A Denotational Model for Probabilistic and Nondeterministic Processes

D. Cazorla F. Cuartero V. Valero F. L. Pelayo
Departamento de Informática *
Escuela Politécnica Superior de Albacete
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 model for the description of probabilistic and nondeterministic processes: PNAL (Probabilistic and Nondeterministic Algebraic Language), an extension of EPL, defined by Matthew Hennessy, including a probabilistic choice operator. An operational semantics and a testing semantics for the language are presented. A denotational semantics for the finite subset of the language, $PNAL_f$, is proposed. Finally, we provide two results: equivalence between nondeterministic processes in finite EPL is preserved in $PNAL_f$, and denotational semantics is fully abstract with respect to the testing semantics.

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

1. Introduction

In the eighties, the fundamentals of process algebras were firmly established thanks to the works of several authors. In these works only qualitative aspects of concurrent systems were considered. Later quantitative aspects have been studied by several researchers. Thus, several extensions to classical process algebra models have been proposed 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 [7] 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 the process behaves like E_i. In [17], this model is extended and three interpretations for concurrent probabilistic processes are presented: reactive, generative and stratified models.

In [8] 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 in the top level of the syntax: probabilistic and nondeterministic processes. The alternating of both kinds of processes is necessary, this is why the model is called *alternating model*. Besides the models already mentioned, several languages based on CCS [4, 11, 13] that analyze several semantic aspects have been proposed. Thus, in [13] a notion of bisimulation between probabilistic transition systems is introduced; this topic is also studied in [4]. In [11] several equivalences between processes are presented.

Based on CSP, we can cite [5, 6], where a *reactive* interpretation is considered, whereas in [16] 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 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.

So, 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 may be quantified (for instance, the probability that a faulty communication channel fails), whereas in other occasions it cannot (for instance, relative speed between two processes). This is why we need a language that allows us to model nondeterministic systems

* This work has been partially supported by the CICYT project "Formal development of distributed systems", TIC97-0669-c03-02.

with or without quantification, depending on the nature of the system.

Following these ideas, in [18] 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 generalizing, for probabilistic and nondeterministic processes, may, $must$ and $test$ preorders in the testing theory of De Nicola and Hennessy. This work is extended in [10], where these preorders are redefined and a denotational characterization for may–testing and must–testing is presented.

In [14] 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. In [3] TPAL is introduced, an algebraic language with time and probabilities, based on LOTOS, where internal choice is replaced by a probabilistic choice. Alternation of both probabilistic and nondeterministic terms is required.

Following the ideas of [18, 10], in this paper we will introduce a probabilistic extension that maintains nondeterminism. In order to deal with probabilities and nondeterminism we need to choose which of the two choices (probabilistic or internal choice) has to be resolved first. Some models [1, 15, 18] consider that probabilistic choice is resolved first, whereas other models [14] consider the opposite approach. A different point of view is considered in [12], where an action–guarded reactive probabilistic choice is introduced and the point at which probabilistic choice occurs has no significance.

The decision of which choice is made first is very important because, depending on the approach adopted, some properties of classical process algebras are not maintained. For example, idempotency of internal choice is only maintained when internal choice is resolved first. Idempotency of external choice is never maintained (e.g. in all the models, if we consider the process $P = a \otimes_{0.5} b$, then $P \,\Box\, P$ increments the probability with which action a is executed).

We have chosen the first approach (i.e. probabilistic choice is resolved before internal choice) for two reasons. The first one because this approach leads to a denotational semantics that is a very intuitive extension of the denotational semantics of EPL (acceptance trees). Even when restricting to processes without probabilistic choices the semantics objects we obtained are identical (except for the probabilistic branches labelled with probability 1). This not occurs in [14] where the semantics of a process without probabilistic choices is not given in terms of refusals but in terms of probability-action graphs.

The second reason is related with normal forms. One of our future works will be concerned with finding a sound and complete proof system with respect to both semantics. For that, we will need to define a kind of normal forms. A very natural definition is formed of a probabilistic choice on top, followed by a pure nondeterministic process, which is a natural extension of EPL normal form.

In our opinion, the advantage of this natural definition of normal form justifies that idempotency property of internal choice is not fully maintained in the general case for probabilistic processes. Moreover, it is difficult to imagine the aspect of the normal forms when considering the approach taken in [14].

We have to emphasize that, when restricting to processes without probabilistic choices (probabilistically stable processes), idempotence for both choices is fully maintained, and then, our model is a full extension of EPL.

The paper is structured as follows. Section 2 presents the syntax of the language as an extension of EPL [9]. In Section 3 we present an operational semantics for the whole language. In Section 4 a testing semantics is presented. In Section 5 a denotational semantics is given for a finite version of the language ($\mathrm{PNAL_f}$) by means of probabilistic acceptance trees and we prove that the equivalence between nondeterministic processes in finite EPL is maintained in our model. We also prove that denotational semantics is fully abstract with respect to testing semantics.

2. Syntax of the language

The language we present, PNAL, is based on EPL [9] and provides three kinds of choice: classical internal choice, denoted by $\oplus$, classical external choice, denoted by $+$, and a new choice operator, a probabilistic choice with probability p: $\otimes_p$.

Let Δ be a set of input actions, $\overline{\Delta} = \{\overline{a} \mid a \in \Delta\}$ 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:

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

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

The operators are interpreted as usual. The probabilistic choice, $P \otimes_p Q$, has a *generative* meaning: with probability

p the process behaves like P and with probability $1-p$ it behaves like Q. The decision is made internally, and every probabilistic choice is considered independent of the others.

3. Operational semantics

Operational semantics is defined in the usual way by using a labelled transition system. The transition rules are divided into 3 parts (Tables 1–3). First we need to introduce some auxiliary concepts.

Definition 1 *We inductively define the probabilistically stable predicate*

i) $\mathbf{0}$*,* $a.P$ *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 the process* P *is probabilistically stable.*

The first table, given in Table 1, defines the inference rules for probabilistic transitions, $P \longrightarrow_p Q$, which means that P may evolve to Q immediately with probability p. 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 indicate that probabilistic choice is resolved before the other operators. In rules P2c, P3c, and P4c, it is assumend that P and Q are probabilistically independent processes; the same occurs in [1, 8, 18]. For rules in Tables 2 and 3 we assume that processes are probabilistically stable. The second table defines the rules for unlabelled transitions, i.e., those representing 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 external choice and parallel composition. Rule I3c indicates that an unbarred action may synchronize with its homologous barred action and the result process evolves internally. Finally, rules I4 and I5 show the operational behaviour for restriction and recursion operators. The third table defines the inference rules for transitions labelled by actions in $\mathcal{A}ct$, $P \xrightarrow{a} Q$, with the usual meaning that P may perform action a and it evolves into Q.

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

Table 1. Inference rules for transitions labelled by probabilities

(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)	$\dfrac{P \xrightarrow{a} P', Q \xrightarrow{\overline{a}} Q'}{P\|Q \longrightarrow P'\|Q'}$		
(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 ($P\downarrow$ 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.

In figure 1 we illustrate the application of the rules over a process; every transition is labelled with the name of the applied rule.

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

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

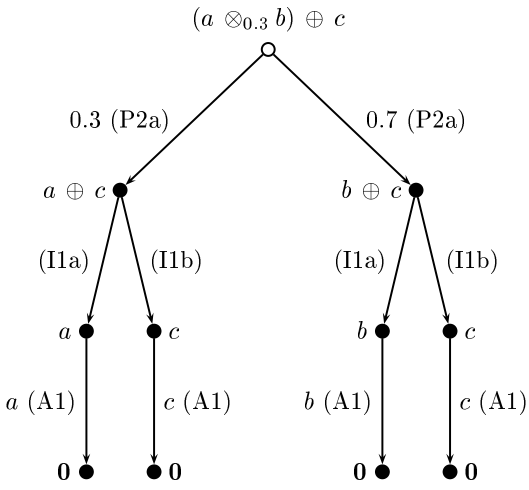

Figure 1. Transition system of the process $(\mathbf{a.0} \otimes_{\mathbf{0.3}} \mathbf{b.0}) \oplus \mathbf{c.0}$

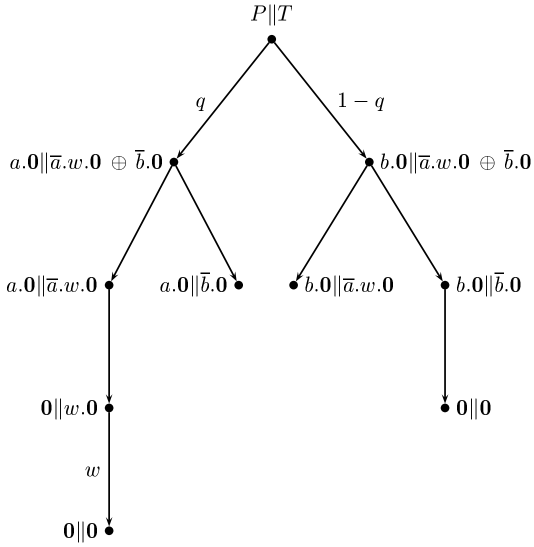

Figure 2. Transition system of the test process $\mathbf{a} \otimes_{\mathbf{q}} \mathbf{b} \,||\, \overline{\mathbf{a}}.\mathbf{w} \oplus \overline{\mathbf{b}}$

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 test T with a certain probability p. On the contrary, if the process is nondeterministic (i.e. it contains internal choices) we cannot compute a unique probability; instead, we have a set of probabilities to pass each test.

In order to relate two processes using the testing semantics, we take one of them as a specification and the other 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 are represented by S, S', etc. In figure 2 we see the transition system of a test process. Intuitively, given a process P and a test T, we consider $\phi_{(P,T)}$ as the set whose elements are the probabilities obtained taking into account each possible evolution of $P||T$. In order to define $\phi_{(P,T)}$ we need some previous definitions, following the ideas presented in [18].

Definition 3 *A resolution R of $P||T$ is a maximal subtree of $P||T$ with branches labelled with probabilities and unlabelled branches, in such a way that for every node S, whenever $S \longrightarrow S_1$ and $S \longrightarrow S_2$, we have $S_1 = S_2$.*

In other words, every nondeterministic node in a resolution has a unique unlabelled output arc. In figure 3 we show one of the resolutions of the test process in figure 2.

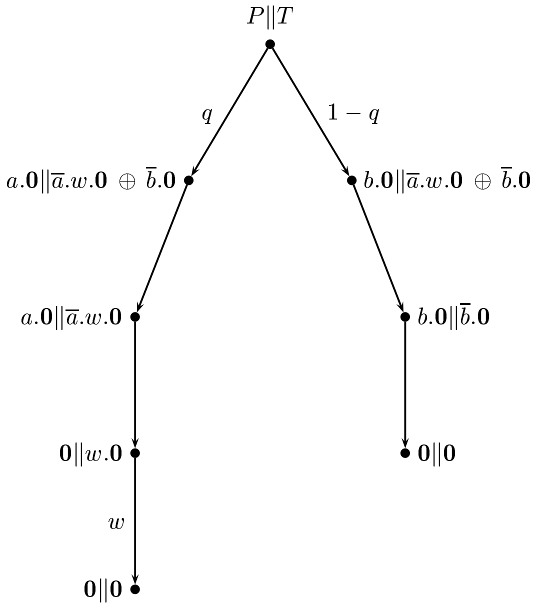

Figure 3. One of the resolutions of the transition system of $\mathbf{a} \otimes_{\mathbf{q}} \mathbf{b} \,||\, \overline{\mathbf{a}}.\mathbf{w} \oplus \overline{\mathbf{b}}$

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 $\longrightarrow_p$ or $\longrightarrow$.

We will use $\top$ to denote a successful state and $\perp$ to denote an unsuccessful state.

We establish the following notation:

1. $\mathcal{R}(S)$ denotes the set of resolutions of S.
2. $\mathcal{C}(R)$ denotes the set of computations of resolution R.
3. $Pr(C;R;S)$ denotes the probability for the test process S to reach a successful state along the computation C in the resolution R.
4. $Pr(R;S)$ denotes the probability for S 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 resolution R. The probability of success in $C \in \widehat{\mathcal{C}}(R)$ is defined as follows*

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

Definition 6 *The probability of success in a resolution is defined as*

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

Definition 7 *Given a process P and a test T, we say that P* may pass *T with the set of probabilities $\phi_{(P,T)} = \{Pr(R;P\|T) \mid R \in \mathcal{R}(P\|T)\}$.*

Definition 8 *We define the following preorders over processes:*

1. *$P \sqsubseteq_{best} Q$ if $\forall T \ \sup\{\phi_{(P,T)}\} \geq \sup\{\phi_{(Q,T)}\}$*
2. *$P \sqsubseteq_{worst} Q$ if $\forall T \ \inf\{\phi_{(P,T)}\} \leq \inf\{\phi_{(Q,T)}\}$*
3. *$P \sqsubseteq_{test} Q$ if $P \sqsubseteq_{best} Q$, $P \sqsubseteq_{worst} Q$ and $\forall T$ if $\phi_{(Q,T)} = \{0\}$ then $\phi_{(P,T)} = \{0\}$*

According to the definition of test preorder given in [10], 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 $\cancel{may}$ $c.w$ so $P \not\sqsubseteq_{may} Q$. The last condition in our definition of test preorder is introduced in order to consider these situations.

The $test$ preorder has the meaning indicated above: $P \sqsubseteq_{test} Q$ if Q is a good implementation of the specification P. The corresponding equivalence for the test preorder is defined as follows:

Definition 9 *$P \equiv_{test} Q$ iff $P \sqsubseteq_{test} Q$ and $Q \sqsubseteq_{test} P$.*

5. Denotational semantics

The aim of this section is to provide our language with a denotational semantics which is fully abstract with respect to the testing semantics. Our domain will be a set of finite probabilistic acceptance trees, fPAT. We will restrict ourselves to the finite case (PNAL$_\mathrm{f}$), which includes **0**, the prefix operator and the three choice operators. Let us informally describe the structure of these trees:

- We will consider trees with two kinds of nodes: probabilistic and nondeterministic. The probabilistic nodes are represented by $\otimes$ and the nondeterministic nodes by $\bullet$.
- The root node is a probabilistic node and there is an alternation of probabilistic and nondeterministic nodes.
- Nondeterministic nodes are labelled with an acceptance set (as defined in [9]). Probabilistic nodes are not labelled.
- The branches leaving a probabilistic node are labelled with probabilities p_i such that $\sum_i p_i = 1$. Every branch leaving a nondeterministic node is labelled by one of the actions appearing in the acceptance set. There do not exist two branches labelled by the same action.
- A terminal node is always a nondeterministic node labelled by the acceptance set $\{\emptyset\}$.

These kind of trees are very similar to those used in [9]. However, in our model we have to include probabilistic information related to the probabilistic distribution associated with every reachable state, so we need to expand the trees in order to include this new information.

Definition 10 *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 $(p,\mathcal{A}) \in (0,1] \times \boldsymbol{D}(\mathcal{A})$.*

Definition 11 *Let $\mathcal{A}$ be an acceptance set. We define $A(\mathcal{A})$, the set of actions belonging to $\mathcal{A}$, as follows:*

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

Definition 12 *A probabilistic acceptance set, $\mathcal{PA}$, is defined as follows:*

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

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

Definition 13 *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 14 *Let* $\mathcal{PA}$ *be a probabilistic acceptance set. We define the set of acceptance sets belonging to* $\mathcal{PA}$ *as follows:*

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

Besides the definitions above, we will need two additional operators borrowed from [9]: 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.

The probabilistic acceptance set of each process at top level, $\mathcal{PA}(P)$, is defined as follows:

Stop. $\mathcal{PA}(\mathbf{0}) = \Big\{ \Big(1, \{\emptyset\}\Big) \Big\}$.

Prefix. $\mathcal{PA}(a.P) = \Big\{ \Big(1, \{\{a\}\}\Big) \Big\}$.

Probabilistic choice. Let $P = P_1 \otimes_q P_2$.

$$\begin{aligned} &\mathcal{PA}(P) = \{(p, \mathcal{A}) \mid \\ &\mathcal{A} \in Accept(\mathcal{PA}(P_1)) \vee \mathcal{A} \in Accept(\mathcal{PA}(P_2)), \\ &p = q \cdot Prob(\mathcal{A}, \mathcal{PA}(P_1)) + (1-q) \cdot Prob(\mathcal{A}, \mathcal{PA}(P_2))\} \end{aligned}$$

Internal choice. Let $P = P_1 \oplus P_2$.

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

External choice. Let $P = P_1 + P_2$.

$$\begin{aligned} \mathcal{PA}(P) = \{(p, \mathcal{A}) \mid &\exists \mathcal{B} \in Accept(\mathcal{PA}(P_1)), \\ &\exists \mathcal{C} \in Accept(\mathcal{PA}(P_2)) \text{ s.t. } \mathcal{A} = \mathcal{B} \; u \; \mathcal{C}, \\ &p = \sum_{\mathcal{A} = \mathcal{B} \; u \; \mathcal{C}} Prob(\mathcal{B}, \mathcal{PA}(P_1)) \cdot Prob(\mathcal{C}, \mathcal{PA}(P_2))\} \end{aligned}$$

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

Once process P performs its first action a (from the state $\mathcal{A}$) it becomes the process $P/(\mathcal{A}, a)$; then in order to get its probabilistic acceptance set, we need an equivalent syntactic expression for $P/(\mathcal{A}, a)$, which can be obtained as follows:

- If P is a process such that $a \notin A(\mathcal{A})$ or $\mathcal{A} \notin Accept(\mathcal{PA}(P))$ then $P/(\mathcal{A}, a)$ is not defined.
- Prefix. Let $P = a.P_1$. The only reachable state is $\{\{a\}\}$ with probability 1. Then $P/(\{\{a\}\}, a) = P_1$.
- Probabilistic choice. Let $P = P_1 \otimes_q P_2$. We can distinguish several cases:

  $\mathcal{A} \in Accept(\mathcal{PA}(P_1))$ and $\mathcal{A} \notin Accept(\mathcal{PA}(P_2))$. In this case we can only perform the action a from P_1, so $P/(\mathcal{A}, a) = P_1/(\mathcal{A}, a)$.

  $\mathcal{A} \notin Accept(\mathcal{PA}(P_1))$ and $\mathcal{A} \in Accept(\mathcal{PA}(P_2))$. Similar to the previous case: $P/(\mathcal{A}, a) = P_2/(\mathcal{A}, a)$.

  $\mathcal{A} \in Accept(\mathcal{PA}(P_1))$ and $\mathcal{A} \in Accept(\mathcal{PA}(P_2))$. In this case the performed action a can be made either by P_1 or P_2, then:

  $$P/(\mathcal{A}, a) = P_1/(\mathcal{A}, a) \otimes_q P_2/(\mathcal{A}, a)$$

- Internal choice. Let $P = P_1 \oplus P_2$. 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 the union of states. Let us consider that the state $\mathcal{A}$ has been obtained as $\mathcal{A} = c(\mathcal{B} \cup \mathcal{C})$ where $\mathcal{B} \in Accept(\mathcal{PA}(P_1))$ and $\mathcal{C} \in Accept(\mathcal{PA}(P_2))$. For every pair of acceptance sets $\mathcal{B}$ and $\mathcal{C}$ such that $\mathcal{A} = c(\mathcal{B} \cup \mathcal{C})$ we can distinguish the following cases:

  - $a \in A(\mathcal{B})$ and $a \notin A(\mathcal{C})$. In this case we can only perform action a from P_1 so: $P/(\mathcal{A}, a) = P_1/(\mathcal{B}, a)$.
  - $a \notin A(\mathcal{B})$ and $a \in A(\mathcal{C})$. Symmetric case: $P/(\mathcal{A}, a) = P_2/(\mathcal{C}, a)$.
  - $a \in A(\mathcal{B})$ and $a \in A(\mathcal{C})$. In this case the performed action a can be made either by P_1 or P_2, then: $P/(\mathcal{A}, a) = P_1/(\mathcal{B}, a) \oplus P_2/(\mathcal{C}, a)$

  The general solution, taking into account that $\mathcal{A}$ may be obtained from different $\mathcal{B}$ and $\mathcal{C}$, is:

  $$\begin{aligned} &P/(\mathcal{A}, a) = \\ &\bigoplus_{\mathcal{B},\mathcal{C}} P_1/(\mathcal{B}, a) \oplus && \text{if } a \in A(\mathcal{B}) \wedge a \notin A(\mathcal{C}) \\ &\bigoplus_{\mathcal{B},\mathcal{C}} P_2/(\mathcal{C}, a) \oplus && \text{if } a \notin A(\mathcal{B}) \wedge a \in A(\mathcal{C}) \\ &\bigoplus_{\mathcal{B},\mathcal{C}} P_1/(\mathcal{B}, a) \oplus P_2/(\mathcal{C}, a) && \text{if } a \in A(\mathcal{B}) \wedge a \in A(\mathcal{C}) \end{aligned}$$

- External choice. Let $P = P_1 + P_2$. In this case, every state $\mathcal{A}$ is built as $\mathcal{A} = \mathcal{B} \; u \; \mathcal{C}$ where $\mathcal{B} \in$

$Accept(\mathcal{PA}(P_1))$ and $\mathcal{C} \in Accept(\mathcal{PA}(P_2))$. External choice is propagated to the following level as an internal choice, so we obtain the same result as in the previous case.

Each probabilistic node in the tree defining the denotational semantics of a process P may be reached by following a "sequence" of pairs acceptance set–action, $\langle \mathcal{A}_1 a_1 \ldots \mathcal{A}_k a_k \rangle$. Therefore, we will identify these nodes by $[\![P]\!](s)$, where s is a sequence. In particular, the root node of the tree will be represented as $[\![P]\!](\langle\rangle)$. Using the same notation we can also identify syntactical terms. So, the process P will be represented as $P/(\langle\rangle)$, $P/(\mathcal{A}, a)$ as $P/(\langle \mathcal{A}a \rangle)$, $(P/(\mathcal{A}, a))\,/(\mathcal{B}, b)$ as $P/(\langle \mathcal{A}a\mathcal{B}b \rangle)$ and so on.

Example 1

- *Let* $P \equiv P_1 \otimes_q P_2 \equiv a.\boldsymbol{0} \otimes_q b.\boldsymbol{0}$.

$$\begin{aligned}
\mathcal{PA}(P) &= \mathcal{PA}(P_1) \otimes_q \mathcal{PA}(P_2) \\
&= \Big\{ \Big(1, \{\{a\}\}\Big) \Big\} \otimes_q \Big\{ \Big(1, \{\{b\}\}\Big) \Big\} \\
&= \Big\{ \Big(q, \{\{a\}\}\Big), \Big(1-q, \{\{b\}\}\Big) \Big\}
\end{aligned}$$

$$\mathcal{PA}(P/(\{\{a\}\}, a)) = \mathcal{PA}(\boldsymbol{0}) = \Big\{ \Big(1, \emptyset\Big) \Big\}$$

$$\mathcal{PA}(P/(\{\{b\}\}, b)) = \mathcal{PA}(\boldsymbol{0}) = \Big\{ \Big(1, \emptyset\Big) \Big\}$$

- *Let* $P \equiv P_1 \oplus P_2 \equiv a.\boldsymbol{0} \oplus b.\boldsymbol{0}$.

$$\begin{aligned}
\mathcal{PA}(P) &= \mathcal{PA}(P_1) \oplus \mathcal{PA}(P_2) \\
&= \Big\{ \Big(1, \{\{a\}\}\Big) \Big\} \oplus \Big\{ \Big(1, \{\{b\}\}\Big) \Big\} \\
&= \Big\{ \Big(1, \{\{a\}, \{b\}, \{a,b\}\}\Big) \Big\}
\end{aligned}$$

$$\mathcal{PA}(P/(\{\{a\}, \{b\}, \{a,b\}\}, a)) = \mathcal{PA}(\boldsymbol{0}) = \Big\{ \Big(1, \emptyset\Big) \Big\}$$

$$\mathcal{PA}(P/(\{\{a\}, \{b\}, \{a,b\}\}, b)) = \mathcal{PA}(\boldsymbol{0}) = \Big\{ \Big(1, \emptyset\Big) \Big\}$$

A graphical representation of the denotational semantics of these processes is given in Fig. 4.

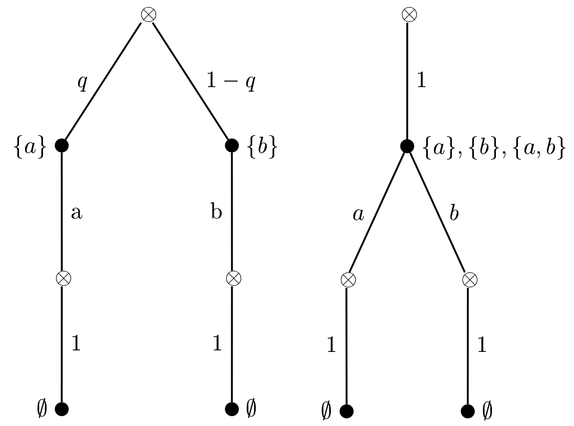

Figure 4. fPAT for $\mathbf{a} \otimes_{\mathbf{q}} \mathbf{b}$ and $\mathbf{a} \oplus \mathbf{b}$

Definition 15 *We define the sequences of the process P, denoted $Seq(P)$, as the set of all the sequences of pairs acceptance set–action we may obtain from $[\![P]\!]_{fPAT}$.*

Definition 16 *We say two processes are equivalent according to the denotational semantics, denoted by $[\![P]\!]_{fPAT} \equiv [\![Q]\!]_{fPAT}$, iff:*

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

The model presented here does not preserve some of the properties of the non-probabilistic model (EPL[9]). For example, internal choice now fails to be idempotent. Let P be the process $a.\mathbf{0} \otimes_{0.5} b.\mathbf{0}$, the probabilistic acceptance set at the top level of P and $P \oplus P$ are

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

The same occurs in other models [15, 18] where a similar approach to probabilistic choice is considered. On the contrary in [14] internal choice remains idempotent; this occurs because internal choice is resolved before probabilistic choice.

In the following theorem we state that the equivalence between probabilistic processes presented in this paper is an extension of that presented in [9] for nondeterministic processes.

Theorem 1 *Let $P, Q \in EPL_f$. We have* $[\![P]\!]_{fAT} \equiv [\![Q]\!]_{fAT} \iff [\![P]\!]_{fPAT} \equiv [\![Q]\!]_{fPAT}$

Finally, we have that the testing semantics and the denotational semantics presented in this paper identify the same processes, i.e., the denotational semantics is fully abstract with respect to the testing semantics.

Theorem 2 *Let $P, Q \in PNAL_f$. We have* $P \equiv_{test} Q \iff [\![P]\!]_{fPAT} \equiv [\![Q]\!]_{fPAT}$

6. 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 consider that probabilistic choice is resolved before 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 property of internal choice). An operational and a testing semantics for the whole language have been presented, and a denotational semantics for a restricted (finite) part of the language has been defined. It has been established that equivalence between nondeterministic processes in $\mathrm{EPL}_{\mathrm{f}}$ is maintained in $\mathrm{PNAL}_{\mathrm{f}}$. Also we have established that both, testing and denotational semantics, are equivalent.

We are currently working in the full version of the language, including recursion, parallel composition and restriction operator.

Our future work will be concerned with finding a sound and complete proof system with respect to both semantics. For that, we will need to define a kind of normal forms. A very natural definition is the following one

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

The aspect of this definition is a natural extension of EPL normal form, just considering a probabilistic choice on top, followed by a pure nondeterministic process.

References

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

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

[3] 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.

[4] I. Christoff. *Testing Equivalence for Probabilistic Processes*. PhD thesis, Uppsala University, Uppsala, 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 *Proceedings of Fourth AMAST Workshop on Real-Time Systems*, 1997.

[7] 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.

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

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

[10] 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.

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

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

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

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

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

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

[17] R. van Glabbeek, S. Smolka, B. Steffen, and C. 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.

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