University of Castilla-La Mancha

A publication of the

Department of Computer Science

Algebraic Theory of Probabilistic and Nondeterministic Processes.

by

Diego Cazorla, Fernando Cuartero,
Valentín Valero, Fernando L. Pelayo

Technical Report **#DIAB-01-01-19** September 2001

DEPARTAMENTO DE INFORMÁTICA
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE CASTILLA-LA MANCHA
Campus Universitario s/n
Albacete - 02071 - Spain
Phone +34.967.599200, Fax +34.967.599224

Algebraic Theory of Probabilistic and Nondeterministic Processes *

D. Cazorla F. Cuartero V. Valero F. L. Pelayo
Departamento de Informática
Universidad de Castilla-La Mancha
Campus Universitario s/n. 02071 Albacete, Spain
{dcazorla, fernando, valentin, fpelayo}@info-ab.uclm.es

Abstract

In this paper we present an algebraic language for the specification of probabilistic and nondeterministic processes, PNAL, which is a probabilistic extension of EPL (*Algebraic Theory of Processes*, M. Hennessy) that maintains nondeterminism.

We have defined both an operational and a testing semantics for PNAL, as well as a denotational semantics and a proof system. We show that the equivalence between nondeterministic processes in EPL is preserved, and we also prove that the testing semantics, the denotational semantics and the proof system are fully abstract.

Keywords: Formal Methods, Process Algebra, Testing, Probability, Nondeterminism, Process Specification, Algebraic reasoning.

1 Introduction

The fundamentals of process algebras were established in the eighties but only qualitative aspects of concurrent systems were considered in these pioneering works. Subsequently quantitative aspects have been proposed, and several extensions to classical process algebra models have been developed including time, priorities and probabilities.

One of the first papers describing processes with a probabilistic behaviour is [2], where the concept of probabilistic experiments is introduced. In [10] PCCS is defined as an algebraic language based on the synchronous version of CCS where the nondeterministic choice is replaced by a probabilistic choice: $\sum[p_i]E_i$, where p_i is the probability of the process behaving like E_i. In [28, 27], this model

*This work has been partially supported by the CICYT project "Performance Evaluation of Distributed Systems", TIC2000-0701-c02-02

is extended and three interpretations for concurrent probabilistic processes are presented: reactive, generative and stratified models.

In [13] TPCCS is presented, which is an extension of CCS with probabilities and time. This model is based on the asynchronous version of CCS and allows us to model more realistic concurrent systems, which are usually asynchronous. Probability is introduced using a probabilistic choice operator. Two types of processes may be distinguished at the top level of the syntax: probabilistic and nondeterministic processes. The alternating of both kinds of processes is necessary, which is why the model is called *the alternating model*.

Besides the models already mentioned, several languages based on CCS [4, 19, 21] that analyze various semantic aspects have been proposed. Thus, in [21] a notion of bisimulation between probabilistic transition systems is introduced; this topic is also studied in [4]. In [19] several equivalences between processes are presented.

Based on CSP, we can cite [5, 6], where a *reactive* interpretation is considered, whereas in [24] a *generative* one is presented. In both models fully abstract testing semantics with respect to denotational semantics, which are based on acceptance trees, are presented.

Nevertheless, the main idea in all these models consists of substituting a nondeterministic behaviour by a probabilistic one in which the different types of behaviours are quantified with probabilities. A precise knowledge of the probabilities associated with each behaviour is therefore required to define a process. However, in practice, this probability distribution could be completely or partially unknown.

It is also necessary therefore to provide operators that allow us to deal with nondeterminism, but, at the same time, taking into account the probabilistic aspects. In the process of modelling a concurrent system we sometimes find nondeterministic behaviours that can be quantified (for instance, the probability of a faulty communication channel failing), whereas on other occasions these nondeterministic behaviours cannot be quantified (for instance, relative speed between two processes). This is why we need a language that allows us to model nondeterministic systems with or without quantification, depending on the nature of the system in question.

In line with these ideas, in [31] a probabilistic extension of CCS is presented. The language adds a probabilistic choice operator ($\oplus_p$) indexed with a probability $p \in (0, 1)$, which models the probabilistic internal choice, whereas the original CCS choice ($+$) is maintained for representing nondeterminism. Intuitively, $P \oplus_p Q$ behaves like P with probability p and like Q with probability $1 - p$. A testing semantics, based on De Nicola and Hennessy testing, is presented. Given a process P and a test T to deal with nondeterminism, the notion "P *may pass* T with a set of probabilities $\phi_{(P,T)}$" is defined. Based on these sets, three preorders are defined by generalizing, for probabilistic and nondeterministic processes, *may*, *must* and *test* preorders in the testing theory of De Nicola and Hennessy. This work is taken further in [17, 18], where these preorders are redefined and denotationals characterization for may–testing and must–testing are presented.

In [22] a probabilistic extension of CSP (PCSP) that maintains nondeterminism is introduced. The model is provided with an operational semantics, and several denotational semantics are given, based on different kinds of observations (trace, broom and barb semantics), where semantic objects are trees with three kinds of nodes: probabilistic, nondeterministic and action nodes.

Another probabilistic extension of CSP is presented in [23], where a denotational semantics of CSP is defined by applying the probabilistic powerdomain construction of Jones and Plotkin [16, 15] over a directed–complete partial order. Probabilistic processes are considered to be probability distributions over processes of CSP. In [20], a generalisation of Larsen and Skou's [21] reactive probabilistic transition system is presented. This model considers three kinds of choices: action–guarded probabilistic choice, external and internal choice. An operational preorder and a equivalence for processes based on testing are proposed.

Based on ACP, we can cite [1]. In that paper, a probabilistic version of ACP is presented leading to a language that combines probability and nondeterminism. An operational semantics is defined based on the alternating model [13]. In the construction of the term models they use a probabilistic bisimulation showing soundness and completeness of the term model with respect to the proposed axiom systems.

In [9] a comparison of the various ways of combining probabilistic choice with classical process algebras is introduced. Several interpretations of nondeterminism and probability are considered. A distinction is drawn between local, "global–don't care", and "global–don't know" nondeterminism versus unconditional, "conditional–don't care" and "conditional–don't know" probability. Different combinations of these induced nine different semantic models.

Different approaches, in the field of probabilistic automaton and Markov decision processes, are considered in [26, 25, 29, 8] where the separation between nondeterministic and probabilistic behaviour is achieved by means of adversaries, schedulers or policies, which resolve the nondeterminism.

The specific aim of this paper is to add probabilistic choice to Hennessy's EPL [14], to show how the semantics of that can be defined, and to investigate the resulting process algebra. This paper is a extended version of [3], where a finite process algebra is considered and denotational semantics of parallel composition is introduced intuitively, without a formal definition. In this paper we consider the full language, and give proofs of theorems, propositions and lemmas.

The paper is structured as follows. Section 2 introduces our language. In Sections 3 and 4 an operational semantics and a testing semantics are presented. In Section 5 a denotational semantics is given by means of probabilistic acceptance trees and we prove that the denotational semantics is fully abstract with respect to the testing semantics. In section 6 a sound and complete proof system is presented. Finally, in section 7 we evaluate the work and compare it with other approaches.

2 PNAL language

The syntax of the language we present, PNAL, is based on EPL [14] and provides three kinds of choice: classical internal choice, denoted by $\oplus$, classical external choice, denoted by $+$, and a probabilistic choice labelled by a probability p: $\otimes_p$.

Let Δ be a set of input actions, $\overline{\Delta} = \{\overline{a} \,|\, a \in \Delta\}$ their corresponding output actions, and $\mathcal{A}ct = \Delta \cup \overline{\Delta}$. Furthermore we assume a set of process variables Id ranged over by X and Y. Terms of PNAL are defined by:

$$P ::= \mathbf{0} \;|\; \Omega \;|\; X \;|\; a.P \;|\; P \oplus P \;|\; P + P \;|\; P \otimes_p P \;|\; P|P \;|\; P\backslash A \;|\; recX.P$$

where $a \in \mathcal{A}ct$, $p \in (0,1)$, $X \in Id$, and $A \subset \Delta$.

The classical operators are interpreted in the usual fashion and the probabilistic choice has a *generative* interpretation, i.e., $P \otimes_p Q$ behaves like P with probability p, and like Q with probability $1-p$, this decision been made internally.

In order to deal with probabilities and nondeterminism we have to decide how to solve a situation where both choices appear. In the literature available we have, basically, two alternatives: in [1, 23, 31] probabilistic choices are solved first, whereas in [22] the opposite approach is taken. Note that this decision is not at all meaningless, because, depending on the approach adopted, some properties of classical process algebras will be preserved or not. For example, idempotency of internal choice is only maintained when internal choice is resolved first. Idempotency of external choice is never maintained, because if we take $P = a \otimes_{0.5} b$, we find that $P \,\Box\, P$ increments the probability of a, for any model we consider.

In all these models we can identify two kinds of nondeterminism: nondeterminism introduced by internal choice (syntactic nondeterminism), and nondeterminism introduced by external choice and parallel composition between processes prefixed by the same action. Let us consider the following example (using PNAL syntax):

$$(a.P \;|\; (a.Q \otimes_p b)) \oplus R$$

If we consider Lowe's approach [22], we first resolve internal choice and later (considering the left hand side process) probabilistic choice; at this point we find a nondeterministic behaviour introduced by the process $a.P|a.Q$, and the resolution of this nondeterminism is delayed after the execution of action a. But, in Lowe's words *(sic)* “it is not enough to resolve all nondeterminism in a given state before the probabilistic choices; we have to resolve all nondeterminism before *any* probabilistic choice ... can we find a model that is (faithful to our intuition)?.Such a model would need to ensure that all nondeterministic branching occurs before any other branching, but it is far from obvious how to write down the semantic equations so as to achieve this”. This is still, indeed, an open problem.

If we consider the approach taken by Morgan *et al.* [23], probabilistic choice is resolved first, and later on both kinds of nondeterminism can be resolved.

Moreover, in that model we find the opposite behaviour, i.e., every probabilistic choice must be resolved before any internal choice. But, in the model in question, probabilistic choice is distributed through all the other operators, so this aim can be achieved easily.

Taking into account both models [22, 23], and the study of asynchronous probabilistic parallel composition introduced in [7] (where "bundle probabilistic transition systems" are introduced), we consider that the more suitable approach is to resolve probabilistic choice first, and then resolve any kind of nondeterminism later on. As we will see in the following sections, this decision leads us to a model that shares some characteristics with the Morgan *et al.* [23] model (e.g., distribution properties of probabilistic choice).

3 Operational semantics

The operational semantics is defined in the usual way, by using a labelled transition system. In order to understand these rules we require a preliminary definition:

Definition 1 *(Probabilistic stability)*
We define in an inductive way the probabilistic stability predicate

i) $\mathbf{0}$, Ω, $a.P$ *and* $recX.P$ *are probabilistically stable.*

ii) $P \oplus Q$, $P + Q$, $P|Q$ *are probabilistically stable iff both* P *and* Q *are.*

iii) $P\backslash A$ *is probabilistically stable iff* P *is.*

We use $P\!\downarrow$ *to denote that* P *is probabilistically stable.*

Transition rules are divided into 3 tables. Table 1 defines the probabilistic choice operator behaviour, by using probabilistic transitions, like $P \longrightarrow_p Q$, which means that P may evolve to Q with probability p immediately. Rules P1a and P1b indicate that process $P \otimes_p Q$ behaves like P with probability p and like Q with probability $1-p$. The remaining rules establish the precedence of the probabilistic choice with respect to the other operators. Processes P and Q are supposed to be probabilistically independent processes in rules P2c, P3c, and P4c, as in [1, 13, 31].

Probabilistically stable conditions for process Q in rule P2a and process P in rule P2b are needed, as seen in the following example. Let us replace rules P2a and P2b by

$$\textbf{(P2a')}\quad \frac{P \longrightarrow_p P'}{P \oplus Q \longrightarrow_p P' \oplus Q} \qquad \textbf{(P2b')}\quad \frac{Q \longrightarrow_q Q'}{P \oplus Q \longrightarrow_q P \oplus Q'}$$

$$
\begin{array}{llll}
\textbf{(P1a)} & \dfrac{}{P \otimes_p Q \longrightarrow_p P} & \textbf{(P1b)} & \dfrac{}{P \otimes_p Q \longrightarrow_{1-p} Q} \\[2ex]
\textbf{(P2a)} & \dfrac{P \longrightarrow_p P', Q\!\downarrow}{P \oplus Q \longrightarrow_p P' \oplus Q} & \textbf{(P2b)} & \dfrac{P\!\downarrow, Q \longrightarrow_q Q'}{P \oplus Q \longrightarrow_q P \oplus Q'} \\[2ex]
\textbf{(P2c)} & \dfrac{P \longrightarrow_p P', Q \longrightarrow_q Q'}{P \oplus Q \longrightarrow_{p \cdot q} P' \oplus Q'} & & \\[2ex]
\textbf{(P3a)} & \dfrac{P \longrightarrow_p P', Q\!\downarrow}{P + Q \longrightarrow_p P' + Q} & \textbf{(P3b)} & \dfrac{P\!\downarrow, Q \longrightarrow_q Q'}{P + Q \longrightarrow_q P + Q'} \\[2ex]
\textbf{(P3c)} & \dfrac{P \longrightarrow_p P', Q \longrightarrow_q Q'}{P + Q \longrightarrow_{p \cdot q} P' + Q'} & & \\[2ex]
\textbf{(P4a)} & \dfrac{P \longrightarrow_p P', Q\!\downarrow}{P|Q \longrightarrow_p P'|Q} & \textbf{(P4b)} & \dfrac{P\!\downarrow, Q \longrightarrow_q Q'}{P|Q \longrightarrow_q P|Q'} \\[2ex]
\textbf{(P4c)} & \dfrac{P \longrightarrow_p P', Q \longrightarrow_q Q'}{P|Q \longrightarrow_{p \cdot q} P'|Q'} & \textbf{(P5)} & \dfrac{P \longrightarrow_p P'}{P\backslash A \longrightarrow_p P'\backslash A}
\end{array}
$$

Table 1: Inference rules for transitions labelled by probabilities

and consider the process $P = (a.\mathbf{0} \otimes_p b.\mathbf{0}) \oplus (c.\mathbf{0} \otimes_q d.\mathbf{0})$. Applying rules P2a', P2b' and P2c we get the following transitions

$$
\begin{array}{ll}
P2a' & P \longrightarrow_p a.\mathbf{0} \oplus (c.\mathbf{0} \otimes_q d.\mathbf{0}) \\
P2a' & P \longrightarrow_{1-p} b.\mathbf{0} \oplus (c.\mathbf{0} \otimes_q d.\mathbf{0}) \\
P2b' & P \longrightarrow_q (a.\mathbf{0} \otimes_p b.\mathbf{0}) \oplus c.\mathbf{0} \\
P2b' & P \longrightarrow_{1-q} (a.\mathbf{0} \otimes_p b.\mathbf{0}) \oplus d.\mathbf{0} \\
P2c & P \longrightarrow_{p \cdot q} a.\mathbf{0} \oplus c.\mathbf{0} \\
P2c & P \longrightarrow_{(1-p) \cdot q} b.\mathbf{0} \oplus c.\mathbf{0} \\
P2c & P \longrightarrow_{p \cdot (1-q)} a.\mathbf{0} \oplus d.\mathbf{0} \\
P2c & P \longrightarrow_{(1-p) \cdot (1-q)} b.\mathbf{0} \oplus d.\mathbf{0}
\end{array}
$$

Adding up the probabilities that label branches leaving from P we get 3, when in our probability space we do not allow probabilities greater than 1. Thus, we need to introduce the probabilistically stable predicate in this rules in order to avoid these situations.

The behaviour of the remaining operators is presented in Tables 2 and 3, where we assume that processes are now probabilistically stable. This condition will be justified in the next section, where we introduce testing semantics. The second table defines the rules for unlabelled transitions, i.e., those representing an internal evolution, $P \longrightarrow Q$, with the usual meaning that P may evolve internally to Q without performing any visible action. Rules I1a and I1b show how the internal choice operator behaves. Rules I2a, I2b, I3a and I3b indicate that internal choice distributes over the external choice and parallel composition. Rule I3c indicates that an unbarred action may synchronize with its homologous barred action and the obtained process evolves internally. Rule I4 shows the operational behaviour for the restriction operator. Finally, rule I5 is the unwinding rule for recursion.

(I1a) $\dfrac{}{P \oplus Q \longrightarrow P}$	**(I1b)** $\dfrac{}{P \oplus Q \longrightarrow Q}$
(I2a) $\dfrac{P \longrightarrow P'}{P + Q \longrightarrow P' + Q}$	**(I2b)** $\dfrac{Q \longrightarrow Q'}{P + Q \longrightarrow P + Q'}$
(I3a) $\dfrac{P \longrightarrow P'}{P\|Q \longrightarrow P'\|Q}$	**(I3b)** $\dfrac{Q \longrightarrow Q'}{P\|Q \longrightarrow P\|Q'}$
(I3c) $\dfrac{P \stackrel{a}{\longrightarrow} P', Q \stackrel{\overline{a}}{\longrightarrow} Q'}{P\|Q \longrightarrow P'\|Q'}$	**(I4)** $\dfrac{P \longrightarrow P'}{P\backslash A \longrightarrow P'\backslash A}$
(I5) $\dfrac{}{recX.P \longrightarrow P[recX.P/X]}$	

Table 2: Inference rules for unlabelled transitions ($\mathbf{P}\downarrow$ and $\mathbf{Q}\downarrow$)

The third table defines the inference rules for transitions labelled by actions in $\mathcal{A}ct$, $P \stackrel{a}{\longrightarrow} Q$, with the usual meaning that P can perform an a and move to Q.

(A1) $\dfrac{}{a.P \stackrel{a}{\longrightarrow} P}$	**(A2a)** $\dfrac{P \stackrel{a}{\longrightarrow} P'}{P + Q \stackrel{a}{\longrightarrow} P'}$
(A2b) $\dfrac{Q \stackrel{a}{\longrightarrow} Q'}{P + Q \stackrel{a}{\longrightarrow} Q'}$	**(A3a)** $\dfrac{P \stackrel{a}{\longrightarrow} P'}{P\|Q \stackrel{a}{\longrightarrow} P'\|Q}$
(A3b) $\dfrac{Q \stackrel{a}{\longrightarrow} Q'}{P\|Q \stackrel{a}{\longrightarrow} P\|Q'}$	**(A4)** $\dfrac{P \stackrel{a}{\longrightarrow} P'}{P\backslash A \stackrel{a}{\longrightarrow} P'\backslash A}$ $(a, \overline{a} \notin A)$

Table 3: Inference rules for transitions labelled by actions $\mathbf{a} \in \mathcal{A}ct$ ($\mathbf{P}\downarrow$ and $\mathbf{Q}\downarrow$)

Definition 2 *The operational semantics of PNAL is defined as the multiset of transitions we can derive by using the rules in Tables 1–3.*

Throughout the paper we will use $\rightsquigarrow$ instead of $\longrightarrow_p$, $\longrightarrow$ and $\xrightarrow{a}$ in order to represent any of these transitions.

Fig. 1 illustrates the application of these rules. Each transition is labelled by the name of the applied rule.

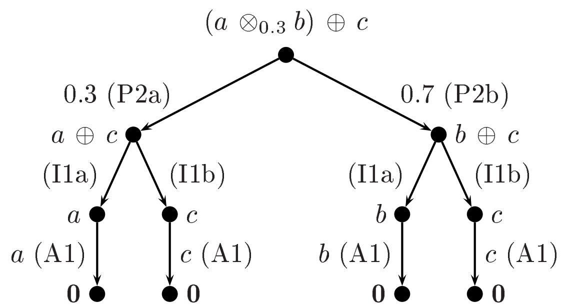

Figure 1: Transition system of the process $(\mathbf{a.0} \otimes_{\mathbf{0.3}} \mathbf{b.0}) \oplus \mathbf{c.0}$

4 Testing semantics

A test is just a process which may additionally perform a special action w to report a successful state. A purely probabilistic process P will pass a test T with a certain probability p. However, when dealing with nondeterministic processes, i.e., including internal choices, we will have to compute a set of probabilities to pass each test, in order to capture all the possibilities.

As usual, in order to relate two processes using the testing semantics, we take one of them as a specification and the other one as its implementation. We will say that a process is a good implementation of a specification if the range of probabilities with which the implementation passes every test is included in the range of probabilities with which the specification passes the same test.

We use $P \| T$ to stand for $(P|T)\backslash \mathcal{A}ct$. $P \| T$ is a process that can only perform probabilistic transitions $\longrightarrow_p$, unlabelled transitions $\longrightarrow$ or w−transitions $\xrightarrow{w}$. These processes are called test processes and will be represented by S, S', etc.

Given a process P and a test T, $\phi_{(P,T)}$ will represent the set of probabilities for accepting T considering every possible evolution of $P \| T$. In order to define $\phi_{(P,T)}$ we need some preliminary definitions, along the lines of the ideas presented in [31].

Definition 3 *A resolution R of $P \| T$ is a maximal subtree of $P \| T$ such that, whenever $S \hookrightarrow S_1$, $S \hookrightarrow S_2$, we have $S_1 = S_2$, where $\hookrightarrow$ represents either $\longrightarrow$ or $\xrightarrow{w}$.*

Every nondeterministic node in a resolution has therefore a unique output arc. Fig. 2 shows the transition system for the test process $a \otimes_q b \,\|\, \overline{a}.w \oplus \overline{b}$ and one of its resolutions (thick lines).

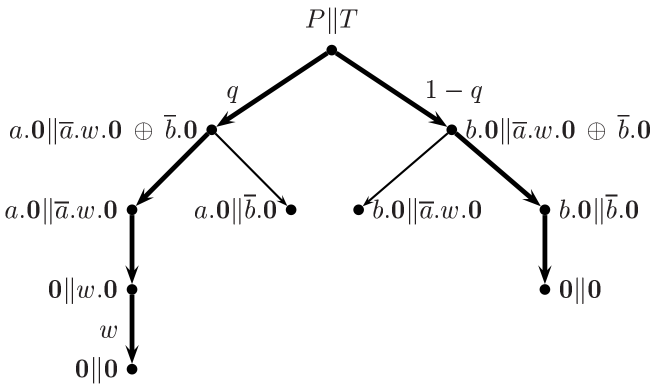

Figure 2: Transition system of a process and one of its resolutions

Definition 4 *A computation C of a test process S is a maximal sequence of transitions, $S \rightsquigarrow S_1 \dots S_{i-1} \rightsquigarrow S_i \dots$ where $\rightsquigarrow$ represents either $\longrightarrow_p$, $\longrightarrow$ or $\xrightarrow{w}$.*

We will use the following notation: $\mathcal{R}(S)$ denotes the set of resolutions of S, $\mathcal{C}(R)$ denotes the set of computations of resolution R, $Pr(C)$ denotes the probability of reaching a successful state throughout the computation C, and $Pr(R)$ denotes the probability to reach a successful state in the resolution R.

Definition 5 *A computation $C = S \rightsquigarrow S_1 \cdots S_{i-1} \rightsquigarrow S_i \dots \rightsquigarrow S_n$ is a successful computation if $S_n \xrightarrow{w}$ and $S_k \stackrel{w}{\not\longrightarrow}$ for all $k < n$. We use $\widehat{\mathcal{C}}(R)$ to denote the set of successful computations of R. We may compute the probability for C to be a successful computation as follows:*

$$
\begin{array}{l}
Pr(S_n \xrightarrow{w} S_{n+1}) = 1 \text{ (success)} \\
Pr(S_i \rightsquigarrow C) = \left\{ \begin{array}{ll} p \cdot Pr(C) & \text{if } \rightsquigarrow = \longrightarrow_p \\ Pr(C) & \text{if } \rightsquigarrow = \longrightarrow \end{array} \right.
\end{array}
$$

And the probability of achieving success in a resolution as:

$$
Pr(R) = \sum_{C \in \widehat{\mathcal{C}}(R)} Pr(C)
$$

Definition 6 *Given a process P and a test T, we say that P* may pass *T with the set of probabilities $\phi_{(P,T)} = \{Pr(R) \mid R \in \mathcal{R}(P \| T)\}$.*

At this point, we can justify the need for processes to be probabilistically stable in order to apply the rules of operational semantics in Tables 2 and 3. Let $P = (a \otimes_q b) \oplus a$ and $T = \overline{a}.w + \overline{b}.w$ be a process and a test, respectively. If we do not have the probabilistically stable premise, we obtain the transition system shown in Figure 3. One of its resolutions is shown in Figure 4.

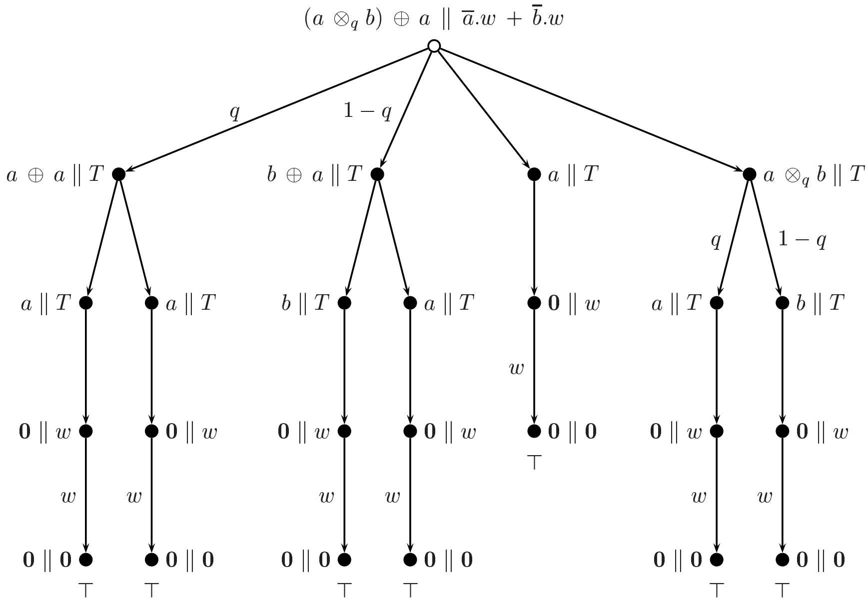

Figure 3: Transition system of $(a \otimes_q b) \oplus a \parallel \overline{a}.w + \overline{b}.w$ (when we do not consider the "probabilistically stable" premise).

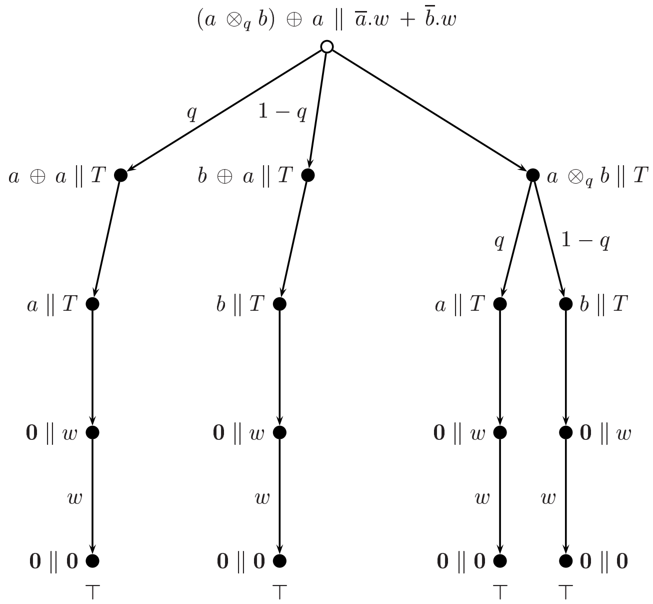

Figure 4: One of the resolution of the transition system in Figure 3.

This resolution has 4 computations, which have the following probabilities of success: q, $1-q$, q and $1-q$. Adding up probabilities we find that the probability of achieving success in the resolution is 2, which is a not valid probability because it is greater than 1. We obtain this result because we allow unlabelled and probabilistic transitions to be executed at the same level. If we avoid these situations, i.e., if we consider the probabilistically stable premise, these kinds of problems do not arise.

We may now introduce the *test* preorder over processes, by using $\phi_{(P,T)}$.

Definition 7 *We define the test preorder as follows:*

$$\begin{aligned} P \sqsubseteq_{test} Q \; \textit{if} \; \forall T \; \textit{s.t.} \; \phi_{(P,T)} \neq \{0\}, \; \textit{then} \\ \phi_{(Q,T)} \neq \{0\}, \sup\{\phi_{(P,T)}\} \geq \sup\{\phi_{(Q,T)}\}, \\ \textit{and} \; \inf\{\phi_{(P,T)}\} \leq \inf\{\phi_{(Q,T)}\} \end{aligned}$$

This preorder imposes the condition that whenever Q cannot accept a test T, P cannot accept it either. This has been introduced because, according to the definition of test preorder given in [17], the processes $P = a \oplus c$ and $Q = a$ are related under the test preorder, i.e., $P \sqsubseteq_{test} Q$. By contrast in Hennessy's testing P *may* $c.w$ but Q $\not{m}ay$ $c.w$ so $P \not\sqsubseteq_{may} Q$. According to our definition both situations are captured.

The *test* relation has the following interpretation: $P \sqsubseteq_{test} Q$ if Q is a good implementation of the specification P. The corresponding equivalence is defined as follows:

Definition 8 $P \equiv_{test} Q$ *iff* $P \sqsubseteq_{test} Q$ *and* $Q \sqsubseteq_{test} P$.

Example 1 *Let* $P = a \otimes_{0.5} \mathbf{0}$ *and* $T = \overline{a}.w$. *We then have* $\phi_{(P,T)} = \{0.5\}$, $\phi_{(P \oplus P,T)} = \{0.25, 0.50, 0.75\}$ *and* $\phi_{(P + P,T)} = \{0.75\}$. *Therefore,* $P \neq P \oplus P$ *and* $P \neq P + P$

Now, we state that the testing semantics here presented is an extension of EPL testing semantics. In order to differentiate both semantics we will use the subscript *Hen* to denote Hennessy's testing preorder and equivalence.

In Hennessy's testing we consider two relations between a process and a test; these relations are P *may* T if the process P may pass or may fail the test, and P *must* T if the process P always passes the test T. In terms of probabilities we can say that:

- P *must* T if $0 \notin \phi_{(P,T)}$, and
- P *may* T if $1 \in \phi_{(P,T)}$

Proposition 1 *Let* $P, Q \in EPL$. *Then*

$$P \sqsubseteq_{Hen} Q \Longleftrightarrow P \sqsubseteq_{test} Q$$

5 Denotational semantics

The aim of this section is to provide our language with a fully abstract denotational semantics with respect to the testing semantics.

5.1 Domain of semantic processes

Our domain will be a set of probabilistic acceptance trees, PAT. Let us informally describe the structure of these trees:

D1 We will consider trees with three kinds of nodes: probabilistic ($\otimes$), closed ($\bullet$), and open ($\circ$).

D2 The root node is a probabilistic one.

D3 Closed nodes are labelled by an acceptance set (as defined in [14]). Open nodes are labelled by a deterministic acceptance set. Probabilistic nodes are unlabelled.

D4 There is a finite number of branches leaving a probabilistic node (1 or more). There is a finite number of branches leaving an open or closed node (0 or more).

D5 Branches leaving a probabilistic node are labelled by probabilities p_i, with $\sum_i p_i = 1$. Every branch leaving a closed or an open node is labelled by one action in the acceptance set, but there are no two branches labelled by the same action.

D6 Probabilistic nodes and nondeterministic nodes (closed and open) appear in an alternative way. Once we have reached an open node, alternation occurs only between probabilistic and open nodes. Terminal nodes are always closed or open.

We can see that these trees are quite similar to those used in [14], but now we have to extend these trees in order to include the probabilistic information associated with every reachable state.

Definition 9 *A semantic process is a tree that holds rules D1–D6.*

Let us now introduce some preliminary definitions and concepts.

Definition 10 *An acceptance set or state, $\mathcal{A}$, is a subset of $\mathcal{P}(\mathcal{A}ct)$. Let $\boldsymbol{D}(\mathcal{A})$ be the domain of acceptance sets for a given set of actions $\mathcal{A}ct$.*

Let $\mathcal{A}$ be an acceptance set. We define $A(\mathcal{A})$, the set of actions belonging to $\mathcal{A}$, as follows:

$$A(\mathcal{A}) = \{a \in \mathcal{A}ct \,|\, \exists B \in \mathcal{A} \text{ such that } a \in B\}$$

We say that s is a sequence of "acceptance set–action pairs" if $s \in (\mathbf{D}(\mathcal{A}) \times \mathcal{A}ct)^*$. We will denote the empty sequence by ε or $\langle\rangle$. As usual, we will consider a sequence concatenation operator, represented as $s\hat{\ }s'$. Given a set of actions $\mathcal{A}ct$, the domain of sequences induced by $\mathcal{A}ct$ will be represented by $\mathbf{D}(s_{\mathcal{A}ct})$.

For the sake of simplicity, we will represent a sequence as follows:

$$\langle \mathcal{A}_1 a_1 \mathcal{A}_2 a_2 \dots \mathcal{A}_n a_n \rangle$$

Given a tree $t \in$ PAT, we can use sequences in order to identify its nodes. Thus, a probabilistic node may be identified by using a sequence. For instance, the sequence $\langle \mathcal{A}_i a_i \rangle$ identifies the probabilistic node we can reach from the root node after following the branch labelled by action a_i, which leaves from the nondeterministic node (open or closed) labelled by the acceptance set $\mathcal{A}_i$. In order to identify nondeterministic nodes we need more information; first we need to reach its parent probabilistic node and then we need to distinguish among the nondeterministic nodes that share the same parent node. Thus, in this case we need a pair sequence–acceptance set, $(s, \mathcal{A})$. For the sake of homogeneity we will allow this pair to be represented as $\langle s\mathcal{A} \rangle$, i.e., $\langle \mathcal{A}_1 a_1 \mathcal{A}_2 a_2 \dots \mathcal{A}_n a_n \mathcal{A} \rangle$.

But this is not enough to identify a nondeterministic node. Consider the example in figure 5. In this case both nodes (open and closed) are identified by

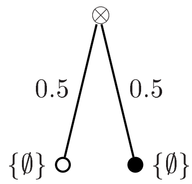

Figure 5: A pair $(s, \mathcal{A})$ identifies two different nodes

the pair $(\langle\rangle, \{\emptyset\})$. Therefore, we need to introduce more information in order to distinguish both nodes. An acceptance set that labels an open node will be represented by $\overline{\mathcal{A}}$, and we will consider that $\mathcal{A} \neq \overline{\mathcal{A}}$. Thus, in figure 5, the open node will be identified by the pair $(\langle\rangle, \overline{\{\emptyset\}})$ and the closed node by $(\langle\rangle, \{\emptyset\})$.

Given a tree $t \in$ PAT, we will use t/s to denote the tree whose root node is the probabilistic node identified by the sequence s. Thus, the root node of t may be alternatively represented as $t/\langle\rangle$.

We will use $Seq(t)$ to denote the set of sequences we may obtain from a tree, and $L(t)$ to denote the language t may accept. $L(t)$ may be defined as follows:

$$\begin{aligned} L(t) = \{ a_1 a_2 \dots a_n \in \mathcal{A}ct^* | \exists \mathcal{A}_i \in \mathbf{D}(\mathcal{A}) \text{ such that} \\ s = \langle \mathcal{A}_1 a_1 \mathcal{A}_2 a_2 \dots \mathcal{A}_n a_n \rangle \in Seq(t) \text{ and } \forall i\, a_i \in A(\mathcal{A}_i) \} \end{aligned}$$

Now, we introduce probabilistic acceptance sets.

Definition 11 *We say that $(p, \mathcal{A})$ is a "probability–acceptance set pair" if $(p, \mathcal{A}) \in (0, 1] \times \boldsymbol{D}(\mathcal{A})$.*

A probabilistic acceptance set, $\mathcal{PA}$, is defined as follows:

$$\mathcal{PA} = \{(p_i, \mathcal{A}_i) \in (0,1] \times \boldsymbol{D}(\mathcal{A}) \mid \sum_i p_i = 1 \wedge \mathcal{A}_i \neq \mathcal{A}_j \text{ if } i \neq j\}$$

The domain of probabilistic acceptance sets is represented by $\boldsymbol{D}(\mathcal{PA})$.

Definition 12 *Let $\mathcal{PA}$ be a probabilistic acceptance set. We define the set of acceptance sets belonging to $\mathcal{PA}$ as follows:*

$$\mathit{Accept}(\mathcal{PA}) = \{\mathcal{A} \in \boldsymbol{D}(\mathcal{A}) \mid \exists p \textit{ such that } (p, \mathcal{A}) \in \mathcal{PA}\}$$

Given a semantic process t, we will use $\mathcal{PA}(t/s)$ to represent the probabilistic acceptance set at the first level of the semantic process t/s.

Definition 13 *Let $t \in PAT$. We define the probability distribution, induced by a probabilistic acceptance set, as the function $\pi : PAT \times \boldsymbol{D}(\mathcal{A}) \longrightarrow [0,1]$ given by*

$$\pi(t/s, \mathcal{A}) = \begin{cases} p & \textit{if } (p, \mathcal{A}) \in \mathcal{PA}(t/s) \\ 0 & \textit{otherwise} \end{cases}$$

Definition 14 *Let two acceptance sets, $\mathcal{A}_1 \in \mathit{Accept}(\mathcal{PA}(t_1/s_1))$ and $\mathcal{A}_2 \in \mathit{Accept}(\mathcal{PA}(t_2/s_2))$. We say that $\mathcal{A}_1$ precedes $\mathcal{A}_2$, denoted by $\mathcal{A}_1 \preceq \mathcal{A}_2$, if one of the following conditions holds:*

- *If $\mathcal{A}_2$ labels an open node, then $\mathcal{A}_1$ labels an open node too, and $A(\mathcal{A}_1) \subseteq A(\mathcal{A}_2)$.*
- *If $\mathcal{A}_2$ labels a closed node, then*
  - *$\mathcal{A}_1$ labels an open node, or*
  - *$\mathcal{A}_1$ labels a closed node and $\mathcal{A}_2 \subseteq \mathcal{A}_1$.*

Intuitively, an open node, as we will see later, represents a possibility of divergent behaviour. Thus, when comparing an open node and a closed node, the open node will always precede the closed one. Therefore, the precedence relation give us some idea of order based on nondeterminism. An open node is more nondeterministic than a closed one. When we compare two open nodes, we need to consider the set of actions that may be performed, apart from divergence. In this case we have the condition $A(\mathcal{A}_1) \subseteq A(\mathcal{A}_2)$, i.e, the node in tree t_1 is more nondeterministic than the node in t_2 because we know less about it. Finally, in the event of both nodes being closed, we have the usual condition, $\mathcal{A}_2 \subseteq \mathcal{A}_1$.

5.2 Partial order and equivalence

In order to define the semantics of recursive processes, we need to introduce an order over semantic processes, denoted by $\leq_{PAT}$.

First, we need to introduce some preliminary definitions.

Definition 15 *Let $t, t' \in PAT$, and $\mathcal{A}' \in Accept\mathcal{PA}(t')$. We define the probabilistic cover of $\mathcal{A}'$ with respect to t' and t as follows:*

$$pC(\mathcal{A}', t', t) = \{\mathcal{A}_i \in Accept(\mathcal{PA}(t)) \mid \mathcal{A}_i \preceq \mathcal{A}' \text{ and } \pi(t', \mathcal{A}') \leq \sum_i \pi(t, \mathcal{A}_i)\}$$

Intuitively, every action the semantics process t' may perform from a state $\mathcal{A}'$ with a certain probability p, may be performed by the semantics process t from several states $\mathcal{A}_i$ that are more nondeterministic than $\mathcal{A}'$, such that $\pi(t', \mathcal{A}') \leq \sum_i \pi(t, \mathcal{A}_i)$. The idea behind this definition is to establish an order over semantic processes that follow the idea given in the testing semantics, i.e., whatever t' can do with certain probability (or set of probabilities), t can do with a wider set of probabilities which includes that of t'.

Definition 16 *Let $t, t' \in PAT$. We say that $t \leq_{PAT} t'$ if:*

- $L(t) \subseteq L(t')$,
- *for every $\mathcal{A}' \in Accept(\mathcal{PA}(t'))$ there is a probabilistic cover $pC(\mathcal{A}', t', t)$,*
- $\bigcup_{\mathcal{A}' \in Accept(\mathcal{PA}(t'))} pC(\mathcal{A}', t', t) = Accept(\mathcal{PA}(t))$
- $t/\langle \mathcal{A}_i, a\rangle \leq_{PAT} t'/\langle \mathcal{A}', a\rangle, \quad \forall a \in A(\mathcal{A}_i)$

In Figure 6 we can see an example of $t \leq_{PAT} t'$.

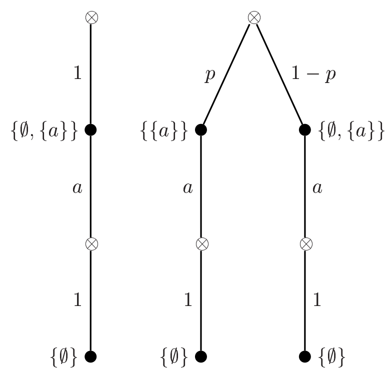

Figure 6: The left tree (t) is smaller than the right tree (t').

Proposition 2 *$(PAT, \leq_{PAT})$ is a partial order.*

Definition 17 *Let A be a set, and $\leq_A$ an order defined over A. A chain is a sequence of elements from A, a_0, a_1, $\ldots$, a_n, $\ldots$, such that*

$$a_0 \leq_A a_1 \leq_A \cdots \leq_A a_n \leq_A \cdots$$

In order to define the limit of a chain, we will use the terminology used in [14]: if X is a set we say the property P is true *almost always* in X if for all but a finite number of $x \in X$, P is true of x. If x_0, x_1,..., is an infinite sequence of elements from X, then there exists some $k \geq 0$ such that P is true of x_n for every $n \geq k$. If X is a chain in some partial order then it follows that there exists some $x \in X$ such that P is true of x' for every $x' \geq x$.

Definition 18 *Let $\{t_i\}_{i\in I}$ be a chain. We define the limit of the chain, represented by $\sqcup t_i$, as follows:*

$$\begin{aligned}
Seq(\sqcup t_i) =& \{s \mid s \in Seq(t_i) \text{ for almost all } t_i \in \{t_i\}\} \\
\mathcal{PA}((\sqcup t_i)/s) =& \{(p, \mathcal{A}) \mid \mathcal{A} \in Accept(\mathcal{PA}(t_i/s)) \text{ for almost all } t_i \in \{t_i\}, \\
& p = \lim_{i\to\infty} \pi(t_i/s, \mathcal{A})\}
\end{aligned}$$

Proposition 3 *The tree $\sqcup t_i$ is well defined.*

Theorem 1 *$(PA, \leq_{PAT})$ is a complete partial order (cpo).*

Once we have defined an order over semantic processes, we may establish an equivalence in PAT.

Definition 19 *$t \equiv_{PAT} t'$ if and only if $t \leq_{PAT} t' \wedge t' \leq_{PAT} t$.*

Proposition 4 *Let t,$t' \in PAT$. $t \equiv_{PAT} t'$ if and only if*

- $Seq(t) = Seq(t')$
- $\mathcal{PA}(t/s) = \mathcal{PA}(t'/s)$, $\forall s \in Seq(t)$

5.3 Semantics of the operators

We define the semantics of PNAL operators in a denotational style, by associating a semantic process $[\![P]\!]$ (i.e., a PAT) to each process P.

In this section we will make extensive use of two operators borrowed from [14]: c (convex closure) and u (pointwise union). Here we recall their formal definition and some properties.

Let $\mathcal{B}$ be an acceptance set. Then, $c(\mathcal{B})$ (convex closure of $\mathcal{B}$) is the least set such that

a) $\mathcal{B} \subseteq c(\mathcal{B})$

b) $X, Y \in c(\mathcal{B})$ implies $X \cup Y \in c(\mathcal{B})$ ($c(\mathcal{B})$ is $\cup$–closed)

c) $X, Y \in c(\mathcal{B})$, $X \subseteq Z \subseteq Y$ implies $Z \in c(\mathcal{B})$ ($c(\mathcal{B})$ is convex–closed)

Some of its more useful properties are

c1. If $\mathcal{B}$ is convex then $c(\mathcal{B}) = \mathcal{B}$

c2. If $\mathcal{B} \subseteq \mathcal{B}'$ then $c(\mathcal{B}) \subseteq c(\mathcal{B}')$

c3. $c(\mathcal{B}) = c(c(\mathcal{B}))$

c4. $c(\mathcal{B}_1) \cup c(\mathcal{B}_2 \cup \mathcal{B}_3) = c(\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3)$

c5. $c(\mathcal{B}_1 \cup \mathcal{B}_2) = c(c(\mathcal{B}_1) \cup c(\mathcal{B}_2))$

c6. If $A \in c(\mathcal{B})$ then $B \subseteq A$ for some $B \in \mathcal{B}$.

Let $\mathcal{B}_1, \mathcal{B}_2$ be acceptance sets. The pointwise union of $\mathcal{B}_1$ and $\mathcal{B}_2$ is defined as follows: $\mathcal{B}_1 \, u \, \mathcal{B}_2 = \{B_1 \cup B_2 \mid B_1 \in \mathcal{B}_1 \text{ and } B_2 \in \mathcal{B}_2\}$. Some useful properties are

du1. $(\mathcal{B}_1 \, u \, \mathcal{B}_2) \, u \, \mathcal{B}_3 = \mathcal{B}_1 \, u \, (\mathcal{B}_2 \, u \, \mathcal{B}_3)$

du2. $\mathcal{B}_1 \, u \, \mathcal{B}_2 = \mathcal{B}_2 \, u \, \mathcal{B}_1$

du3. $\mathcal{B} \, u \, \mathcal{B} = \mathcal{B}$, if $\mathcal{B}$ is convex

du4. $\mathcal{B}_1 \, u \, (\mathcal{B}_2 \cup \mathcal{B}_3) = (\mathcal{B}_1 \, u \, \mathcal{B}_2) \cup (\mathcal{B}_1 \, u \, \mathcal{B}_3)$

Its interaction with the operator c is given by:

c7. If $\mathcal{B}_1, \mathcal{B}_2$ are convex, $\mathcal{B}_1 \, u \, \mathcal{B}_2$ is convex

c8. $c(\mathcal{B}_1 \, u \, \mathcal{B}_2) = c(\mathcal{B}_1) \, u \, c(\mathcal{B}_2)$

Now, we may define the semantics of PNAL operators.

Stop. The semantic process $[\![\mathbf{0}]\!]$ is a tree with two nodes, one probabilistic (the root) and the other a closed node labelled by the acceptance set $\{\emptyset\}$. The arc connecting them is labelled by 1 (see figure 7).

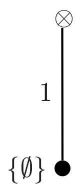

Figure 7: PAT of **0**

The probabilistic acceptance set at the first level is $\mathcal{PA}(\mathbf{0}) = \Big\{(1, \{\emptyset\})\Big\}$. Continuations $\mathcal{PA}(\mathbf{0}/s)$ where $s \neq \langle\rangle$ are not defined.

Divergence. The semantic process $[\![\Omega]\!]$ is a tree with two nodes, one probabilistic (the root) and the other an open node labelled by the acceptance set $\overline{\{\emptyset\}}$. The arc connecting them is labelled by 1 (see figure 8).

The probabilistic acceptance set at the first level is $\mathcal{PA}(\Omega) = \Big\{(1, \overline{\{\emptyset\}})\Big\}$. In this case we have $\mathcal{PA}(\Omega/s) = \mathcal{PA}(\Omega)$ for every sequence s.

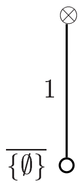

Figure 8: PAT of Ω

Prefix operator. The semantic process $a.t$ is defined from t by adding two new nodes to it: one probabilistic (the new root node) and one closed labelled by the acceptance set $\{\{a\}\}$. The arc connecting them is labelled by 1, and the arc connecting the closed node with the root node of t is labelled by a (see figure 9).

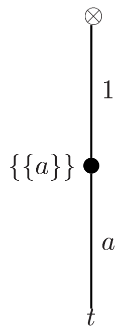

Figure 9: PAT of $a.t$

The probabilistic acceptance set at the first level is $\mathcal{PA}(a.t) = \Big\{\big(1, \{\{a\}\}\big)\Big\}$. Continuations are given by $a.t/\langle\{\{a\}\}a\rangle\hat{\ }s = t/s$.

Probabilistic choice. Let $t_1, t_2 \in PAT$. The first level of $t_1 \otimes_q t_2$ is obtained considering the branches at the first level in t_1 and t_2, weighted by q and $1-q$ respectively, and adding up the branches that lead to the same acceptance set.

The probabilistic acceptance set at the first level is:

$$\begin{aligned}\mathcal{PA}(t_1 \otimes_q t_2) = \{(p, \mathcal{A}) \,|\, &\mathcal{A} \in \mathit{Accept}(\mathcal{PA}(t_1)) \vee \mathcal{A} \in \mathit{Accept}(\mathcal{PA}(t_2)),\\ &p = q \cdot \pi(t_1, \mathcal{A}) + (1-q) \cdot \pi(t_2, \mathcal{A})\}\end{aligned}$$

In order to obtain the continuations, we must distinguish several cases where action a is performed from state $\mathcal{A}$:

- $\mathcal{A} \in \mathit{Accept}(\mathcal{PA}(t_1))$ and $\mathcal{A} \notin \mathit{Accept}(\mathcal{PA}(t_2))$. In this case only t_1 can execute the action a, so:

$$t/\langle\mathcal{A}, a\rangle = t_1/\langle\mathcal{A}, a\rangle$$

- $\mathcal{A} \notin \mathit{Accept}(\mathcal{PA}(t_1))$ and $\mathcal{A} \in \mathit{Accept}(\mathcal{PA}(t_2))$. This is the symmetric case:

$$t/\langle \mathcal{A}, a\rangle = t_2/\langle \mathcal{A}, a\rangle$$

- $\mathcal{A} \in \mathit{Accept}(\mathcal{PA}(t_1))$ and $\mathcal{A} \in \mathit{Accept}(\mathcal{PA}(t_2))$ Now action a can be carried out either by t_1 or t_2, then

$$t/\langle \mathcal{A}, a\rangle = t_1/\langle \mathcal{A}, a\rangle \otimes_q t_2/\langle \mathcal{A}, a\rangle$$

Internal choice. In order to define the internal choice operator, we need first to introduce some additional notation. We extend the convex closure operator, c, in order to take into account which kind of node (open or closed) is labelled by an acceptance set. The new operator, c_Ω , is defined as follows:

$$\begin{aligned}
\mathcal{A}\, c_\Omega\, \mathcal{B} =& c(\mathcal{A} \cup \mathcal{B})\\
\overline{\mathcal{A}}\, c_\Omega\, \mathcal{B} =& \overline{\{A(\overline{\mathcal{A}}) \cup A(\mathcal{B})\}}\\
\mathcal{A}\, c_\Omega\, \overline{\mathcal{B}} =& \overline{\{A(\mathcal{A}) \cup A(\overline{\mathcal{B}})\}}\\
\overline{\mathcal{A}}\, c_\Omega\, \overline{\mathcal{B}} =& \overline{\{A(\overline{\mathcal{A}}) \cup A(\overline{\mathcal{B}})\}}
\end{aligned}$$

Let $t_1, t_2 \in PAT$. The states at the first level of $t = t_1 \oplus t_2$ are obtained as $\mathcal{A} = \mathcal{B}\, c_\Omega\, \mathcal{C}$, where $\mathcal{B}$ and $\mathcal{C}$ are acceptance sets at the first level of t_1 and t_2 respectively. Note that every acceptance set $\mathcal{A}$ may be obtained from different pair of states $\mathcal{B}$ and $\mathcal{C}$.

The probabilistic acceptance set at the first level is:

$$\begin{aligned}
\mathcal{PA}(t_1 \oplus t_2) = \{(p, \mathcal{A}) \mid\, & \exists \mathcal{B} \in \mathit{Accept}(\mathcal{PA}(t_1)), \exists \mathcal{C} \in \mathit{Accept}(\mathcal{PA}(t_2))\\
& \text{such that } \mathcal{A} = \mathcal{B}\, c_\Omega\, \mathcal{C},\\
& p = \sum_{\mathcal{A} = \mathcal{B}\, c_\Omega\, \mathcal{C}} \pi(t_1, \mathcal{B})) \cdot \pi(t_2, \mathcal{C})\}
\end{aligned}$$

In order to obtain the continuations after the sequence $\langle \mathcal{A} a\rangle$, $t/\langle \mathcal{A} a\rangle$, we can distinguish the following cases:

- $a \in A(\mathcal{B})$ and $a \notin A(\mathcal{C})$. In this case t_1 executes a, giving us: $t/(\mathcal{A}, a) = t_1/(\mathcal{B}, a)$.
- $a \notin A(\mathcal{B})$ and $a \in A(\mathcal{C})$. Symmetric case: $t/(\mathcal{A}, a) = t_2/(\mathcal{C}, a)$.
- $a \in A(\mathcal{B})$ and $a \in A(\mathcal{C})$. Either t_1 or t_2 execute the action, in which case: $t/(\mathcal{A}, a) = t_1/(\mathcal{B}, a) \oplus t_2/(\mathcal{C}, a)$

Therefore, as $\mathcal{A}$ may be obtained from several different sets $\mathcal{B}$ and $\mathcal{C}$, we have:

$$\begin{aligned}
t/(\mathcal{A}, a) = & \bigoplus_{\substack{\mathcal{B}, \mathcal{C}, \mathcal{B}\, c_\Omega\, \mathcal{C} = \mathcal{A}\\ a \in A(\mathcal{B}) \wedge a \notin A(\mathcal{C})}} t_1/(\mathcal{B}, a) \oplus \bigoplus_{\substack{\mathcal{B}, \mathcal{C}, \mathcal{B}\, c_\Omega\, \mathcal{C} = \mathcal{A}\\ a \notin A(\mathcal{B}) \wedge a \in A(\mathcal{C})}} t_2/(\mathcal{C}, a) \oplus\\
& \bigoplus_{\substack{\mathcal{B}, \mathcal{C}, \mathcal{B}\, c_\Omega\, \mathcal{C} = \mathcal{A}\\ a \in A(\mathcal{B}) \wedge a \in A(\mathcal{C})}} (t_1/(\mathcal{B}, a) \oplus t_2/(\mathcal{C}, a))
\end{aligned}$$

External choice. This operator is very similar to internal choice. The only difference is at the first level of the tree. As in the previous case, we need to extend the pointwise union operator, u , in order to take into account which kind of node (open or closed) is labelled by an acceptance set. The new operator, u_Ω , is defined as follows:

$$
\begin{aligned}
\mathcal{A} \, u_\Omega \, \mathcal{B} &= \mathcal{A} \; u \; \mathcal{B} \\
\overline{\mathcal{A}} \, u_\Omega \, \mathcal{B} &= \overline{\{A(\overline{\mathcal{A}}) \cup A(\mathcal{B})\}} \\
\mathcal{A} \, u_\Omega \, \overline{\mathcal{B}} &= \overline{\{A(\mathcal{A}) \cup A(\overline{\mathcal{B}})\}} \\
\overline{\mathcal{A}} \, u_\Omega \, \overline{\mathcal{B}} &= \overline{\{A(\overline{\mathcal{A}}) \cup A(\overline{\mathcal{B}})\}}
\end{aligned}
$$

Let $t_1, t_2 \in PAT$. The states at the first level of $t = t_1 + t_2$ are obtained as $\mathcal{A} = \mathcal{B} \, u_\Omega \, \mathcal{C}$, where $\mathcal{B}$ and $\mathcal{C}$ are acceptance sets at the first level of t_1 and t_2 respectively. As in the internal choice case, every acceptance set $\mathcal{A}$ may be obtained from different pairs of states $\mathcal{B}$ and $\mathcal{C}$.

The probabilistic acceptance set at the first level is:

$$
\begin{aligned}
\mathcal{PA}(t_1 + t_2) = \{(p, \mathcal{A}) \mid \exists \mathcal{B} \in \mathit{Accept}(\mathcal{PA}(t_1)), \exists \mathcal{C} \in \mathit{Accept}(\mathcal{PA}(t_2))& \\
\text{such that } \mathcal{A} = \mathcal{B} \, u_\Omega \, \mathcal{C},& \\
p = \sum_{\mathcal{A} = \mathcal{B} \, u_\Omega \, \mathcal{C}} \pi(t_1, \mathcal{B})) \cdot \pi(t_2, \mathcal{C})\}&
\end{aligned}
$$

Continuations are identical to the internal choice case, i.e.,

$$
(t_1 + t_2)/\langle \mathcal{A} a \rangle = (t_1 \oplus t_2)/\langle \mathcal{A} a \rangle
$$

Restriction. In order to define the semantics of the process $t \backslash C$ we will use the following notation: $\overline{C} = \{\overline{a} \mid a \in C\}$ and $\mathcal{A} \backslash C = \{B - (C \cup \overline{C}) \mid B \in \mathcal{A}\}$.

Intuitively, the semantic process we obtain when we apply restriction with respect to a set C, is a tree such that every action belonging to C or $\overline{C}$ is removed from acceptance sets and arcs. This may lead to a situation where different acceptance sets become identical, so we first need to add up probabilities, and later, in order to obtain continuations, we need to propagate probabilities to the following levels. In the last case we need to use a normalization factor. An example is shown in figure 10

The acceptance set at the first level is:

$$
\mathcal{PA}(t \backslash C) = \{(p, \mathcal{A}) \mid \mathcal{A} = \mathcal{B} \backslash C \wedge p = \sum_{\mathcal{A} = \mathcal{B} \backslash C} \pi(t, \mathcal{B})\}
$$

Continuations, if $a \notin C$, are defined as follows:

$$
(t \backslash C)/\langle \mathcal{A} a \rangle = \bigotimes_{\mathcal{A} = \mathcal{B} \backslash C} \left[\frac{\pi(t, \mathcal{B})}{\pi(t \backslash C, \mathcal{A})} \right] (t/\langle \mathcal{B} a \rangle) \backslash C
$$

If $a \in C$ then continuation is not defined.

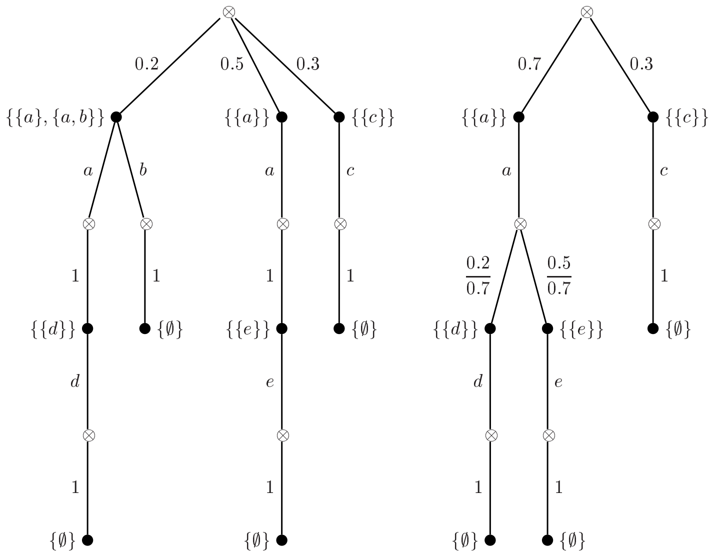

Figure 10: t and $t \setminus \{b\}$.

Parallel composition The definition of the function over PAT corresponding to $|$ is quite difficult, and a considerable amount of notation is needed. We will first introduce intuitively the concepts related to this operator and then we will give its formal definition.

Let us consider the processes $P_1 = a.(c \otimes_{0.4} e) \otimes_{0.3} b$, $P_2 = \overline{a}.d$ and $P = P_1|P_2$. In Figure 11 we can see the PAT of P_1, P_2 and the first level of $P_1|P_2$.

At the first level, P may perform the actions that P_1 or P_2 may perform at their first level. Furthermore, in the event of communication between P_1 and P_2, some actions of the following level in P_1 and P_2 are promoted to the first level in P. We have to consider all the possibilities, i.e., every action may or may not communicate with its homologous barred action, and this introduce some kind of nondeterminism.

In this example, actions a and $\overline{a}$ may communicate, so second level action, like c, e and d, may be performed at the first level. Let us assume that a and $\overline{a}$ communicate, and the probabilistic choice after action a in P_1 is resolved following the branch labelled by probability 0.4. In this case, with probability $0.3 \times 0.4 = 0.12$, the semantic process P reaches a nondeterministic acceptance set $\mathcal{A}$ such that $A(\mathcal{A}) = \{a, \overline{a}, c, d\}$. In case of communication, only actions c and d may be performed. Nondeterminism is introduced in $\mathcal{A}$ taking into account the convex closure of the union of sets $\{c, d\}$ and $\{a, \overline{a}, c, d\}$.

In order to give the formal definition of parallel composition we need to introduce some preliminary definitions.

Definition 20 *Let $t \in PAT$. The set of maximal sequences of t is defined as*

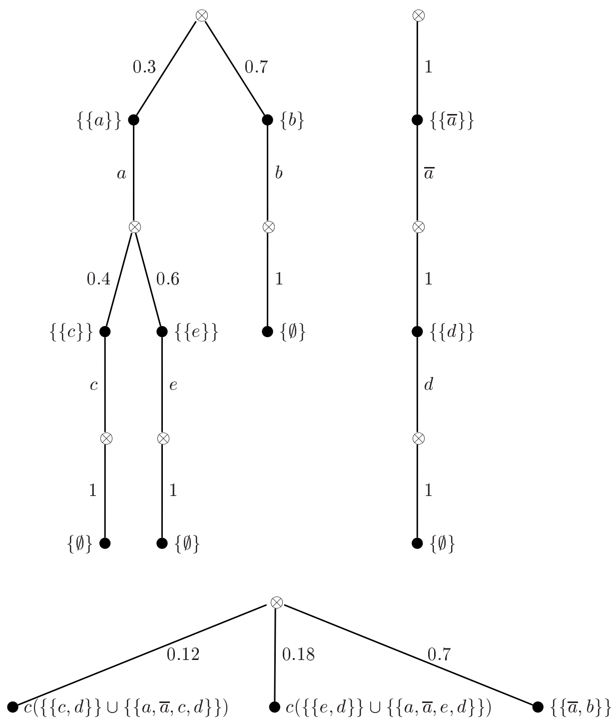

Figure 11: PAT of P_1, P_2 and first level of $P_1|P_2$

follows:

$$\begin{aligned} Seq_{Max}(t) = \{s \in Seq(t) \mid \forall \mathcal{A} \in Accept(\mathcal{PA}(t/s)) \\ \text{we have } \mathcal{A} = \{\emptyset\} \text{ or } \mathcal{A} = \overline{\{\emptyset\}}\} \end{aligned}$$

We say that t_m is a maximal subtree of t if $\forall s \in Seq_{Max}(t_m)$ we have $s \in Seq_{Max}(t)$

Now we introduce the concept of *probabilistic resolution*, a maximal subtree such that every probabilistic node has only one outgoing branch.

Definition 21 *Let $t \in PAT$. A probabilistic resolution t_e of t is a maximal subtree of t such that $|\mathcal{PA}(t_e/s)| = 1$ for every sequence $s \in Seq(t_e)$. The set of probabilistic resolutions of $t \in PAT$ will be denoted by $P\mathrm{Res}(t)$. $\boldsymbol{D}(P\mathrm{Res})$ is the domain of probabilistic resolutions.*

In the following definitions, we will relate a probabilistic resolution with an acceptance tree (AT, [14]) and a probability.

Definition 22 *Let $s = \langle \mathcal{A}_1 a_1 \mathcal{A}_2 a_2 \ldots \mathcal{A}_n a_n \rangle$ be a sequence. We obtain the trace of this sequence by removing the information of the acceptance sets, i.e., $trace(s) = a_1 a_2 \ldots a_n$.*

We will use $\overline{trace}(s)$ to denote the trace such that each action is replaced by its homologous barred action. As usual, we will assume that $\overline{\overline{a}} = a$.

Given a probabilistic resolution t_r, we obtain its associated AT, denoted $NPT(t_r)$ (non probabilistic tree), as follows:

i) $L(t_{AT}) = L(t_r)$ and

ii) for every $s \in Seq(t_r)$, if $\mathcal{B} \in Accept(\mathcal{PA}(t_r/s))$ then $\mathcal{A}(t_{AT}(trace(s))) = \mathcal{B}$

And its associated probability, denoted $\pi_{PR}(t_r)$, as follows:

$$\pi_{PR}(t_r) = \begin{cases} \pi(t_r, \mathcal{B}) \cdot \displaystyle\prod_{\substack{b \in A(\mathcal{B}) \\ \mathcal{B} \in Accept(\mathcal{PA}(t_r))}} \pi_{PR}(t/\langle \mathcal{B} b \rangle) \\ p \qquad \text{if } \exists p \in (0,1] \text{ s.t. } \mathcal{PA}(t_r) = \{(p, \{\emptyset\}) \text{ or } \mathcal{PA}(t_r) = \{(p, \overline{\{\emptyset\}})\} \end{cases}$$

We can now obtain the first level of the PAT $t|t'$, as follows:

$$\mathcal{PA}(t|t') = \big\{(r, \mathcal{B}) \mid \exists t_i \in PRes(t) \text{ and } \exists t'_j \in PRes(t') \text{ s.t. } \\ \mathcal{B} = \mathcal{A}((NPT(t_i)|NPT(t'_j))(\varepsilon)) \text{ and } r = \sum_{\mathcal{B}} \pi_{PR}(t_i) \cdot \pi_{PR}(t'_j)\big\}$$

In order to obtain $(t|t')/\langle \mathcal{B} b \rangle$ we need to introduce more definitions.

Definition 23 *We define the subset of probabilistic resolutions from which we obtain the same acceptance set $\mathcal{B}$ as follows:*

$$SubsetPRes(t, t', \mathcal{B}) = \{t_i \in PRes(t) \mid \exists t'_j \in PRes(t') \text{ s.t. } \\ \mathcal{B} = \mathcal{A}((NPT(t_i) \mid NPT(t'_j))(\varepsilon))\}$$

Definition 24 *Let $\mathcal{PA} = \{(p_1, \mathcal{A}_1), \ldots, (p_n, \mathcal{A}_n)\}$ be a probabilistic acceptance set. We define the normalization function as follows:*

$$Norm(\mathcal{PA}) = \left\{ (q_i, \mathcal{A}_i) \mid \exists (p_i, \mathcal{A}_i) \in \mathcal{PA} \text{ and } q_i = \frac{p_i}{\sum_j p_j} \right\}$$

Now, we define a function, $PATfromPRes(S_{PR})$, which allows us a to build a PAT from a set of probabilistic resolutions.

Definition 25 *Let $S_{PR} \in \boldsymbol{D}(PRes)$ be a set of probabilistic resolutions. The tree $t = PATfromPRes(S_{PR})$ is obtained as follows:*

i) $Seq(t) = \displaystyle\bigcup_{t_i \in S_{PR}} Seq(t_i)$

ii) $\mathcal{PA}(t/s) = Norm\left(\displaystyle\bigcup_{s \in Seq(t_i)} \mathcal{PA}(t_i/s) \right)$

Definition 26 *Let s and s′ be sequences and $a \in \mathcal{A}ct$. We define the Boolean synchronization function as follows:*

$$\begin{aligned} sync(s, s', a) = &(trace(s) = \overline{trace}(s')\hat{\ }a) \ or \\ &(trace(s') = \overline{trace}(s)\hat{\ }, a) \end{aligned}$$

Finally we define $(t|t')/\langle \mathcal{B}b \rangle$. Let $t_1 = PATfromPRes(SubsetPRes(t, t', \mathcal{B}))$ and $t'_1 = PATfromPRes(SubsetPRes(t', t, \mathcal{B}))$, then

$$(t|t')/\langle \mathcal{B}b \rangle = \bigoplus_{s \in Seq(t_1), s' \in Seq(t'_1), sync(s,s',b)=true} (t_1/s) \ | \ (t'_1/s')$$

Proposition 5 *Every PNAL operator is monotonous and continuous*

Recursion. To define the semantics of $recX.P(X)$, where $P(X)$ is a process in which X is a free variable, we use the fixed point technique.

As a consequence of theorem 1 and proposition 5 we may define the semantics of the recursion operator as the least fixed point of the corresponding functional, which coincides with the limit of the finite approximations:

$$[\![recX.P(X)]\!] = \sqcup [\![P_i]\!]$$

where $P_0 = \Omega, P_1 = P(\Omega), \ldots, P_i = P^i(\Omega)$.

One interesting consequence of mixing up probabilities and nondeterminism is that, while probabilistic choice allows us to distinguish between the different reachable states of a process, internal (and also external) choice works in the opposite way, gathering states that previously were different. This may also affect continuations, as can be seen in the following example.

Example 2 *Let $P = P_1 \oplus P_2$, where $P_1 = b.c$ and $P_2 = a.d \otimes_{0.4} (a.e \oplus b.f)$.*

In this example all the reachable states in processes P_1 and P_2 are gathered into the state $\{\{a\}, \{b\}, \{a, b\}\}$ in process P, which is reached with probability 1.

This means that continuations after executing action a are obtained as

$$P_2/(\{\{a\}\})a) \oplus P_2/(\{\{a\}, \{b\}, \{a, b\}\}a)$$

i.e., distinguishable continuations in P_2 now become indistinguishable due to the internal choice. PATs of these processes are shown in Figure 12.

Finally, we state that the equivalence relation between probabilistic processes is an extension of that presented in [14] for nondeterministic processes.

Theorem 2 *Let $P, Q \in EPL$ be two processes,*

$$[\![P]\!]_{AT} = [\![Q]\!]_{AT} \text{ if and only if } [\![P]\!]_{PAT} = [\![Q]\!]_{PAT}$$

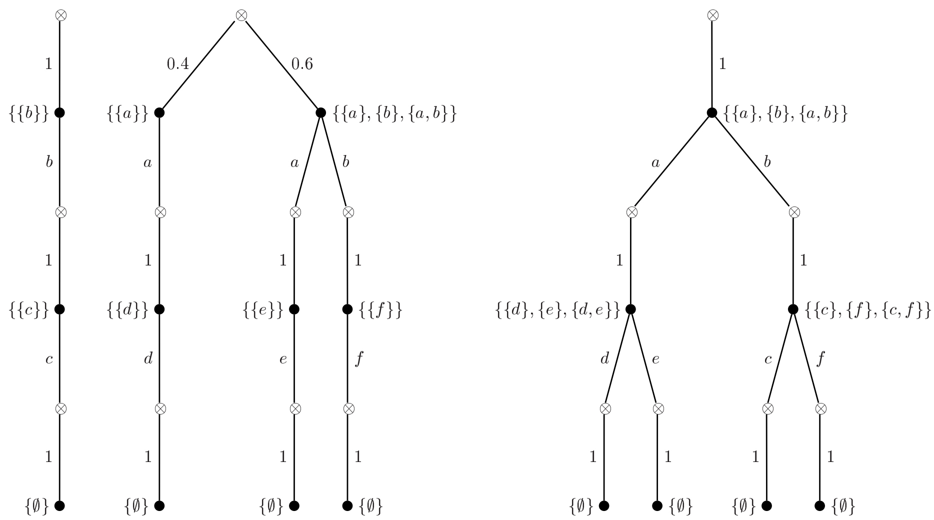

Figure 12: PAT of $b.c$, $a.d \otimes_{0.4} (a.e \oplus b.f)$ and $b.c \oplus (a.d \otimes_{0.4} (a.e \oplus b.f))$

5.4 Full abstraction

In this section we will prove that both the testing semantics and the denotational semantics identify the same processes, i.e., the denotational semantics is fully abstract with respect to the testing semantics.

First, we will consider a reduced version of the language, PNAL_1, that includes $\mathbf{0}$, prefix operator and the three choices, and the set of test we can derive from PNAL_1, called $\mathcal{T}_{PNAL_1}$. We will prove that denotational and testing semantics are fully abstract when we restrict ourselves to this kind of process and test. Afterwards, we will consider an extension of PNAL_1, PNAL_2, which includes recursion. As in the previous case, the derived set of tests will be called $\mathcal{T}_{PNAL_2}$, and we will extend the previous result to recursive processes and tests. We will denote $\equiv_{test_1}$ and $\equiv_{test_2}$ the testing equivalences induced by the tests in $\mathcal{T}_{PNAL_1}$ and $\mathcal{T}_{PNAL_2}$ respectively.

Finally, we need to consider restriction operator and parallel composition in order to extend the full abstraction result to processes in PNAL. We leave this to the following section, where we introduce proof system, and we prove that restriction and parallel composition are derived operators, and thus, every term in PNAL may be specified in terms of PNAL_2.

In order to prove this result we need to introduce some concepts. First, we define the pass of a test over a semantic process. Later, we give a constructive method to obtain a PAT of a process of which we know only the responses to tests, i.e., $\sup\{\phi_{((P,T))}\}$ and $\inf\{\phi_{((P,T))}\}$. Finally, we prove that the results of the tests over syntactic and semantic processes are identical, and also that the PAT obtained from responses to tests and from syntactic terms are equivalent.

First, we introduce the set of probabilities with which a semantic process passes a test.

Definition 27 *Given a process $P \in PNAL_1$ and a test $T \in \mathcal{T}_{PNAL_1}$, the set of probabilities $\phi^D_{(P,T)}$ is defined as follows:*

$$\begin{aligned}\phi^D_{(P,T)} = &\{Prob(\{\{w\}\}, [\![P \| T]\!]) \,, \\ & Prob(\{\{w\}\}, [\![P \| T]\!]) + Prob(\{\emptyset, \{w\}\}, [\![P \| T]\!])\}\end{aligned}$$

Now, let us see how to construct the PAT obtained from $\phi_{(P,T)}$, denoted by $[\![P]\!]^T$:

Construction of $[\![P]\!]^T$. The set of the successful tests $T = a_1.a_2.\cdots.a_k.w$ gives us the language the process accepts, $L(P)$.

Once we know $L(P)$, we also know the set of actions the process may perform at each moment, and the set of possible acceptance sets ($\mathcal{A}_i$), but we do not know the probabilities associated with these acceptance sets (q_i). In order to obtain this set of probabilities we will use an essential set of tests. As an intermediate step, we will use logical expressions associated with acceptance sets. We shall now see how we may obtain this set of essential logical expressions.

Let C be the set of possible actions and $\mathcal{A}_i$ an acceptance set such that $\emptyset \notin \mathcal{A}_i$ and $C \in \mathcal{A}_i$. Every set $B_j \in \mathcal{A}_i$ and every acceptance set $\mathcal{A}_i$ may be related, respectively, to a logical expression in the form $E(B_j) = \bigwedge_{a_i \in B_j} a_i$, and $E(\mathcal{A}_i) = \bigvee_{B_j} E(B_j)$. The logical expression $E(\mathcal{A}_i)$ is true only for the acceptance set $\mathcal{A}_i$ and for every subset of $\mathcal{A}_i$.

We will use the following notation:

- E^A_A is the set of logical expressions that contain only actions in A.
- E^B_A is the set of logical expressions in the form $E^A_A \wedge (\bigwedge_{a \in B-A} \neg a)$, where $A \subseteq B$.
- $\mathcal{A}_A$ is the collection of acceptance sets that contain only actions in A.
- q_A are the probabilities associated to $\mathcal{A}_A$.

The whole set of logical expressions may be obtained by using the following procedure, starting with step 1 and repeating steps 2 and 3. In each cycle, only one action is added.

1) $E^\emptyset_\emptyset = \{true\}$.

2) Inheritance. If $a \notin A$ then $E^{A \cup \{a\}}_A = \{e \wedge \neg a \,|\, e \in E^A_A\}$

3) New expressions. $E^A_A = \{E(\mathcal{A}_i) \,|\, A \in \mathcal{A}_i\} \cup \{true\}$

Finally, the set of essential logical expressions is $\bigcup_{A \subseteq \mathcal{A}ct} E_A^{\mathcal{A}ct}$.

For example, let $\{a, b\}$ be the set of possible actions. The collection of possible acceptance sets is: $\{\emptyset\}$, $\{\emptyset, \{a\}\}$, $\{\{a\}\}$, $\{\emptyset, \{b\}\}$, $\{\{b\}\}$, $\{\emptyset, \{a\}, \{b\}, \{a, b\}\}$, $\{\{a\}, \{b\}, \{a, b\}\}$, $\{\{a\}, \{a, b\}\}$, $\{\{b\}, \{a, b\}\}$, $\{\{a, b\}\}$.

The set of associated logical expressions is:
$E_\emptyset^\emptyset = \{true\}$,
$E_\emptyset^{\{a\}} = \{true \wedge \neg a\} = \{\neg a\}$, $E_{\{a\}}^{\{a\}} = \{true, a\}$,
$E_\emptyset^{\{b\}} = \{true \wedge \neg b\} = \{\neg b\}$, $E_{\{b\}}^{\{b\}} = \{true, b\}$,
$E_\emptyset^{\{a,b\}} = \{\neg a \wedge \neg b\}$, $E_{\{a\}}^{\{a,b\}} = \{a \wedge \neg b, \neg b\}$, $E_{\{b\}}^{\{a,b\}} = \{b \wedge \neg a, \neg a\}$,
$E_{\{a,b\}}^{\{a,b\}} = \{true, a \vee b \vee (a \wedge b), a \vee (a \wedge b), b \vee (a \wedge b), a \wedge b\}$.

and the set of essential logical expressions is: $\bigcup_{A \subseteq \{a,b\}} E_A^{\{a,b\}}$.

Now, we associate one real number in $[0, 1]$ to every essential logical expression. We do this by using the following rules:

- We associate $\inf\{\phi_{(P,w)}\} = 1$ to the logical expression *true*.
- Let F be a logical expression with non-negative premises. We associate $\inf\{\phi_{(P,T_F)}\}$ to this expression, where T_F is a test obtained from F as follows:
  - Every action a_i is replaced with $a_i w$.
  - Every $\wedge$ connective is replaced with $\oplus$.
  - Every $\vee$ connective is replace with $+$.
- Let F' be a logical expression in the form $F \wedge (\bigwedge_i \neg a_i)$. We associate $\inf\{\phi_{(P,T_{F'})}\}$ with F' where $T_{F'}$ is a test in the form $T_F + (\sum_i a_i)$.
- Let F'' be a logical expression in the form $\neg F$. We associate $1-\sup\{\phi_{(P,T_F)}\}$ to F''.

We extend the notation previously introduced: $T_A^{\mathcal{A}ct}$ is the set of real numbers in $[0, 1]$ which are associated with the logical expressions $E_A^{\mathcal{A}ct}$.

Now, we can write a linear system by using the responses to the tests. The unknown variables are the probabilities associated to every acceptance set. Each equation is related to a value in $T_A^{\mathcal{A}ct}$, and consequently to an acceptance set.

We need to check whether we may distinguish between different states or not. To that end we sort the equations in the system by using two criteria. Each equation is related to a value in $T_A^{\mathcal{A}ct}$, so the first criterion we may use is based on the actions in the set A. We start with the equations related to $T_{\mathcal{A}ct}^{\mathcal{A}ct}$ (group of equations $\mathcal{A}ct$) and we finish with the equation related to $T_\emptyset^{\mathcal{A}ct}$ (group $\emptyset$). Later, every set of equations related to the group A_i is sorted by using an inclusion relation between the acceptance sets in $\mathcal{A}_{A_i}$; the first equation corresponds to the set $\{A\}$. If two sets are not related, the order is not significant. This order may be extended to the probabilities q_i.

Let $\mathbb{A}$ be the matrix of the system. In this matrix we may distinguish submatrices of the form A_α^β which relate $T_\alpha^{\mathcal{A}ct}$ to q_β.

Each test may be satisfied for its associated acceptance set $\mathcal{A}$ and for every subset of $\mathcal{A}$. Due to the introduced order, every submatrix A_α^α is always lower–triangular and therefore non–singular. On the other hand, the inheritance method prevents a test of the group A from being satisfied by an acceptance set of a group B such that $B - A \neq \emptyset$. The system we obtain is:

$$\begin{bmatrix} T_{a_1\cdots a_n}^{a_1\cdots a_n} \\ T_{a_1\cdots a_{n-1}}^{a_1\cdots a_n} \\ \cdots \\ T_{a_2\cdots a_n}^{a_1\cdots a_n} \\ \cdots \\ T_{a_1}^{a_1\cdots a_n} \\ \cdots \\ T_{a_n}^{a_1\cdots a_n} \\ T_{\emptyset}^{a_1\cdots a_n} \end{bmatrix} = \left[\ \mathbb{A}\ \right] \times \begin{bmatrix} q_{a_1 a_2\cdots a_n} \\ q_{a_1 a_2\cdots a_{n-1}} \\ \cdots \\ q_{a_2 a_3\cdots a_n} \\ \cdots \\ q_{a_1} \\ \cdots \\ q_{a_n} \\ q_\emptyset \end{bmatrix}$$

where $[\mathbb{A}]$ is the matrix shown in Fig. 13:

$$\begin{bmatrix} A_{a_1\cdots a_n}^{a_1\cdots a_n} & A_{a_1\cdots a_n}^{a_1\cdots a_{n-1}} & \cdots & A_{a_1\cdots a_n}^{a_2\cdots a_n} & \cdots & A_{a_1\cdots a_n}^{a_1} & \cdots & A_{a_1\cdots a_n}^{a_n} & A_{a_1\cdots a_n}^{\emptyset} \\ 0 & A_{a_1\cdots a_{n-1}}^{a_1\cdots a_{n-1}} & \cdots & 0 & \cdots & A_{a_1\cdots a_{n-1}}^{a_1} & \cdots & A_{a_1\cdots a_{n-1}}^{a_n} & A_{a_1\cdots a_{n-1}}^{\emptyset} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & A_{a_2\cdots a_n}^{a_2\cdots a_n} & \cdots & A_{a_2\cdots a_n}^{a_1} & \cdots & A_{a_2\cdots a_n}^{a_n} & A_{a_2\cdots a_n}^{\emptyset} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & \cdots & A_{a_1}^{a_1} & \cdots & 0 & A_{a_1}^{\emptyset} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & A_{a_n}^{a_n} & A_{a_n}^{\emptyset} \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & A_{\emptyset}^{\emptyset} \end{bmatrix}$$

Figure 13: Matrix $[\mathbb{A}]$

Finally, we have to prove that the system has a unique solution. Solving the system in a bottom-up way we may obtain $q_\emptyset$, whose values we then use to obtain $q_{a_1}, \ldots, q_{a_n}$, and so on. Finally, we can write:

$$\left[\ T_{a_1\cdots a_n}^{a_1\cdots a_n}\ \right] - \left[\ A_{a_1\cdots a_n}^{a_1\cdots a_{n-1}} \quad \cdots \quad A_{a_1\cdots a_n}^{\emptyset}\ \right] \times \begin{bmatrix} q_{a_1\cdots a_{n-1}} \\ \cdots \\ q_\emptyset \end{bmatrix} = \left[\ A_{a_1\cdots a_n}^{a_1\cdots a_n}\ \right] \times \left[\ q_{a_1\cdots a_n}\ \right]$$

and this system has a unique solution because $A_{a_1\cdots a_n}^{a_1\cdots a_n}$ is a non–singular matrix.

Now we know the probability of every acceptance set in the first level of the tree. In order to obtain these values for the whole tree we need to know $\phi_{(P/(a,\mathcal{A}),T)}$.

Definition 28 *(Test process after an action a from a state $\mathcal{A}$, $P/(a,\mathcal{A})\|T$)*

- $P = \mathit{0} \qquad P/(a,\mathcal{A})\|T = \{0\}$
- $P = a.P_1$
  $P/(b,\mathcal{A})\|T = \begin{cases} P_1\|T & \text{if } a = b \wedge \mathcal{A} = \{\{a\}\} \\ \{0\} & \text{otherwise} \end{cases}$
- $P = P_1 \otimes_q P_2$.
  $P/(a,\mathcal{A})\|T = \{q \cdot x_i + (1-q)\cdot y_i \mid x_i \in P_1/(a,\mathcal{A})\|T \wedge y_i \in P_2/(a,\mathcal{A})\|T\}$
- $P = P_1 \oplus P_2$.
  $$P/(a,\mathcal{A})\|T = \bigcup_{\mathcal{A}=c(\mathcal{B}\cup\mathcal{C})} [P_1/(a,\mathcal{B})\|T \cup P_2/(a,\mathcal{C})\|T]$$
- $P = P_1 + P_2$.
  $$P/(a,\mathcal{A})\|T = \bigcup_{\mathcal{A}=\mathcal{B}\ u\ \mathcal{C}} [P_1/(a,\mathcal{B})\|T \cup P_2/(a,\mathcal{C})\|T]$$

Definition 29 *(Obtaining $[\![P]\!]^T$ from testing semantics information)*
Using the constructive method introduced and Def. 28 we obtain the probability associated with each acceptance set for a process P/s (where s is a sequence) . The tree $[\![P]\!]^T$ is built just putting together this information (every sequence identify only one node).

Lemma 1 *Let $P \in PNAL_1$ be a process. For every test $T \in \mathcal{T}_{PNAL_1}$:*

$$\inf\{\phi_{(P,T)}\} = \inf\{\phi^D_{(P,T)}\} \text{ and } \sup\{\phi_{(P,T)}\} = \sup\{\phi^D_{(P,T)}\}$$

Lemma 2 *Let $P \in PNAL_1$ be a process. Then $[\![P]\!] = [\![P]\!]^T$.*

Theorem 3 $\forall P, Q \in PNAL_1$, $P \equiv_{test_1} Q \Longleftrightarrow P \equiv_{PAT} Q$.

In order to extend this result to the recursive case, we need to prove that recursive tests do not give any information, the set of essential tests thus remaining unchanged.

Proposition 6 *Let $P, Q \in PNAL_2$. Then*

$$P \equiv_{test_1} Q \text{ if and only if } P \equiv_{test_2} Q$$

Theorem 4 *(Full abstraction)* $\forall P, Q \in PNAL_2$, $P \equiv_{test_2} Q \Longleftrightarrow P \equiv_{PAT} Q$.

As mentioned earlier, we need to consider restriction operator and parallel composition in order to extend the full abstraction result to processes in PNAL. This point will be covered in the next section, where we introduce the proof system and we prove that restriction and parallel composition are, as usual, derived operators.

6 Proof system

In this section we present a proof system, which is sound and complete with respect to the denotational semantics. This proof system is based on that of EPL, by including the axioms related to the probabilistic choice operator. Nevertheless, some of the axioms of EPL are not preserved in the general case. For instance, the internal and the external choice now fail to be idempotent as we showed before, and distributivity of the internal over the external choice (and vice versa) is not maintained. However, these axioms can be used when we are working with a special kind of process (probabilistically stable processes) and they are very useful when proving completeness.

6.1 Axioms and soundness

Table 4 contains the set of axioms related to the following operators: **0**, prefix, the three choices and recursion. Axioms in Table 4 show that probabilistic choice is commutative (P1), associative in a probabilistic way (P2), and idempotent (P3). These axioms allow us to generalize probabilistic choice to an arbitrary number of arguments. Thus we can write $\bigotimes_i [p_i]P_i$ to denote a probabilistic choice among n processes, each one with probability $p_i > 0$. Other rules state that probabilistic choice is propagated to the following level when both processes are prefixed by the same action (P4), probabilistic choice distributes through the internal and the external choice (PI1, PE1), and that probabilistic choice is greater than internal choice (PI2).

Note that the axioms maintained from EPL are as follows: the internal and the external choice are commutative and associative (I1,I2,E1,E2), the external choice has a zero (E3), the internal and the external choice between processes prefixed with the same action are propagated to the following level (I3,IE1), internal choice is smaller than external choice (IE2), and finally, divergence does not differentiate between internal and external choice (ΩIE1).

Finally, we have axiom PI3, which will be very useful when proving completeness.

The axioms in Table 5 (marked by the symbol $\downarrow$), do not hold in the general case, but if we restrict P, Q and R to a kind of process, which we call *probabilistically stable processes*, they become true. These axioms are: idempotency of the internal and the external choice (I4$\downarrow$, E4$\downarrow$), and distributivity of the internal choice over the external choice and vice versa (IE3$\downarrow$, IE4$\downarrow$).

Example 3 *Let $P = a.\boldsymbol{0} \otimes_{0.5} b.\boldsymbol{0}$, We have*

$$\begin{aligned}
\mathcal{PA}(P/\varepsilon) &= \{(0.5, \{\{a\}\}), (0.5, \{\{b\}\})\} \\
\mathcal{PA}((P \oplus P)/\varepsilon) &= \{(0.25, \{\{a\}\}), (0.25, \{\{b\}\}), (0.5, \{\{a\}, \{b\}, \{a, b\}\})\} \\
\mathcal{PA}((P + P)/\varepsilon) &= \{(0.25, \{\{a\}\}), (0.25, \{\{b\}\}), (0.5, \{\{a, b\}\})\}
\end{aligned}$$

Therefore, $P \neq P \oplus P$ and $P \neq P + P$. PATs of these processes are shown in Figure 14.

P1	$P \otimes_p Q \equiv Q \otimes_{1-p} P$
P2	$(P \otimes_p Q) \otimes_q R \equiv P \otimes_{p \cdot q} (Q \otimes_{\frac{(1-p)\cdot q}{1-p\cdot q}} R)$
P3	$P \otimes_p P \equiv P$
P4	$a.P \otimes_p a.Q \equiv a(P \otimes_p Q)$
PI1	$P \oplus (Q \otimes_p R) \equiv (P \oplus Q) \otimes_p (P \oplus R)$
PE1	$P + (Q \otimes_p R) \equiv (P + Q) \otimes_p (P + R)$
I1	$P \oplus Q \equiv Q \oplus P$
I2	$P \oplus (Q \oplus R) \equiv (P \oplus Q) \oplus R$
I3	$a.P \oplus a.Q \equiv a(P \oplus Q)$
E1	$P + Q \equiv Q + P$
E2	$P + (Q + R) \equiv (P + Q) + R$
E3	$P + \mathbf{0} \equiv P$
IE1	$a.P + a.Q \equiv a.P \oplus a.Q$
PI2	$P \oplus Q \leq P \otimes_p Q$
IE2	$P \oplus Q \leq P + Q$
PI3	$(P \oplus Q) \otimes_p R \leq P \otimes_p R$
	if $L((P \oplus Q) \otimes_p R) = L(P \otimes_p R)$ and $P\downarrow$, $Q\downarrow$ and $R\downarrow$
ΩIE1	$\Omega \oplus P \equiv \Omega + P$

Table 4: Set of axioms for **0**, prefix, choice operators and recursion

I4↓	$P \oplus P \equiv P$
E4↓	$P + P \equiv P$
IE3↓	$P + (Q \oplus R) \equiv (P + Q) \oplus (P + R)$
IE4↓	$P \oplus (Q + R) \equiv (P \oplus Q) + (P \oplus R)$

Table 5: Axioms for probabilistically stable processes

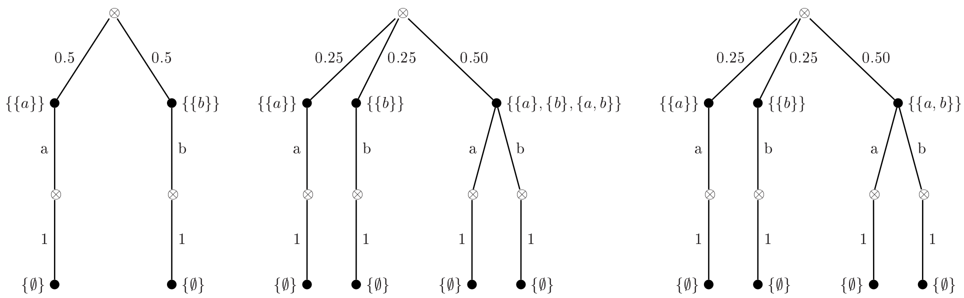

Figure 14: PAT of $P = a.\mathbf{0} \otimes_{0.5} b.\mathbf{0}$, $P \oplus P$ and $P + P$.

We may also apply, over probabilistically stable processes, the derived equations shown in Table 6, which are useful to prove completeness.

Finally, in Table 7 we can see the axioms for restriction and parallel com-

Der1⇃	$P \oplus Q \equiv P \oplus Q \oplus (P \oplus Q)$
Der2⇃	$P \oplus (P + Q + R) \equiv$
	$\equiv P \oplus (P + Q) \oplus (P + Q + R)$
Der3⇃	$(a.P_1 + R_1) \oplus (a.P_2 + R_2) \equiv$
	$\equiv (a.(P_1 \oplus P_2) + R_1) \oplus (a.(P_1 \oplus P_2) + R_2)$

Table 6: Some derived equations for probabilistically stable processes in PNAL

position, which, as usual, are derived operators. In rule PaE1, the function $comm(P,Q)$ (defined in [14]) is false if there is no possibility of communication between P and Q.

Note that axioms PaI1 and PaE1, which are introduced in [14], hold only for processes in EPL; applying repeatedly P4, PI1, PE1, ReP1, and PaP1 (distribution axioms) each process in the form $P|Q$ may be transformed into $\bigotimes_{i,j}[p_i \cdot q_j]P_i|Q_j$, where $P_i, Q_j \in$ EPL.

RE1	$(P + Q)\backslash C \equiv P\backslash C + Q\backslash C$
RI1	$(P \oplus Q)\backslash C \equiv P\backslash C \oplus Q\backslash C$
R1	$a.P\backslash C \equiv \begin{cases} a.(P\backslash C) & \text{if } a \notin C \vee \overline{a} \notin C \\ \mathbf{0} & \text{otherwise} \end{cases}$
R01	$\mathbf{0}\backslash C \equiv \mathbf{0}$
RP1	$(P \otimes_q Q)\backslash C \equiv P\backslash C \otimes_q Q\backslash C$
PaP1	$P\|(Q \otimes_q R) \equiv (P\|Q) \otimes_q (P\|R)$
Pa01	$\mathbf{0}\|P \equiv P$
PaI1	$(P \oplus Q)\|R \equiv P\|R \oplus Q\|R, \qquad P, Q, R \in$ EPL
PaE1	if P, Q are $\sum\{a_i P_i, i \in I\}$, $\sum\{b_j Q_j, j \in J\}$ then $P\|Q \equiv \begin{cases} ext(P,Q) \text{ if } comm(P,Q) \text{ is false} \\ (ext(P,Q) + int(P,Q)) \oplus int(P,Q) \\ \qquad \text{otherwise} \end{cases}$ where $ext(P,Q) = \sum\{a_i(P_i\|Q), i \in I\} + \sum\{b_j(P\|Q_j), j \in J\}$, $int(P,Q) = \bigoplus\{P_i\|Q_j, a_i = \overline{b_j}\}$. and $P, Q \in$ EPL

Table 7: Axioms for restriction and parallel composition

Example 4 *Consider the process in Figure 11 (page 22).*

$$\begin{aligned}(a.(c \otimes_{0.4} e) \otimes_{0.3} b)|\overline{a}.d &\stackrel{P4}{\equiv} ((a.c \otimes_{0.4} a.e) \otimes_{0.3} b)|\overline{a}.d \\ &\stackrel{P2}{\equiv} ([0.12]a.c \otimes [0.18]a.e \otimes [0.3]b)|\overline{a}.d \\ &\stackrel{PaP1}{\equiv} [0.12]a.c|\overline{a}.d \otimes [0.18]a.e|\overline{a}.d \otimes [0.3]b|\overline{a}.d\end{aligned}$$

Finally, we apply axiom PaE1 as usual.

The set of rules in the proof system are shown in Table 8. Rules $R1$–$R8$ are the classical rules in EPL. In rule R8, $App(P) = \{P^n \mid n \in \mathbb{N}\}$ is the set finite approximations of P. There is a new rule $R9$, related to recursion operator, which has been introduced for a technical reason: real numbers with their natural order do not constitute an ω–algebraic domain. This rule is needed in order to prove completeness (lemma 3).

R1] Reflexivity	$\dfrac{}{P \leq P}$	
R2] Antisymmetry	$\dfrac{P \leq P' \qquad P' \leq P}{P \equiv P'}$	
R3] Transitivity	$\dfrac{P \leq P' \qquad P' \leq P''}{P \leq P''}$	
R4] Substitution	$\dfrac{P_1 \leq P_1', \ldots, P_k \leq P_k'}{f(P_1, \ldots, P_k) \leq f(P_1', \ldots, P_k')}$	for every f in PNAL
R5] Inequation	$\dfrac{}{P \leq P'}$	for every inequation $P \leq P'$
R6] Ω-rule	$\dfrac{}{\Omega \leq P}$	for every $P \in$ PNAL
R7] Recursion	$\dfrac{}{recX.P \equiv P[recX.P/X]}$	
R8] w-Induction	$\dfrac{\text{for every } d_i \in App(P), d_i \leq P'}{P \leq P'}$	
R9] w-Approximation	$\dfrac{\alpha(n) \to 0}{P \equiv \sqcup(\Omega \otimes_{\alpha(n)} P)}$	

Table 8: Rules of the proof system

As we indicated above, this proof system is sound with respect to the denotational semantics:

Theorem 5 *Axioms and rules are sound with respect to the denotational semantics.*

6.2 Completeness

As usual, to prove the completeness, we look for the adequate normal forms. Essentially, these normal forms represent the different ways a process has to complete its execution. This leads us to a generalized probabilistic choice at the top, followed by a generalized internal choice between a set of states, which is followed by a generalized prefixed external choice between the actions in this set, whose continuations are also in normal form.

Definition 30 *(Normal forms)*

- *Processes* ***0*** *and* Ω *are in normal form.*
- *If* $\mathcal{A}_i$ *is a convex set and for every* $a \in A(\mathcal{A}_i)$ *there is a normal form* $n(\mathcal{A}_i, a)$*, then*

$$\bigotimes_i [q_i] \bigoplus_{A \in \mathcal{A}_i} \sum_{a \in A} a.n(\mathcal{A}_i, a)$$

  is a normal form.

These normal forms share some characteristics of classical models (CSP, EPL), as well as characteristics of other probabilistic models [6]. Thus, the continuations after performing the same action in different probabilistic branches do not need to be identical (usual in probabilistic models), and the acceptance sets $\mathcal{A}_i$ must be convex sets as usual in classical models.

Theorem 6 *(of normal forms) For every process* $P \in PNAL$ *there exists one and only one normal form* $nf(P)$ *(modulo associativity and commutativity) such that* $P \equiv nf(P)$*.*

Lemma 3 *Let* $P, Q \in PNAL$ *be two processes in normal form. Then*

$$[\![P]\!]_{PAT} \leq [\![Q]\!]_{PAT} \Longrightarrow P \leq Q$$

Theorem 7 *Let* $P, Q \in PNAL$ *be two processes, then*

$$[\![P]\!]_{PAT} \leq [\![Q]\!]_{PAT} \Longrightarrow P \leq Q$$

In the next example, given two processes n and m, we show how we can use the proof system in order to prove that $n \leq m$.

Example 5 *Let* $n = (a \oplus b) \otimes_{0.7} (a \oplus (a + b))$ *and* $m = a \otimes_{0.4} (a + b)$*. We shall prove that* $n \leq m$*.*

For process n *we have*

$$\begin{array}{ll} \mathcal{A}_1 = \{\{a\}, \{b\}, \{a, b\}\} & p_1 = 0.7 \\ \mathcal{A}_2 = \{\{a\}, \{a, b\}\} & p_2 = 0.3 \end{array}$$

and for process m

$$\begin{array}{ll} \mathcal{B}_1 = \{\{a\}\} & q_1 = 0.4 \\ \mathcal{B}_2 = \{\{a, b\}\} & q_2 = 0.6 \end{array}$$

Sets $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *are probabilistically covered by* $\mathcal{A}_1$ *and* $\mathcal{A}_2$. *Following the technique presented in the proof of Lemma 3, we can write the following linear system of equations*

$$\begin{aligned} 0.4 &= k_{11} \cdot 0.7 + k_{21} \cdot 0.3 \\ 0.6 &= k_{12} \cdot 0.7 + k_{22} \cdot 0.3 \\ k_{11} + k_{12} &= 1 \\ k_{21} + k_{22} &= 1 \end{aligned}$$

One solution to this system is $k_{22} = 0$ $k_{12} = \frac{0.6}{0.7}$, $k_{11} = \frac{0.1}{0.7}$, *and* $k_{21} = 1$. *With these values we obtain* ($p_{ij} = p_i \times k_{ij}$)*:* $p_{11} = 0.1$, $p_{12} = 0.6$, $p_{21} = 0.3$ *and* $p_{22} = 0$.

Applying repeatedly axioms P3, PI3 and again P3 over process n *we obtain:*

$$\begin{aligned} n &= [0.1](a \oplus b) \otimes [0.6](a \oplus b) \otimes [0.3](a \oplus (a + b)) \\ &\leq [0.1]a \otimes [0.6](a + b) \otimes [0.3]a \\ &= [0.4]a \otimes [0.6](a + b) \\ &= m \end{aligned}$$

Finally, by using theorems 4, 5 and 7, we can relate the proof system with the testing semantics as follows:

Corollary 1 *Let* $P, Q \in PNAL$ *be two processes, then*

$$P \equiv_{test} Q \text{ if and only if } P \equiv Q$$

7 Conclusions and comparisons

In this paper we have presented an algebraic theory of probabilistic and nondeterministic processes which is a probabilistic extension of Hennessy's theory presented in [14]. An operational and a testing semantics have been introduced, as well as a denotational semantics and a proof system. The testing semantics, the denotational semantics and the proof system are shown to be fully abstract.

To our knowledge, this paper is the first that presents a general framework of probabilistic and nondeterministic processes based on Hennessy's model. Similar approaches have been presented, based on different classical process algebras. Thus, in [1] an operational semantics, a probabilistic bisimulation and a proof system are introduced for a probabilistic extension of ACP. In [23], a denotational semantics and a proof system are introduced for a probabilistic extension of CSP. In [22], only an operational and a denotational semantics are defined for another probabilistic version of CSP which does not include a hiding operator. In [20] only an action–guarded probabilistic choice is considered, and they therefore avoid the problems associated with an internal choice of probabilistic processes (or vice versa), i.e., internal and probabilistic choices are separated. Finally, in [31, 17] a probabilistic extension of CCS is presented, and an operational semantics, a testing semantics and a denotational semantics are defined.

In order to deal with probabilism and nondeterminism we have chosen an approach that considers that a probabilistic choice is resolved before an internal choice. Thus, the transition system we obtain is very similar to "bundle probabilistic transition systems" (BPTS) presented in [7], which are the converse of the simple model of [26, 25]. In our case (as in BPTS), a set of nondeterministic alternatives is chosen with a certain probability, while in [26, 25], an action is chosen nondeterministically and then, for each action, a distribution of probabilities over the successor states is given (reactive probabilistic choice).

A similar operational semantics is defined in [31], but in this case some of the premises of the rules include negative conditions over transitions, which is problematic (see [12]). In our model we have avoided this problem by introducing the "probabilistic stable" predicate, which is defined syntactically.

The decision of resolving probabilistic choices first leads us to a model in which some of the properties of classical process algebras are not maintained. For instance, the internal choice now fails to be idempotent, as the following example shows. Let P be the process $a.\mathbf{0} \otimes_{0.5} b.\mathbf{0}$, we then have:

$$\begin{aligned} \mathcal{PA}(P) =& \{(0.5, \{\{a\}\}), (0.5, \{\{b\}\})\} \\ \mathcal{PA}(P \oplus P) =& \{(0.25, \{\{a\}\}), (0.25, \{\{b\}\}), (0.5, \{\{a\}, \{b\}, \{a, b\}\})\} \end{aligned}$$

This is, however, a known consequence of introducing probabilities and nondeterminism by taking the approach of first resolving the probabilistic choices. We encounter the same problem in some of the aforementioned papers [1, 23, 31], and also in [30], where a probabilistic extension of CCS is presented by introducing intervals of probabilities that allow us to specify processes with uncertain information.

The normal forms are defined in a very natural way. They consist in a generalized probabilistic choice at the top, followed by a generalized internal choice between a set of states, which is followed by a generalized prefixed external choice between the actions in this set, whose continuations are also in normal form. But, as we could see in Section 6, distribution axioms allow us to define an alternative normal form that consists in a probabilistic choice followed by a non–probabilistic process in EPL's normal form, i.e., each probabilistic choice must be resolved before any internal choice. The same conclusion is obtained in [23] for a probabilistic extension of CSP.

Finally, the proof system we have obtained is very similar to Andova's proof system [1]. One noteworthy feature is that, in her case, the proof system is sound and complete with respect to probabilistic bisimulation, while in our case it is sound and complete with respect to the testing semantics.

Acknowledgements

The authors would like to thank Manuel Nuñez for some very rewarding discussions about probabilistic and nondeterministic processes.

Appendix A. Proofs

Proof of Proposition 1. First we assume that $P \sqsubseteq_{Hen} Q$ holds, and we prove that $P \sqsubseteq_{test} Q$. For each test, we can distinguish four different cases:

- $P\ \mathit{may}\ T$: this means, in terms of probabilities, that $\phi_{(P,T)} = \{0,1\}$ or $\phi_{(P,T)} = \{1\}$.
- $P\ \mathit{must}\ T$: this means that $\phi_{(P,T)} = \{0\}$.
- $P \mathit{m\not{a}y} T$: this means that $\phi_{(P,T)} = \{0\}$.
- $P \mathit{m\not{u}st} T$: this means that $\phi_{(P,T)} = \{0,1\}$ or $\phi_{(P,T)} = \{0\}$.

Furthermore, each test may or may not satisfy the relations *may* and *must*. If we consider the processes P and Q, such that $P \sqsubseteq_{Hen} Q$, we have, again, four different cases:

- $P\ \mathit{may}\ T$ and $P\ \mathit{must}\ T$: in this case the only possibility, probabilistically speaking, is $\phi_{(P,T)} = \{1\}$. Therefore $\phi_{(Q,T)} = \{1\}$.
- $P\ \mathit{may}\ T$ and $P \mathit{m\not{u}st} T$: in this case $\phi_{(P,T)} = \{0,1\}$. For the process Q we have two possibilities, $\phi_{(Q,T)} = \{0,1\}$ or $\phi_{(Q,T)} = \{1\}$.
- $P \mathit{m\not{a}y} T$ and $P\ \mathit{must}\ T$: this case is impossible.
- $P \mathit{m\not{a}y} T$ and $P \mathit{m\not{u}st} T$: in this case $\phi_{(P,T)} = \{0\}$ and $\phi_{(Q,T)}$ may have any value.

According to these cases, for every test such that $\phi_{(P,T)} \neq \{0\}$, conditions given in definition 7 hold, and therefore $P \sqsubseteq_{test} Q$.

Now we consider the reverse implication. If $P \sqsubseteq_{test} Q$, we have two possibilities:

- $\phi_{(P,T)} = \{0,1\}$; then $\phi_{(Q,T)} = \{0,1\}$ or $\phi_{(Q,T)} = \{1\}$.
- $\phi_{(P,T)} = \{0,1\}$; then $\phi_{(Q,T)} = \{1\}$.

In both cases it holds that $1 \in \phi_{(P,T)}$ implies $1 \in \phi_{(Q,T)}$, i.e., $P\ \mathit{may}\ T$ implies $Q\ \mathit{may}\ T$. Furthermore, $0 \notin \phi_{(P,T)}$ implies $0 \notin \phi_{(Q,T)}$, i.e., $P\ \mathit{must}\ T$ implies $Q\ \mathit{must}\ T$. Taking into account both results we have $P \sqsubseteq_{Hen} Q$. □

Proof of Proposition 3. We have to prove that $\sqcup t_i$ belongs to the semantic domain. Rules D1, D2, D3 and D6 hold trivially, because every t_i of the chain satisfies them. The rest of the rules are related to finding the limit when we obtain more deterministic trees.

Given a tree in PAT, there are four methods we can use to obtain a tree that is more deterministic than the one given:

- By replacing an acceptance set $\mathcal{A}$ with another one $\mathcal{A}'$ such that $\mathcal{A}' \subseteq \mathcal{A}$ and $A(\mathcal{A}') = A(\mathcal{A})$. Eventually, it will be necessary to add up probabilities in the event of two or more branches leading to nodes labelled by the same acceptance set.

- By dividing a probabilistic branch, labelled by q, which leads to a closed node labelled by $\mathcal{A}$, into two or more probabilistic branches, labelled by p_i, such that $\sum_i p_i = q$, which lead to closed nodes variously labelled by $\mathcal{B}_i$ such that $\forall i \mathcal{B}_i \subseteq \mathcal{A}$ and $A(\mathcal{B}_i) = A(\mathcal{A})$.

- By dividing a probabilistic branch, labelled by q, which leads to an open node labelled by $\overline{\mathcal{A}}$, into two or more probabilistic branches, labelled by p_i, such that $\sum_i p_i = q$, which lead to open nodes variously labelled by $\overline{\mathcal{B}}_i$ such that $\forall i A(\overline{\mathcal{B}}_i) = A(\overline{\mathcal{A}})$.

- By replacing an open node with a closed node labelled by $\mathcal{A}$. For each action in $A(\mathcal{A})$ we have a leaving branch that leads to a PAT.

We can see that, in any case, the limit exists.

Case 1: Acceptance sets are finite, so that, from one tree t_d in the chain it must be true that $\mathcal{PA}(t'/s) = \mathcal{PA}(t_d/s)$ for every t' such that $t_d \leq_{PAT} t'$; otherwise, we could have an infinite sequence $\mathcal{A}_1 \supset \mathcal{A}_2 \supset \ldots$, but this sequence is not possible because acceptance sets are finite. At the limit, we see that every closed node is labelled by a deterministic acceptance set, i.e., $\mathcal{A} = A(\mathcal{A})$.

Cases 2 and 3: Since acceptance sets are finite, every probabilistic branch may be divided into a finite number of branches. Thus, the limit exists.

Case 4: This case is ω–algebraic in depth (probabilities only appear in width), so we use the general result which says that the semantics of a process defined recursively is the limit of the chain of finite approximations. $\square$

Proof of Theorem 1. We have to prove that there exists a least element, and that every chain has a least upper bound.

The least element is a tree t with two nodes, one probabilistic (the root) and the other an open node labelled by the acceptance set $\overline{\{\emptyset\}}$. The arc connecting them is labelled by 1. For this tree we have $Seq(t) = \{\varepsilon\}$ and $\pi(t, \overline{\{\emptyset\}}) = 1$. We shall prove that this is the least element.

Let be $t' \in$ PAT. We then have:

- $L(t) = \{\varepsilon\} \subseteq L(t')$
- $pC(\mathcal{A}', t', t) = \{\overline{\{\emptyset\}}\}$ for every $\mathcal{A}' \in Accept(\mathcal{PA}(t'))$
- $\displaystyle\bigcup_{\mathcal{A}' \in Accept(\mathcal{PA}(t'))} pC(\mathcal{A}', t', t) = Accept(\mathcal{PA}(t)) = \{\overline{\{\emptyset\}}\}$

Therefore, $t \leq_{\text{PAT}} t'$ for every $t' \in$ PAT. Now, let us see that the least upper bound is $\sqcup t_i$. First we will see that it is in fact an upper bound, i.e., $\forall\, t_i \in \{t_i\}$, $t_i \leq_{\text{PAT}} \sqcup t_i$. If this were not true, we could find a tree t_s in the chain from where, for every t' such that $t_s \leq_{\text{PAT}} t'$, we will have that $t' \not\leq_{\text{PAT}} \sqcup t_i$.

This means that we can find an infinite number of trees in the chain for which conditions given to define $\sqcup t_i$ (*almost always*) do not hold. Therefore, $\sqcup t_i$ must be an upper bound.

In order to prove that this is a *lub*, let us consider an upper bound $t_M \neq \sqcup t_i$ such that $t_M \leq_{\text{PAT}} \sqcup t_i$. This means that at least one of the following situations holds:

- There exists at least one close node in t_M labelled by an acceptance set that is more nondeterministic that its equivalent node in $\sqcup t_i$.
- There exists at least one open node in t_M such that its equivalent node in $\sqcup t_i$ is closed.
- There exists at least one open node in t_M labelled by an acceptance set that precedes its equivalent (also open) in $\sqcup t_i$.

The first case is not possible because, at the limit, every closed node is labelled by a deterministic acceptance set. The second case is not possible because otherwise we could find a tree t_j in the chain whereby $\forall j \geq i \;\; t_j \not\leq_{\text{PAT}} t_M$ (because the equivalent node is closed, as in $\sqcup t_i$). Finally, the third case is not possible because we could find a tree t_i in the chain such that an open node in t_M would precede its equivalent open node in t_i, and therefore $t_i \not\leq_{\text{PAT}} t_M$.

Therefore, it holds that $\sqcup t_i \leq_{\text{PAT}} t_M$ and $\sqcup t_i$ is a *lub*. □

Proof of Proposition 4 (Sketch). The left to right implication is trivial. The right to left implication can be shown by contradiction. Let us suppose that $t \equiv_{\text{PAT}} t'$ and that right to left implication does not hold. This means that we can find at least a probabilistic acceptance set (after a sequence s) such that $\mathcal{PA}(t/s) \neq \mathcal{PA}(t'/s)$. As $t \leq_{PAT} t'$, each acceptance set $\mathcal{A}' \in \mathit{Accept}(\mathcal{PA}(t'/s))$ has a probabilistic cover $pC(\mathcal{A}', t', t)$. At the same time, since $t' \leq_{PAT} t$, we see that every acceptance set $\mathcal{A} \in \mathit{Accept}(\mathcal{PA}(t/s))$ has a probabilistic cover $pC(\mathcal{A}, t, t')$. The only situation in which both conditions hold is when $\mathcal{PA}(t/s) = \mathcal{PA}(t'/s)$.

Therefore, $\mathit{Seq}(t) = \mathit{Seq}(t')$ and $\mathcal{PA}(t/s) = \mathcal{PA}(t'/s)$ for every sequence s. □

Proof of Proposition 5 (Sketch). Here we only present the proof corresponding to internal choice. As very similar ideas can be applied to prove the remaining cases, these proofs are omitted.

In order to prove the internal choice case, we first need to prove that c_Ω is monotonous, i.e., if $\mathcal{A} \preceq \mathcal{B}$ then $\mathcal{A}\, c_\Omega\, \mathcal{C} \preceq \mathcal{B}\, c_\Omega\, \mathcal{C}$. We can distinguish the following cases:

- In the case of the acceptance sets labelling closed nodes ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$) we have to prove that: $\mathcal{A} \subseteq \mathcal{B} \Rightarrow c(\mathcal{A} \cup \mathcal{C}) \subseteq c(\mathcal{B} \cup \mathcal{C})$. This expression is proved simply by taking into account that

$$\mathcal{A} \subseteq \mathcal{B} \Rightarrow \mathcal{A} \cup \mathcal{C} \subseteq \mathcal{B} \cup \mathcal{C}$$

and the property $c2$ of convex closure (page 16)

$$\mathcal{B} \subseteq \mathcal{B}' \Rightarrow c(\mathcal{B}) \subseteq c(\mathcal{B}')$$

- If $\mathcal{C}$ labels an open node ($\overline{\mathcal{C}}$), the expression we have to prove is $A(\mathcal{A}) \cup A(\overline{\mathcal{C}}) \subseteq A(\mathcal{B}) \cup A(\overline{\mathcal{C}})$, which follows directly from $\mathcal{A} \subseteq \mathcal{B} \Rightarrow A(\mathcal{A}) \subseteq A(\mathcal{B})$ and the properties of union of sets.

- If $\mathcal{B}$ labels an open node ($\overline{\mathcal{B}}$), then, since $\mathcal{A} \preceq \overline{\mathcal{B}}$, it must hold that $\mathcal{A}$ also labels an open node ($\overline{\mathcal{A}}$). The expression to prove in this case is $A(\overline{\mathcal{A}}) \cup A(\mathcal{C}) \subseteq A(\overline{\mathcal{B}}) \cup A(\mathcal{C})$, which is straightforward.

- Finally, if $\mathcal{A}$ labels an open node ($\overline{\mathcal{A}}$), and $\mathcal{B}$ and $\mathcal{C}$ closed ones, we have $\overline{\mathcal{A}}\, c_\Omega\, \mathcal{C} = \overline{\{A(\overline{\mathcal{A}}) \cup A(\mathcal{C})\}}$ preceding $\mathcal{B}\, c_\Omega\, \mathcal{C} = c(\mathcal{B} \cup \mathcal{C})$, because the former labels an open node and the latter a closed node.

We can now prove monotonicity and continuity of internal choice operator.

- Monotonicity: $t \leq_{\mathrm{PAT}} t' \Longrightarrow t \oplus t'' \leq_{\mathrm{PAT}} t' \oplus t''$

  We use induction on the depth of the tree.

  As $t \leq_{\mathrm{PAT}} t'$, at the first level it holds that for every $\mathcal{A}' \in \mathit{Accept}(\mathcal{PA}(t'))$ there is a probabilistic cover $pC(\mathcal{A}')$ whose elements are called $\mathcal{A}_i$.

  If we denote the states reachable at the first level of t'' as $\mathcal{C}_k$, $\mathcal{C}_k \in \mathit{Accept}(\mathcal{PA}(t''))$, due to the monotonicity of c_Ω operator we have

  $$\mathcal{A}_i \preceq \mathcal{A}' \Rightarrow \mathcal{A}_i\, c_\Omega\, \mathcal{C}_k \preceq \mathcal{A}'\, c_\Omega\, \mathcal{C}_k$$

  For probabilities, if q_k is the probability of t'' to reach state $\mathcal{C}_k$ at the first level, we have

  $$\pi(t', \mathcal{A}') \leq \sum_i \pi(t, \mathcal{A}_i) \Rightarrow q_k \cdot \pi(t', \mathcal{A}') \leq q_k \cdot \sum_i \pi(t, \mathcal{A}_i)$$

- Continuity: $\sqcup t_i \oplus t'' \equiv_{\mathrm{PAT}} \sqcup (t_i \oplus t'')$

  Let $\{t_i\}_{i \in I}$ be a chain of semantic processes. As internal choice is continuous it follows that $\{t_i \oplus t''\}_{i \in I}$ is also a chain of semantic processes.

  For every semantic process $t_i \oplus t''$ we know

  $$\pi(t_i \oplus t'', \mathcal{A}) = \sum_{\mathcal{A} = \mathcal{B}\, c_\Omega\, \mathcal{C}} \pi(t_i, \mathcal{B}) \cdot \pi(t'', \mathcal{C})$$

At the limit

$$\begin{aligned}
\pi(\sqcup(t_i \oplus t''), \mathcal{A}) &= \lim \pi(t_i \oplus t'', \mathcal{A}) \\
&= \lim \sum_{\mathcal{A}=\mathcal{B}\ c_\Omega\ \mathcal{C}} \pi(t_i, \mathcal{B}) \cdot \pi(t'', \mathcal{C}) \\
&= \sum_{\mathcal{A}=\mathcal{B}\ c_\Omega\ \mathcal{C}} \lim(\pi(t_i, \mathcal{B}) \cdot \pi(t'', \mathcal{C})) \\
&= \sum_{\mathcal{A}=\mathcal{B}\ c_\Omega\ \mathcal{C}} (\lim \pi(t_i, \mathcal{B})) \cdot \pi(t'', \mathcal{C}) \\
&= \sum_{\mathcal{A}=\mathcal{B}\ c_\Omega\ \mathcal{C}} \pi(\sqcup t_i, \mathcal{B})) \cdot \pi(t'', \mathcal{C}) \\
&= \pi(\sqcup t_i \oplus t'', \mathcal{A})
\end{aligned}$$

□

Proof of Theorem 2 (Sketch). We have only to consider that:

- In nondeterministic processes there are no probabilistic choices, and every probabilistic node will have a single outgoing arc labelled by probability 1.
- The construction of acceptance sets and probabilistic acceptance sets is identical for every operator of EPL.

□

Proof of Lemma 1 (Sketch). We use structural induction on the structure of the terms. We shall first look at the internal choice case:

- $P = P_1 \oplus P_2$. The set of resolutions of the test process $P \| T$ is the union of sets or resolutions of $P_1 \| T$ and $P_2 \| T$. Then, we have

  $$\inf\{\phi_{(P,T)}\} = \inf\{\inf\{\phi_{(P_1,T)}\}, \inf\{\phi_{(P_2,T)}\}\}$$

  $$\sup\{\phi_{(P,T)}\} = \sup\{\sup\{\phi_{(P_1,T)}\}, \sup\{\phi_{(P_2,T)}\}\}$$

  On the other hand, $[\![P \| T]\!]$ is obtained as $[\![P_1 \| T]\!] \oplus [\![P_2 \| T]\!]$. In this case also the set of resolutions over the tree is the union of the resolutions of $[\![P_1 \| T]\!]$ and $[\![P_2 \| T]\!]$:

  $$\inf\{\phi^D_{(P,T)}\} = \inf\{\inf\{\phi^D_{(P_1,T)}\}, \inf\{\phi^D_{(P_2,T)}\}\}$$

  $$\sup\{\phi^D_{(P,T)}\} = \sup\{\sup\{\phi^D_{(P_1,T)}\}, \sup\{\phi^D_{(P_2,T)}\}\}$$

□

Proof of Lemma 2 (Sketch). We use induction on the structure of the terms. We will focus only on the internal choice case. $P = P_1 \oplus P_2$. We know that:

$$\mathcal{PA}([\![P]\!]) = \{(p, \mathcal{A}) \mid \mathcal{A} = c(\mathcal{B} \cup \mathcal{C}), \mathcal{B} \in \mathit{Accept}(\mathcal{PA}(P_1)), \mathcal{C} \in \mathit{Accept}(\mathcal{PA}(P_2)),$$
$$p = \sum_{\mathcal{A}=c(\mathcal{B}\cup\mathcal{C})} \pi([\![P_1]\!], \mathcal{B}) \cdot \pi([\![P_2]\!], \mathcal{C})\}.$$

With respect to the testing semantics, once we know $\mathit{Accept}(\mathcal{PA}([\![P_1]\!]))$ and $\mathit{Accept}(\mathcal{PA}([\![P_2]\!]))$, we can obtain $\mathit{Accept}(\mathcal{PA}([\![P]\!]))$ by using the constructive definition of $[\![P]\!]^T$. The set of reachable acceptance sets from $[\![P]\!]$ may be taken as $\mathcal{A} = c(\mathcal{B} \cup \mathcal{C})$ where $\mathcal{B}$ and $\mathcal{C}$ are the reachable acceptance sets from $[\![P_1]\!]$ and $[\![P_2]\!]$.

Given a test T, $\inf\{\phi_{(P,T)}\}$ is obtained by adding up the contributions of every state which is always successful. For every state $\mathcal{A}_i$, its associated probability is calculated as the product of the probabilities related to the states $\mathcal{B}_i$ and $\mathcal{C}_i$ from which we obtain $\mathcal{A}_i = c(\mathcal{B}_i \cup \mathcal{C}_i)$ because we need both states $\mathcal{B}_i$ and $\mathcal{C}_i$ simultaneously to obtain $\mathcal{A}_i$. $\square$

Proof of Theorem 3. This follows directly from lemmas 1 and 2 $\square$

Proof of Proposition 6. The right to left implication is obvious. The proof of the left to right implication is done by contradiction, following the ideas presented in [11].

First we assume that $P \equiv_{test_1} Q$ and that there exists a recursive test T such that $\sup\{\phi_{(P,T)}\} \neq \sup\{\phi_{(Q,T)}\}$ or $\inf\{\phi_{(P,T)}\} \neq \inf\{\phi_{(Q,T)}\}$. Let us assume that $\sup\{\phi_{(P,T)}\} \neq \sup\{\phi_{(Q,T)}\}$ (the proof when $\inf\{\phi_{(P,T)}\} \neq \inf\{\phi_{(Q,T)}\}$ is similar).

Each element in the set $\phi_{(P,T)}$ is related to a resolution. Let R_P be the resolution from which we obtain $\sup\{\phi_{(P,T)}\}$. We can do the same with $\phi_{(Q,T)}$; let R_Q be the resolution from which we obtain $\sup\{\phi_{(Q,T)}\}$. By using definition 5, we have:

$$\begin{aligned}\sup\{\phi_{(P,T)}\} &= \lim_{n\to\infty} \sum_{C \in \hat{C}(R_P)} \{Pr(C) \mid \mathit{length}(C) < n\} \\ &\neq \lim_{n\to\infty} \sum_{C \in \hat{C}(R_Q)} \{Pr(C) \mid \mathit{length}(C) < n\} = \sup\{\phi_{(Q,T)}\}\end{aligned}$$

where $\mathit{length}(C)$ represents the number of transitions in computation C.

Let $\#\mathit{trans}(C)$ be the addition of the number of probabilistic transitions ($\longrightarrow_p$) and the number of transitions labelled by w ($\xrightarrow{w}$) in C. If we have the previous inequality it is obvious that we also have:

$$\begin{aligned}\sup\{\phi_{(P,T)}\} &= \lim_{n\to\infty} \sum_{C\in\hat{C}(R_P)} \{Pr(C) \mid \#trans(C) < n\} \\ &\neq \lim_{n\to\infty} \sum_{C\in\hat{C}(R_Q)} \{Pr(C) \mid \#trans(C) < n\} = \sup\{\phi_{(Q,T)}\}\end{aligned}$$

If these two limits are different, then n_0 is such that for all $n \geq n_0$ we have

$$\begin{aligned}&\sum_{C\in\hat{C}(R_P)} \{Pr(C) \mid \#trans(C) < n\} \neq \\ &\qquad \neq \lim_{n\to\infty} \sum_{C\in\hat{C}(R_Q)} \{Pr(C) \mid \#trans(C) < n\}\end{aligned}$$

Let us consider the finite test T' resulting from T by unwinding n_0 times each occurrence of recursion in T, and then replacing any occurrence of recursion by $\mathbf{0}$. We see that T' can execute the same sequences of transitions C such that $\#trans(C) \leq n_0$ as T; the only difference is that in the sequences corresponding to T there can be additional unlabelled transitions, but they will have probability 1. Thus, $P \parallel T'$ (respectively $Q \parallel T'$) can execute the same computations as $P \parallel T$ (respectively $Q \parallel T$), and it therefore follows, by the previous inequality, that $\sup\{\phi_{(P,T')}\} \neq \sup\{\phi_{(P,T)}\}$, which implies $P \not\equiv_{fT} Q$ □

Proof of Theorem 4. This follows directly from Theorem 3 and Proposition 6.

Proof of Theorem 5 (Sketch). We will prove only some of the axioms and rules. Since very similar ideas can be applied to prove the soundness of the remaining axioms and rules, these proof are omitted. We consider three processes P, Q and R such that:

$$\mathcal{PA}(P/s) = \{(p_1, \mathcal{A}_1), (p_2, \mathcal{A}_2), \ldots, (p_l, \mathcal{A}_l)\}$$

$$\mathcal{PA}(Q/s) = \{(q_1, \mathcal{B}_1), (q_2, \mathcal{B}_2), \ldots, (q_m, \mathcal{B}_m)\}$$

$$\mathcal{PA}(R/s) = \{(r_1, \mathcal{C}_1), (r_2, \mathcal{C}_2), \ldots, (r_n, \mathcal{C}_n)\}$$

- **Axiom P2**, $(P \otimes_p Q) \otimes_q R = P \otimes_{p\cdot q} (Q \otimes_{\frac{(1-p)\cdot q}{1-p\cdot q}} R)$

  $$\begin{aligned}&\mathcal{PA}(((P \otimes_p Q) \otimes_q R)/s) = \\ &\quad = \{(p\cdot q\cdot p_1, \mathcal{A}_1), \ldots, (p\cdot q\cdot p_l, \mathcal{A}_l), ((1-p)\cdot q\cdot q_1, \mathcal{B}_1), \ldots, \\ &\quad ((1-p)\cdot q\cdot q_m, \mathcal{B}_m), ((1-q)\cdot r_1, \mathcal{C}_1), \ldots, ((1-q)\cdot r_n, \mathcal{B}_n)\}\end{aligned}$$

  $$\begin{aligned}&\mathcal{PA}(((P \otimes_{p\cdot q} (Q \otimes_{\frac{(1-p)\cdot q}{1-p\cdot q}} R))/s) = \\ &\quad = \{(p\cdot q\cdot p_1, \mathcal{A}_1), \ldots, (p\cdot q\cdot p_l, \mathcal{A}_l), \ldots, ((1-p\cdot q)\frac{(1-p)\cdot q}{1-p\cdot q}\cdot q_m, \mathcal{B}_m), \\ &\quad \ldots, ((1-p\cdot q)\cdot(1-\frac{(1-p)\cdot q}{1-p\cdot q})\cdot r_n, \mathcal{C}_n)\} \\ &\quad = \{(p\cdot q\cdot p_1, \mathcal{A}_1), \ldots, (p\cdot q\cdot p_l, \mathcal{A}_l), ((1-p)\cdot q\cdot q_1, \mathcal{B}_1), \ldots, \\ &\quad ((1-p)\cdot q\cdot q_m, \mathcal{B}_m), ((1-q)\cdot r_1, \mathcal{C}_1), \ldots, ((1-q)\cdot r_n, \mathcal{B}_n)\}\end{aligned}$$

- **Axiom PI1** $P \oplus (Q \otimes_p R) = (P \oplus Q) \otimes_p (P \oplus R)$

  $$\begin{aligned}&\mathcal{PA}((P \oplus (Q \otimes_p R))/s) = \\ &\quad = \{(p_1, \mathcal{A}_1), \ldots, (p_l, \mathcal{A}_l)\} \oplus \{(p\cdot q_1, \mathcal{B}_1), \ldots, ((1-p)\cdot r_n, \mathcal{C}_n)\} \\ &\quad = \{(p_1\cdot p\cdot q_1, \mathcal{A}_1\, c_\Omega\, \mathcal{B}_1)), \ldots, (p_1\cdot p\cdot q_m, \mathcal{A}_1\, c_\Omega\, \mathcal{B}_m)), \\ &\quad (p_1\cdot(1-p)\cdot r_1, \mathcal{A}_1\, c_\Omega\, \mathcal{C}_1), \ldots, (p_1\cdot(1-p)\cdot r_n, \mathcal{A}_1\, c_\Omega\, \mathcal{C}_n)\}\end{aligned}$$

  $$\begin{aligned}&\mathcal{PA}(((P \oplus Q) \otimes_p (P \oplus R))/s) = \\ &\quad = \{(p_1\cdot q_1, \mathcal{A}_1\, c_\Omega\, \mathcal{B}_1), \ldots, (p_l\cdot q_m, \mathcal{A}_l\, c_\Omega\, \mathcal{B}_m)\} \otimes_p \\ &\quad \otimes_p \{(p_1\cdot r_1, \mathcal{A}_1\, c_\Omega\, \mathcal{C}_1), \ldots, (p_l\cdot r_n, \mathcal{A}_l\, c_\Omega\, \mathcal{C}_n)\} \\ &\quad = \{(p\cdot p_1\cdot q_1, \mathcal{A}_1\, c_\Omega\, \mathcal{B}_1), \ldots, (p\cdot p_1\cdot q_m, \mathcal{A}_1\, c_\Omega\, \mathcal{B}_m), \\ &\quad ((1-p)\cdot p_1\cdot r_1, \mathcal{A}_1\, c_\Omega\, \mathcal{C}_1), \ldots, ((1-p)\cdot p_1\cdot r_n, \mathcal{A}_1\, c_\Omega\, \mathcal{C}_n)\}\end{aligned}$$

- **Rule R9.** The semantic process $\sqcup(\Omega \otimes_{\alpha(n)} P)$ is the limit of the chain $\{Q^i\}_{i\in I}$, where $Q^i = \Omega \otimes_{\alpha(i)} P$.

  By using definition 18 (limit of a chain) we can deduce that

  $$\pi(\sqcup Q^n, \Omega) = \lim_{n\to\infty} \alpha(n) = 0$$

  and then, $\sqcup Q^n = \Omega \otimes_0 P = P$.

$\square$

Proof of Theorem 6 (Sketch). We proceed by induction on the depth of the term. The base case is immediate (**0** and Ω are already in normal form). For the induction step we shall consider the remaining operators. We will focus on internal and external choices.

- $P = Q \oplus R$. Consider:

$$Q \equiv \bigotimes_{i=1}^{n} [q_i] \bigoplus_{A \in \mathcal{A}_i} \sum_{a \in A} a.n(\mathcal{A}_i, a)$$

$$R \equiv \bigotimes_{j=1}^{m} [r_i] \bigoplus_{B \in \mathcal{B}_j} \sum_{b \in B} b.m(\mathcal{B}_j, b)$$

Then,

$$P \equiv \left(\bigotimes_{i=1}^{n} [q_i] \bigoplus_{A \in \mathcal{A}_i} \sum_{a \in A} a.n(\mathcal{A}_i, a) \right) \oplus \left(\bigotimes_{j=1}^{m} [r_i] \bigoplus_{B \in \mathcal{B}_j} \sum_{b \in B} b.m(\mathcal{B}_j, b) \right)$$

Applying repeatedly axioms PI1, P2 and P4 we obtain:

$$P \equiv \bigotimes_{i=1,j=1}^{i=n,j=m} [q_i \cdot r_j] \Big(\bigoplus_{A \in \mathcal{A}_i} \sum_{a \in A} a.n(\mathcal{A}_i, a) \;\oplus\; \bigoplus_{B \in \mathcal{B}_j} \sum_{b \in B} b.m(\mathcal{B}_j, b) \Big)$$

Finally we have to apply (repeatedly) the derived equations Der1↓, Der2↓, Der3↓ over the non probabilistic part to obtain a normal form.

- $P = Q + R$. We have:

$$P \equiv \bigotimes_{i=1}^{n} [q_i] \bigoplus_{A \in \mathcal{A}_i} \sum_{a \in A} a.n(\mathcal{A}_i a) \;+\; \bigotimes_{j=1}^{m} [r_i] \bigoplus_{B \in \mathcal{B}_j} \sum_{b \in B} b.m(\mathcal{B}_j b)$$

Applying repeatedly axioms PE1, P2 and P4 we obtain:

$$\bigotimes_{i=1,j=1}^{i=n,j=m} [q_i \cdot r_j] \Big(\bigoplus_{A \in \mathcal{A}_i} \sum_{a \in A} a.n(\mathcal{A}_i, a) \;+\; \bigoplus_{B \in \mathcal{B}_j} \sum_{b \in B} b.m(\mathcal{B}_j, b) \Big)$$

Applying, over the non probabilistic part, the axioms IE3↓, IE1 and the derived equation Der3↓ we obtain:

$$\bigotimes_{i=1,j=1}^{i=n,j=m} [q_i \cdot r_j] \bigoplus_{C \in \mathcal{C}_{ij}} \sum_{c \in C} c.q(\mathcal{C}_{ij}, c)$$

where $\mathcal{C}_{ij} = \mathcal{A}_i \; u \; \mathcal{B}_j$ and

$$q(\mathcal{C}_{ij}, c) =$$
$$\begin{cases} n(\mathcal{A}_i, a) + m(\mathcal{B}_j, b) & \text{if } c \in A(\mathcal{A}_i) \cap A(\mathcal{B}_j) \\ n(\mathcal{A}_i, a) & \text{if } c \in A(\mathcal{A}_i),\, c \notin A(\mathcal{B}_j) \\ m(\mathcal{B}_j, b) & \text{if } c \notin A(\mathcal{A}_i),\, c \in A(\mathcal{B}_j) \end{cases}$$

has a normal form by the induction hypothesis.

□

Proof of Lemma 3. We can distinguish several cases depending on whether P and Q are finite or infinite.

Case 1: P and Q are finite.

If one of them is $\mathbf{0}$ or Ω, then both are $\mathbf{0}$ or Ω (respectively) and the implications holds. If $P = \Omega$ the result is straightforward by applying rule R6.

If P and Q are different from $\mathbf{0}$ and Ω, since they are in normal form, we have

$$P = s \bigotimes_i [p_i] \bigoplus_{A \in \mathcal{A}_i} \sum_{a \in A} a.P(\mathcal{A}_i, a)$$

$$Q = \bigotimes_j [q_j] \bigoplus_{B \in \mathcal{B}_j} \sum_{b \in B} b.Q(\mathcal{B}_j, b)$$

Applying repeatedly axiom P3 over process P we obtain

$$P = \bigotimes_{i,j} [p_{ij}] \bigoplus_{A \in \mathcal{A}_{ij}} \sum_{a \in A} a.n(\mathcal{A}_{ij}, a)$$

where $\sum_j p_{ij} = p_i$, $\sum_i p_{ij} = q_j$, $\mathcal{A}_{ij} = \mathcal{A}_i$ and $\mathcal{A}_{ij} \in pC(\mathcal{B}_j, [\![Q]\!], [\![P]\!])$ for every i and j.

The p_{ij} may be obtained just resolving the linear system

$$\begin{cases} q_j = \sum_i k_{ij} \cdot p_i & \\ \sum_j k_{ij} = 1 & \\ k_{ij} = 0 & \text{if } \mathcal{A}_i \notin pC(\mathcal{B}_j, [\![Q]\!], [\![P]\!]) \end{cases}$$

and calculating $p_{ij} = k_{ij} \cdot p_i$.

This linear system has, at least, one solution. To prove this we need only to view it as a problem of network flux distribution, which, in turn, is a generalization of the transport problem. We may take $(p_i, \mathcal{A}_i)$ as the producers (sources) of flux, and $(q_i, \mathcal{B}_i)$ as the consumers. This problem always has a solution in balanced networks. In our case, this condition holds because $\sum_i p_i = \sum_j q_j = 1$.

Now, applying axiom PI3, we can replace acceptance sets $\mathcal{A}_{ij}$ by $\mathcal{B}_j$ in order to obtain a more deterministic process

$$P \leq \bigotimes_{i,j} [p_{ij}] \bigoplus_{A \in \mathcal{B}_j} \sum_{a \in A} a.P(\mathcal{A}_{ij}, a)$$

We now apply axiom P3 in order to gather probabilistic branches

$$P \leq \bigotimes_j [q_j] \bigoplus_{A \in \mathcal{B}_j} \sum_{a \in A} a.P(\mathcal{A}_{ij}, a)$$

where $q_j = \sum_i p_{ij}$.

Since $[\![P]\!]_{\mathrm{fPAT}} \leq [\![Q]\!]_{\mathrm{fPAT}}$, according to definition 16 it holds that $\forall a \in A(\mathcal{B}_j)$ and $\forall \mathcal{A}_{ij} \in pC(\mathcal{B}_j, [\![m]\!], [\![n]\!])$

$$[\![P/\langle \mathcal{A}_{ij}, a\rangle]\!]_{\mathrm{PAT}} \leq [\![Q/\langle \mathcal{B}_j, a\rangle]\!]_{\mathrm{PAT}}$$

Applying induction we have

$$P(\mathcal{A}_{ij}, a) \leq Q(\mathcal{B}_j, a)$$

and therefore

$$P \leq \bigotimes_j [q_j] \bigoplus_{A \in \mathcal{B}_j} \sum_{a \in A} a.Q(\mathcal{B}_j, a) = Q$$

Case 2: P infinite and Q finite

Finite approximations of P verify that

$$[\![P^0]\!] \leq [\![P^1]\!] \leq \cdots [\![P^n]\!] \leq [\![P]\!] \leq [\![Q]\!]$$

Since every P^n is finite we have $P^n \leq_{PS} Q$, and applying rule R8 we obtain $P \leq Q$.

Case 3: P finite and Q infinite

A possible way of proving the desired result would be to show the existence of one n such that $[\![P]\!] \leq [\![Q^n]\!]$ as occurs in ω–algebraic domains. But this is not true in our model (the same occurs in [6]). Therefore, we need a different approach; in particular we require rule R9.

We know that $[\![P]\!] \leq [\![Q]\!] \equiv \sqcup [\![Q^n]\!]$, and we will use induction on the number of recursions. The base case is a process of the form $Q = recX.(X \otimes_q R)$, where R is a finite process.

Finite approximations of Q are

$$\begin{array}{rcl} Q^0 & = & \Omega \\ Q^1 & = & \Omega \otimes_q R \\ Q^2 & = & (\Omega \otimes_q R) \otimes_q R = \Omega \otimes_{q^2} R \\ & \vdots & \\ Q^n & = & \Omega \otimes_{q^n} R \end{array}$$

where $q, q^2, \ldots, q^n$ are a geometric sequence, and $\lim_{n \to \infty} q^n = 0$, because $0 < q < 1$.

Applying rule R9 we have

$$R = \sqcup(\Omega \otimes_{q^n} R) = \sqcup Q^n = Q$$

and the problem is reduced to case 1.

In the most general case, R is infinite of the form $R = recX.(X \otimes_q R')$, where R' es infinite of the same form as R. In this case it is enough to use induction on the nested recursions.

Case 4: P and Q are infinite.

Using the previous case, we can deduce that for every finite approximation P^n of P, it holds that $P^n \leq Q$, and therefore applying rule R8 we have $P \leq Q$.
□

Proof of Theorem 7. This follows directly from Theorem 6 and Lemma 3.
□

References

[1] S. Andova. Process algebra with probabilistic choice. In *Proceedings of ARTS'99, LNCS 1601*, 1999.

[2] B. Bloom and A.R. Meyer. A remark on bisimulation between probabilistic processes. In Meyers and Tsailin, editors, *Logik at Botik*. Springer-Verlag, 1989.

[3] D. Cazorla, F. Cuartero, V. Valero, and F.L. Pelayo. A Process Algebra for Probabilistic and Nondeterministic Processes. *Information Processing Letters*, 80(1):15–23, October 2001.

[4] I. Christoff. *Testing Equivalence for Probabilistic Processes*. PhD thesis, Uppsala University, Sweden, 1990.

[5] F. Cuartero, D. de Frutos, and V. Valero. PCSP: A denotational model for probabilistic processes. In *Proceedings of Third AMAST Workshop on Real-Time Systems*, 1996.

[6] F. Cuartero, D. de Frutos, and V. Valero. A sound and complete proof system for probabilistic processes. In *ARTS'96, LNCS 1231*, 1997.

[7] P.R. D'Argenio, H. Hermanns, and J.-P. Katoen. On generative parallel composition. *Electronic Notes in Theoretical Computer Science*, 22:25, 1999.

[8] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1998.

[9] J.I den Hartog and E.P. de Vink. Mixing up nondeterminism and probability: a preliminary report. In *Proc. PROBMIV'98, Electronic Notes in TCS*, 1998.

[10] 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*, Sea of Galilee, Israel, 1990.

[11] C. Gregorio and M. Núñez. Denotational semantics for probabilistic refusal testing. In *Proc. PROBMIV'98, Electronic Notes in TCS vol.22*, 1998.

[12] J.F. Groote. Transition system specifications with negative premises. *Theoretical Computer Science*, 118:263–299, 1993.

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

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

[15] C. Jones. *Probabilistic Non–determinism.* PhD thesis, University of Edinburgh, 1990.

[16] C. Jones and G.D. Plotkin. A probabilistic powerdomain of evaluations. In *Proceedings of 4th Annual Symposium on Logic in Computer Science*, 1989.

[17] B. Jonsson and W. Yi. Compositional testing preorders for probabilistic processes. In *10th IEEE Symposium on Logic in Computer Science*, pages 431–443, San Diego, Ca., 1995.

[18] B. Jonsson and W. Yi. Fully abstract characterization of probabilistic may testing. In *5th Int. AMAST Workshop on Real-Time and Probabilistic Systems, ARTS'99*, 1999.

[19] C. Jou and S.A. Smolka. Equivalences, congruences and complete axiomatizations for probabilistic processes. In *Proceedings of CONCUR'90, LNCS 458*, 1990.

[20] M.Z. Kwiatkowska and G. Norman. A fully abstract metric-space denotational semantics for reactive probabilistic processes. In *Proc. COMPROX'98, Electronic Notes in TCS vol.13*, 1998.

[21] K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

[22] G. Lowe. Representing nondeterministic and probabilistic behaviour in reactive processes. Technical Report PRG-TR-11-93, Oxford university computing laboratory, 1993.

[23] C. Morgan, A. McIver, K. Seidel, and J.W. Sanders. Refinement-oriented probability for CSP. *Formal Aspects of Computing*, 8(6):617–647, 1996.

[24] M. Núñez, D. de Frutos, and L. Llana. Acceptance trees for probabilistic processes. In *CONCUR'95, LNCS 962*, pages 249–263, 1995.

[25] R. Segala. A compositional trace-based semantics for probabilistic automata. In *Proceedings of CONCUR'95, LNCS 962*, 1995.

[26] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In *Proceedings of CONCUR'94, LNCS 836*, 1994.

[27] R. van Glabbeek, S.A. Smolka, and B.U. Steffen. Reactive, generative ans stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, 1995.

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

[29] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of 26th FOCS*, pages 327–338, 1985.

[30] W. Yi. Algebraic reasoning for real-time probabilistic processes with uncertainty information. In *Formal Techniques in Real-Time and Fault-Tolerant Systems, LNCS 863*, pages 680–693, Kiel, Germany, 1994.

[31] W. Yi and K.G. Larsen. Testing probabilistic and nondeterministic processes. In *Protocol Specification, Testing and Verification XII*, pages 47–61, 1992.