$ are applied when both processes are stable and (at least) one of them may execute some observable action. The value $\hat{q}$ is obtained by normalizing the probability q of performing this external transition, taking into account whether one or both processes can perform external transitions.

Example 1 Let $P = (a; Nil) +_p Nil$. We have $P \xrightarrow{a}_1 Nil$, while if we would not use this *normalization* we would obtain $P \xrightarrow{a}_p Nil$. □

Rules $(PAR1-3)$ are similar to $(EXT1-3)$. If none of the processes can perform internal transitions, then rules $(PAR4-6)$ are applied. $(PAR4-5)$ deal with *interleaving* actions, while $(PAR6)$ deals with *synchronization* actions. As usual, in these last three rules we use a *normalization factor*, $\mu(P,Q,A)$, in order to obtain that the sum of all the external transitions is 1 (or zero if no transition is possible):

$$\mu(P,Q,A) = \sum_{a\in A} \{\!|\, p\cdot q \mid \exists P', Q' :\ P \xrightarrow{a}_p P' \wedge Q \xrightarrow{a}_q Q' \,|\!\} + \sum_{b\notin A} \{\!|\, p \mid \exists P' :\ P \xrightarrow{b}_p P' \,|\!\} + \sum_{b\notin A} \{\!|\, q \mid \exists Q' :\ Q \xrightarrow{b}_q Q' \,|\!\}$$

As a consequence of this definition of operational semantics, we have that internal and external transitions are not mixed, and then we have the following

Lemma 1 Let P be a process. If there exist p, P' such that $P \succ\!\!\longrightarrow_p P'$ then there do not exist q, a, P'' such that $P \xrightarrow{a}_q P''$, or equivalently if there exist p, a, P' such that $P \xrightarrow{a}_p P'$ then there do not exist q, P'' such that $P \succ\!\!\longrightarrow_q P''$. □

We finish this section generalizing the choice operators to deal with an arbitrary (finite) number of arguments. For the generalized external choice we will use a restricted form, in which all the arguments are prefixed by different actions. These operators will be used, in particular, when we define the notion of normal form.

Definition 3 Let $P_1, P_2, \ldots, P_n$ be processes, and $a_1, a_2, \ldots, a_n \in Act$ different actions. We inductively define the *generalized external choice* by

1. $\sum_{i=1}^{1} [1]\, a_1; P_1 = a_1; P_1$ 2. $\sum_{i=1}^{n} [p_i]\, a_i; P_i = (a_1; P_1) +_{p_1} (\sum_{i=1}^{n-1} [\frac{p_{i+1}}{1-p_1}]\, a_{i+1}; P_{i+1})$

where $p_1, p_2, \ldots, p_n > 0$ are such that $\sum p_i = 1$.
We inductively define the *generalized internal choice* by

1. $\bigoplus_{i=1}^{0} [p_i]\, P_i = \Omega$ 2. $\bigoplus_{i=1}^{1} [1]\, P_1 = P_1$

3. $\bigoplus_{i=1}^{n} [p_i]\, P_i = \bigoplus_{i=1}^{n} [\frac{p_i}{p}]\, P_i \oplus_p \Omega$ [if $p = \sum p_i < 1 \ \wedge\ n > 0$]

4. $\bigoplus_{i=1}^{n} [p_i]\, P_i = P_1 \oplus_{p_1} (\bigoplus_{i=1}^{n-1} [\frac{p_{i+1}}{1-p_1}]\, P_{i+1})$ [if $\sum p_i = 1 \ \wedge\ n > 1$]

where $p_1, p_2, \ldots, p_n > 0$ are such that $\sum p_i \leq 1$. □

Let us remark that the sum of the probabilities associated with a generalized internal choice may be less than 1. The difference between 1 and this value indicates the probability of divergence. In this case the third clause is applied first so that the sum of the probabilities associated with the remaining generalized internal choice is equal to 1 (afterwards the second or the fourth clauses will be used). We consider that the empty summation (i.e. $\sum_{i=1}^{0} P_i$) represents the process *Nil*.

2.1 Testing Semantics

As in the nonprobabilistic case tests will be just processes where the alphabet Act is extended with a new action ω indicating *successful* termination. The operational semantics of tests is the same as the one for processes (considering ω as an ordinary action). Now we have to define how a process interacts with a test. As usual, this interaction is modeled by the parallel composition of the process and the test. We will denote the composition of a process P and a test T by $P \mid T$, and it is defined as $P \mid T = P \parallel_{Act} T$. Note that ω is not included in the synchronization alphabet. Now we will define a function computing the probability with which a test is passed by a process.

Definition 4 Let P_0 be a process and T_0 be a test. A *computation* is a sequence of transitions $C = P_0 \mid T_0 \longmapsto_{p_1} P_1 \mid T_1 \longmapsto_{p_2} \cdots P_{n-1} \mid T_{n-1} \longmapsto_{p_n} P_n \mid T_n \cdots$, where $\longmapsto_p$ denotes either $\rightarrowtail_p$, or $\xrightarrow{a}_p$ for some $a \in Act \cup \{\omega\}$. If C is finite we say that $length(C) = n$.

Let C be a computation such that $length(C) = n$. We say that C is *successful* if $P_{n-1} \mid T_{n-1} \xrightarrow{\omega}_p P_n \mid T_n$, and there is no other occurrence of ω in C, that is, $\nexists n' < n, p' : \; P_{n'-1} \mid T_{n'-1} \xrightarrow{\omega}_{p'} P_{n'} \mid T_{n'}$.

We denote by $\tilde{C}_{P|T}$ the *multiset* of *successful computations* of $P \mid T$. We define the *probability of a successful computation* S as $Pr(S) = \prod_{i=1}^{length(S)} p_i$. Finally, we define the *probability* with which the process P *passes* the test T as $pass(P,T) = \sum_{S \in \tilde{C}_{P|T}} Pr(S)$.

Given two processes P and Q, we say that they are *testing equivalent*, and we write $P \approx Q$, iff for all test T we have $pass(P,T) = pass(Q,T)$. □

Note that $pass(P,T) = \lim_{n\to\infty} \sum \{\!| Pr(S) \mid S \in \tilde{C}_{P|T} \wedge length(S) < n |\!\}$. Let us remark that the role played by tests of the form $a +_p (\tau;\omega)$ in other models (e.g. [5, 24]) is played in our model by tests of the form $a +_p \omega$, which are not trivially passed within our framework. For example, the process $P = a$ *passes* the test above with a probability $1 - p$.

In the following, we will show that the whole family of tests can be reduced to a simpler class of tests. Although this fact is not important for the axiomatic semantics, it strongly simplifies soundness proofs. First, we have that *infinite* tests (i.e. tests having occurrences of recursion) are not necessary.

Lemma 2 $P \approx Q$ iff for all *finite* test T we have $pass(P,T) = pass(Q,T)$. □

Now we define a set of *essential* tests, called *probabilistic barbs*, with sufficient discriminatory power to distinguish any pair of non-equivalent processes. These probabilistic barbs are very similar to *probabilistic traces* [24] if we consider the latter as probabilistic tests.

Definition 5 The set of *probabilistic barbs*, denoted by $\mathcal{PB}$, is defined by means of the following BNF expression:

$$T ::= \sum_{i=1}^{s}[p_i]\ (a_i; Nil) +_p \omega \ \mid \ \sum_{i=1}^{s}[p_i]\ a_i; T_i \qquad \text{where } T_i = \begin{cases} T & \text{if } i = s \\ Nil & \text{otherwise} \end{cases}$$

where $p \in (0,1)$, $\sum p_i = 1$, and $a_i \in Act$. □

Theorem 1 $P \approx P'$ iff for all $T \in \mathcal{PB}$ we have $pass(P,T) = pass(P',T)$. □

In the rest of this paper, mainly in some of the proofs, we will use the denotational semantics for PPA given in [19]. Anyway, previous knowledge of this semantics is not necessary in order to understand the bulk of this paper. In this paper we use the following:

- The denotational semantics of a syntactic process P is denoted by $[\![P]\!]$.
- The semantic order relation and its induced equivalence are denoted by $\sqsubseteq_{\mathbf{PAT}}$ and $=_{\mathbf{PAT}}$ respectively.
- The probability with which a process P reaches a node labeled by the state A of its semantic tree after a sequence s is denoted by $p([\![P]\!], s, A)$. In particular, if $P = P_1 \oplus_p P_2$, then $p([\![P]\!], s, A) = p \cdot p([\![P_1]\!], s, A) + (1-p) \cdot p([\![P_2]\!], s, A)$.
- The denotational semantics of recursive processes is given by their finite approximations.
- *(Full Abstraction)* Let P, Q be PPA processes. Then, $P \approx Q$ iff $[\![P]\!] =_{\mathbf{PAT}} [\![Q]\!]$.

3 Axiomatization for Finite Processes

In this section we will define an axiom system inducing an equivalence relation, denoted by $\equiv$, among the terms of the language PPA_{fin} which is the subset of PPA where neither $\|_A$ nor $recX.P$ have been included. We will also use an order relation $\sqsubseteq$ to define this equivalence relation. This system includes axioms expressing algebraic properties of the operators as well as relations among the operators like distributivity. We will also present some axioms which are sound in the nonprobabilistic framework but not in our case. Soundness of rules (axioms) dealing with $\equiv$ will be shown with respect to the testing equivalence, and we will frequently use Theorem 1, while soundness of the ones corresponding to $\sqsubseteq$ will be shown with respect to the fully abstract denotational semantics equivalence defined in [19]. Although we will mix soundness (and completeness) proofs with respect to either the testing or the denotational semantics, this process is correct. First, we will prove $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$ iff $\vdash P \sqsubseteq Q$. From this result, given that both $\sqsubseteq_{\mathbf{PAT}}$ and $\sqsubseteq$ are preorders, we will trivially get $[\![P]\!] =_{\mathbf{PAT}} [\![Q]\!]$ iff $\vdash P \equiv Q$ an so, by full abstraction, we finally obtain the desired result $P \approx Q$ iff $\vdash P \equiv Q$.

The first axioms of our system are similar to those in [12], and they express that internal choice is idempotent, commutative and associative, while external choice is commutative and *Nil* is its identity element. Commutativity and associativity are intended up to a suitable rebalance of probabilities. Soundness is trivial.

(**II**) $P \oplus_p P \equiv P$ (**CI**) $P \oplus_p Q \equiv Q \oplus_{1-p} P$

(**AI**) $P \oplus_p (Q \oplus_q R) \equiv (P \oplus_{p'} Q) \oplus_{q'} R$, where $q' = p + q - p \cdot q$ and $p' = \frac{p}{q'}$

(**CE**) $P +_p Q \equiv Q +_{1-p} P$ (**NE**) $P +_p Nil \equiv P$

Now, we present some *axioms* that are not sound in our probabilistic model, although they were in nonprobabilistic testing models. First, in general, the external choice operator is not idempotent as the following example shows:

Example 2 Consider the processes $P = a \oplus_{\frac{1}{2}} b$ and $P' = P +_{\frac{1}{2}} P$, and the test $T = a;\omega$. We have $pass(P,T) = \frac{1}{2}$ while $pass(P',T) = \frac{3}{4}$. □

This fact also appears in models dealing with *replication* where the choice between the same process is not equivalent to the original process. On the other hand we have the following:

Proposition 1 Let P be a stable process. Then, for any $p \in (0,1)$ we have $P \approx (P +_p P)$. □

Moreover, associativity of the external choice does not hold, even if we introduce a rebalance of probabilities similar to that used in axiom (**AI**).

Example 3 Consider $P = a +_{\frac{1}{2}} (b +_{\frac{1}{2}} Nil)$ and $P' = (a +_{\frac{2}{3}} b) +_{\frac{3}{4}} Nil$, and let $T = a;\omega +_{\frac{1}{2}} b;Nil$. We have $pass(P,T) = \frac{1}{2}$, but $pass(P',T) = \frac{2}{3}$. This is so because $P \approx (a +_{\frac{1}{2}} b)$ while $P' \approx (a +_{\frac{2}{3}} b)$, and obviously $(a +_{\frac{1}{2}} b) \not\approx (a +_{\frac{2}{3}} b)$. □

This lack of associativity could create problems when trying to define normal forms, but fortunately non-associativity only appears in the presence of *Nil*. We can easily solve the problem since, by axiom (**NE**), we can remove all the occurrences of the process *Nil* in external choices. In short, we have a restricted form of associativity that will be enough in order to transform any finite process into normal form.

Proposition 2 Let P_1, P_2, P_3 be processes such that for all i we have $P_i \not\longrightarrow$, that is, stable processes which are not operationally equivalent to *Nil*. Then, $P_1 +_p (P_2 +_q P_3) \approx (P_1 +_{p'} P_2) +_{q'} P_3$, where $q' = p + q - p \cdot q$ and $p' = \frac{p}{q'}$. □

Next we will introduce axioms dealing with divergence. Soundness proofs with respect to $\sqsubseteq_{\mathbf{PAT}}$ are again trivial.

(**D**) $\Omega \sqsubseteq P$ (**DI**) $P \oplus_p \Omega \sqsubseteq P$ (**DE**) $P +_p \Omega \equiv \Omega$

Note that, in contrast with the nonprobabilistic case, $P \oplus_p \Omega \not\equiv \Omega$. For example, consider $P = a; Nil$, and $T = a; \omega$. We have $pass(P \oplus_p \Omega, T) = p$, while $pass(\Omega, T) = 0$.

Now, we will consider the distributive laws between the external and the internal choice operators. The soundness of the following axiom is easy to prove:

$$(\mathbf{DEI})\ P_1 +_p (P_2 \oplus_q P_3) \equiv (P_1 +_p P_2) \oplus_q (P_1 +_p P_3)$$

The previous axiom can be generalized to deal with generalized internal choices:

$$(\mathbf{DEIG})\ P +_p (\bigoplus_{i=1}^{n} [p_i]\ P_i) \equiv \bigoplus_{i=1}^{n} [p_i]\ (P +_p P_i)$$

On the contrary, the converse distributivity does not hold in general. This is illustrated by the following example.

Example 4 Let $P = a \oplus_{\frac{1}{2}} (b +_{\frac{1}{2}} c)$ and $Q = (a \oplus_{\frac{1}{2}} b) +_{\frac{1}{2}} (a \oplus_{\frac{1}{2}} c)$. We have $pass(P, a; \omega) = \frac{1}{2}$ while $pass(Q, a; \omega) = \frac{3}{4}$. □

As in the nonprobabilistic case, in order to prove the completeness of the logic system we will introduce the adequate notion of normal form. Given that in our normal forms we will have generalized external choices instead of binary ones, we need an axiom for composing two generalized external choices by a binary external choice, called (**EBE**), and one for composing two generalized external choices having the same associated actions and the same probabilities by an internal choice, called (**IBE**).

Let $A = \{a_1, \ldots, a_n\} \subseteq Act$ and $B = \{b_1, \ldots, b_m\} \subseteq Act$. Let us consider the processes $P = \sum_{i=1}^{n} [p_i]\ a_i; P_i$ and $Q = \sum_{j=1}^{m} [q_j]\ b_j; Q_j$. Then, the following axiom is sound:

$$(\mathbf{EBE})\ P +_p Q \equiv R$$

where $R = \sum_{k=1}^{l} [r_k]\ c_k; R_k$, $C = \{c_1, \ldots, c_l\} = A \cup B$, and

$$r_k = \begin{cases} p \cdot p_i & \text{if } c_k = a_i \in A - B \\ (1-p) \cdot q_j & \text{if } c_k = b_j \in B - A \\ p \cdot p_i + (1-p) \cdot q_j & \text{if } c_k = a_i = b_j \in A \cap B \end{cases}$$

$$R_k = \begin{cases} P_i & \text{if } c_k = a_i \in A - B \\ Q_j & \text{if } c_k = b_j \in B - A \\ P_i \oplus_{p'} Q_j & \text{if } c_k = a_i = b_j \in A \cap B \ \wedge\ p' = \frac{p \cdot p_i}{p \cdot p_i + (1-p) \cdot q_j} \end{cases}$$

Let $\{(a_1, p_1), (a_2, p_2), \ldots, (a_n, p_n)\}$ be a non empty state. Then the following axiom is sound:

$$(\mathbf{IBE})\ (\sum_{i=1}^{n} [p_i]\ a_i; P_i) \oplus_p \sum_{i=1}^{n} [p_i]\ a_i; Q_i \equiv \sum_{i=1}^{n} [p_i]\ a_i; (P_i \oplus_p Q_i)$$

Next we present the soundness proof of the axiom **EBE** (the proof of **IBE** is easier). First, we give an auxiliary definition.

Definition 6 Let $T = \sum_{i=1}^{u}[s_i]\ (t_i; Nil) +_s \omega \mid \sum_{i=1}^{u}[s_i]\ t_i; T_i$ be a probabilistic barb. We define its *set of initial actions*, denoted by $\tilde{T}$, as $\tilde{T} = \{t_1, \ldots, t_u\}$. Given a set of actions $C \subseteq Act$, we define the set T_C as $T_C = \{t_i \mid t_i \in C \cap \tilde{T}\}$. □

Lemma 3 The axiom (**EBE**) is sound.

Proof. In order to clarify the notation, $p(P, a_i)$ stands for p_i and $p(Q, b_j)$ stands for q_j. Note that applying the rules $(EXT4)$ and $(EXT5)$ we have $P \xrightarrow{a}_q P'$ implies $P +_p Q \xrightarrow{a}_{p \cdot q} P'$ and $Q \xrightarrow{a}_q P'$ implies $P +_p Q \xrightarrow{a}_{(1-p)\cdot q} P'$. We will show that for any $T \in \mathcal{PB}$ we have $pass(P +_p Q, T) = pass(R, T)$.

If T is a probabilistic barb of the form $T = \sum_{i=1}^{u}[s_i]\ (t_i; Nil) +_s \omega$, then

$$
\begin{aligned}
&pass(P +_p Q, T) = \\
&= \frac{1-s}{(1-s) + \sum_{t_i \in T_A} s \cdot s_i \cdot p \cdot p(P, t_i) + \sum_{t_i \in T_B} s \cdot s_i \cdot (1-p) \cdot p(Q, t_i)} \\
&= \frac{1-s}{(1-s) + \sum_{t_i \in T_{A-B}} s \cdot s_i \cdot p \cdot p(P, t_i) + \sum_{t_i \in T_{B-A}} s \cdot s_i \cdot (1-p) \cdot p(Q, t_i) + \sum_{t_i \in T_{A \cap B}} s \cdot s_i \cdot (p \cdot p(P, t_i) + (1-p) \cdot p(Q, t_i))} \\
&= pass(R, T)
\end{aligned}
$$

Let $T = \sum_{i=1}^{u}[s_i]\ t_i; T_i$ be a probabilistic barb such that if $i = u$ then $T_i = T'$ for some probabilistic barb T', while $T_i = Nil$ otherwise, we distinguish four cases:

1. $\exists\ 1 \leq i \leq n :\ a_i = t_u \in A - B$
2. $\exists\ 1 \leq j \leq m :\ b_j = t_u \in B - A$
3. $\exists\ 1 \leq i \leq n,\ 1 \leq j \leq m : a_i = b_j = t_u \in A \cap B$
4. $t_u \notin A \cup B$.

In the last case we trivially get $pass(P +_p Q, T) = pass(R, T) = 0$. The proof for the first three cases is very similar, so we present, as an example, the proof for the third case:

$$
\begin{aligned}
&pass(P +_p Q, T) \\
&= \frac{s_u \cdot p \cdot p_i \cdot pass(P_i, T') + s_u \cdot (1-p) \cdot q_j \cdot pass(Q_j, T')}{\sum_{t_i \in T_A} s_i \cdot p \cdot p(P, t_i) + \sum_{t_i \in T_B} s_i \cdot (1-p) \cdot p(Q, t_i)} \\
&= \frac{s_u \cdot (p \cdot p_i + (1-p) \cdot q_j) \cdot pass(P_i \oplus_{q'} Q_j, T')}{\sum_{t_i \in T_{A-B}} s_i \cdot p \cdot p(P, t_i) + \sum_{t_i \in T_{B-A}} s_i \cdot (1-p) \cdot p(Q, t_i) + \sum_{t_i \in T_{A \cap B}} s_i \cdot (p \cdot p(P, t_i) + (1-p) \cdot p(Q, t_i))} \\
&= pass(R, T), \qquad \text{where } q' = \frac{p \cdot p_i}{p \cdot p_i + (1-p) \cdot q_j}
\end{aligned}
$$

□

(**O1**) $\frac{P \sqsubset Q \ \wedge \ Q \sqsubset P}{P \equiv Q}$	(**O2**) $\frac{P \equiv Q}{P \sqsubseteq Q, \ Q \sqsubseteq P}$	(**O3**) $\frac{P \sqsubset Q \ \wedge \ Q \sqsubset R}{P \sqsubset R}$
(**C1**) $\frac{P \sqsubset Q}{a;P \sqsubseteq a;Q}$	(**C2**) $\frac{P \sqsubset Q \ \wedge \ P' \sqsubset Q'}{P +_p P' \sqsubseteq Q +_p Q'}$	(**C3**) $\frac{P \sqsubset Q \ \wedge \ P' \sqsubset Q'}{P \oplus_p P' \sqsubseteq Q \oplus_p Q'}$
(**RE**) $\frac{}{P \equiv P}$	(**OI1**) $\frac{P \sqsubset Q}{P \sqsubseteq P \oplus_p Q}$	(**OI2**) $\frac{P \sqsubset Q}{P \oplus_p Q \sqsubseteq Q}$

Fig. 2. Inference Rules.

In addition to the previous axioms, we need a set of rules indicating that the relation $\equiv$ fulfills some *good* properties. The inference rules of our logic system are given in Figure 2. Rules (**O1-3**) indicate that $\sqsubseteq$ is an order relation. Rules (**C1-3**) say that $\sqsubseteq$ is a precongruence with respect to the basic operators of the language. (**RE**) says that $\equiv$ is reflexive. Finally, (**OI1-2**) indicate that internal choice occupies an intermediate position between the corresponding processes. Soundness of (**O1-3**), (**C1-3**), and (**RE**) rules is trivial with respect to $\sqsubseteq_{\mathbf{PAT}}$, given that the latter is defined compositionally, while the soundness of (**OI1-2**) can be easily shown with respect to $\sqsubseteq_{\mathbf{PAT}}$.

Definition 7 Given two processes P and Q, we write $\vdash P \sqsubseteq Q$ (resp. $\vdash P \equiv Q$) if $P \sqsubseteq Q$ (resp. $P \equiv Q$) can be derived from the axioms given before and the rules given in Figure 2. □

Given that the previous axioms and rules are sound, we automatically get

Theorem 2 (Soundness for PPA_{fin} **)**
For any P, $Q \in \mathrm{PPA}_{fin}$ we have $\vdash P \sqsubseteq Q$ implies $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$. As a corollary, we also have $\vdash P \equiv Q$ implies $[\![P]\!] =_{\mathbf{PAT}} [\![Q]\!]$, and by using full abstraction of $=_{\mathbf{PAT}}$, $\vdash P \equiv Q$ implies $P \approx Q$. □

This result indicates that if we can derive the equivalence between two finite processes from the axiom system, then these two processes are testing equivalent. In the remainder of the section we will prove that our axiomatization is also *complete*, that is, if two finite processes are testing equivalent, then the equivalence of these processes with respect to $\equiv$ can be derived from the given axiomatization.

In order to simplify the completeness proof we will use a notion of *normal form*, and we will prove that every PPA_{fin} process can be transformed into a normal form by applying the axioms and rules of our axiom system. Our normal forms are similar to those in [12], that is, they will be generalized internal choices of generalized external choices. The actions associated with the generalized external choices *prefix* normal forms, so that normal forms will be processes which have a strict alternation between generalized internal choices and generalized external choices. Moreover, we will not allow two generalized external choices associated with the same internal choice to have the same set of actions and the

same probability distribution associated with them. Actually, our normal forms are the syntactic expression of the semantic processes described in [19].

Definition 8 *Normal Forms* are those PPA_{fin} processes defined by means of the following BNF expression:

$$N ::= \bigoplus_{i=1}^{n} [p_i] \sum_{j=1}^{r_i} [p_{i,j}]\, a_{i,j}\,;N$$

where $n \geq 0$, $\sum_{i=1}^{n} p_i \leq 1$, and

- $\forall\ 1 \leq i \leq n : p_i > 0 \wedge r_i \geq 0$, and if $r_i > 0$ then $\sum_{j=1}^{r_i} p_{i,j} = 1 \wedge \forall\ 1 \leq j \leq r_i : p_{i,j} > 0$
- $\forall\ 1 \leq i \leq n : \forall\ 1 \leq k, l \leq r_i, k \neq l : a_{i,k} \neq a_{i,l}$
- $\forall\ 1 \leq u, v \leq n, u \neq v : \{(a_{u,j}, p_{u,j})\}_{j=1}^{r_u} \neq \{(a_{v,j}, p_{v,j})\}_{j=1}^{r_v}$ □

Note that, in contrast with [12], we do not force the continuations after the same action in different states to be equal. We will use the following alternative notation for normal forms:

$$N ::= \bigoplus_{A \in \mathcal{A}} [p_A] \sum_{(a,p_a) \in A} [p_a]\, a; N_{a,A}$$

where $\mathcal{A}$ is a finite subset of $\mathcal{P}(Act \times (0,1])$ such that for all $A \in \mathcal{A}$, if $A \neq \emptyset$ then $\sum \{\!|\, p_a \mid (a,p_a) \in A \,|\!\} = 1$. The next result states that any PPA_{fin} process can be transformed into a normal form by using the axiom system.

Theorem 3 Given $P \in \text{PPA}_{fin}$, there exists a normal form N such that $\vdash P \equiv N$.

Proof. The proof is done by structural induction, and we only present the case for internal choice. The proof for *Nil*, Ω, and prefix is trivial, while the proof for external choice is similar to the one for internal choice.

If $P = P_1 \oplus_p P_2$, then by induction hypothesis P_1 and P_2 can be transformed into normal forms N_1 and N_2 respectively, such that $P_1 \equiv N_1$ and $P_2 \equiv N_2$, where

$$N_1 = \bigoplus_{A \in \mathcal{A}} [p_A] \sum_{(a,p_a) \in A} [p_a]\, a; P_{a,A} \qquad \text{and} \qquad N_2 = \bigoplus_{B \in \mathcal{B}} [q_B] \sum_{(b,q_b) \in B} [q_b]\, b; Q_{b,B}$$

Applying the rules (**C3**) and (**O1-2**) we obtain $P_1 \oplus_p P_2 \equiv N_1 \oplus_p N_2$. Now, applying the axiom (**IBE**), if necessary, and given that any generalized internal choice can be decomposed into binary internal choices and vice versa, we obtain the normal form

$$N = \bigoplus_{C \in \mathcal{C}} [r_C] \sum_{(c,r_c) \in C} [r_c]\, c; R_{c,C}$$

where $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ and for any $C \in \mathcal{C}$ we have three possibilities:

$$\begin{array}{lll}
C = A \in \mathcal{A} - \mathcal{B} & \Rightarrow & r_C = p \cdot p_A \ \wedge\ \forall\ c \in C :\ R_{c,C} = P_{c,A} \\
C = B \in \mathcal{B} - \mathcal{A} & \Rightarrow & r_C = (1-p) \cdot q_B \ \wedge\ \forall\ c \in C :\ R_{c,C} = Q_{c,B} \\
C = A = B \in \mathcal{A} \cap \mathcal{B} & \Rightarrow & r_C = p \cdot p_A + (1-p) \cdot q_B \ \wedge\ \forall\, c \in C : R_{c,C} = P_{c,A} \oplus_{\frac{p \cdot p_A}{r_C}} Q_{c,B}
\end{array}$$

In the first two cases we obtain that $R_{c,C}$ are already normal forms, while in the last case we can apply the induction hypothesis to the corresponding processes $P_{c,A}$ and $Q_{c,B}$ in order to get a normal form. Therefore, we have got a normal form N such that $N_1 \oplus_p N_2 \equiv N$, and so, applying the rules (**O1-3**), we obtain $P_1 \oplus_p P_2 \equiv N$. □

Next we present a result stating that if two (semantic) processes are related by $\sqsubseteq_{\mathbf{PAT}}$, then the corresponding syntactic processes are also related by $\sqsubseteq$.

Lemma 4 Let $P, Q \in \mathrm{PPA}_{fin}$. Then, $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$ implies $P \sqsubseteq Q$.

Proof. By Theorem 3, P and Q can be transformed into normal forms by using the axiom system. So we can restrict ourselves to the study of the equivalent normal forms. Let us take

$$P = \bigoplus_{A \in \mathcal{A}} [p_A] \sum_{(a,p_a) \in A} [p_a]\ a; P_{a,A} \qquad \text{and} \qquad Q = \bigoplus_{B \in \mathcal{B}} [q_B] \sum_{(b,q_b) \in B} [q_b]\ b; Q_{b,B}$$

where $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(\Sigma \times (0,1])$. Note that $p(P, \epsilon, A) = p_A$ and $p(Q, \epsilon, B) = q_B$. The proof is done by using structural induction over the *complexity* of processes. By complexity we mean the *depth* of processes, that is, the maximum number of times that a generalized internal choice (followed by a generalized external choice) appears in a row. If two processes have the same depth, we consider that a process is more complex than another one if the reachable states of the latter are contained in the ones of the former. We have three possibilities:

• $\mathcal{A}$ and $\mathcal{B}$ are different • $\mathcal{A} = \mathcal{B}$ and $\exists C : p_C \neq q_C$ • $\mathcal{A} = \mathcal{B}$ and $\forall C \in \mathcal{A} : p_C = q_C$

We present the proof only for the first case. So we suppose that $\mathcal{A} \neq \mathcal{B}$. Given that $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$, there exists a state B' such that $B' \in \mathcal{B} - \mathcal{A}$. Moreover, the probability in Q of any state belonging to $\mathcal{B}$ must be greater than or equal to the corresponding one in P. Moreover, since $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$, we have $\mathcal{A} \subseteq \mathcal{B}$. Then we have

$$\sum_{A \in \mathcal{A}} p_A \leq \sum_{A \in \mathcal{A}} q_A < \sum_{A \in \mathcal{A}} q_A + q_{B'} \leq \sum_{B \in \mathcal{B}} q_B \leq 1$$

and so the probability of P diverging in its first step is greater than or equal to $q_{B'}$. Now, using the axiom (**AI**), we can rewrite P and Q as:

$$P \equiv \left(\bigoplus_{A \in \mathcal{A}} [\tfrac{p_A}{1-q_{B'}}] \sum_{(a,p_a) \in A} [p_a]\ a; P_{a,A} \right) \oplus_{1-q_{B'}} \Omega$$

$$Q \equiv \left(\bigoplus_{B \in \mathcal{B}-B'} [\tfrac{q_B}{1-q_{B'}}] \sum_{(b,q_b) \in B} [q_b]\ b; Q_{b,B} \right) \oplus_{1-q_{B'}} \left(\sum_{(b',q_{b'}) \in B'} [q_{b'}]\ b'; Q_{b',B'} \right)$$

Applying axiom (**D**), we have

$$\Omega \sqsubseteq \sum_{(b',q_{b'}) \in B} [q_{b'}]\ b'; Q_{b',B'} \tag{1}$$

Given that $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$, we obtain

$$[\![\bigoplus_{A \in \mathcal{A}} [\frac{p_A}{1-q_{B'}}] \sum_{(a,p_a) \in A} [p_a]\ a; P_{a,A}]\!] \sqsubseteq_{\mathbf{PAT}} [\![\bigoplus_{B \in \mathcal{B}-B'} [\frac{q_B}{1-q_{B'}}] \sum_{(b,q_b) \in B} [q_b]\ b; Q_{b,B}]\!]$$

and applying the induction hypothesis, given that the states of the right hand side process are contained in those of the process Q, we have

$$\bigoplus_{A \in \mathcal{A}} [\frac{p_A}{1-q_{B'}}] \sum_{(a,p_a) \in A} [p_a]\ a; P_{a,A} \sqsubseteq \bigoplus_{B \in \mathcal{B}-B'} [\frac{q_B}{1-q_{B'}}] \sum_{(b,q_b) \in B} [q_b]\ b; Q_{b,B} \tag{2}$$

Then applying the rule (**C3**) to equations (1) and (2) we conclude $P \sqsubseteq Q$. □

By using the equivalence between $=_{\mathbf{PAT}}$ and $\approx$, and this result we obtain

Theorem 4 (Completeness for PPA_{fin} **)**
For any processes $P, Q \in \mathrm{PPA}_{fin}$ we have $P \approx Q$ implies $\vdash P \equiv Q$. □

4 Extension of the System to Infinite Processes

In this section we extend the previous results to deal with recursion, adding to PPA_{fin} recursive processes (we call this language PPA_{rec}). We will work with the approximations by *finite processes* of recursive processes, which are defined like in [12].

Definition 9 Let P be a PPA_{rec} process. For any $n \in \mathbb{N}$, we define the n-th *finite approximation* of P as $P^0 = \Omega$, and for $n \geq 0$:

- $X^{n+1} = X$, if $X \in Id$
- $Nil^{n+1} = Nil$
- $\Omega^{n+1} = \Omega$
- $(a; P)^{n+1} = a; P^{n+1}$
- $(P \oplus_p Q)^{n+1} = P^{n+1} \oplus_p Q^{n+1}$
- $(recX.P)^{n+1} = P^{n+1}\{(recX.P)^n / X\}$
- $(P +_p Q)^{n+1} = P^{n+1} +_p Q^{n+1}$

□

Note that for PPA_{fin} processes it holds that their finite approximations are equal to themselves. Also note that each finite approximation is a finite process, and therefore we can use the results given in the previous section when reasoning about finite approximations. The previous axiom system is extended with three new rules:

$$(\mathbf{R1})\ \frac{}{P\{recX.P/X\} \sqsubseteq recX.P} \qquad (\mathbf{R2})\ \frac{\forall\ n \in \mathbb{N}:\ P^n \sqsubseteq R}{P \sqsubseteq R}$$

$$(\mathbf{R3})\ \frac{\forall\ n \in \mathbb{N}:\ P \oplus_{\frac{n-1}{n}} \Omega \sqsubseteq R}{P \sqsubseteq R}$$

The first two rules already appeared in [12], and their soundness proofs easily follow from the definition of the denotational semantics of recursive processes.[2] Concretely, soundness of (**R1**) is trivial because $[\![P\{recX.P/X\}]\!] = \bigsqcup_{n=1}^{\infty} [\![P^n]\!]$, while (**R2**) is sound because we are working within a *cpo*, and so $[\![P]\!]$ is the least upper bound of $\{[\![P^i]\!]\}_{i=0}^{\infty}$.

The rule (**R3**) is added to our system because of *technical* reasons. This rule is necessary because the semantics of finite syntactic processes (i.e. without occurrences of the recursion operator) is given by non compact elements in the semantic domain. We will comment more thoroughly this rule when we use it.

Lemma 5 The rule (**R3**) is sound.

Proof. Let us suppose that for all $n \in \mathbb{N}$ we have $[\![P \oplus_{\frac{n-1}{n}} \Omega]\!] \sqsubseteq_{\mathbf{PAT}} [\![R]\!]$. That is, for any $n \in \mathbb{N}$, any sequence s, and any state A, $p([\![P \oplus_{\frac{n-1}{n}} \Omega]\!], s, A) \leq p([\![R]\!], s, A)$. From the definition of the internal choice semantic function, we have $p([\![P \oplus_{\frac{n-1}{n}} \Omega]\!], s, A) = \frac{n-1}{n} \cdot p([\![P]\!], s, A) + \frac{1}{n} \cdot p([\![\Omega]\!], s, A) = \frac{n-1}{n} \cdot p([\![P]\!], s, A)$.

[2] As usually, the (denotational) semantics of a recursive process is given by the limit of its finite approximations, that is, $[\![recX.\ P]\!] = \sqcup_{n=0}^{\infty} [\![P^n]\!]$.

Taking into account the two previous facts, we have that for any s and A:

$$p([\![P]\!], s, A) = \lim_{n\to\infty} \tfrac{n-1}{n} \cdot p([\![P]\!], s, A) \leq p([\![R]\!], s, A)$$

which implies $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![R]\!]$. □

Theorem 5 (Soundness for PPA_{rec} **)**
Let P, Q be PPA_{rec} processes. We have that $\vdash P \equiv Q$ implies $P \approx Q$. □

Now we will prove completeness of the axiomatization. First we present a result (whose proof is essentially like in [12]), and then we extend Lemma 4 for the case when one of the processes is not finite.

Lemma 6 Let $P \in \mathrm{PPA}_{rec}$. For any approximation P^n of P we have $\vdash P^n \sqsubseteq P$. □

Lemma 7 Let $P \in \mathrm{PPA}_{rec}$, and $Q \in \mathrm{PPA}_{fin}$. $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$ implies $P \sqsubseteq Q$.

Proof. Given that the finite approximations of P are a chain, such that $[\![P]\!]$ is its least upper bound, we have $[\![P^0]\!] \sqsubseteq_{\mathbf{PAT}} \cdots \sqsubseteq_{\mathbf{PAT}} [\![P^n]\!] \cdots \sqsubseteq_{\mathbf{PAT}} [\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$. Given that the processes P^n and Q are finite, we can apply the previous results for finite processes, concluding that for all n we have $P^n \sqsubseteq Q$, and applying (**R2**) we have $P \sqsubseteq Q$. □

Now, let us consider the case where P is finite but Q is not. Given that the usual way to assign semantics to recursive processes is by means of their finite approximations, the most straight way for proving $P \sqsubseteq Q$ would be to guarantee that there exists m such that the m-th finite approximation of the process Q fulfills $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q^m]\!]$. Then, given that P and Q^m are finite, we can apply Lemma 4, deducing $P \sqsubseteq Q^m$. Besides, we have $Q^m \sqsubseteq Q$ (Lemma 6), and so, applying (**O3**), we would obtain $P \sqsubseteq Q$. If finite processes were mapped into compact (also called finite) elements in the semantic domain, then the existence of such an m would be guaranteed, given that if R is a compact element and $R \sqsubseteq_{\mathbf{PAT}} \sqcup R^n$ then there exists R^i such that $R \sqsubseteq_{\mathbf{PAT}} R^i$, but unfortunately this is not the case, as the following example shows.

Example 5 Consider $P = recX.((a; Nil) \oplus_{\frac{1}{2}} X)$, and $Q = a; Nil$. It is easy to check that the finite approximations of P are given by $P^n = (a; Nil) \oplus_{1-\frac{1}{2^n}} \Omega$.

By definition we have $[\![P]\!] = \sqcup [\![P^n]\!]$, and so we trivially get $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} \sqcup [\![P^n]\!]$. Moreover, $[\![P]\!]$ *describes* a syntactic finite process, because $[\![P]\!] =_{\mathbf{PAT}} [\![Q]\!]$, and so, we should be able to conclude $P \equiv Q$. By the previous lemma we have $P \sqsubseteq Q$, but there does not exist m such that $[\![Q]\!] \sqsubseteq_{\mathbf{PAT}} [\![P^m]\!]$, otherwise we would have

$$1 = p([\![Q]\!], \epsilon, \{(a, 1)\}) \leq p([\![P^m]\!], \epsilon, \{(a, 1)\}) = 1 - \frac{1}{2^m}$$

which is not the case. So, we have found a finite (syntactic) process, $a; Nil$, which semantics is the least upper bound of the infinite nontrivial chain $\{[\![P^n]\!]\}_{n=1}^{\infty}$. □

The previous example shows that in general we must use another way in order to deduce $P \sqsubseteq Q$ from $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$. This is the reason why the rule (**R3**) was included in our logic system. This is an important difference with respect to [12] where finite processes are mapped into compact elements. Note that even if we delete probabilities, the previous example is not correct in the classical testing theory, given that Ω is a *zero* of $\oplus$ (Ω is also a zero of the external choice and parallel operators), and so the rule (**R3**) is not sound in that setting. Let us remark that the only compact element of the semantic domain is the one corresponding to divergence, given that for any process P different from Ω we can always construct a succession, for instance $P^n = P \oplus_{\frac{n}{n+1}} \Omega$, such that P is *lower* than the limit (actually $[\![P]\!] = \sqcup [\![P^n]\!]$) while for any n we have $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![P^n]\!]$ does not hold.

Lemma 8 Let $P \in \mathrm{PPA}_{fin}$ and $Q \in \mathrm{PPA}_{rec}$. $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$ implies $P \sqsubseteq Q$.

Proof. We have $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!] = \sqcup [\![Q^n]\!]$. If there exists m such that $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q^m]\!]$, then the proof can be done as previously indicated. So, let us suppose that there does not exist such an m. Given that $\{Q^n\}$ are a chain, for any sequence s and any state A, we have $p([\![P]\!], s, A) \leq p([\![Q]\!], s, A) = \lim_{n\to\infty} p([\![Q^n]\!], s, A)$.

Let us consider those sequences s and those states A such that $p([\![P]\!], s, A) > 0$. We have that for all $k > 0$, $(1 - \frac{1}{k}) \cdot p([\![P]\!], s, A) < \lim_{n\to\infty} p([\![Q^n]\!], s, A)$. Note that $(1 - \frac{1}{k}) \cdot p([\![P]\!], s, A) = p([\![P \oplus_{1-\frac{1}{k}} \Omega]\!], s, A)$.

Given that P is a finite process, the set of (s, A) pairs verifying $p([\![P]\!], s, A) > 0$ is finite. So, for each $k \in \mathbb{N}$ there exists $n_k \in \mathbb{N}$ such that for any sequence s and any state A, such that $p([\![P]\!], s, A) > 0$, we have $p([\![P \oplus_{1-\frac{1}{k}} \Omega]\!], s, A) \leq p([\![Q^{n_k}]\!], s, A)$. Obviously, if $p([\![P]\!], s', A') = 0$ then the previous result also holds, and so we have $[\![P \oplus_{1-\frac{1}{k}} \Omega]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q^{n_k}]\!]$ and given that $P \oplus_{1-\frac{1}{k}} \Omega$ and Q^{n_k} are finite processes, we can apply Lemma 4 obtaining $P \oplus_{1-\frac{1}{k}} \Omega \sqsubseteq Q^{n_k}$, for all $k \in \mathbb{N}$. Again, given that for all $n \in \mathbb{N}$, $Q^n \sqsubseteq Q$, we have for all $k \in \mathbb{N}$, $P \oplus_{1-\frac{1}{k}} \Omega \sqsubseteq Q$, and so, applying (**R3**), we conclude $P \sqsubseteq Q$. □

Theorem 6 Let P, Q be PPA_{rec} processes. $[\![P]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$ implies $P \sqsubseteq Q$.

Proof. If either P or Q is finite, then we apply Lemmas 4, 7, and 8. Otherwise, by the definition of (semantic) finite approximations we have that for all $n \in \mathbb{N}$, $[\![P^n]\!] \sqsubseteq_{\mathbf{PAT}} [\![P]\!]$, and given that $\sqsubseteq_{\mathbf{PAT}}$ is a preorder, we have $[\![P^n]\!] \sqsubseteq_{\mathbf{PAT}} [\![Q]\!]$. Now, by Lemma 8, we have $P^n \sqsubseteq Q$, and applying (**R2**) we conclude $P \sqsubseteq Q$. □

Again, by the previous result and the equivalence between $=_{\mathbf{PAT}}$ and $\approx$ we get

Theorem 7 (Completeness for PPA_{rec} **)**
For any processes $P, Q \in \mathrm{PPA}_{rec}$, we have $P \approx Q \implies \vdash P \equiv Q$. □

5 Extension of the System to the Parallel Operator

In this section we give some axioms for the parallel operator showing that it can be considered as a derived one in the sense that it can be completely eliminated

in finite processes (by transforming processes where the parallel operator appears into equivalent processes without occurrences of the parallel operator), and that the occurrences of the parallel operator in recursive processes can be *sunk* in such a way that there will be occurrences of the parallel operator but not in the *head* of the expression (that is, these processes can be transformed into *head normal form*).

These axioms indicate that the parallel operator is commutative and distributes over the internal choice. Moreover, we have an axiom indicating that the parallel operator is strict, and that it can be eliminated if both processes are *Nil*. We also have an *expansion* axiom similar to that in nonprobabilistic process algebras. Finally, we have a rule indicating that this operator is congruent. Let us comment that even though a parallel operator was not included in [19], and so its (denotational) semantic function is not included there, this function is not used in this paper given that soundness proofs of the following axioms are easy with respect to $\approx$ (the corresponding semantic function is given in [17]).

(**CP**) $P \parallel_A Q \equiv Q \parallel_A P$ (**DPIG**) $P \parallel_A (\bigoplus_{i=1}^{n} [p_i]\ P_i) \equiv \bigoplus_{i=1}^{n} [p_i]\ (P \parallel_A P_i)$

(**DP**) $P \parallel_A \Omega \equiv \Omega$ (**NP**) $Nil \parallel_A Nil \equiv Nil$

Let $A = \{a_1, \ldots, a_n\} \subseteq Act$ and $B = \{b_1, \ldots, b_m\} \subseteq Act$. If we consider the processes $P = \sum_{i=1}^{n} [p_i]\ a_i; P_i$ and $Q = \sum_{j=1}^{m} [q_j]\ b_j; Q_j$, then the following axiom is sound

$$(\mathbf{EP})\ P \parallel_X Q \equiv R$$

where $R = \sum_{k=1}^{l} [\frac{r_k}{\mu(P,Q,X)}]\ c_k; R_k$, $C = \{c_1, \ldots, c_l\} = (A \cup B) - X \cup (A \cap B \cap X)$, and

$$r_k = \begin{cases} p_i \cdot q_j & \text{if } c_k = a_i = b_j \in X \\ p_i & \text{if } c_k = a_i \in (A - B) - X \\ q_j & \text{if } c_k = b_j \in (B - A) - X \\ p_i + q_j & \text{if } c_k = a_i = b_j \in (A \cap B) - X \end{cases}$$

$$R_k = \begin{cases} P_i \parallel_X Q_j & \text{if } c_k = a_i = b_j \in X \\ P_i \parallel_X Q & \text{if } c_k = a_i \in (A - B) - X \\ P \parallel_X Q_j & \text{if } c_k = b_j \in (B - A) - X \\ (P_i \parallel_X Q) \oplus_{p'} (P \parallel_X Q_j) & \text{if } c_k = a_i = b_j \in (A \cap B) - X \ \wedge\ p' = \frac{p_i}{p_i + q_j} \end{cases}$$

Note that this last axiom can be applied to the process *Nil* (i.e. empty generalized external choices), so that the combination of (**NP**), (**DP**), and (**EP**) will allow to remove trailing occurrences of the parallel operator.

Finally, we have that $\equiv$ is a congruence for the parallel operator, that is, we have the following rule:

$$(\mathbf{C4})\ \frac{P \equiv P'}{P \parallel_A Q \equiv P' \parallel_A Q}$$

6 Conclusions

We have presented a sound and complete axiomatization of probabilistic testing with generative probabilities. The rules and axioms have been presented in three steps: first, we studied finite processes without parallel composition, then, we extended the previous axiomatization to deal with recursively defined processes, and finally, we gave sound axioms which indicate that the parallel operator can be considered as a derived one from the rest of operators.

A possible extension of our work would be to include some kind of *hiding* or *restriction* operator. Our results in [17] show that such inclusion is far from easy if we want to have this operator as a derived one. Specifically, consider the following processes

$$P = (a +_p c; b)\backslash c \qquad Q = b \oplus_{p'} \left(\bigoplus_{i=1}^{n} [q_i](a +_{p_i} b) \right)$$

If we make an interpretation of hiding *a la* CCS, that is, considering that P behaves like $a +_p \tau; b$, and we use a suitable definition of testing (e.g. [5]) we have that given p, in general, there do not exist values $0 < p', q_1 \ldots q_n, p_1 \ldots p_n < 1$ such that P and Q are probabilistic testing equivalent. We have an additional result. After a complicated redefinition of the parallel operator (using *prenormalization* factors) we got that P could be equivalent to the process $(a +_1 b) \oplus_p b$, where $+_1$ indicates a priority operator similar to that in [16]. We worked out a definition of the new semantic model but it was so unmanageable that we decided not to include priorities in our framework.

Acknowledgments: I would like to thank my advisor, David de Frutos, for his advice and encouragement. I also would like to thank Scott Smolka for many valuable comments as member of my PhD thesis committee which have improved the quality of this paper.

References

1. J.C.M. Baeten, J.A. Bergstra, and S.A. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities. *Information and Computation*, 121(2):234–255, 1995.
2. C. Baier and H. Hermanns. Weak bisimulation for fully probabilistic processes. In *Computer Aided Verification'97, LNCS 1254*, pages 119–130. Springer, 1997.
3. I. Christoff. Testing equivalences and fully abstract models for probabilistic processes. In *CONCUR'90, LNCS 458*, pages 126–140. Springer, 1990.
4. R. Cleaveland, I. Lee, P. Lewis, and S.A. Smolka. A theory of testing for soft real-time processes. In *8th International Conference on Software Engineering and Knowledge Engineering*, 1996.
5. R. Cleaveland, S.A. Smolka, and A.E. Zwarico. Testing preorders for probabilistic processes. In *19th ICALP, LNCS 623*, pages 708–719. Springer, 1992.
6. F. Cuartero, D. de Frutos, and V. Valero. A sound and complete proof system for probabilistic processes. In *4th International AMAST Workshop on Real-Time Systems, Concurrent and Distributed Software, LNCS 1231*, pages 340–352. Springer, 1997.

7. P. R. D'Argenio, H. Hermanns, and J.-P. Katoen. On generative parallel composition. In *Workshop on Probabilistic Methods in Verification, PROBMIV'98*, pages 105–121, 1998.
8. R. de Nicola and M.C.B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.
9. A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of Working Conference on Programming Concepts and Methods, IFIP TC 2*. North Holland, 1990.
10. C. Gregorio, L. Llana, M. Núñez, and P. Palao. Testing semantics for a probabilistic-timed process algebra. In *4th International AMAST Workshop on Real-Time Systems, Concurrent, and Distributed Software, LNCS 1231*, pages 353–367. Springer, 1997.
11. C. Gregorio and M. Núñez. Denotational semantics for probabilistic refusal testing. In *Workshop on Probabilistic Methods in Verification, PROBMIV'98*, pages 123–137, 1998.
12. M. Hennessy. *Algebraic Theory of Processes*. MIT Press, 1988.
13. C.-C. Jou and S.A. Smolka. Equivalences, congruences and complete axiomatizations for probabilistic processes. In *CONCUR'90, LNCS 458*, pages 367–383. Springer, 1990.
14. K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.
15. K.G. Larsen and A. Skou. Compositional verification of probabilistic processes. In *CONCUR'92, LNCS 630*, pages 456–471. Springer, 1992.
16. G. Lowe. Probabilistic and prioritized models of timed CSP. *Theoretical Computer Science*, 138:315–352, 1995.
17. M. Núñez. *Semánticas de Pruebas para Álgebras de Procesos Probabilísticos*. PhD thesis, Universidad Complutense de Madrid, 1996.
18. M. Núñez and D. de Frutos. Testing semantics for probabilistic LOTOS. In *Formal Description Techniques VIII*, pages 365–380. Chapman & Hall, 1995.
19. M. Núñez, D. de Frutos, and L. Llana. Acceptance trees for probabilistic processes. In *CONCUR'95, LNCS 962*, pages 249–263. Springer, 1995.
20. E.W. Stark and S.A. Smolka. A complete axiom system for finite-state probabilistic processes, 1996.
21. C. Tofts. A synchronous calculus of relative frequency. In *CONCUR'90, LNCS 458*, pages 467–480. Springer, 1990.
22. R. van Glabbeek, S.A. Smolka, and B. Steffen. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, 1995.
23. W. Yi and K.G. Larsen. Testing probabilistic and nondeterministic processes. In *Protocol Specification, Testing and Verification XII*, pages 47–61. North Holland, 1992.
24. S. Yuen, R. Cleaveland, Z. Dayar, and S.A. Smolka. Fully abstract characterizations of testing preorders for probabilistic processes. In *CONCUR'94, LNCS 836*, pages 497–512. Springer, 1994.