Acceptance Trees for Probabilistic Processes*

Manuel Núñez, David de Frutos, and Luis Llana

Dept. de Informática y Automática
Facultad CC. Matemáticas
Universidad Complutense de Madrid, Spain
e-mail: {manuelnu,defrutos,llana}@eucmvx.sim.ucm.es

Abstract. In this paper we study the extension of classical testing theory to a probabilistic process algebra. We consider a *generative* interpretation of probabilities for a language with two choice operators (one internal and the other external), which are annotated with a probability $p \in (0,1)$. We define a testing semantics for our language, and we write $P\, pass_p\, T$ to denote that the process P *passes* the test T with a probability p. We also give a set of *essential* tests which has the same *strength* as the full family of tests. Next we give an alternative characterization of the testing semantics, based on the idea of *acceptance sets*, and we prove that the new equivalence is equal to the testing equivalence. Finally, we present a fully abstract denotational semantics based on *acceptance trees*.

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.

[GJS90] presents a probabilistic version of SCCS called PCCS where the sum operator is extended with probabilities (i.e. $\sum_{i \in I} [p_i] E_i$, $p_i \in (0,1]$ and $\sum p_i = 1$). [vGSST90] discusses the *reactive, generative*, and *stratified* models for PCCS. According to the interaction between the processes and the environment, the *reactive* model is defined as the model where the environment may only offer a single action at a time. In the *generative* model, the environment can offer several actions at the same time, and the process chooses between them according to some probability distribution. In the *stratified* model, the branching structure of the probabilistic choices is captured.

In [HJ90], a probabilistic and timed extension of CCS is studied. They have two choice operators: A probabilistic internal one and a (nonprobabilistic) external nondeterministic one. They define *alternating processes* which have a (strict) alternation between probabilistic and nondeterministic states. In [YL92], CCS is extended with a nondeterministic probabilistic choice operator. A new notion of

* Research partially supported by the CICYT project TIC 94-0851-C02-02

testing semantics is defined, where a process *can pass* a test with a set of probabilities. A process *must pass* a test if the minimum of the set of probabilities is equal to one, and *may pass* the test if the maximum is greater than zero. In [JY95] denotational characterizations of the induced testing preorders are given.

[Cua93] presents a probabilistic (reactive) version of CSP whose testing semantics is studied. He also gives a fully abstract denotational semantics. Also, [Sei92, Low95] develop probabilistic extensions of CSP, but their semantic frameworks are very far from our own.

In the framework of probabilistic labeled transition systems (*plts*), [Chr90] presents three equivalences based on testing. [CSZ92] presents a testing semantics for *plts*; as in the nonprobabilistic case, processes and tests are essentially the same, while in [Chr90] or in [Cua93] they were different. [YCDS94] presents an alternative characterization of the testing preorder given in [CSZ92], introducing a set of *essential* tests, called *probabilistic traces*, which are proved to have the same *strength* as the full family of tests.

Finally, [WSS94] studies probabilistic processes modeled by probabilistic I/O automata. Even though this syntactic framework is not too close to our approach, several of their results are rather thorough and it seems interesting to translate them to the field of probabilistic process algebras.

In this paper, we focus on the *generative* model, and we introduce a probabilistic extension of classical testing theory, as proposed in [Hen88]. In order to concentrate on the characteristics of the probabilistic extension, we restrict ourselves to the study of a subset of the language in [Hen88]. We consider the language built from *Nil*, Ω, prefix, choice (internal and external) and recursion, where the choice operators are extended with a probability. The main features of our work are:

1. In our language, as in [Hen88], we have two different choice operators that are extended with a probability.
2. We define a testing equivalence in the standard way, where tests are identical to processes but extended with a new action ω which denotes that the test has succeeded. Thus, we have in particular that several actions can be offered at the same time. This is why we have found natural to consider a *generative* interpretation of probabilities, although a *stratified* interpretation could be considered.
3. We present an alternative characterization of the testing equivalence based on acceptance sets [Hen88].
4. We develop a denotational semantics based on acceptance trees [Hen85], which is proved to be fully abstract with respect to the testing semantics.

The rest of this paper is structured as follows. Section 2 presents our language, its operational semantics, the associated testing semantics, and a set of *essential* tests. In Section 3, we describe the probabilistic extension of *acceptance sets*. In Section 4, we prove that the equivalences defined in the previous two sections are the same. Section 5 presents a denotational semantics which is fully abstract with respect to the testing semantics, while in Section 6 we give our conclusions and some outlines for future work.

2 A Testing Semantics

As we said in the Introduction, we will extend with probabilities the operators of a subset of the language presented in [Hen88] (this language first appeared in [dNH87]). In this paper we consider the simple language without parallel composition or restriction, and we call this language PPA (Probabilistic Process Algebra). We have two probabilistic choice operators: an internal choice operator, and an external one. Extremal values (0 and 1) of probabilities are not allowed in these two choice operators, because they cannot be simply treated in an adequate way within the generative model (see [SS90] where extremal values are used to capture priorities, but in a stratified model).

2.1 Syntax of PPA

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

$$P ::= Nil \mid \Omega \mid X \mid a;P \mid P \oplus_p P \mid P +_p P \mid recX.P$$

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

Now, we give an intuitive explanation of each operator:

- Nil is a process that cannot perform any action. Ω is a divergent process.
- $a;P$ is a process that first performs action a, and then behaves as P.
- $P \oplus_p Q$ is a process that behaves as P with probability equal to p, and as Q with probability equal to $1-p$. This choice is made internally nondeterministically, i.e. without interaction with the environment.
- $P +_p Q$ is a process that can behave either as P or as Q, but this choice depends on the environment. This means that if the environment offers actions that only one of the processes P or Q can perform, the corresponding process will be chosen. On the other hand, if both processes can perform actions from those offered by the environment, then the choice is made partially nondeterministically using the probability p.
- $recX.P$ is used to define recursive behaviors.

2.2 Operational Semantics

The set of rules that define the operational semantics is given in Fig. 1. There are two types of transitions. The intuitive meaning of a probabilistic transition $P \stackrel{a}{\longrightarrow}_p Q$ is that if the environment offers all the actions in Act, then the probability with which P performs a and then behaves as Q is equal to p; the meaning of $P \succ\!\!\longrightarrow_p Q$ is that the process P evolves to Q with probability p, without interaction with the environment.

For the sake of simplicity, we have omitted indices in the definition of transitions. Usually (see [GJS90]) indices are used to distinguish among different occurrences of the same transition; instead we have supposed that if a transition can be derived in several ways, each derivation generates a different instance of this transition (this could be formalized using multisets of transitions).

$$(PRE)\frac{}{a;P \xrightarrow{a}_1 P} \qquad (INT1)\frac{}{P \oplus_p Q \rightarrowtail_p P} \qquad (INT2)\frac{}{P \oplus_p Q \rightarrowtail_{1-p} Q}$$

$$(EXT1)\frac{P \rightarrowtail_q P' \wedge\ Q_\oplus = 0}{P +_p Q \rightarrowtail_q P' +_p Q} \qquad (EXT2)\frac{Q \rightarrowtail_q Q' \wedge\ P_\oplus = 0}{P +_p Q \rightarrowtail_q P +_p Q'} \qquad (EXT3)\frac{P \rightarrowtail_{q_1} P' \wedge\ Q \rightarrowtail_{q_2} Q'}{P +_p Q \rightarrowtail_{q_1 \cdot q_2} P' +_p Q'}$$

$$(EXT4)\frac{P \xrightarrow{a}_q P' \wedge\ Q_\oplus = 0}{P +_p Q \xrightarrow{a}_{p \cdot \hat{q}} P'} \qquad (EXT5)\frac{Q \xrightarrow{a}_q Q' \wedge\ P_\oplus = 0}{P +_p Q \xrightarrow{a}_{(1-p)\cdot \hat{q}} Q'}$$

$$(REC)\frac{}{rec\ X.P \rightarrowtail_1 P\{rec\ X.P/X\}} \qquad (DIV)\frac{}{\Omega \rightarrowtail_1 \Omega}$$

$$R_\oplus = \sum_{R'} \{\!\{\, s \mid R \rightarrowtail_s R' \,\}\!\},\ R_+ = \sum_{R'} \{\!\{\, s \mid \exists a : R \xrightarrow{a}_s R' \,\}\!\},\ \hat{q} = \frac{q}{p \cdot P_+ + (1-p)\cdot Q_+}$$

Fig. 1. Operational Semantics of PPA.

Example 1. Let $P = a; Nil +_{\frac{1}{2}} a; Nil$. Then, this process has two times the transition $P \xrightarrow{a}_{\frac{1}{2}} Nil$. Also, $Q = Q' \oplus_{\frac{1}{2}} Q'$ has two times the transition $Q \rightarrowtail_{\frac{1}{2}} Q'$.

While the rules for prefix, internal choice, divergence and recursion do not need any explanation, we will briefly explain the rules for the external nondeterministic operator. The first three rules say that whenever any of the arguments of an external choice can evolve via an internal transition, this kind of transitions are performed until both arguments reach a stable state (i.e. no more internal transitions are possible). Let us remark that $+_p$ is not a static operator; these three rules simply indicate that internal choices do not *solve* external choices.

The other two rules are applied when none of the processes can perform more internal actions and (at least) one of the arguments may evolve by executing an external action. The value $\hat{q}$ is obtained from the probability q with which this external transition is executed, taking into account if both processes can perform external transitions. Whenever both of them can perform external actions, the value p (the value associated with the external choice) is considered.

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

Lemma 2. *Let P be a process.* $\exists p, P' :\ P \rightarrowtail_p P' \implies \nexists q, a, P'' :\ P \xrightarrow{a}_q P''$.

Note that we have not the converse; for example, *Nil* performs no transitions (neither internal nor external). Also note that the statement in the previous lemma is equivalent to $\exists q, a, P'' : P \xrightarrow{a}_q P'' \Longrightarrow \nexists p, P' : P \rightarrowtail_p P'$.

Corollary 3. *Let P be a process. Then, the following holds*

$$(P_\oplus = 0 \vee P_\oplus = 1) \wedge (P_+ = 0 \vee P_+ = 1) \wedge (P_\oplus \neq 1 \vee P_+ \neq 1)$$

2.3 Probabilistic Tests

Once the operational semantics has been defined, in order to define a testing semantics we must define the concept of (*probabilistic*) *test*. As in the nonprobabilistic case, tests will be just processes where the alphabet *Act* is extended with a new action ω which indicates successful termination. The operational semantics of tests is the same as that of processes (considering ω as an ordinary

$$\frac{P \rightarrowtail_p P' \wedge T_\oplus = 0}{P \mid T \longmapsto_p P' \mid T} \qquad \frac{T \rightarrowtail_p T' \wedge P_\oplus = 0}{P \mid T \longmapsto_p P \mid T'} \qquad \frac{P \rightarrowtail_p P' \wedge T \rightarrowtail_q T'}{P \mid T \longmapsto_{p \cdot q} P' \mid T'}$$

$$\frac{P \xrightarrow{a}_p P' \wedge T \xrightarrow{a}_q T'}{P \mid T \longmapsto_{r_1} P' \mid T'} \qquad \frac{T \xrightarrow{\omega}_p T' \wedge P_\oplus = 0}{P \mid T \xrightarrow{\omega}_{r_2} Nil}$$

where $r_1 = \frac{p \cdot q}{\mu(P,T)}$ and $r_2 = \frac{p}{\mu(P,T)}$

Fig. 2. Rules for the parallel composition.

action). Now, we have to define how a process interacts with a test. As usual, this interaction is modeled by parallel composition of the process with the test. The rules which define this parallel composition are given in Fig. 2.

Let us briefly explain these rules. The first three rules deal with internal transitions and say that if one and only one of the processes can perform an internal transition, this transition is performed (first two rules). If both can perform an internal transition, then both perform their transitions at the *same time* (third rule). This does not imply that our calculus is synchronous. Since the two arguments of the parallel composition must solve their internal choices before evolving by an external transition, there is no problem in forcing the simultaneous resolution of the internal choices in both arguments.

If none of the processes can perform internal transitions, then the last two rules are applied. The first rule deals with *synchronization*. The second one deals with the special action ω. After this action is performed, no more actions can be performed by the composition (success of test). Note that in this rule, we also have a clause $P_\oplus = 0$, so that a divergent process (that is a process which has no finite maximal computation) would *not pass* any test.

In these last two rules, we use a *normalization factor* $\mu(P,T)$. This normalization factor is similar to that in [CSZ92] (considering that this factor is only used when just visible actions may be performed), and it is defined by

$$\mu(P,T) = \sum_a \{\!|\, p \cdot q | \exists P', T' :\ P \xrightarrow{a}_p P' \wedge T \xrightarrow{a}_q T' \,|\!\} + \sum \{\!|\, p | \exists T' :\ T \xrightarrow{\omega}_p T' \,|\!\}$$

Definition 4. Let P be a process and T a test. A *computation* is a *maximal* sequence of transitions $C = P \mid T \longmapsto_{p_1} P_1 \mid T_1 \longmapsto_{p_2} \cdots P_{n-1} \mid T_{n-1} \overset{*}{\longmapsto}_{p_n} R$, where $*$ denotes either an empty label or the special action ω. A sequence is said to be *maximal* when there do not exist $p > 0, R'$ such that $R \overset{*}{\longmapsto}_p R'$.

When the last transition is of the form $P_{n-1} \mid T_{n-1} \xrightarrow{\omega}_{p_n} Nil$, we say that the computation is *successful*. We denote by $\tilde{C}$ the *set* of *successful computations* from C. The probability of a successful computation S, $Pr(S)$, is inductively defined as

$$Pr(Nil) = 1 \text{ (i.e. } T \text{ has succeeded)}$$
$$Pr(P \mid T \overset{*}{\longmapsto}_p C) = p \cdot Pr(C)$$

We write $P\, pass_p\, T$ if $\sum_{S \in \tilde{C}} Pr(S) = p$. □

Now, given a family of tests we can define the corresponding notion of testing equivalence with respect to this family.

Definition 5. Given a set of probabilistic tests $\mathcal{T}$, and two processes P and Q, we say $P \approx_{\mathcal{T}} Q$ iff $\forall\, T \in \mathcal{T} :\ P\ pass_p\ T \wedge\ Q\ pass_q\ T \Longrightarrow p = q$. If the family of tests is the whole set of tests, we just write $P \approx Q$. □

2.4 A set of *essential* tests

In this section, we construct a family of tests $\mathcal{PB}$, which is equivalent to the whole family of tests, in the sense that $P \approx_{\mathcal{PB}} P' \Longleftrightarrow P \approx P'$. This family of tests will be close to the notion of *probabilistic traces* given in [YCDS94]. First, we need the following

Lemma 6. *Let P be a process, T, T' be tests, and $a \in Act$. Then,*

1. $P\ pass_q\ (T \oplus_p T') \Longleftrightarrow P\ pass_{q_1}\ T \wedge P\ pass_{q_2}\ T' \wedge p \cdot q_1 + (1-p) \cdot q_2 = q$
2. $P\ pass_q\ (a;T) +_p (a;T') \Longleftrightarrow P\ pass_q\ a;(T \oplus_p T')$

Now we will generalize the choice operators to n arguments. Note that in this definition we do not allow nondeterminism caused by prefixing two arguments of a generalized external choice by the same action.

Definition 7. 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 in the generalized internal choice, the sum of probabilities may be less than 1. The difference between 1 and this value indicates the probability of divergence. For example, $\bigoplus_{i=1}^{2} ([\frac{1}{3}]\ P_1)\ ([\frac{1}{3}]\ P_2) = (P_1 \oplus_{\frac{1}{2}} P_2) \oplus_{\frac{2}{3}} \Omega$.

Taking into account that internal choices can be removed from tests and that only deterministic tests are needed (by Lemma 6), that infinite tests are redundant, and following [YCDS94], we obtain the following family of tests.

Definition 8. The set of *probabilistic barbed tests*, $\mathcal{PB}$, is defined by the BNF expression

$$T ::= \omega; Nil \mid \sum_{i=1}^{n} [p_i]\, a_i; R_i \qquad \text{where} \quad R_i = \begin{cases} Nil & \text{if } 1 \le i \le n-1 \\ T & \text{if } i = n \end{cases}$$

$p_i > 0$, $\sum p_i = 1$, $a_i \in Act$, and $a_i \neq a_j$ for $i \neq j$ □

The following theorem is a simple adaptation of Theorem 2 in [YCDS94], although its proof requires more care, because now divergent processes are allowed.

Theorem 9. $P \approx_{\mathcal{PB}} Q \iff P \approx Q$.

3 Probabilistic Acceptance Sets

In this section we present an alternative characterization of the testing equivalence defined in the previous section. We will use a version of *acceptance sets* [Hen88] introducing probabilities in order to capture the information given by probabilistic tests. The main changes in the definitions given in [Hen88], in order to adapt them to the probabilistic case, are the following:

- States are not sets of actions but sets of pairs (action, probability).
- In the nonprobabilistic case, acceptance sets are defined as the reachable states after a sequence of actions has been performed. In the probabilistic case, sequences of actions are not valid any more; instead we have to take sequences of pairs (state, action) where the action must belong to those contained in the state (see Example 2).
- The notion of equivalence must be modified taking into account the probabilistic information that appears in acceptance sets (see Example 3).

While the first change is clear, the other two need more explanation.

In the nonprobabilistic case, *states* (i.e. the different sets contained in the acceptance sets) can be just characterized by sequences of actions because the *continuations* of a process after an action is performed cannot be distinguished, and thus they must be joined into a common continuation. This is illustrated by the following

Example 2. Let P be the nonprobabilistic process $a; d \oplus ((a; b) + c)$, where Nil's have been omitted. We have that P is equivalent to $P' = a; (d \oplus b) \oplus ((a; (d \oplus b)) + c)$ with respect to the equivalence induced by acceptance sets. That is, continuations after a has been performed are joined.

But this does not happen to the probabilistic case. Let us consider $P = a; d \oplus_{\frac{1}{2}} ((a; b) +_{\frac{1}{2}} c)$, and suppose that there exist R_1, R_2 such that P is (testing) equivalent to $P' = a; R_1 \oplus_{\frac{1}{2}} ((a; R_1) +_{\frac{1}{2}} c; R_2)$. If we consider $T = (a; b; \omega) +_{\frac{1}{3}} c$, we have $P\, pass_{\frac{1}{6}}\, T$. Since $P \approx P'$, we obtain $\frac{1}{6} = \frac{1}{2} \cdot q + \frac{1}{2} \cdot \frac{1}{3} \cdot q$, where $R_1\, pass_q\, b; \omega$, and thus $q = \frac{1}{4}$ (i.e. $R_1\ pass_{\frac{1}{4}}\ b; \omega$). But, if we consider $T' = a; b; \omega$, using a similar argument we obtain $R_1\ pass_{\frac{1}{2}}\ b; \omega$, which is a contradiction.

In general we can distinguish two (or more) continuations of the same action if they correspond to different states, and so we must include states in the sequences defining probabilistic acceptance sets.

Example 3. Let P be the nonprobabilistic process $a \oplus b$, where *Nil* 's have been omitted. Then, $\mathcal{A}(P,\epsilon) = \{\{a\},\{b\}\}$, $\mathcal{A}(P,a) = \mathcal{A}(P,b) = \{\emptyset\}$ and if $s \neq \epsilon, a, b$ then $\mathcal{A}(P,s) = \emptyset$. This process is equivalent to $Q = (a \oplus b) \oplus (a + b)$.

But if we consider the probabilistic version of P, $P' = a \oplus_p b$, we have that P' has two reachable states: $\{a\}$ (with probability equal to p) and $\{b\}$ (with probability equal to $1-p$), while state $\{a,b\}$ is not reachable. But for any process $Q' = (a \oplus_p b) \oplus_q (a +_r b)$, the state $\{(a,r),(b,1-r)\}$ is reachable with probability equal to $(1-q) > 0$, and so there is no such Q' equivalent to P'.

Definition 10. Let $A \subseteq Act \times (0,1]$. We define the *multiset of actions* of A as $Act(A) = \{\!\!\{\, a \mid \exists p :\ (a,p) \in A \,\}\!\!\}$. We say that A is a (*probabilistic*) *state* if every $a \in Act$ appears at most one time in $Act(A)$ and either $\sum \{\!\!\{\, p \mid (a,p) \in A \,\}\!\!\}$ is equal to 0 (when $A = \emptyset$) or it is equal to 1. For each state A, we define the *probability* of a in A, denoted by $pro(a,A)$, as p if $(a,p) \in A$ and as 0 if $a \notin Act(A)$. □

Next we define the set of stable processes to which a process may evolve after performing a sequence of actions possibly interspersed with internal transitions.

Definition 11. Given a stable process P, we define its (immediately) *reachable state* as the set $S(P) = \{(a,p) \mid p = \sum_R \{\!\!\{\, p_i \mid P \xrightarrow{a}_{p_i} R \,\}\!\!\} \wedge p > 0\}$.

Given two processes P, P', where P' is stable, we say that P evolves to P' by a *generalized internal transition* with probability p if $P \rightarrowtail^*_p P'$ can be derived by applying the following rules:

$$P \rightarrowtail^*_1 P \text{ if } P \text{ is stable} \quad (\text{i.e. } \nexists q, Q :\ P \rightarrowtail_q Q)$$
$$P \rightarrowtail^*_p P' \text{ if } \exists q, q', Q :\ P \rightarrowtail_q Q \rightarrowtail^*_{q'} P' \wedge p = q \cdot q'$$

Finally, given $s = \langle A_1\, a_1, A_2\, a_2, \ldots, A_n\, a_n \rangle$, we define the relation $P \overset{s}{\Longrightarrow}_p P'$ by:

$P \overset{\epsilon}{\Longrightarrow}_p P'$ iff $P \rightarrowtail^*_p P'$

$P \overset{s}{\Longrightarrow}_p P'$ iff $\exists Q_1, P_1, p_1, q_1 : P \rightarrowtail^*_{p_1} Q_1 \xrightarrow{a_1}_{q_1} P_1 \overset{s'}{\Longrightarrow}_{p'} P' \wedge S(Q_1) = A_1 \wedge p = \frac{p' \cdot p_1 \cdot q_1}{r_1}$

where A_i is a state, $a_i \in Act(A_i)$, $s' = \langle A_2\, a_2, \ldots, A_n\, a_n \rangle$, $r_1 = pro(a_1, A_1)$. □

As in the case of $\rightarrowtail_p$ and $\xrightarrow{a}_p$ we must take care of repetitions when generating both $\rightarrowtail^*_p$ and $\overset{s}{\Longrightarrow}_q$. Note that $P \rightarrowtail^*_p P'$ iff P' is stable and P can evolve to P' by a finite sequence of internal transitions. Also note that $P \overset{s}{\Longrightarrow}_p P'$ iff P performs the sequence s, i.e. it performs the actions a_i from Q_i such that $S(Q_i) = A_i$, and then evolves to P' by a generalized internal transition. The value p is the *global* probability of the derived computation. It is obtained by multiplying the probabilities of the selected branches from all the nondeterministic choices solved along the computation. It is important to note that besides the syntactic internal choices which generate the generalized internal transitions, we must take care of the nondeterminism induced by the different observable transitions labeled by the same action leaving an stable state.

Definition 12. Let P be a (probabilistic) process and s be a sequence of pairs (state, action). We define the (*probabilistic*) *acceptance sets* of P *after* s as

$$\mathcal{A}(P,s) = \{(A, p_A/q_s) \mid p_A = \textstyle\sum_{P'} \{\!\!\{\, p_i \mid P \overset{s}{\Longrightarrow}_{p_i} P' \ \wedge S(P') = A \,\}\!\!\} \ \wedge p_A > 0\}$$

where $q_\epsilon = 1$ and $q_{s' \circ \langle B\, b \rangle} = \sum_{Q'} \{\!\!\{\, q_i \mid P \overset{s'}{\Longrightarrow}_{q_i} Q' \ \wedge S(Q') = B \,\}\!\!\}$ □

In order to calculate the probabilistic acceptance sets of P after a sequence s, first we calculate the reachable states by P after the sequence s, and then we sum the probabilities of reaching these states divided by the total probability of reaching the last state in s. Note that $\mathcal{A}(\Omega, s) = \emptyset$ for any s (even if $s = \epsilon$), since $\Omega \stackrel{s}{\Longrightarrow}_p P$ for no P, $p > 0$.

Definition 13. (**Alternative Characterization**)
Let P, P' be processes. We write $P \cong P'$ if for all $s = \langle A_1\, a_1, A_2\, a_2, \ldots, A_n\, a_n \rangle$ (n can be equal to zero) we have $(A, p) \in \mathcal{A}(P, s) \Longleftrightarrow (A, p) \in \mathcal{A}(P', s)$. □

Example 4. Let $P = (a +_{\frac{1}{3}} b) \oplus_{\frac{1}{4}} (b; (c \oplus_{\frac{1}{3}} d))$, where *Nil* 's have been omitted.

$\mathcal{A}(P, \epsilon) = \{\ (A, \frac{1}{4}), (B, \frac{3}{4})\ \}$, where $A = \{(a, \frac{1}{3}), (b, \frac{2}{3})\}$ and $B = \{(b, 1)\}$
$\mathcal{A}(P, \langle A\, b \rangle) = \mathcal{A}(P, \langle A\, a \rangle) = \{\ (\emptyset, 1)\ \}$
$\mathcal{A}(P, \langle B\, b \rangle) = \{\ (\{(c, 1)\}, \frac{1}{3}), (\{(d, 1)\}, \frac{2}{3})\ \}$
...

Thus, for instance we have $P' = ((a +_{\frac{1}{3}} b) \oplus_{\frac{1}{2}} (b; c)) \oplus_{\frac{1}{2}} (b; d) \cong P$.

4 Characterization Theorem

Now, our main task is to prove that the two equivalence relations given by Defs. 5 and 13 are equivalent. In order to do it, we associate to each process its *computations tree* (not to be confused with *computations* in Definition 4). These trees are built by alternating *internal* and *external* states. Generalized internal transitions leave from internal states reaching external states, while observable transitions leave from external states reaching internal states. By associating the generalized internal choice to internal states, and the generalized external choice to external states, these trees can be seen as (possibly infinite) syntactic processes in normal form. By extending, in a trivial way, the operational semantics to this kind of processes, we can also define testing semantics for them.

Definition 14. Let P be a process. We define the *derived* (generalized) *process associated to* P, denoted by $\widehat{\mathcal{A}}(P)$, as $\widehat{\mathcal{A}}(P) = \widehat{\mathcal{A}}(P, \epsilon)$ where

$$\widehat{\mathcal{A}}(P, s) = \bigoplus_{i=1}^{n} [p_i] \sum_{j=1}^{r_i} [p_{i,j}]\, a_{i,j}\, ; \widehat{\mathcal{A}}(P, s \circ \langle A_i\, a_{i,j} \rangle)$$

where $\sum_{j=1}^{0} P_i$ just denotes *Nil*, $\mathcal{A}(P, s) = \{(A_1, p_1), (A_2, p_2), \ldots, (A_n, p_n)\}$ and $A_i = \{(a_{i,1}, p_{i,1}), (a_{i,2}, p_{i,2}), \ldots, (a_{i,r_i}, p_{i,r_i})\}$. □

Lemma 15. $P \cong P' \Leftrightarrow \widehat{\mathcal{A}}(P) = \widehat{\mathcal{A}}(P')$.

Note the close relation between $\widehat{\mathcal{A}}(P)$ and the probabilistic acceptance sets of the form $\mathcal{A}(P, s)$. So, $\widehat{\mathcal{A}}(P)$ can be seen as the (*pseudo*) *normal form* of P.

Example 5. The process P in Example 4 is in normal form, that is $P = \widehat{\mathcal{A}}(P)$. The process P' in Example 4 is not in normal form, but $\widehat{\mathcal{A}}(P') = P$.

Theorem 16. *For each process P we have $P \approx \widehat{\mathcal{A}}(P)$.*

Proof. (Sketch) The proof is easy but cumbersome, noting that computations of $\widehat{\mathcal{A}}(P)$ are just computations of P where each sequence of internal transitions of P has become a generalized internal transition of P, and thus an internal transition of $\widehat{\mathcal{A}}(P)$. □

We have to prove that the notions of testing and acceptance sets equivalence are *equivalent*. By Lemma 15, it is enough to consider processes of the form $\widehat{\mathcal{A}}(P)$. By Theorem 9 we can restrict the whole set of tests to probabilistic barbed tests. First, we need the uniqueness result given in Lemma 17. This result is similar to that of Lemma 9 in [WSS94].

Lemma 17. *Suppose f and f' are two functions of $n \geq 0$ variables $x_1, x_2 \ldots x_n$, defined as follows:*

$$f = \sum_{i \in I} \frac{c_i}{1 + \sum_{j=1}^{n} d_{j,i} \cdot x_j} \qquad f' = \sum_{i' \in I'} \frac{c'_{i'}}{1 + \sum_{j=1}^{n} d'_{j,i'} \cdot x_j}$$

where I, I' are finite sets of indices, $c_i, c'_{i'} > 0$, and for each $r, s \in I$, such that $r \neq s$, the tuples $(d_{1,r}, d_{2,r}, \ldots, d_{r,r})$ and $(d_{1,s}, d_{2,s}, \ldots, d_{r,s})$ are distinct. If $f = f'$, then there exists a bijection $h : I \longrightarrow I'$ such that $d_{j,i} = d'_{j,h(i)}$ and $c_i = c'_{h(i)}$ for all $i \in I$ and $1 \leq j \leq n$.

Theorem 18. $\widehat{\mathcal{A}}(P) \approx_{\mathcal{PB}} \widehat{\mathcal{A}}(P') \iff \widehat{\mathcal{A}}(P) = \widehat{\mathcal{A}}(P')$.

Proof. (Sketch) The right to left implication is trivial. For the left to right implication, let $Act = \{a_1, \ldots, a_n\}$ (note that Act is finite). Then

$$\widehat{\mathcal{A}}(P) = \bigoplus_{i=1}^{m} [p_i] \sum_{j=1}^{n} [p_{i,j}]\, a_j\, ; C_{i,k} \qquad \text{and} \qquad \widehat{\mathcal{A}}(P') = \bigoplus_{i'=1}^{m'} [p'_{i'}] \sum_{j=1}^{n} [p'_{i',j}]\, a_j\, ; C'_{i',k}$$

where to simplify the notation we take $p_{i,j} = 0$ if a_j did not appear in the i-th summand of $\widehat{\mathcal{A}}(P)$. For each probability distribution $\bar{q} = \langle q_1, q_2, \ldots q_n \rangle$ ($q_i \geq 0$, $\sum q_i = 1$), and for each $1 \leq k \leq n$, such that $q_k \neq 0$, we consider the barbed tests

$$T_k^{\bar{q}} = \sum_{j=1}^{n} [q_j]\; a_j\, ; T_{j,k} \qquad \text{where} \qquad T_{j,k} = \begin{cases} \omega & , \text{ if } k = j \\ Nil & , \text{ if } k \neq j \end{cases}$$

Composing the processes and the tests and using $\widehat{\mathcal{A}}(P) \approx_{\mathcal{PB}} \widehat{\mathcal{A}}(P')$, we have

$$p = \sum_{i=1}^{m} p_i \cdot \frac{p_{i,k} \cdot q_k}{\sum_{j=1}^{n} p_{i,j} \cdot q_j} = \sum_{i'=1}^{m'} p'_{i'} \cdot \frac{p'_{i',k} \cdot q_k}{\sum_{j=1}^{n} p'_{i',j} \cdot q_j}$$

where $\widehat{\mathcal{A}}(P)\ pass_p\ T_k^{\bar{q}}$ (and thus $\widehat{\mathcal{A}}(P')\ pass_p\ T_k^{\bar{q}}$). Whenever $p_{i,k} = 0$ we also take $(p_{i,k} \cdot q_k) / \sum p_{i,j} \cdot q_j = 0$. After some algebraic manipulations, we can apply Lemma 17 to conclude $\mathcal{A}(\widehat{\mathcal{A}}(P), \epsilon) = \mathcal{A}(\widehat{\mathcal{A}}(P'), \epsilon)$. Taking deeper barbed tests, and using a similar argument, we can also prove $\mathcal{A}(\widehat{\mathcal{A}}(P), s) = \mathcal{A}(\widehat{\mathcal{A}}(P'), s)$ for any arbitrary sequence s. □

Corollary 19. *Let P, P' be processes. Then $P \approx P' \iff P \cong P'$.*

5 A Fully Abstract Denotational Semantics

In this section we present an interpretation which is fully abstract with respect to the testing equivalence $\approx$. This interpretation will be based on *acceptance trees* [Hen85], and as in the nonprobabilistic case, the key to this definition is the alternative characterization given by $\cong$.

The semantic domain (denoted by $\mathbf{PAT}_{\mathrm{Act}}$) of *probabilistic acceptance trees over Act* (*pat*) is the set of rooted trees with two kinds of nodes: internal nodes (labeled with $\oplus$) and external nodes (labeled with $+$) that satisfy:

- The root is an internal node.
- Arcs outgoing from internal nodes are labeled with different states (see Def. 10) together with a probability. The sum of these probabilities must be less than or equal to 1. These arcs go to external nodes.
- Arcs outgoing from external nodes are labeled with the actions in the state labeling the ingoing arc. For any action in that state there must be an (unique) arc labeled with this action. These arcs go to internal nodes.

We usually will denote by $R, R_1, \ldots$ the elements of $\mathbf{PAT}_{\mathrm{Act}}$. Let us remark that it is possible that several outgoing arcs from an internal node are labeled with states which have the same set of actions but with different probabilities associated with the actions in the state. For example, a process may have the states $\{(a, \frac{1}{2}), (b, \frac{1}{2})\}$ and $\{(a, \frac{1}{3}), (b, \frac{2}{3})\}$. In internal nodes, the sum of probabilities associated with outgoing arcs can be less than one. The difference between this sum and 1 denotes the probability of divergence. Some examples of probabilistic acceptance trees are given in Fig. 3. We will characterize the nodes of these trees as the reachable nodes after a sequence of pairs (action,state).

Definition 20. Let R be a *pat* and A a state. We define the *probability* with which R (immediately) *reaches* the state A, denoted by $p(R, A)$, as p_A if there is an outgoing arc labeled $[p_A]\ A$ from the root node of R. If there is no such arc, then $p(R, A) = 0$.

Let A be a state, such that $p_A = p(R, A) > 0$, and $a \in act(A)$. We define the *continuation after the execution of a in A*, denoted by $R/(A, a)$, as the tree whose root is the internal node reached by the arc labeled with a originating in the external node reached by the arc labeled with $[p_A]\ A$.

Let $s = \langle A_1 a_1, A_2 a_2, \ldots, A_n a_n \rangle$ be a sequence such that $a_i \in act(A_i)$ and A_i are states. We define the *probability* with which R *reaches the external node represented by the sequence s and A*, denoted by $p(R, s, A)$, as

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

□

It is easy to see that from the values of $p(R, s, A)$ we can rebuild the tree R. We will use this fact in the rest of the section, in order to (implicitely) define certain *pat*'s.

Definition 21. Let R_1 and R_2 be *pat*'s. We write $R_1 \sqsubseteq R_2$ iff for any sequence s we have $p(R_1, s, A) \leq p(R_2, s, A)$. We write $R_1 \doteq R_2$ if $R_1 \sqsubseteq R_2$ and $R_2 \sqsubseteq R_1$. □

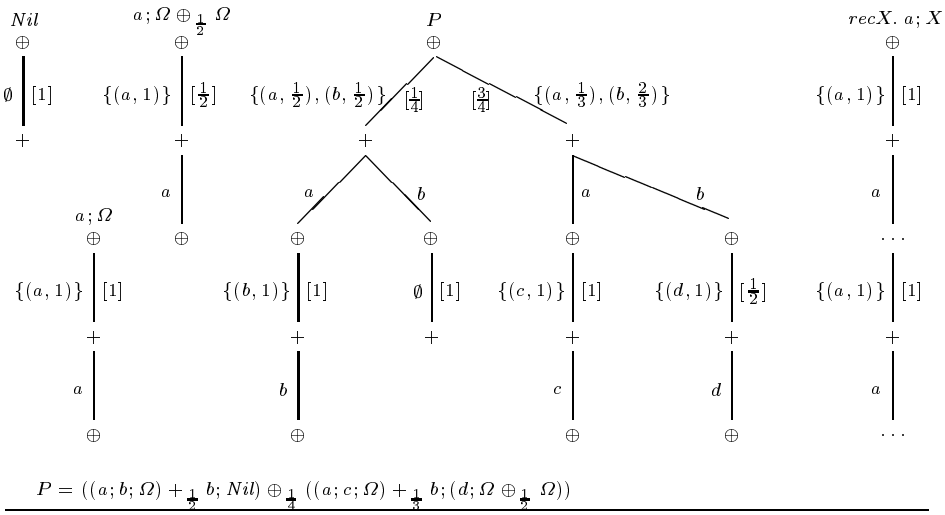

Fig. 3. Probabilistic acceptance trees.

Proposition 22. *The relation $\sqsubseteq$ is an order where the empty tree (the tree with no arcs) is the least element.*

Theorem 23. $(\mathbf{PAT}_{\mathrm{Act}}, \sqsubseteq)$ *is a complete partial order (cpo).*

Proof. (Sketch) Given a directed set $\{R_i\}$ of *pat*'s, it is not difficult to check, using basic notions of Real Analysis, that R defined by $p(R, s, A) = \sup p(R_i, s, A)$ is indeed a *pat* and that R is the *lub* of the set $\{R_i\}$. □

Once we have introduced the semantic domain for PPA processes, we define the semantic operators corresponding to the syntactic ones of the language. As usual, we denote by $[\![P]\!]$ the semantics of process P.

Nil and Ω

Nil has only an (immediately) reachable state: the state $\emptyset$ (reachable with probability equal to 1). Thus, $[\![Nil]\!]$ is a tree with an internal node, and under it, an external node.

$$p([\![Nil]\!], s, A) = \begin{cases} 1 & \text{if } s = \epsilon \ \wedge A = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Ω has no reachable states. Thus, $[\![\Omega]\!]$ is a tree with one internal node, with no node under it. That is, $p([\![\Omega]\!], s, A) = 0$ for any sequence s and any state A.

Prefix

For every $a \in Act$ we have a semantic function $a;\_ : \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$. The *pat* $a; R$ is the tree R preceded by an internal and an external node which *corresponds* to a. Thus, the definition of $a; R$ is:

$$p(a; R, s, A) = \begin{cases} 1 & \text{if } s = \epsilon \ \wedge A = \{(a,1)\} \\ p(R, s', A) & \text{if } s = \langle(\{(a,1)\}, a)\rangle \circ s' \\ 0 & \text{otherwise} \end{cases}$$

Internal Choice

For a probability p, $\oplus_p : \mathbf{PAT}_{\mathrm{Act}} \times \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ is a function which returns a tree which is the *union* of the trees corresponding to its arguments considering the probability p:

$$p(R_1 \oplus_p R_2, s, A) = p \cdot p(R_1, s, A) + (1-p) \cdot p(R_2, s, A)$$

External Choice

Before defining the functions $+_p$, we will define an auxiliary operator to join states according to a probability.

Definition 24. Let X, Y be states and $p \in (0,1)$. We define the *union* of states X and Y with *associated probability* p as:

$$X \cup_p Y = \begin{cases} X & \text{if } Y = \emptyset \\ Y & \text{if } X = \emptyset \\ \{(a, p \cdot pro(a, X) + (1-p) \cdot pro(a, Y))\} & \text{otherwise} \end{cases}$$

□

As in the nonprobabilistic case, when we are defining the union of two trees, we must distinguish between defining *in the root* and *under the root*. In the root, we must consider the union of the (initial) states of the two trees composed with the external choice. Then,

$$p(R_1 +_p R_2, \epsilon, A) = \sum_{A = B \cup_p C} p(R_1, B) \cdot p(R_2, C)$$

That is, a state A is reachable (immediately) if there exist states B (reachable by R_1) and C (reachable by R_2) such that $A = B \cup_p C$. The probability with which A is reachable is equal to the sum of the product of the probabilities with which B's and C's are reached. As a consequence of this definition, we have that the function corresponding to the external choice is strict, because if an argument is equivalent to Ω, then the result will be Ω (Ω has not reachable states). That is, $[\![P +_p \Omega]\!] = [\![\Omega +_p P]\!] = [\![\Omega]\!]$ for any $0 < p < 1$.

Now we have to define the rest of the tree under the root.

$$\begin{aligned} p(R_1 +_p R_2, \langle A\, a\rangle \circ s', X) = & \sum_{A = B \cup_p C} \frac{p \cdot pro(a,B)}{p \cdot pro(a,B) + (1-p) \cdot pro(a,C)} \cdot p(R_1, s_B, X) \cdot p(R_2, C) \\ & + \sum_{A = B \cup_p C} \frac{(1-p) \cdot pro(a,C)}{p \cdot pro(a,B) + (1-p) \cdot pro(a,C)} \cdot p(R_2, s_C, X) \cdot p(R_1, B) \end{aligned}$$

where $s_B = \langle B\, a\rangle \circ s'$ and $s_C = \langle C\, a\rangle \circ s'$.

Intuitively, in order to calculate the reachable states after a sequence of the form $s = \langle A_1\, a_1, A_2\, a_2, \ldots, A_n\, a_n \rangle$, we must know if the first action a_1 could be executed by R_1 and/or R_2 and how the state A_1 can be reached by $R_1 +_p R_2$. Once we know which *pat performs* a_1, these states must be reachable by this *pat*.

Proposition 25. *For each* $a \in Act$, *the function* $a;\_ :: \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ *is continuous. The functions* $\oplus_p$, $+_p :: \mathbf{PAT}_{\mathrm{Act}} \times \mathbf{PAT}_{\mathrm{Act}} \longrightarrow \mathbf{PAT}_{\mathrm{Act}}$ *are continuous for each* $p \in (0,1)$.

Recursion

As usual when definiting a denotational semantics, the meaning of recursively processes defined by expressions of the form $recX.P(X)$ is obtained as the limit of the finite approximations $P_0 = \Omega, P_1 = P(\Omega), \ldots, P_n = P^n(\Omega)$. Because all the operators included in the term $P(X)$ are continuous, this limit is the least fixed point of the equation $X = P(X)$. That is, we define

$$[\![recX.\ P(X)]\!] = \bigsqcup_{n=0}^{\infty} [\![P_n]\!]$$

The close relation between the denotational semantics of a process and its operational behavior is underscored in the following

Theorem 26. *Let* P *be a process.* $\mathcal{A}(P,s) = \{(A_1,p_1),(A_2,p_2)\ldots(A_n,p_n)\}$ *iff* $\forall\ 1 \leq i \leq n:\ p_i = \frac{p([\![P]\!],s,A_i)}{p_s} \ \wedge\ p([\![P]\!],s,A) = 0$ if $A \neq A_i$, *where* $p_\epsilon = 1$ *and* $p_{s' \circ \langle B\, b \rangle} = p([\![P]\!], s', B)$

Proof. (Sketch) The proof is done by induction on the length of the sequence s. The only difficult case is that of external choice. □

Corollary 27. *Let* P, Q *be processes. Then,* $P \cong Q \Leftrightarrow [\![P]\!] \doteq [\![Q]\!]$.

Corollary 28. (Full Abstraction for $\mathbf{PAT}_{\mathrm{Act}}$)
Let P, Q *be processes. Then,* $P \approx Q \Leftrightarrow [\![P]\!] \doteq [\![Q]\!]$.

6 Conclusion and Future work

In this paper we have developed a probabilistic extension of classical testing semantics [Hen88]. We have presented an alternative characterization which is equivalent to the testing equivalence. We also have given a denotational semantics which has been proved to be fully abstract with respect to the testing equivalence.

As future work we plan to extend the studied language to a language where parallel and restriction operators are considered. We are working with two possibilities for the parallel operator: a parallel operator with one probability (as in [Cua93]) or the one with two probabilities described in [BBS92]. We are also very interested in the verification of properties for processes in our language, using HML extended with probabilities. A good starting point is the logic described in [LS92]. Finally, we will extend the complete axiomatization given in [Cua93] to our semantic model.

Acknowledgements

We would like to thank Scott A. Smolka for his very useful comments on a draft version of this paper and for his warm reception when the first author visited Stony Brook. We are also indebted to the anonymous referees for their valuable comments.

References

[BBS92] J.C.M. Baeten, J.A. Bergstra, and S.A. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities. In *CONCUR'92, LNCS 630*, pages 472–485, 1992.

[Chr90] I. Christoff. Testing equivalences and fully abstract models for probabilistic processes. In *CONCUR'90, LNCS 458*, pages 126–140, 1990.

[CSZ92] R. Cleaveland, S.A. Smolka, and A.E. Zwarico. Testing preorders for probabilistic processes. In *19 th ICALP, LNCS 623*, pages 708–719, 1992.

[Cua93] F. Cuartero. *CSP probabilístico (PCSP). Un modelo probabilístico de procesos concurrentes.* PhD thesis, Universidad Complutense de Madrid, 1993.

[dNH87] R. de Nicola and M. Hennessy. CCS without τ's. In *TAPSOFT'87, LNCS 249*, pages 138–152, 1987.

[GJS90] 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*, 1990.

[Hen85] M. Hennessy. Acceptance trees. *Journal of the ACM*, 32(4):896–928, 1985.

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

[HJ90] H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *11th IEEE Real-Time Systems Symposium*, pages 278–287, 1990.

[JY95] B. Jonsson and W. Yi. Compositional testing preorders for probabilistic processes. In *10th IEEE Symposium on Logic In Computer Science*, 1995.

[Low95] G. Lowe. Probabilistic and prioritized models of timed CSP. *Theoretical Computer Science*, 138:315–352, 1995.

[LS92] K.G. Larsen and A. Skou. Compositional verification of probabilistic processes. In *CONCUR'92, LNCS 630*, pages 456–471, 1992.

[Sei92] K. Seidel. *Probabilistic Communicating Processes.* PhD thesis, Oxford University, 1992.

[SS90] S.A. Smolka and B. Steffen. Priority as extremal probability. In *CONCUR'90, LNCS 458*, pages 456–466, 1990.

[vGSST90] R. van Glabbeek, S.A. Smolka, B. Steffen, and C.M.N. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *5th IEEE Symposium on Logic In Computer Science*, pages 130–141, 1990.

[WSS94] S.-H. Wu, S.A. Smolka, and E.W. Stark. Composition and behaviors of probabilistic I/O automata. In *CONCUR'94, LNCS 836*, pages 513–528, 1994.

[YCDS94] 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, 1994.

[YL92] W. Yi and K.G. Larsen. Testing probabilistic and nondeterministic processes. In *PSTV XII*, pages 47–61, 1992.