University of Castilla-La Mancha

A publication of the

Department of Computer Science

A Process Algebra for Probabilistic and Nondeterministic Processes.

by

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

Technical Report **#DIAB-00-01-11** November 2000

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

A Process Algebra for 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 [6] that maintains nondeterminism. Thus, we have both kinds of choices, and we have decided that probabilistic ones are first resolved, and after that nondeterministic choices are then considered.

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 finite EPL (EPL$_f$) is preserved, and we also prove that the testing semantics, the denotational semantics and the proof system are fully abstract.

1. Introduction

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.

In [4], the classical nondeterministic choice is replaced by a probabilistic choice: $\sum [p_i]E_i$, where p_i is the probability for the process to behave like E_i. In [14], this model is extended and three interpretations for concurrent probabilistic processes are presented: reactive, generative and stratified models.

In [5] an extension of CCS with probabilities and time is presented, based on the asynchronous version of CCS, and a generative probabilistic choice operator is introduced together with the classical deterministic one. A reactive operator is used in [3] in a similar way, while a probabilistic testing using a generative interpretation is introduced in [13]. Besides the models already mentioned, several languages, semantic aspects, notions on probabilistic bisimulation and other related topics are presented in [8, 10, 2].

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

Thus, it is also necessary to provide operators that allow us to deal with nondeterminism, but at the same time, we should not forget the probabilistic aspects. In the process of modelling a concurrent system we sometimes find nondeterministic behaviours that can be quantified (for instance, the probability for a faulty communication channel to fail), whereas on other occasions they cannot (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.

Following these ideas, in [15] a probabilistic extension of CCS is presented. A probabilistic choice is added, whereas the original CCS choice is maintained for representing nondeterminism. 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 generalizing, for probabilistic and nondeterministic processes, may, $must$ and $test$ preorders. This work is extended in [7], where a denotational characterization for may–testing and must–testing is presented.

Another work combining probabilities and nondeterminism is [11]. 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.

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 finite 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 give our conclusions and some outlines for future work.

2. The language PNAL

The syntax of the language we present, PNAL, is based on EPL [6] and provides three kinds of choice: classical internal choice, denoted by $\oplus$, classical external choice, denoted by $+$, and a probabilistic choice labelled with 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:

$$\begin{aligned} P ::= \mathbf{0} \mid X \mid a.P \mid P \oplus P \mid P + P \mid \\ P \otimes_p P \mid P|P \mid P\backslash A \end{aligned}$$

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

The classical operators are interpreted as usual 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 $1 - p$, and this decision is 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 we have, basically, two alternatives: in [1, 12, 15] probabilistic choices are solved firstly, whereas in [11] the opposite approach is considered. Also, a different point of view is considered in [9], where an action–guarded reactive probabilistic choice is introduced and the point at which probabilistic choice occurs has no significance.

Let us observe that this decision is not meaningless at all, because depending on the adopted approach, 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 have that $P \,\Box\, P$ increments the probability of a, for any model we consider.

We have chosen the first approach (probabilistic choices are resolved before internal ones) for several reasons. We have observed that, when combining probabilistic and internal choices, it is not possible to define a framework in which the following two equations are simultaneously satisfied:

$$\begin{aligned} a \otimes_p (b \oplus c) = (a \otimes_p b) \oplus (a \otimes_p c) \\ a \oplus (b \otimes_p c) = (a \oplus b) \otimes_p (a \oplus c) \end{aligned}$$

Thus, it has been necessary to take a decision about which of these two properties holds. Taking into account that one of our objectives is to extend the theory of nondeterministic processes ([6]), we have found that this is only possible when the second equation is true. Thus, we have decided that, in our model, the second property is maintained. This implies that probabilistic choice must be resolved at the first place.

Furthermore, we also obtain that normal forms are defined in a very natural way. The simplicity of these normal forms is one of the main advantages of our model, in contrast to other proposals [11], where it is really difficult to capture the aspect of the normal forms there presented. The price that one has to pay to get these simple normal forms is that idempotency is not preserved by the internal choice, but this usually occurs when dealing with probabilistic processes. But, let us observe that this property will hold for non–probabilistic processes (i.e., processes without probabilistic choices), so in this case we have that our model extends EPL.

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) $\mathbf{0}$ *and* $a.P$ *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, like in [1, 5, 15].

The behaviour of the remaining operators is presented in Tables 2 and 3, where we assume that processes are now probabilistically stable. 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

(P1a) $\frac{}{P \otimes_p Q \longrightarrow_p P}$	**(P1b)** $\frac{}{P \otimes_p Q \longrightarrow_{1-p} Q}$
(P2a) $\frac{P \longrightarrow_p P',\, Q\downarrow}{P \oplus Q \longrightarrow_p P' \oplus Q}$	**(P2b)** $\frac{P\downarrow,\, Q \longrightarrow_q Q'}{P \oplus Q \longrightarrow_q P \oplus Q'}$
(P2c) $\frac{P \longrightarrow_p P',\, Q \longrightarrow_q Q'}{P \oplus Q \longrightarrow_{p\cdot q} P' \oplus Q'}$	
(P3a) $\frac{P \longrightarrow_p P',\, Q\downarrow}{P + Q \longrightarrow_p P' + Q}$	**(P3b)** $\frac{P\downarrow,\, Q \longrightarrow_q Q'}{P + Q \longrightarrow_q P + Q'}$
(P3c) $\frac{P \longrightarrow_p P',\, Q \longrightarrow_q Q'}{P + Q \longrightarrow_{p\cdot q} P' + Q'}$	
(P4a) $\frac{P \longrightarrow_p P',\, Q\downarrow}{P\vert Q \longrightarrow_p P'\vert Q}$	**(P4b)** $\frac{P\downarrow,\, Q \longrightarrow_q Q'}{P\vert Q \longrightarrow_q P\vert Q'}$
(P4c) $\frac{P \longrightarrow_p P',\, Q \longrightarrow_q Q'}{P\vert Q \longrightarrow_{p\cdot q} P'\vert Q'}$	**(P5)** $\frac{P \longrightarrow_p P'}{P\backslash A \longrightarrow_p P'\backslash A}$

Table 1. Inference rules for transitions labelled with probabilities

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. Finally, rule I4 shows the operational behaviour for the restriction operator.

(I1a) $\frac{}{P \oplus Q \longrightarrow P}$	**(I1b)** $\frac{}{P \oplus Q \longrightarrow Q}$
(I2a) $\frac{P \longrightarrow P'}{P + Q \longrightarrow P' + Q}$	**(I2b)** $\frac{Q \longrightarrow Q'}{P + Q \longrightarrow P + Q'}$
(I3a) $\frac{P \longrightarrow P'}{P\vert Q \longrightarrow P'\vert Q}$	**(I3b)** $\frac{Q \longrightarrow Q'}{P\vert Q \longrightarrow P\vert Q'}$
(I3c) $\frac{P \xrightarrow{a} P',\, Q \xrightarrow{\overline{a}} Q'}{P\vert Q \longrightarrow P'\vert Q'}$	**(I4)** $\frac{P \longrightarrow P'}{P\backslash A \longrightarrow P'\backslash A}$

Table 2. Inference rules for unlabelled transitions ($P\downarrow$ and $Q\downarrow$)

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

(A1) $\frac{}{a.P \xrightarrow{a} P}$	**(A2a)** $\frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'}$
(A2b) $\frac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'}$	**(A3a)** $\frac{P \xrightarrow{a} P'}{P\vert Q \xrightarrow{a} P'\vert Q}$
(A3b) $\frac{Q \xrightarrow{a} Q'}{P\vert Q \xrightarrow{a} P\vert Q'}$	**(A4)** $\frac{P \xrightarrow{a} P'}{P\backslash A \xrightarrow{a} P'\backslash A}\ (a, \overline{a} \notin A)$

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

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. Every transition is labelled with the name of the applied rule.

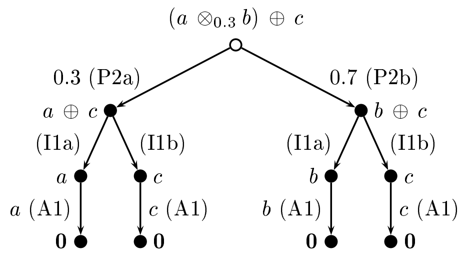

Figure 1. Transition system of the process $(\mathbf{a.0} \otimes_{\mathbf{0.3}} \mathbf{b.0}) \oplus \mathbf{c.0}$

4. Testing semantics

A test is just a process which additionally may 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. But, 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 to do that.

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 each possible evolution of $P\|T$. In order to define $\phi_{(P,T)}$ we need some preliminary definitions, following similar ideas to those presented in [15].

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 $\xrightarrow{w}$.*

Therefore, every nondeterministic node in a resolution has 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).

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

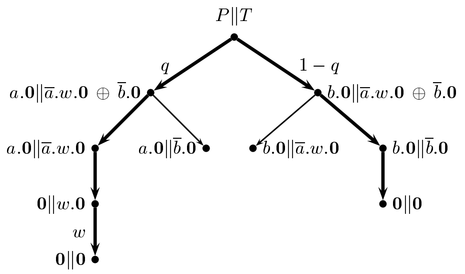

Figure 2. Transition system of a process and one of its resolutions

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 to reach a successful state along the computation C, and $Pr(R)$ denotes the probability to reach a successful state in the resolution R.

Definition 5 *A computation $C = S \rightsquigarrow S_1 \cdots S_{i-1} \rightsquigarrow S_i \ldots \rightsquigarrow S_n$ 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:*

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

And the probability to get a success in a resolution:

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

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

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

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

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

This preorder impose that whenever Q cannot accept a test T, P cannot accept it either. This has been introduce because, according to the definition of test preorder given in [7], the processes $P = a \oplus c$ and $Q = a$ are related under the test preorder, i.e., $P \sqsubseteq_{test} Q$. On the contrary in Hennessy's testing $P\ may\ c.w$ but $Q\ m\!\!\!/ay\ c.w$ 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 be $P = a \otimes_{0.5} \boldsymbol{0}$ and $T = \overline{a}.w$. We have that $\phi_{(P,T)} = \{0.5\}$, $\phi_{(P \oplus P,T)} = \{0.25, 0.50, 0.75\}$ and $\phi_{(P + P,T)} = \{0.75\}$. Therefore, $P \neq P \oplus P$ and $P \neq P + P$*

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

- We will consider trees with two kinds of nodes: probabilistic ($\otimes$), and nondeterministic ($\bullet$).
- The root node is a probabilistic one and probabilistic and nondeterministic nodes appear in an alternative way.
- Nondeterministic nodes are labelled with an acceptance set (as defined in [6]), while probabilistic nodes are unlabelled.
- Branches leaving a probabilistic node are labelled with probabilities p_i, with $\sum_i p_i = 1$. Every branch leaving a nondeterministic node is labelled with one action in the acceptance set, but there are not two branches labelled with the same action.
- Terminal nodes are always nondeterministic, and are labelled with the acceptance set $\{\emptyset\}$.

We can see that these trees are quite similar to those used in [6], but now we have to extend these trees in order to include the probabilistic information associated with every reachable state.

Let us introduce now acceptance sets and probabilistic acceptance sets.

Definition 9 *An acceptance set $\mathcal{A}$ 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 $\mathcal{A}ct$ .We say that $(p, \mathcal{A})$ is a "probability–acceptance set pair" if $(p, \mathcal{A}) \in (0, 1] \times \boldsymbol{D}(\mathcal{A})$.*

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

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

Definition 10 *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 11 *We define the probability distribution, induced by a probabilistic acceptance set, as the function* $Prob : \boldsymbol{D}(\mathcal{A}) \times \boldsymbol{D}(\mathcal{PA}) \longrightarrow [0,1]$ *given by*

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

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

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

We will use the following notation: $\overline{C} = \{\overline{a} \mid a \in C\}$ and $\mathcal{A} \backslash C = \{B - (C \cup \overline{C}) \mid B \in \mathcal{A}\}$.

Besides the definitions above, we will need two additional operators borrowed from [6]: operators c and u. Due to the lack of space, we will not give here their formal definitions. Instead, we will remember their informal meaning: c is the convex closure of a set, and u is the pointwise union of sets of sets.

We may now define the probabilistic acceptance set of each process at the top level, $\mathcal{PA}(P)$. Parallel composition will be studied separately because extra explanation is needed.

$$\mathcal{PA}(\mathbf{0}) = \{(1, \{\emptyset\})\}, \qquad \mathcal{PA}(a.t) = \{(1, \{\{a\}\})\}.$$

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

$$\begin{aligned} \mathcal{PA}(t_1 \oplus t_2) = \{(p, \mathcal{A}) \mid\ & \exists \mathcal{B} \in Accept(\mathcal{PA}(t_1)), \\ & \exists \mathcal{C} \in Accept(\mathcal{PA}(t_2)) \text{ s.t. } \mathcal{A} = c(\mathcal{B} \cup \mathcal{C}), \\ & p = \sum_{\mathcal{A} = c(\mathcal{B} \cup \mathcal{C})} Prob(\mathcal{B}, \mathcal{PA}(t_1)) \cdot Prob(\mathcal{C}, \mathcal{PA}(t_2))\} \end{aligned}$$

$$\begin{aligned} \mathcal{PA}(t_1 + t_2) = \{(p, \mathcal{A}) \mid\ & \exists \mathcal{B} \in Accept(\mathcal{PA}(t_1)), \\ & \exists \mathcal{C} \in Accept(\mathcal{PA}(t_2)) \text{ s.t. } \mathcal{A} = \mathcal{B}\ u\ \mathcal{C}, \\ & p = \sum_{\mathcal{A} = \mathcal{B}\ u\ \mathcal{C}} Prob(\mathcal{B}, \mathcal{PA}(t_1)) \cdot Prob(\mathcal{C}, \mathcal{PA}(t_2))\} \end{aligned}$$

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

The probabilistic acceptance set defined above allows us to distinguish between the different reachable states, and also provides us with the probability associated with every state.

Let us now introduce the semantic process $t/(\mathcal{A}, a)$, which models the behaviour of t, once it has executed the action a at the state given by $\mathcal{A}$. We define $t/(\mathcal{A}, a)$ as follows:

- If t is a process such that either $a \notin A(\mathcal{A})$ or $\mathcal{A} \notin Accept(\mathcal{PA}(t))$ then $t/(\mathcal{A}, a)$ is not defined.
- If $t = a.t_1$, the only reachable state is $\{\{a\}\}$ with probability 1. Then $t/(\{\{a\}\}, a) = t_1$.
- $t = t_1 \otimes_q t_2$. We must distinguish several subcases:
  – $\mathcal{A} \in Accept(\mathcal{PA}(t_1))$ and $\mathcal{A} \notin Accept(\mathcal{PA}(t_2))$. In this case only t_1 can execute the action a, so $t/(\mathcal{A}, a) = t_1/(\mathcal{A}, a)$.
  – $\mathcal{A} \notin Accept(\mathcal{PA}(t_1))$ and $\mathcal{A} \in Accept(\mathcal{PA}(t_2))$. It is the symmetric case: $t/(\mathcal{A}, a) = t_2/(\mathcal{A}, a)$.
  – $\mathcal{A} \in Accept(\mathcal{PA}(t_1))$ and $\mathcal{A} \in Accept(\mathcal{PA}(t_2))$. Now a can be made either by t_1 or t_2, then:
  $$t/(\mathcal{A}, a) = t_1/(\mathcal{A}, a) \otimes_q t_2/(\mathcal{A}, a)$$
- $t = t_1 \oplus t_2$. We have that, according to the definition of $\mathcal{PA}$ for the internal choice operator, each one of its states, $\mathcal{A}$, is obtained as the convex closure of a union of states, $\mathcal{A} = c(\mathcal{B} \cup \mathcal{C})$, where $\mathcal{B} \in Accept(\mathcal{PA}(t_1))$ and $\mathcal{C} \in Accept(\mathcal{PA}(t_2))$. Then, we can distinguish the following cases:
  - $a \in A(\mathcal{B})$ and $a \notin A(\mathcal{C})$. Then, it is t_1 which executes a, and thus: $t/(\mathcal{A}, a) = t_1/(\mathcal{B}, a)$.
  - $a \notin A(\mathcal{B})$ 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, then:
    $t/(\mathcal{A}, a) = t_1/(\mathcal{B}, a) \oplus t_2/(\mathcal{C}, a)$

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

$$t/(\mathcal{A}, a) = \begin{array}{ll} \displaystyle\bigoplus_{\substack{\mathcal{B},\mathcal{C} \\ \mathcal{A}=c(\mathcal{B}\cup\mathcal{C})}} t_1/(\mathcal{B}, a) \oplus & \text{if } a \in A(\mathcal{B}) \wedge a \notin A(\mathcal{C}) \\ \displaystyle\bigoplus_{\substack{\mathcal{B},\mathcal{C} \\ \mathcal{A}=c(\mathcal{B}\cup\mathcal{C})}} t_2/(\mathcal{C}, a) \oplus & \text{if } a \notin A(\mathcal{B}) \wedge a \in A(\mathcal{C}) \\ \displaystyle\bigoplus_{\substack{\mathcal{B},\mathcal{C} \\ \mathcal{A}=c(\mathcal{B}\cup\mathcal{C})}} t_1/(\mathcal{B}, a) \oplus t_2/(\mathcal{C}, a) & \text{if } a \in A(\mathcal{B}) \wedge a \in A(\mathcal{C}) \end{array}$$

- $t = t_1 + t_2$. We have that every state $\mathcal{A}$ is built as $\mathcal{A} = \mathcal{B}\ u\ \mathcal{C}$, where $\mathcal{B} \in Accept(\mathcal{PA}(t_1))$ and $\mathcal{C} \in Accept(\mathcal{PA}(t_2))$, and we can repeat the same case distinction as in the internal choice, thus obtaining exactly the same expression for $t/(\mathcal{A}, a)$.
- $t \backslash C$. If $a \in C$ then $(t \backslash C)/(\mathcal{A}, a)$ is not defined. If $a \notin C$ then:
  $$(t \backslash C)/\langle \mathcal{A} a \rangle = \bigotimes_{\mathcal{A} = \mathcal{B} \backslash C} \left[\frac{Prob(t, \mathcal{B})}{Prob(t \backslash C, \mathcal{A})} \right] (t/\langle \mathcal{B} a \rangle) \backslash C$$

Once we have defined the probabilistic acceptance sets of a process, not only for its first action, but all over its possible executions, we are interested in all the possible sequences of actions it can perform. Let us now introduce some notation for that. As each probabilistic node in the tree representing t can be reached by following a sequence of pairs (acceptance set, action), due to the alternating nature of these trees, we may identify these nodes by $t(s)$, where $s = \langle \mathcal{A}_1, a_1, \ldots, \mathcal{A}_k, a_k \rangle$ is the sequence leading us to the node. We can use a similar notation for syntactical terms. Then, $(P/(\mathcal{A}, a))\,/(\mathcal{B}, b)$ will be represented shortly by $P/(\langle \mathcal{A}, a, \mathcal{B}, b \rangle)$

Parallel composition. The definition of the function over fPAT 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 later 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 3 we can see the fPAT of P_1, P_2 and the first level of $P_1 | P_2$.

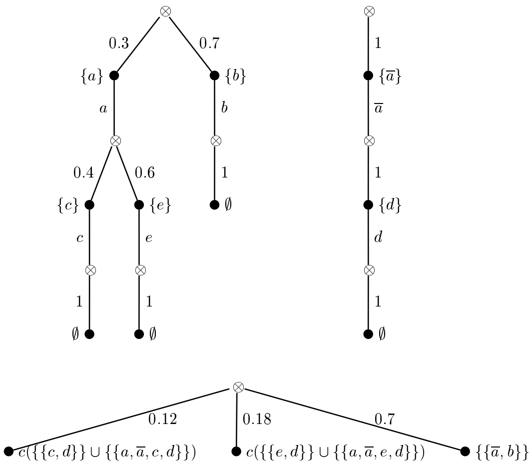

Figure 3. fPAT of P_1, P_2 and first level of $P_1 | P_2$

In the first level, P may perform the actions that P_1 or P_2 may perform at their first level. Besides, in case 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 one 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 introduced 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 consider that a and $\overline{a}$ communicate, and the probabilistic choice after action a in P_1 is resolved following the branch labelled with 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 $\mathcal{A}$ considering the convex closure of the union of sets $\{c, d\}$ and $\{a, \overline{a}, c, d\}$.

We need to introduce some preliminary definitions.

Definition 13 *Let $t \in$ fPAT. We define the sequences of t, denoted by $Seq(t)$, as the set of all the sequences of pairs (acceptance set,action) that we may obtain from t.*

The set of maximal sequences of t is defined as follows: $Seq_{Max}(t) = \{s \in Seq(t) \mid \forall \mathcal{A} \in Accept(\mathcal{PA}(t/s))$ *we have that* $\mathcal{A} = \{\emptyset\}\}$

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

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

Definition 14 *Let be $t \in$ 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 $PRes(t)$. $\boldsymbol{D}(PRes)$ is the domain of probabilistic resolutions.*

In the following definitions, we will relate a probabilistic resolution with an finite acceptance tree (fAT) and a probability.

Definition 15 *Let be $s = \langle \mathcal{A}_1 a_1 \mathcal{A}_2 a_2 \ldots \mathcal{A}_n a_n \rangle$ 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 every action is replaced by its homologous barred action. As usual, we will consider that $\overline{\overline{a}} = a$, and we wil use $\hat{}$ as traces concatenation operator.

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

i) $L(t_{fAT}) = L(t_r)$ 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:

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

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

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

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

Definition 16 *We define the subset of probabilistic resolutions from which we obtain the same acceptance set $\mathcal{B}$ as follows*

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

Definition 17 *Let be $\mathcal{PA} = \{(p_1,\mathcal{A}_1),\ldots,(p_n,\mathcal{A}_n\}$ 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, $PATfromPRes(S_{PR})$, that allows us a to build a fPAT from a set of probabilistic resolutions.

Definition 18 *Let $S_{PR} \in \boldsymbol{D}(PRes)$ be a set of probabilistic resolutions. The tree $t = PATfromPRes(S_{PR})$ is obtained as follows:*

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

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

Definition 19 *Let be s and s' sequences and $a \in \mathcal{A}ct$. We define the boolean syncronization function as follows:*

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

Finally we define $(t|t')/\langle \mathcal{B} b\rangle$.

Let be $t_1 = PATfromPRes(SubsetPRes(t,t',\mathcal{B}))$ and $t'_1 = PATfromPRes(SubsetPRes(t',t,\mathcal{B}))$, then

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

Definition 20 *We define the sequences of P, denoted by $Seq(P)$, as the set of all the sequences of pairs (acceptance set,action) that we may obtain from $[\![P]\!]_{fPAT}$.*

Definition 21 *We say that two processes are equivalent according to the denotational semantics, which will be denoted by $[\![P]\!]_{fPAT} = [\![Q]\!]_{fPAT}$, iff:*

- *$Seq(P) = Seq(Q)$, and*
- *For every sequence s, $\mathcal{PA}(P/(s)) = \mathcal{PA}(Q/(s))$.*

Let us note that the model presented here does not preserve some of the properties of the non-probabilistic model (EPL[6]). 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 have that

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

But this is a known consequence of introducing probabilities and nondeterminism considering the approach of resolving firstly the probabilistic choices [12, 15].

We now state that this equivalence relation between probabilistic processes is an extension of that presented in [6] for nondeterministic processes.

Theorem 1 *Let $P,Q \in EPL_f$ be two processes,*

$$[\![P]\!]_{fAT} = [\![Q]\!]_{fAT} \textit{ if and only if } [\![P]\!]_{fPAT} = [\![Q]\!]_{fPAT}$$

Proof (Sketch) We have only to consider that:

- In nondeterministic processes there are not probabilistic choices, and every probabilistic node will have a single outgoing arc labelled with probability 1.
- The construction of acceptance sets and probabilistic acceptance sets is identical for every operator of EPL.

□

Also, we have 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. In order to prove this theorem we need to introduce some concepts. First we introduce the set of probabilities with which a semantic process passes a test.

Definition 22 (Set of probabilities $\phi^D_{(P,T)}$) *Given a semantic process P and a test T, the set of probabilities $\phi^D_{(P,T)}$ is defined as follows:*

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

Let us see how to construct the fPAT obtained from $\phi_{(P,T)}$, $[\![P]\!]^T$:

Construction of $[\![P]\!]^T$ The set of successful test $T = a_1.a_2.\cdots.a_k.w$ 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 every moment, and the set of possible acceptance sets ($\mathcal{A}_i$), but we do not know the probabilities associated to 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 logic expressions associated to acceptance sets. Let us see how we may obtain this set of essential logic 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 logic 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 logic 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 logic expressions that contain only actions in A.
- E_A^B is the set of logic 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 logic expressions may be obtained by using the following procedure, starting with step 1 and repeating steps 2 and 3. In every cycle, only one action is added.

1) $E_\emptyset^\emptyset = \{true\}$.
2) Inheritance. If $a \notin A$ then $E_A^{A\cup\{a\}} = \{e \wedge \neg a \mid e \in E_A^A\}$
3) New expressions. $E_A^A = \{E(\mathcal{A}_i) \mid A \in \mathcal{A}_i\} \cup \{true\}$

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

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

- We associate $\inf\{\phi_{(P,w)}\} = 1$ to the logic expression $true$.
- Let F be a logic 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 logic expression in the form $F \wedge (\bigwedge_i \neg a_i)$. We associate $\inf\{\phi_{(P,T_{F'})}\}$ to F' where $T_{F'}$ is a test in the form $T_F + (\sum_i a_i)$.
- Let F'' be a logic 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 to logic 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. Every 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. In order to do that we sort the equations in the system by using two criteria. Every 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.

Every 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 nonsingular. On the other hand, the inheritance method prevents a test of the group A to be satisfied by an acceptance set of a group B such that $B - A \neq \emptyset$. The system we obtain is:

$$\begin{bmatrix} T^{a_1\cdots a_n}_{a_1\cdots a_n} \\ T^{a_1\cdots a_n}_{a_1\cdots a_{n-1}} \\ \cdots \\ T^{a_1\cdots a_n}_{a_2\cdots a_n} \\ \cdots \\ T^{a_1\cdots a_n}_{a_1} \\ \cdots \\ T^{a_1\cdots a_n}_{a_n} \\ T^{a_1\cdots a_n}_{\emptyset} \end{bmatrix} = \begin{bmatrix} \mathbb{A} \end{bmatrix} \times \begin{bmatrix} q_{a_1 a_2\cdots a_n} \\ q_{a_1 a_2\cdots a_{n-1}} \\ \cdots \\ q_{a_2 a_3\cdots a_n} \\ \cdots \\ q_{a_1} \\ \cdots \\ q_{a_n} \\ q_{\emptyset} \end{bmatrix}$$

where $[\mathbb{A}]$ is the matrix shown in Fig. 4:

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

$$\begin{bmatrix} T^{a_1\cdots a_n}_{a_1\cdots a_n} \end{bmatrix} - \begin{bmatrix} A^{a_1\cdots a_{n-1}}_{a_1\cdots a_n} & \cdots & A^{\emptyset}_{a_1\cdots a_n} \end{bmatrix} \times \begin{bmatrix} q_{a_1\cdots a_{n-1}} \\ \cdots \\ q_\emptyset \end{bmatrix} = \begin{bmatrix} A^{a_1\cdots a_n}_{a_1\cdots a_n} \end{bmatrix} \times \begin{bmatrix} q_{a_1\cdots a_n} \end{bmatrix}$$

and this system has a unique solution because $A^{a_1\cdots a_n}_{a_1\cdots a_n}$ is a nonsingular 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 23 *(Test process after an action a from a state $\mathcal{A}$, $P/(a,\mathcal{A})\|T$)*

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

Definition 24 *(Obtaining $[\![P]\!]^T$ from testing semantics information)*

Using the constructive method introduced and Def. 23 we obtain for a process P/s (where s is a sequence) the probability associated to every acceptance set. The tree $[\![P]\!]^T$ is built just putting together this information (every sequence identify only one node).

Lemma 1 *Let P be a process. For every test T:* $\inf\{\phi_{(P,T)}\} = \inf\{\phi^D_{(P,T)}\}$ *and* $\sup\{\phi_{(P,T)}\} = \sup\{\phi^D_{(P,T)}\}$.

Proof (Sketch) We use structural induction. Let us see the external choice case:

- $P = P_1 + P_2$. We distinguish two different cases. If $A(Accept(\mathcal{PA}(P_1))) \cap A(Accept(\mathcal{PA}(T))) \neq \emptyset$ and $A(Accept(\mathcal{PA}(P_2))) \cap A(Accept(\mathcal{PA}(T))) = \emptyset$ then:
  $$\inf\{\phi_{(P,T)}\} = \inf\{\phi_{(P_1,T)}\}$$
  $$\sup\{\phi_{(P,T)}\} = \sup\{\phi_{(P_1,T)}\}$$
  In $[\![S]\!]$, the acceptance set is obtained as $(\mathcal{A}_{P_1}\ u\ \mathcal{A}_{P_2}) \Cap \mathcal{A}_T$. In this case only process P_1 contributes to the intersection so:
  $$\inf\{\phi^D_{(P,T)}\} = \inf\{\phi^D_{(P_i,T)}\}$$
  $$\sup\{\phi^D_{(P,T)}\} = \sup\{\phi^D_{(P_i,T)}\}$$
  If $A(Accept(\mathcal{PA}(P_1))) \cap A(Accept(\mathcal{PA}(T))) \neq \emptyset$ and $A(Accept(\mathcal{PA}(P_2))) \cap A(Accept(\mathcal{PA}(T))) \neq \emptyset$ then the set of resolutions of $P\|T$ is the union of the resolutions of $P_1\|T$ and $P_2\|T$. Then:
  $$\inf\{\phi_{(P,T)}\} = \inf\{\inf\{\phi_{(P_1,T)}\}, \inf\{\phi_{(P_2,T)}\}\}$$
  $$\sup\{\phi_{(P,T)}\} = \sup\{\sup\{\phi_{(P_1,T)}\}, \sup\{\phi_{(P_2,T)}\}\}$$
  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 $[\![P_2\|T]\!]$:
  $$\inf\{\phi^D_{(P,T)}\} = \inf\{\inf\{\phi^D_{(P_1,T)}\}, \inf\{\phi^D_{(P_2,T)}\}\}$$
  $$\sup\{\phi^D_{(P,T)}\} = \sup\{\sup\{\phi^D_{(P_1,T)}\}, \sup\{\phi^D_{(P_2,T)}\}\}$$

$\square$

Lemma 2 *Let P be a process. Then:*

$$[\![P]\!] = [\![P]\!]^T$$

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

$$\begin{aligned} &\mathcal{PA}(P) = \{(p,\mathcal{A}) : \mathcal{A} = c(\mathcal{B}\cup\mathcal{C}), \\ &\mathcal{B} \in Accept(\mathcal{PA}(P_1)), \mathcal{C} \in Accept(\mathcal{PA}(P_2)), \\ &p = \sum_{\mathcal{A}=c(\mathcal{B}\cup\mathcal{C})} Prob(\mathcal{B},\mathcal{PA}(P_1)) \cdot Prob(\mathcal{C},\mathcal{PA}(P_2))\}. \end{aligned}$$

With respect to the testing semantics, once we know $Accept(\mathcal{PA}(P_1))$ and $Accept(\mathcal{PA}(P_2))$, we can obtain $Accept(\mathcal{PA}(P))$ by using the constructive definition of

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

Figure 4. Matrix $[\mathbb{A}]$

$[\![P]\!]^T$. The set of reachable acceptance sets from P may be considered 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 simultaneously both states $\mathcal{B}_i$ and $\mathcal{C}_i$ to obtain $\mathcal{A}_i$. □

Theorem 2 *(Full abstraction for fPAT)*
$\forall P, Q \in PNAL_f \quad P \equiv_T Q \iff P \equiv_D Q$

Proof Follows directly from lemmas 1 and 2 □

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 with 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 processes (probabilistically stable processes) and they are very useful when proving completeness.

Table 4, contains the set of axioms related to the following operators: **0**, prefix and the three choices. 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 say that probabilistic choice is propagated to the following level when both processes are prefixed by the same action (P4), and that the probabilistic choice distributes over the internal and the external choice (PI1, PE1).

Let us observe that the axioms maintained from EPL are: the internal and the external choice are commutative and associative (I1,I2,E1,E2), the external choice has a zero (E3), and, finally, the internal and the external choice between processes prefixed with the same action are propagated to the following level (I3,IE1). The rest of axioms (those with the symbol $\downarrow$), do not hold in the general case, but if we restrict P, Q and R to a kind of processes, 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$\downarrow$).

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	$a.P + a.Q \equiv a.P \oplus a.Q$
I4$\downarrow$	$P \oplus P \equiv P$
E4$\downarrow$	$P + P \equiv P$
IE3$\downarrow$	$P + (Q \oplus R) \equiv (P + Q) \oplus (P + R)$
IE4$\downarrow$	$P \oplus (Q + R) \equiv (P \oplus Q) + (P \oplus R)$

Table 4. Set of axioms for $\mathbf{0}$, prefix and choice operators

We may also apply, over probabilistically stable processes, the derived equations given in [6], which are useful to prove completeness. Finally, in Table 5 we can see the axioms for restriction and parallel composition, which, as usual, are derived operators. Note that axioms PaI1 and PaE1, which are introduced in [6], hold only for processes in EPL$_f$; applying repeatedly P4, PI1, PE1, ReP1, and PaP1 (distribution axioms) every process in the form $P|Q$ may be transformed into $\bigotimes_{i,j}[p_i \cdot q_j]P_i|Q_j$, where $P_i, Q_j \in \mathrm{EPL_f}$.

RE1	$(P + Q)\backslash C = P\backslash C + Q\backslash C$
RI1	$(P \oplus Q)\backslash C = P\backslash C \oplus Q\backslash C$
R1	$a.P\backslash C = \begin{cases} a.(P\backslash C) & \text{if } a \notin C \vee \overline{a} \notin C \\ \mathbf{0} & \text{otherwise} \end{cases}$
R01	$\mathbf{0}\backslash C = \mathbf{0}$
RP1	$(P \otimes_q Q)\backslash C = P\backslash C \otimes_q Q\backslash C$
PaP1	$P|(Q \otimes_q R) = (P|Q) \otimes_q (P|R)$
Pa01	$\mathbf{0}|P = P$
PaI1	$(P \oplus Q)|R = P|R \oplus Q|R, \quad P, Q, R \in \mathrm{EPL_f}$
PaE1	if P, Q are $\sum\{a_iP_i, i \in I\}$, $\sum\{b_jQ_j, j \in J\}$ then $P|Q = \begin{cases} ext(P,Q) \text{ if } comm(P,Q) \text{ is false} \\ (ext(P,Q) + int(P,Q)) \oplus int(P,Q) \\ \text{otherwise} \end{cases}$ 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 $P, Q \in \mathrm{EPL_f}$

Table 5. Axioms for restriction and parallel composition

Example 2 *Let us consider the process in figure 3.*

$$(a.(c \otimes_{0.4} e) \otimes_{0.3} b)|\overline{a}.d \overset{P4}{=} ((a.c \otimes_{0.4} a.e) \otimes_{0.3} b)|\overline{a}.d$$
$$\overset{P2}{=} ([0.12]a.c \otimes [0.18]a.e \otimes [0.3]b)|\overline{a}.d$$
$$\overset{PaP1}{=} [0.12]a.c|\overline{a}.d \otimes [0.18]a.e|\overline{a}.d \otimes [0.3]b|\overline{a}.d$$

Finally, we apply axiom PaE1 as usual.

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

Theorem 3 *Axioms are sound with respect to the denotational semantics.*

Proof (sketch) We will prove only some of the axioms (P2, PI1, PE1). Very similar ideas can be applied to prove the soundness of the remaining axioms, thus these proof are omitted. Then, we consider three processes P, Q and R such that:

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

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

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

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

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

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

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

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

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

- **PE1** $\quad P + (Q \otimes_p R) = (P + Q) \otimes_p (P + R)$

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

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

$\square$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 among a set of states, which is followed by a generalized prefixed external choice among the actions in this set, whose continuations are also in normal form.

Definition 25 *(Normal forms)*

- *Process **0** is 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 some characteristics of other probabilistic models [3]. 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 as usual in classical models.

Lemma 3 *Every term in PNAL can be transformed, by using the given proof system, into another equivalent term in normal form.*

Proof We proceed by induction on the depth of the term. The base case is immediate (**0** is already in normal form). For the induction step let us consider the different operators. Prefix operator and probabilistic choice cases are trivial so we will focus on the rest.

- For $P = Q \oplus R$ we have:

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

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

Then, we have

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

Applying repeatedly axiom PI1 we obtain:

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

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

- For $P = Q + R$ 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 axiom PE1 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$\downarrow$, IE1 and the derived equation Der3$\downarrow$ 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.

$\square$

Theorem 4 *Let $P, Q \in$ PNAL be two processes, then*

$$[\![P]\!]_{fPAT} = [\![Q]\!]_{fPAT} \text{ if and only if } P \equiv Q$$

Proof By Lemma 3 we can restrict the proof to normal forms. If P and Q are in normal form we have:

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

$$Q \equiv \bigotimes_{i=1}^{m} [r_i] \bigoplus_{B\in\mathcal{B}_i} \sum_{b\in B} b.m(\mathcal{B}_i, b)$$

$[\![P]\!] = [\![Q]\!]$ so by Definition 21 we have:

- $Seq(P) = Seq(Q)$, and
- For every sequence s, $\mathcal{PA}(P/(s)) = \mathcal{PA}(Q/(s))$.

The base case of induction is $P = Q = \mathbf{0}$, so it is trivial that $P \equiv Q$.

When $s = \varepsilon$, we have that $q_i = r_i$ and $\mathcal{A}_i = \mathcal{B}_i$ for every index i.

Finally, $[\![P]\!](\langle \mathcal{A}_i a \rangle) = [\![Q]\!](\langle \mathcal{A}_i a \rangle)$, so by induction hypothesis we can assume that $n(\mathcal{A}_i, a) \equiv m(\mathcal{A}_i, a)$. $\square$

Finally, by using theorems 2 and 4, 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 future work

We have presented an algebraic language for probabilistic and nondeterministic processes, called PNAL, which is a probabilistic extension of EPL. 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. This decision leads us to a model in which some of the properties of classical process algebras are not maintained (e.g. idempotency of the internal choice). An operational and a testing semantics have been introduced, as well as a denotational semantics and a proof system. It has been established that the testing semantics, the denotational semantics and the proof system are fully abstract.

Our work in progress is concerned with extending these results to the full language, which includes recursion.

References

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

[2] A. Bueno, V. Valero, and F. Cuartero. TPAL: A timed-probabilistic model for concurrent processes. In *Proceedings of APSEC'97/ICSC'97*, Hong-Kong, 1997.

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

[4] A. Giacalone, C.-C. Jou, and S. 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.

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

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

[7] 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.

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

[9] M. Kwiatkowska and G. Norman. A testing equivalence for reactive probabilistic processes. In *Proc. EXPRESS'98, Electronic Notes in TCS vol.16*, 1998.

[10] K. Larsen and A. Skou. Bisimulation through probabilistic testing. In *Proceedings of 16th ACM Symp. on Principles of Programming Languages*, Austin, TX, 1989.

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

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

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

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

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