Testing Semantics for Probabilistic LOTOS

Manuel Núñez and David de Frutos
Facultad de CC. Matemáticas, Dept. de Informática y Automática
Universidad Complutense de Madrid. E-28040 Madrid. Spain.
email: {manuelnu,defrutos}@eucmax.sim.ucm.es

Abstract

In this paper we present a probabilistic extension of LOTOS which is upward compatible with LOTOS. We present testing semantics for the reactive and generative models described in [vGSST90]. While there is a certain *lose* of the meaning of probabilities in the reactive model, testing with probabilistic tests proves to be *too strong*, because it does not relate behavior expressions which we expect to be equivalent. This is why we introduce the *limited generative* model, where tests are not allowed to have explicit probabilities. We give a fully abstract characterization for the reactive model, while we give alternative characterizations (based on a set of essential tests) for the generative and limited generative models. We also present some algebraic laws for each of the models, including some laws which establish the difference between the three models.

Keywords

Distributed Systems; LOTOS; Probabilistic Processes; Testing Semantics

1 INTRODUCTION

LOTOS [LOT88] has been widely used to specify concurrent systems. Nevertheless, it is not completely adequate for the specification of *real* systems. In order to specify such systems, LOTOS has been extended with probabilistic (e.g. [MFV93, KLL94]) and time behaviors (e.g. [QdFA93, dFLL$^+$94]). These extensions are very useful for the specification of communication protocols, real-time systems and fault-tolerant systems. For example, suppose that one wants to specify a communication protocol over a faulty medium. Using LOTOS without probabilities to specify the system, it cannot be inferred that the message will arrive for sure. By introducing probabilities, if the probability that a message is lost is fixed, and the protocol (e.g. the ABP protocol) iterates the sending of the message until the sender knows that the message has arrived, then it can be inferred that with a probability equal to one, the message will (sooner or later) arrive.

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

In the field of probabilistic process algebras, a classification of the different models has been proposed [vGSST90], which is based on the way of the interaction between the processes and the environment. In this paper, we extend some LOTOS operators with a probability, and we study how two of these models (the reactive and generative ones) can be studied in the framework of testing semantics. But we will point out that these models do not fulfill some natural properties and that is why we define another model, the *limited generative* model, showing that those properties are fulfilled in this new model.

The main advantage of our work, in contrast with [vGSST90], is that we do not need either to restrict the (syntactic) form of behavior expressions nor to define a different operational semantics for each model; the syntax of probabilistic LOTOS and its operational semantics will be the same for the reactive, the generative and the limited generative models. The differences between the models come from the different families of tests that are considered in each model, and thus the same (syntactic) specification will have *different* meanings, if considered in different models.

RELATED WORK

[vGSST90] discusses the *reactive, generative* and *stratified* models for PCCS. According to the interaction between the processes and the environment, it is defined the *reactive* model as the model where the environment may only offer a single action at a time. In the *generative* model, the environment can simultaneously offer several actions and the process chooses between them according to some probability distribution. The authors define for each model strong bisimulations and give abstraction functions from one model to another.

[GJS90] presents a probabilistic version of synchronous CCS (called PCCS) where the sum operator is extended with probabilities (i.e. $\sum_{i\in I}[p_i]E_i$, $p_i \in (0,1]$ and $\sum p_i = 1$). In [YL92], CCS is extended with a nondeterministic probabilistic choice operator. A new notion of testing semantics is defined, where a process *can pass* a test with a set of probabilities. In a recent work [JY95], fully abstract characterizations of the testing preorders defined in [YL92] are presented.

Using probabilistic labeled transition systems, [Chr90] presents four partial orders based on testing. [CSZ92] presents a testing semantics for probabilistic processes; as in the nonprobabilistic case, processes and tests are essentially the same, while in [Chr90] they were different. [YCDS94] presents an alternative characterization of the testing preorder given in [CSZ92], introducing a set of essential tests, called probabilistic traces, which are proved to have the same strength that the full family of tests.

[NdFL95] presents a testing semantics for a probabilistic process algebra with two choice operators. An alternative characterization based on *acceptance sets* and a fully abstract denotational semantics based on *acceptance trees* are given.

Also in the framework of LOTOS, some probabilistic extensions have been studied. [MFV93] presents a probabilistic version of LOTOS, where a binary (nondeterministic) probabilistic choice operator is introduced. In [KLL94], a true concurrency semantics (based on event structures) is given to a probabilistic extension of a subset of LOTOS. Again, the extension consists in the inclusion of a nondeterministic probabilistic choice operator.

The structure of the rest of this paper is as follows. Section 2 presents the operational semantics of our language and the associated testing semantics. Section 3 examines the reactive model, including a denotational semantics which is proved to be fully abstract, and the generative and limited generative models, presenting alternative characterizations of the testing preorders. Section 4 shows how the standard *may* and *must* testing preorders can be recovered from the generative model. Section 5 presents some algebraic laws which respect the different testing equivalences. Section 6 discusses possible denotational semantics for the generative and limited generative models. Finally, in Section 7 we present our conclusions and directions of future work.

2 A PROBABILISTIC EXTENSION OF LOTOS

We will work within a probabilistic version of a subset of LOTOS [LOT88]. In contrast with previous approaches, we introduce a probability in the LOTOS (binary) choice operator, and thus this operator remains deterministic, in the sense that a *choice* between both sides of the choice operator is influenced by the environment (this is shown in Example 1). We also introduce a probability in the parallel operator, which indicates the *weight* of both sides of the parallel operator when an *interleaving* event is offered by the environment.

2.1 Syntax of PL (Probabilistic LOTOS)

As usual when giving the syntax of LOTOS, we will consider a universe of gates $\mathcal{G}$ that includes all the gate names; $g, g', g_1, \ldots$ range over $\mathcal{G}$. $G, G', G_1, \ldots$ range over sets of gate names. We will also consider an internal event i which is not visible for an observer. Then the set $\mathcal{E} = \mathcal{G} \cup i$ will be the universe of events; $e, e', e_1, \ldots$ range over $\mathcal{E}$. Finally, we consider a finite set Id of process variables.

Definition 1 The set of *PL behavior expressions* is defined by the BNF-expression:

$$B ::= \text{stop} \mid X \mid e \,;\, B \mid B[\,]_p B \mid B \mid[G]\mid_p B \mid \text{hide } G \text{ in } B \mid X := B$$

where $p \in (0,1)$, $e \in \mathcal{E}$, $G \subseteq \mathcal{G}$ and $X \in Id$.

From now on, we usually omit the term probabilistic when referring to probabilistic behavior expressions (or probabilistic tests).

Example 1 Let $B = g \,;\, B_1[\,]_p g' \,;\, B_2$. Intuitively speaking, if the environment only offers g (resp. g'), then B performs g (resp. g') with a probability equal to 1. If the environment offers both g and g' (with the same probability), then B performs g with a probability equal to p and performs g' with a probability equal to $1-p$.

2.2 Operational Semantics

In order to define a testing semantics for *PL*, first we will define the adequate operational semantics. The meaning of a probabilistic transition $B \xrightarrow{e}_p B'$ is that if all the gate

$$(PRE)\ \frac{}{e;B\xrightarrow{e}_1 B}$$

$$(CHO1)\frac{B_1\xrightarrow{e}_q B_1'}{B_1[\,]_p B_2\xrightarrow[\frac{p\cdot q}{r(B_1,B_2,\emptyset,p)}]{e} B_1'} \qquad (CHO2)\frac{B_2\xrightarrow{e}_q B_2'}{B_1[\,]_p B_2\xrightarrow[\frac{(1-p)\cdot q}{r(B_1,B_2,\emptyset,p)}]{e} B_2'}$$

$$(PAR1)\frac{B_1\xrightarrow{e}_q B_1' \wedge e\notin G}{B_1|[G]|_p B_2\xrightarrow[\frac{p\cdot q}{r(B_1,B_2,G,p)}]{e} B_1'|[G]|_p B_2} \qquad (PAR2)\frac{B_2\xrightarrow{e}_q B_2' \wedge e\notin G}{B_1|[G]|_p B_2\xrightarrow[\frac{(1-p)\cdot q}{r(B_1,B_2,G,p)}]{e} B_1|[G]|_p B_2'}$$

$$(PAR3)\frac{B_1\xrightarrow{g}_s B_1' \wedge B_2\xrightarrow{g}_t B_2' \wedge g\in G}{B_1|[G]|_p B_2\xrightarrow[\frac{s\cdot t}{r(B_1,B_2,G,p)}]{g} B_1'|[G]|_p B_2'}$$

$$(HID1)\frac{B\xrightarrow{e}_q B' \wedge e\notin G}{\text{hide } G \text{ in } B\xrightarrow{e}_q \text{hide } G \text{ in } B'} \qquad (HID2)\frac{B\xrightarrow{g}_q B' \wedge g\in G}{\text{hide } G \text{ in } B\xrightarrow{i}_q \text{hide } G \text{ in } B'}$$

$$(REC)\frac{B\xrightarrow{e}_q B' \wedge X{:=}B}{X\xrightarrow{e}_q B'}$$

Figure 1 Operational Semantics of *PL*

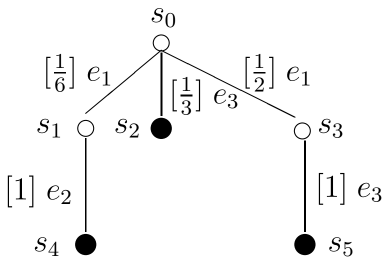
names are available from the environment, the probability with which B performs e and then behaves as B' is equal to p. The rules which define the operational semantics are given in Figure 1.

We have preferred to avoid the use of indices that are normally used to distinguish between the different derivations of the same probability transition. But we will take into account the possibility of having the same transition more than one time by using multisets of transitions instead of set of transitions.

Example 2 Let $B = g\,;\text{stop}[\,]_{\frac{1}{2}}g\,;\text{stop}$. Then we will have the transition $B \xrightarrow{g}_{\frac{1}{2}} \text{stop}$ two times.

In the rules for the choice and parallel operators we use a *normalization factor*. Intuitively speaking, $r(B_1, B_2, G, p)$ calculates the sum of the probabilities associated with the events which do not belong to G that B_1 (resp. B_2) may perform, multiplied by p (resp. $1-p$). This factor also considers the probability with which B_1 and B_2 (simultaneously) may perform events in G. That is, the normalization factor calculates the *whole* probability with which both behavior expressions may perform events, considering the possible restrictions imposed by the synchronization set, and taking also into account the probability p. Formally, this factor is defined by

$$r(B_1,B_2,G,p) = p\cdot\sum_{B_1'}\{\!|\, q \mid \exists e\notin G : B_1 \xrightarrow{e}_q B_1' \,|\!\} + (1-p)\cdot\sum_{B_2'}\{\!|\, q \mid \exists e\notin G : B_2 \xrightarrow{e}_q B_2' \,|\!\}$$
$$+\sum_{B_1',B_2'}\{\!|\, s\cdot t \mid \exists g\in G : B_1 \xrightarrow{g}_s B_1' \ \wedge\ B_2 \xrightarrow{g}_t B_2' \,|\!\}$$



$S = \{s_0, \ldots, s_5\} \qquad \mu(s_0, e_1, s_1) = \frac{1}{6}$
$\mu(s_0, e_3, s_2) = \frac{1}{3} \qquad \mu(s_0, e_1, s_3) = \frac{1}{2}$
$\mu(s_0, e, t) = 0$ if $(e, t) \notin \{(e_1, s_1), (e_3, s_2), (e_1, s_3)\}$
$\mu(s_1, e_2, s_4) = 1$
$\mu(s_1, e, t) = 0$ if $(e, t) \neq (e_2, s_4)$
$\mu(s_4, e, t) = 0$ for all $(e, t) \in \mathcal{E} \times S$
$\ldots$

Figure 2 *plts* of $(e_1\ ;\ e_2\ ;\ \text{stop}[\,]_{\frac{1}{3}} e_3\ ;\ \text{stop})[\,]_{\frac{1}{2}} e_1\ ;\ e_3\ ;\ \text{stop}$

While the rules for the *prefix, hiding* and *recursion* operators do not need any explanation, we briefly explain the rules for the choice and parallel operators. In the rules $(CHO1)$ and $(CHO2)$, the normalization factor has as parameter the empty set of gate names because all the events could be performed by both behavior expressions. Then, if any of the behavior expressions may perform no event, the whole probability *goes* to the other one; in this case the normalization factor evaluates either to p or to $1-p$. Otherwise, the probabilities associated with the events performed by the left hand side (resp. the right one) are multiplied by p (resp. $1-p$); in this case the normalization factor evaluates to 1. Something similar happens to $(PAR1)$ and $(PAR2)$, but considering as parameter of the normalization factor the synchronization set. $(PAR3)$ considers the case when a synchronization event may be performed by both sides of the parallel operator.

As usual, from the defined operational semantics we induce the corresponding notion of labeled transition system.

Definition 2 A *probabilistic labeled transition system* (*plts*) over $\mathcal{E}$ is a 3-tuple (S, μ, s_0) where S is a set of states, $\mu : S \times \mathcal{E} \times S \longrightarrow [0, 1]$ is the *probability distribution function* and $s_0 \in S$ is the *initial state*. The function μ fulfils that for all $s \in S$, $\sum_{e,s'} \mu(s, e, s')$ is equal to 1 or equal to zero (blocked state).

Given a syntactic behavior expression B we can build its associated *plts* as usual: the initial state is labeled with B, and if $B \xrightarrow{e}_p B'$ then $\mu(B, e, B') = p$. Then, the same method is applied to every B' (see Figure 2 for an example). Thus, at the semantic level, a behavior expression may be identified with the *plts* defining its operational semantics.

2.3 Testing Semantics

A probabilistic test is just a behavior expression (*plts*) with a set of successful states. As in [CSZ92, YCDS94] we slightly differ from the standard formulation of tests given in [Hen88], because successful states, instead of a distinguished acceptance event, are used to denote the passing of tests. Anyway, these successful states can be simulated adding a new event *ok* to the set of events and obliging the tests to be behavior expressions such that this new event must always appear guarded by some other event or to be a successful test (i.e. $T := ok;\text{stop}$). In fact, in the rest of this paper we will mainly use this alternative characterization, and for short, we will just write *ok* to denote the test *ok* ; stop.

Definition 3 A *probabilistic test* (*pt*) over $\mathcal{E}$ is a 4-tuple (S, μ, s_0, Suc) where (S, μ, s_0) is a *plts* and $Suc \subseteq S$ is the set of *successful states*. In the graphical representation, successful states will be represented by two concentric circles.

We have to define the way a behavior expression and a test interact. Even if a testing semantics was defined for LOTOS in [BSS86], we have preferred to use some other definitions and notations, closer to those by Hennessy [Hen88]. Testing is performed by studying the computations of the behavior expression that is obtained by putting a test in parallel with the behavior expression to be tested, taking as associated probability $\frac{1}{2}$ (the choice of this probability has been done to keep the symmetry between the internal events of the behavior expression and those of the test) and as synchronization set the full set of gate names (i.e. the acceptance event *ok* is not included). Test application will be represented by $B \mid T$, and is (formally) defined as follows:

$$B \mid T = \text{hide } \mathcal{G} \text{ in } (B \,|[\mathcal{G}]|_{\frac{1}{2}}\, T)$$

Now, we define a function which computes the probability with which a behavior expression *passes* a test.

Definition 4 Let B be a behavior expression and T a test. As usual, we will call *computations* to the (possibly infinite) sequences of transitions from $S = B \mid T$ of the form $C = S \xrightarrow{e_1}_{p_1} S_1 \xrightarrow{e_2}_{p_2} S_2 \ldots$ Note that for all j, $e_j = i$ or $e_j = ok$.

We distinguish *infinite computations* and *blocked computations,* which are those finite computations $C = S \xrightarrow{e_1}_{p_1} S_1 \xrightarrow{e_2}_{p_2} S_2 \ldots S_{n-1} \xrightarrow{e_n}_{p_n} S_n$ that cannot be extended, that is $S_n \xrightarrow{e}_{p} S'$ for no e, p, S'. In this case, we will say that C has *length* n.

Given a blocked computation $C = S \xrightarrow{e_1}_{p_1} S_1 \xrightarrow{e_2}_{p_2} S_2 \ldots S_{n-1} \xrightarrow{e_n}_{p_n} S_n$, we inductively define its probability $Pr(C)$ by

$$Pr(C) = \begin{cases} p & \text{if } \mathit{length}(C) = 1 \wedge C = S \xrightarrow{e}_{p} S' \\ p \cdot Pr(C') & \text{if } \mathit{length}(C) > 1 \wedge C = S \xrightarrow{e}_{p} C' \end{cases}$$

We call *successful computations* to those blocked computations of S such that there exists some $j \in \{1, \ldots, n\}$ such that $e_j = ok$. We will denote by $\widetilde{S}$ the *set of successful computations of* S. We say that a computation is *unsuccessful* whenever it is infinite, or blocked but not successful.

Definition 5 Let B be a behavior expression and T a test, and let us consider $S = B|T$. We say that $B\, \mathit{pass}_p\, T$ if $\sum_{C \in \widetilde{S}} Pr(C) = p$.

The probability of blocked unsuccessful computations can be defined in a similar way, taking $B\, \mathit{blocked}_p\, T$ if $\sum_{C \in \widehat{S}} Pr(C) = p$ where $\widehat{S}$ denotes that set of computations. Finally, we can define the probability of infinite computations of S as $1 - (\sum_{C \in \widetilde{S}} Pr(C) + \sum_{C \in \widehat{S}} Pr(C))$. Thus, we have that the probability of unsuccessful computations of S is equal to $1 - \sum_{C \in \widetilde{S}} Pr(C)$.

It is important to note that the fact that successful computations cannot be extended is essential when considering divergent behavior expressions. For instance, we have

$X := (i \,;\, X)[\,]_p(a \,;\, \mathrm{stop})\ pass_0\ ok$, because $X \mid ok \xrightarrow{ok}_{1/2} X \mid \mathrm{stop}$ is not a successful computation, since it can be infinitely extended by $X \mid \mathrm{stop} \xrightarrow{i}_1 X \mid \mathrm{stop} \ldots$

Now, given a family of tests we can define the corresponding notion of testing preorder between probabilistic behavior expressions with respect to the family.

Definition 6 Given a set of probabilistic tests $\mathcal{T}$, and two behavior expressions B_1 and B_2, we write $B_1 \sqsubseteq_{\mathcal{T}} B_2$ if $\forall\, T \in \mathcal{T} :\ B_1\, pass_p\, T \wedge\ B_2\, pass_q\, T \Longrightarrow p \leq q$. Besides we write $B_1 \approx_{\mathcal{T}} B_2$ iff $B_1 \sqsubseteq_{\mathcal{T}} B_2$ and $B_2 \sqsubseteq_{\mathcal{T}} B_1$.

3 DIFFERENT TESTING SEMANTICS

In this section we define testing semantics for the reactive and generative models (as proposed in [vGSST90]) and for the limited generative model (a variant of the generative model).

3.1 Reactive Model

Let us remember that in the reactive model, the environment (i.e. the tests) only can offer one gate name each time. Using [Mil80] terminology, *only one button can be pressed at the same time*. This gives rise to tests being just traces (see Figure 3 for the general schema).

Definition 7 (Reactive Tests) The set of *reactive tests* (denoted by $\mathcal{R}$) is defined by the BNF expression $T =\ ok \mid g \,;\, T$.

We write $B_1 \sqsubseteq_{\mathcal{R}} B_2$ iff $\forall\, T \in \mathcal{R} :\ B_1\, pass_p\, T \wedge\ B_2\, pass_q\, T \Longrightarrow p \leq q$.

A fully abstract characterization can be given in terms of probabilistic traces. These are defined by extending ordinary traces of behavior expressions with a number which indicates the probability with which the behavior expression can perform the trace (if this trace is offered by the environment).

Definition 8 Given a behavior expression B, we define its *set of probabilistic traces* as:

$$[\![B]\!] = \{(s,p) \mid s \in \mathcal{G}^* \wedge B\, pass_p\, s \circ ok\}$$

where if $s = \langle g_1, g_2, \ldots, g_n\rangle$ then $s \circ ok$ is the reactive test $g_1 \,;\, g_2 \,;\, \ldots g_n \,;\, ok$.

Even if we have used the notion of passing a test to calculate the probability of each trace, it is clear that this value could be directly calculated from the syntactic definition of the behavior expression, although this would unnecessarily complicate the corresponding definition.

Definition 9 Let $X, Y \subseteq (\mathcal{G}^* \times [0,1])$. We write $X \tilde{\subseteq} Y$ if for all $(s,p) \in X$ there exists $q \geq p$ such that $(s,q) \in Y$. We write $B_1 \ll_{\mathcal{R}} B_2$ whenever $[\![B_1]\!] \tilde{\subseteq} [\![B_2]\!]$.

Note that within this reactive model we can distinguish a blocked behavior expression (e.g. stop) from a divergent behavior expression (e.g. $X := i ; X$), because the semantics of the first one includes $(\epsilon, 1)$ while in the latter the empty trace has null probability. This is so, because a divergent behavior has no successful computation, and thus it cannot pass the test *ok* (more exactly, this test is passed with a probability equal to 0). In fact, we have $X := i ; X \ll_{\mathcal{R}}$ stop. Also note that, in general, $(i ; B)[\,]_p(i ; B')$ and $B[\,]_p B'$ are not equivalent.

Theorem 1 (Full Abstraction) $B_1 \sqsubseteq_{\mathcal{R}} B_2$ iff $B_1 \ll_{\mathcal{R}} B_2$.

Proof. The proof is easy from the definition of $[\![B]\!]$ due to the close relation between probabilistic traces and reactive tests. $\square$

While the reactive model is very easy to work with, it presents a serious problem. There is no way of relating different gate names, because the meaning of probabilities in the choice operator is partially lost, as shown by the forthcoming Proposition 1.

Definition 10 Given a behavior expression B, we define the set of its *initial* events as $ini(B) = \{e \in \mathcal{E} \mid \exists p, B' : B \xrightarrow{e}_p B'\}$.

Proposition 1 Let B_1, B_2 be behavior expressions such that $B_1 \xrightarrow{i}_p B_1' \wedge B_2 \xrightarrow{i}_q B_2'$ for no p, q, B_1', B_2' and such that $ini(B_1) \cap ini(B_2) = \emptyset$. Then, for any $0 < p, q < 1$ we have: $B_1[\,]_p B_2 \approx_{\mathcal{R}} B_1[\,]_q B_2$

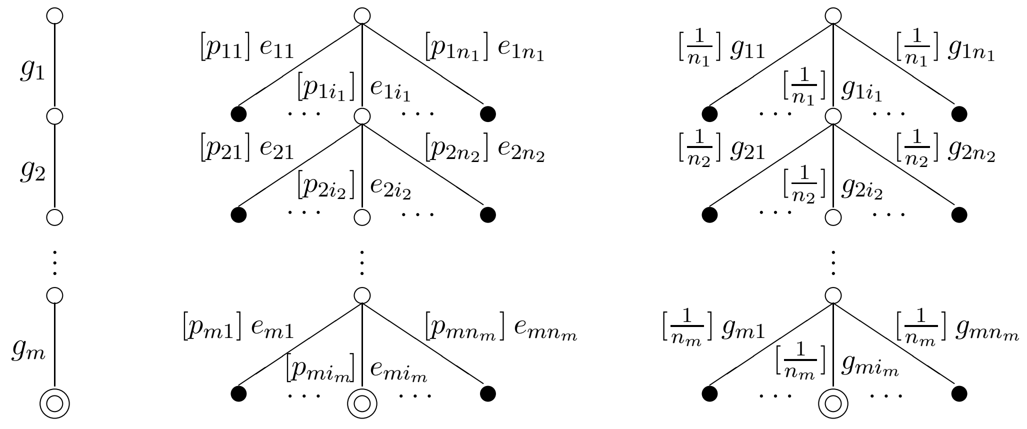
For instance, if $g \neq g'$, we have $(g ; B)[\,]_{\frac{1}{2}}(g' ; B') \approx_{\mathcal{R}} (g ; B)[\,]_{\frac{1}{3}}(g' ; B')$. This *loss* of probabilistic information leads us to the study of the generative model.

3.2 Generative Model

Let us remember that in the generative model, the environment (i.e. the tests) may offer more than one event at a time, and with different probabilities. This means that *more than a button can be simultaneously pressed, and buttons can be pressed with different strengths.* Then, the adequate family of tests to capture this semantics, which we will denote by $\mathcal{GE}$, is the full set of tests given by Definition 3, and thus they (almost) correspond to arbitrary behavior expressions.

Definition 11 $B \sqsubseteq_{\mathcal{GE}} B'$ iff $\forall\, T \in \mathcal{GE} : \; B\, pass_p\, T \wedge \; B'\, pass_q\, T \Longrightarrow p \leq q$.

Let us note that in contrast with the generative model described in [YCDS94], this relation is not an equivalence relation because we allow divergent behavior expressions, and so, for instance, $X := i ; X \sqsubseteq_{\mathcal{GE}}$ stop, while stop $\not\sqsubseteq_{\mathcal{GE}} X := i ; X$. Nevertheless, it is true that if we restrict ourselves to divergence free behavior expressions, we have a similar result to that in [YCDS94]:


g_1
g_2
g_m
$[p_{11}]\ e_{11}$
$[p_{1i_1}]\ e_{1i_1}$
$[p_{1n_1}]\ e_{1n_1}$
$[p_{21}]\ e_{21}$
$[p_{2i_2}]\ e_{2i_2}$
$[p_{2n_2}]\ e_{2n_2}$
$[p_{m1}]\ e_{m1}$
$[p_{mi_m}]\ e_{mi_m}$
$[p_{mn_m}]\ e_{mn_m}$
$[\frac{1}{n_1}]\ g_{11}$
$[\frac{1}{n_1}]\ g_{1i_1}$
$[\frac{1}{n_1}]\ g_{1n_1}$
$[\frac{1}{n_2}]\ g_{21}$
$[\frac{1}{n_2}]\ g_{2i_2}$
$[\frac{1}{n_2}]\ g_{2n_2}$
$[\frac{1}{n_m}]\ g_{m1}$
$[\frac{1}{n_m}]\ g_{mi_m}$
$[\frac{1}{n_m}]\ g_{mn_m}$

where $\forall\, j\ (1 \leq j \leq m) : \sum_{k=1}^{n_j} p_{jk} = 1$ and $\forall\, j\ (1 \leq j \leq m) : i \neq k \Rightarrow g_{ji} \neq g_{jk}$.

Figure 3 Reactive tests, probabilistic barbs and limited probabilistic barbs.

Theorem 2 Let B, B' be divergence free behavior expressions (i.e. such that they can never perform an infinite sequence of i's), then we have

$$B \sqsubseteq_{\mathcal{GE}} B' \Longleftrightarrow B' \sqsubseteq_{\mathcal{GE}} B \ (\Longleftrightarrow B \approx_{\mathcal{GE}} B')$$

An alternative characterization of this semantics can be given in terms of probabilistic *barbs* (called probabilistic traces in [YCDS94]), which are in fact a set of *essential* tests. To define them, we have first to extend the binary choice operator to an n-ary one.

Definition 12 Let $B_1, B_2, \ldots, B_n$ be behavior expressions and $0 < p_1, p_2, \ldots, p_n < 1$ such that $\sum_{i=1}^{n} p_i = 1$. We inductively define the behavior expression $\sum_{i=1}^{n} [p_i] B_i \quad (n \geq 2)$ as:

- $\sum_{i=1}^{2} [p_i] B_i = B_1 [\,]_{p_1} B_2$
- $\sum_{i=1}^{n} [p_i] B_i = B_1 [\,]_{p_1} \sum_{i=2}^{n} [\frac{p_i}{1-p_1}] B_i \qquad (n > 2)$

Definition 13 (Probabilistic Barbs) The set of *probabilistic barbs* (denoted by $\mathcal{PB}$) is defined by the BNF expression $T = \ ok \mid \sum_{i=1}^{n} [p_i]\ e_i\ ; R_i$, where $\sum_{i=1}^{n} p_i = 1$, $R_i =$ stop $(1 \leq i \leq n-1)$, and $R_n = T$.

That is, a probabilistic barb is either equal to the test *ok* or it is a test such that all the outgoing transitions go to stop, but a single one whose continuation is another probabilistic barb. The general form of probabilistic barbs is presented in Figure 3.

Theorem 3 (Alternative Characterization) $B \sqsubseteq_{\mathcal{GE}} B'$ iff $B \sqsubseteq_{\mathcal{PB}} B'$.

Proof. The proof is an adaptation to our syntax of that given in [YCDS94], although some additional care is needed, because divergent behavior expressions are now allowed. □

In the generative model *all* the probabilistic information appearing in behavior expressions is captured. In fact, that is done in a *too strict* way, and thus we distinguish behavior expressions which intuitively should be equivalent. As we will see, that is because the full family of tests has enough power to discriminate the internal structure of probabilistic nondeterminism. This is illustrated by the following

Example 3 Let B and B' be defined as follows:

$$B = i\,;(g\,;\mathrm{stop}[\,]_{\frac{1}{4}}g'\,;\mathrm{stop})[\,]_{\frac{1}{2}}i\,;(g\,;\mathrm{stop}[\,]_{\frac{3}{4}}g'\,;\mathrm{stop})$$
$$B' = g\,;\mathrm{stop}[\,]_{\frac{1}{2}}g'\,;\mathrm{stop}$$

We would expect B and B' to be equivalent, because applying a *reasonable* pseudo-distributive law to B we obtain B'. This is better understood using a *CSP-like* notation. In such a case we would have $B = (g\Box_{\frac{1}{4}}g') \sqcap_{\frac{1}{2}} (g\Box_{\frac{3}{4}}g')$ and $B' = g\Box_{\frac{1}{2}}g'$, where stop's have been omitted. But these two processes can be distinguished by any of the tests $T = g\,;ok[\,]_p g'\,;\mathrm{stop}$, for any $p \in (0,1)$ with $p \neq \frac{1}{2}$.

The fact that for $p = \frac{1}{2}$ both behavior expressions are not distinguished suggested us to study the case in which tests are restricted to be *equitable*. Thus we obtain the so called *limited generative model* which we present in the next section.

3.3 Limited Generative Model

In this model, we force tests to be deterministic and such that all the outgoing transitions from any state are labeled with equal probabilities. This means that *several buttons can be simultaneously pressed, but all of them with the same strength.*

Definition 14 We define the set of *limited generative tests* (denoted by $\mathcal{LG}$) as the least set of tests $T_{\mathcal{LG}} = (S, \mu, s_0, Suc)$ satisfying the following conditions:

- For any $s \in S$, and any $g \in \mathcal{G}$, there do not exist s', s'' ($s' \neq s''$) such that $\mu(s,g,s') > 0$ and $\mu(s,g,s'') > 0$.
- For any $s \in S$, there does not exist s' such that $\mu(s,i,s') > 0$.
- For any $s \in S$, let $n_s = |\{\!\!\{\ g\ |\exists r' : \mu(s,g,r') > 0\ \}\!\!\}|$ (i.e. n_s is the number of outgoing transitions from s). Then, $\forall s' \in S, g \in \mathcal{G} : \mu(s,g,s') > 0 \Rightarrow \mu(s,g,s') = \frac{1}{n_s}$.

In the previous definition, with the first condition we express that for any state $s \in S$ and any gate name g there exists at most one state s' such that $\mu(s,g,s') > 0$. The second one says that there are no transitions labeled with i leaving any state . The last condition says that for any state, all the outgoing transitions are labeled with the same probability.

As in the previous case, an alternative characterization of this semantics can be given in terms of *limited probabilistic barbs*, which are those probabilistic barbs with no transition labeled with i, and such that all the outgoing transitions from the same state are labeled by different gate names, but have the same probability. See Figure 3 for their general schema.

Definition 15 (Limited Probabilistic Barbs) The set of *limited probabilistic barbs* (denoted by $\mathcal{LPB}$) is defined by the BNF expression $T = \ ok \mid \sum_{i=1}^{n}[\frac{1}{n}]\ g_i\ ;\ R_i$, where $i \neq j \Rightarrow g_i \neq g_j$, $R_i = \text{stop}\ (1 \leq i \leq n-1)$, and $R_n = T$.

Theorem 4 (Alternative Characterization) $B \sqsubseteq_{\mathcal{LG}} B'$ iff $B \sqsubseteq_{\mathcal{LPB}} B'$.

Proof. It is similar to that of Theorem 3. $\square$

Under this model, we have that the processes considered in Example 3 are now equivalent. That is, $i\ ;\ (g[\,]_{\frac{1}{4}}g')[\,]_{\frac{1}{2}}i\ ;\ (g[\,]_{\frac{3}{4}}g') \approx_{\mathcal{LG}} g[\,]_{\frac{1}{2}}g'$.

4 RECOVERING NONPROBABILISTIC FROM PROBABILISTIC TESTING

In this section we relate the *may* and *must* testing preorders for nonprobabilistic [*] LOTOS and our own probabilistic models. The classical notions of nonprobabilistic test, composition of nonprobabilistic behavior expressions and nonprobabilistic tests, and successful computation can be recovered from those in Section 2 by *forgetting* the probabilistic information on them (formal definitions of these concepts can be found in [dFNQ95] where a testing semantics for LOTOS is presented). Next we remind the definitions of the *may* and *must* testing preorders.

Definition 16 We say that a behavior expression B *may satisfy* a test T, and we will write B may T, iff $B \mid T$ has some successful computation. We say that B *must satisfy* a test T, and we will write B must T, iff any complete (i.e. *blocked* or infinite) computation of $B \mid T$ is a successful computation.

Now, we translate the previous definition to the probabilistic case, just considering that a behavior expression *may pass* a probabilistic test if it is passed with positive probability, and that it *must pass* the test if it is passed with a probability equal to 1.

Definition 17 Let B_p be a probabilistic behavior expressions and T_p a probabilistic test, and suppose that $B_p\, pass_r\, T_p$. Then we write B_p may T_p if $r > 0$, and B_p must T_p if $r = 1$.

Lemma 1 B_p may T_p iff $B_p \mid T_p$ has a successful computation. B_p must T_p iff the probability of unsuccessful computations of $B_p \mid T_p$ is equal to zero.

[*]In the rest of this section we will distinguish between probabilistic and nonprobabilistic behavior expressions (tests), denoting them by $B_p, B'_p, \ldots$ $(T_p, T'_p, \ldots)$ and $B, B', \ldots$ $(T, T', \ldots)$ respectively.

The following definition covers both the probabilistic and the nonprobabilistic cases, just by taking in each case the corresponding family of tests.

Definition 18 Let B and B' be behavior expressions (probabilistic or not). We define:

- $B \sqsubseteq_{\text{may}} B'$ if $\forall T : B \text{ may } T \Rightarrow B' \text{ may } T$
- $B \sqsubseteq_{\text{must}} B'$ if $\forall T : B \text{ must } T \Rightarrow B' \text{ must } T$

The following Lemma states that in order to characterize the probabilistic *may* preorder, it is enough to consider *reactive* tests (i.e. trace tests). This is similar to the nonprobabilistic case where traces are enough to characterize the *may* preorder [Hen88].

Lemma 2 $B_p \sqsubseteq_{\text{may}} B'_p$ iff $\forall T \in \mathcal{R} : (B_p \, pass_r \, T \wedge r > 0) \Rightarrow (B_p \, pass_s \, T \wedge s > 0)$.

Definition 19 Let B be a behavior expression. We say that a probabilistic behavior expression B_p is a *probabilistic extension* of B if B_p can be obtained from B by assigning probabilities to the choice and parallel operators that appear in B.

Example 4 Let $B = (g_1 ; \text{stop}[\,]g_2 ; \text{stop}) \,|[G]|\, (g_3 ; \text{stop})$. Then the following probabilistic behavior expressions are two of the probabilistic extensions of B:

$$(g_1 ; \text{stop}[\,]_{\frac{1}{3}} g_2 ; \text{stop}) \,|[G]|_{\frac{3}{5}}\, (g_3 ; \text{stop})$$
$$(g_1 ; \text{stop}[\,]_{\frac{1}{4}} g_2 ; \text{stop}) \,|[G]|_{\frac{1}{2}}\, (g_3 ; \text{stop})$$

Theorem 5 Let B, B' be two divergence free and finite state behavior expressions. Then for any probabilistic extensions B_p of B and B'_p of B', we have:

- $B \sqsubseteq_{\text{may}} B' \Longleftrightarrow B_p \sqsubseteq_{\text{may}} B'_p$
- $B \sqsubseteq_{\text{must}} B' \Longleftrightarrow B_p \sqsubseteq_{\text{must}} B'_p$

Proof. (Sketch) The *may* case is immediate due to the correspondence between the successful computations of $B \mid T$ and $B_p \mid T_p$. In this case, the hypothesis of divergence free and finite state are not necessary. Nevertheless, in the *must* case we must be more careful, since a probabilistic version of an infinite computation of $B \mid T$ may or may not have a positive probability. In fact, for behavior expressions not verifying the hypothesis, the result is false in general, as proves the following example. □

Example 5 Let $B := (i;B)[\,](a;\text{stop})$ and $B' := (i;B')[\,](b;\text{stop})$. Clearly, $B \sqsubseteq_{\text{must}} B'$ (in fact they are (must nonprobabilistic) equivalent, because they *must* pass no test) while for their probabilistic extensions $B_p := (i ; B_p)[\,]_{\frac{1}{2}}(a ; \text{stop})$ and $B'_p := (i ; B'_p)[\,]_{\frac{1}{2}}(b ; \text{stop})$ we have $B_p \not\sqsubseteq_{\text{must}} B'_p$ (for example, B_p must $a ; ok$). On the other hand, $a ; \text{stop} \sqsubseteq_{\text{must}} B_p$ (in fact they are also (must probabilistic) equivalent), while $a ; \text{stop} \not\sqsubseteq_{\text{must}} B$.

5 ALGEBRAIC LAWS

In this section we present algebraic laws for each of the semantic equivalences $\approx_i$ with $i \in \{\mathcal{R},\ \mathcal{GE},\ \mathcal{LG}\}$. We shall write $B \approx B'$ meaning that $B \approx_i B'\ \forall i \in \{\mathcal{R},\ \mathcal{GE},\ \mathcal{LG}\}$. First we give a theorem which relates the three models.

Theorem 6 $B \sqsubseteq_{\mathcal{GE}} B' \Longrightarrow B \sqsubseteq_{\mathcal{LG}} B' \Longrightarrow B \sqsubseteq_{\mathcal{R}} B'$

In Section 3, we have presented some examples proving that in general the converse implications are false.

Next we present the commutative and associative laws for the operators of our language.

Proposition 2 The following laws hold:
Commutative Laws

$$\begin{array}{lcl} B[\,]_p B' & \approx & B'[\,]_{1-p} B \\ B\,|[G]|_p\, B' & \approx & B'\,|[G]|_{1-p}\, B \\ \text{hide } G' \text{ in } (\text{hide } G \text{ in } B) & \approx & \text{hide } G \text{ in } (\text{hide } G' \text{ in } B) \end{array}$$

Associative Laws

$$\begin{array}{lcl} B[\,]_p(B'[\,]_q B'') & \approx & (B[\,]_{p'} B')[\,]_{q'} B'' \\ B\,|[G]|_p\,(B'\,|[G]|_q\, B'') & \approx & (B\,|[G]|_{p'}\, B')\,|[G]|_{q'}\, B'' \end{array}$$

where $p' = \frac{p}{p+q\cdot(1-p)}$ and $q' = p + q \cdot (1-p)$

Although in these models associative laws hold, probabilistic models with nonassociative operators could be also considered (see [NPM94]). Now we will present the distributive laws.

Proposition 3 The following laws hold:

$$\begin{array}{lcl} e\,;(i\,;B[\,]_p i\,;B') & \approx & (e\,;B)[\,]_p(e\,;B') \\ (B[\,]_p B')\,|[G]|_q\, B'' & \approx & (B\,|[G]|_q\, B'')[\,]_p(B'\,|[G]|_q\, B'') \end{array}$$

Obviously, the possible distributive law $g\,;(B[\,]_p B') \approx_i (g\,;B)[\,]_p(g\,;B')$ is false for any of the models. The following laws claim that stop is the neutral element for the choice operator and the parallel operator when the synchronization set is empty.

Proposition 4 The following laws hold:

$$\begin{array}{lcl} B[\,]_p\text{stop} & \approx & B \\ B\,|[\,]|_p\,\text{stop} & \approx & B \end{array}$$

Now we will present an useful law for the parallel operator. It says that whenever two processes that cannot perform i's along their evolutions are composed in parallel with total synchronization, the probability in the parallel operator may be *ignored.*

Proposition 5 Let B_1, B_2 be two behavior expressions which cannot perform i's along their evolutions [†]. Then the following law holds:

$$B_1 \,||[\mathcal{G}]|_p\, B_2 \approx B_1 \,||[\mathcal{G}]|_q\, B_2, \quad \text{for any } p, q \in (0,1)$$

When defining the interaction between a process and a test, we take $\frac{1}{2}$ as the probability in the parallel operator connecting them. In fact, for a language without i's, like CSP, in which (internal) choices are always related to an explicit *choice operator*, it would not matter the value used in the composition.

The following law says that in the reactive and limited generative models, choices prefixed by i's (i.e. internal choices) are *worse* than the corresponding choices without the prefix of these i's.

Proposition 6 The following law holds:

$$(i\,;B)[\,]_p(i\,;B') \sqsubseteq_i B[\,]_p B', \quad \text{where } i \in \{\mathcal{R},\ \mathcal{LG}\}$$

Somewhat surprisingly, this law is not true for the generative model. Let us consider $B = i\,; g[\,]_{\frac{1}{2}} i\,; g'$, $B' = g[\,]_{\frac{1}{2}} g'$, where stop*'s* have been omitted. For $T = g\,; ok[\,]_{\frac{1}{3}} g'\,;\text{stop}$ we have $B\,\mathit{pass}_{\frac{1}{2}}\,T$ while $B'\,\mathit{pass}_{\frac{1}{3}}\,T$. Intuitively speaking, this law is true for the limited generative model because tests cannot be sufficiently *biased* in favour of *bad* terminations, as we have done with T for the generative case.

6 DENOTATIONAL MODELS

Although we have presented some alternative characterizations for the generative and limited generative models, these characterizations are not sufficiently abstract to generate from them a fully abstract denotational semantics in an easy way. To define such a semantics for the generative case, we have considered in [NdFL95] a class of trees similar to acceptance trees in [Hen88], but where probabilities are introduced in an adequate way. These trees represent internal probabilistic choices among a family of states labeled with a set of actions, from which leaves a probabilistic external choice among the actions in each state.

It is important to note that we can have several states labeled with the same set of actions under the same node of the tree, and we cannot join these states if they are labeled with different probability distributions. This fact corresponds to the situation illustrated by Example 3. In our generative model, using a *CSP-like* notation, $(a\Box_{\frac{1}{4}}b) \sqcap_{\frac{1}{2}} (a\Box_{\frac{3}{4}}b)$ have two states $\{a_{\frac{1}{4}}, b_{\frac{3}{4}}\}$ and $\{a_{\frac{3}{4}}, b_{\frac{1}{4}}\}$, reachable with probability equal to $\frac{1}{2}$. Instead, in the denotational semantics for the limited generative model, these two states should be joined in a single state $\{a_{\frac{1}{2}}, b_{\frac{1}{2}}\}$ which is reachable with probability equal to 1. This is so because under the limited generative model $(a\Box_{\frac{1}{4}}b) \sqcap_{\frac{1}{2}} (a\Box_{\frac{3}{4}}b)$ is equivalent to $a\Box_{\frac{1}{2}}b$.

[†]Formally, B cannot perform i's along its evolution if there do not exist $g_1, g_2, \ldots, g_n,\ p_1, p_2, \ldots, p_n, p$ and $B_1, B_2, \ldots, B_n, B'$ such that $B \xrightarrow{g_1}_{p_1} B_1 \xrightarrow{g_2}_{p_2} B_2 \ldots \xrightarrow{g_n}_{p_n} B_n \xrightarrow{i}_{p} B'$.

7 CONCLUSIONS AND FUTURE WORK

We have presented testing characterizations of the reactive and generative models described in [vGSST90], and we have introduced another probabilistic model (the limited generative model) where tests have no explicit probabilities (formally, the outgoing transitions from each state are labeled with the same probability). For the reactive model, we have given a denotational semantics, which is proved to be fully abstract with respect to the induced testing semantics. For the generative and limited generative models we have presented a family of essential tests, which give rise to an alternative characterization. We also have shown how nondeterministic (nonprobabilistic) information can be recovered from the probabilistic models. Finally, we have given several laws which respect the different testing equivalences, and we have sketched a (fully abstract) denotational semantics for the generative and limited generative models.

As future work, we seek to define precisely these denotational semantics for the studied models, and find complete axiomatizations based on the algebraic laws given in Section 5. We are also interested on the study of some other models between the reactive and limited generative models, and between the limited generative and generative models, completing a hierarchy of probabilistic models. We also would like to define the adequate extensions of HML [HM85] characterizing our different semantics, and apply them to the verification of properties of probabilistic processes. These logics should be similar to that presented in [LS92].

Acknowledgements The authors would like to thank the anonymous referees of this paper for their valuable comments.

REFERENCES

[BSS86] E. Brinksma, G. Scollo, and C. Steenbergen. LOTOS specifications, their implementations and their tests. In *PSTV VI*, pages 349–360, 1986.

[Chr90] I. Christoff. *Testing Equivalences for Probabilistic Processes.* PhD thesis, Department of Computer Systems. Uppsala University, 1990.

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

[dFLL$^+$94] D. de Frutos, G. Leduc, L. Léonard, L. Llana, C. Miguel, J. Quemada, and G. Rabay. Belgian-Spanish proposal for a time extended LOTOS. In *Working Draft on Enhancements to LOTOS (Annex E).* ISO/IEC JTC1/SC21/WG1, 1994.

[dFNQ95] D. de Frutos, M. Núñez, and J. Quemada. Characterizing termination in LOTOS via testing. *In PSTV XV*, pages 225–240, 1995.

[GJS90] A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of Working Conference on Programming Concepts and Methods, IFIP TC 2*, 1990.

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

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

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

[KLL94] J.P. Katoen, R. Langerak, and D. Latella. Modeling systems by probabilistic process algebra: An event structures approach. In *FORTE VI*, 1994.

[LOT88] LOTOS. A formal description technique based on the temporal ordering of observational behaviour. IS 8807, TC97/SC21, 1988.

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

[MFV93] C. Miguel, A. Fernández, and L. Vidaller. LOTOS extended with probabilistic behaviours. *Formal Aspects of Computing*, 5:253–281, 1993.

[Mil80] R. Milner. *A Calculus of Communicating Systems*. LNCS 92, 1980.

[NdFL95] M. Núñez, D. de Frutos, and L. Llana. Acceptance trees for probabilistic processes. *In CONCUR'95*, 1995.

[NPM94] M. Núñez, P. Palao, and M.T. Morazan. Associativity in probabilistic process algebras. Technical Report DIA-UCM 14/94, Universidad Complutense de Madrid, 1994.

[QdFA93] J. Quemada, D. de Frutos, and A. Azcorra. TIC: A TImed Calculus. *Formal Aspects of Computing*, 5:224–252, 1993.

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

[YCDS94] S. Yuen, R. Cleaveland, Z. Dayar, and S.A. Smolka. Fully abstract characterizations of testing preorders for probabilistic processes. In *CONCUR'94, LNCS 836*, pages 497–512, 1994.

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

Mr. Manuel Núñez graduated in "Mathematics (Computer Science)" in 1992 from the Universidad Complutense de Madrid (UCM). Presently he is developing his PhD Thesis which is devoted to the study of Testing Semantics for Probabilistic Process Algebras, and hopefully he expects to shortly finish it. His main research area is "Process Algebras", and more specifically the semantics of probabilistic processes.

Prof. David de Frutos graduated in "Pure Mathematics and Computer Science" in 1981 from the Universidad Complutense de Madrid (UCM) and achieved a PhD in "Mathematics (Computer Science)" in 1985 with a Thesis devoted to "Denotational Semantics of Probabilistic Constructions (Probabilistic Powerdomains)". He is presently Professor of Computer Science from 1991, and Head of the Computer Science Department of UCM. His main research topic is "Formal Models of Concurrency", and presently he is specially devoted to the study of the semantics of timed and probabilistic processes.