An Asynchronous Calculus for Generative-Reactive Probabilistic Systems*

A. ALDINI AND M. BRAVETTI
Università di Bologna, Italy

Abstract

We present an asynchronous calculus for describing and analyzing probabilistic distributed systems that combines, in a simple way, probabilistic internal/external choices and multi-way synchronization, by employing the generative-reactive approach of $\mathrm{EMPA_{gr}}$. Such an approach is based on a master-slave discipline which leads to a clear notion of control over action synchronizations. Moreover the modeling experience showed it to be a very suitable mechanism for describing real systems. With respect to the algebra $\mathrm{EMPA_{gr}}$, we show that, when restricting to a pure probabilistic setting, it is possible to simplify the operational semantics of the algebra by expressing execution probabilities via explicit parameterization of the choice and the parallel operator, instead of attaching weights with actions. This practice is particularly adequate in the case of the parallel operator, where $P \,\|^{p}\, Q$ represents a system where the relative advancing speed of process P w.r.t. process Q is determined by the probabilities p and $1-p$, respectively. Finally we show that employing probabilistic operators instead of weighted actions has a great impact on the structure of the equational characterization of strong probabilistic bisimulation. As a result we present an axiomatization for the calculus which is sound and complete for finite processes.

1 Introduction

The modeling experience in the specification of probabilistic concurrent systems (see e.g. [1, 2]) has revealed the importance of mechanisms like probabilistic internal/external choices and multi-way synchronization for expressing the behavior of real systems. In this paper we present an asynchronous calculus for describing and analyzing probabilistic distributed systems that combines, in a natural way, such mechanisms in an untimed probabilistic setting. In the last years several models for representing probabilistic systems (see e.g. [9, 15] and the references therein) and several probabilistic extensions of classical process algebras (see e.g. [3, 9, 8])

*The full version of the paper is available at ftp://ftp.cs.unibo.it/pub/techreports/2000-03.ps.gz. This research has been partially funded by MURST progetto TOSCA.

based on such models have been proposed. In particular we briefly recall two different models of probability, which have been presented in [9], by resorting to the terminology of Milner [14], where action type based synchronization is described in terms of *button pushing experiments*. In this view the environment experiments on a process by pushing one of several buttons, where a button represents an action type. According to the *reactive model* of probability, a process reacts internally to a button push performed by its environment on the basis of a probability distribution depending on the button which is pushed. According to the *generative model* of probability, the process itself autonomously decides, on the basis of a probability distribution, which button will go down and how to behave after such event. When two processes, which behave in a reactive way, synchronize on an action a, each of them reacts internally to the synchronization according to the probability distribution associated with the actions of type a it can perform. Whenever the two processes can synchronize on more than one action type, each of them leaves the decision to the environment, hence the choice of the synchronizing action type turns out to be non-deterministic. This kind of synchronization is simple and natural [15], but does not lead to a mechanism for the choice of the button to be pushed (external choice), thus leaving the system, in a sense, under-specified. On the other hand two processes which behave in a generative way, independently decide the action type over which they want to synchronize, hence there may be no agreement on the action type. The proposed solutions [9, 3] are based on the fact that synchronization on different action types is allowed. In the case the model of synchronization adopted is such that processes can synchronize only on the same action type then there is no simple and intuitive solution to the problem [15]. [1]

As an example, let us consider a system composed of two sequential processes which must synchronize over actions of type a and b. Suppose that one process may execute an action a with probability p and an action b with probability $1-p$, while the other process may execute an action a with probability q and an action b with probability $1-q$. This scenario could be represented by a term like $a +^p b \|_{\{a,b\}} a +^q b$. The problem is: How do we choose the synchronization to be performed ? The choice cannot be made independently by the two processes, because e.g. if the lefthand process locally chooses a and the righthand process locally chooses b then no synchronization is possible between the two processes. In order to have a clear model of synchronization, it is important that, once chosen the set of processes that must cooperate, each of such processes can *independently* choose one of its local actions. As pointed out in [15] the problem is due to the fact that both processes want to decide the button to be pushed. A simple and intuitive solution to the problem (as suggested also in [15]) is to adopt a mixed generative-reactive approach by considering an *asymmetric* form of synchronization where a process which behaves generatively may synchronize only with processes which behave reactively [6]. The intuition is that the process

[1] In spite of this a rather artificial solution has been proposed in [8].

which behaves generatively decides which button will go down (and how to behave afterwards) and the process which behaves reactively just reacts internally to the button push of the other process. The integration of the generative and reactive approaches is naturally obtained (similarly as done in [16] for a probabilistic version of I/O automata [13]) by designating some actions, called *generative* actions, as predominant over the other ones, called *reactive* actions, and by imposing that generative actions can synchronize with reactive actions only. Syntactically we distinguish reactive actions from generative actions, by using the subscript $*$ stating the reactive nature of the action. As an example, let us consider the system $a +^p b \,||_{\{a,b\}} (a_* +^q a_*) + (b_* +^r b_*)$. According to the generative-reactive approach, there are two steps. In the first step we probabilistically choose, locally to the lefthand process, between the two generative actions a and b according to probabilities p and $1-p$ (generative choice); while in the second step we probabilistically choose, locally to the righthand process, either between the two reactive actions a according to probabilities q and $1-q$ or between the two reactive actions b according to probabilities r and $1-r$, depending on whether generative action a or b wins in the first step (reactive choice).

The probabilistic transition systems we consider, called *generative-reactive* transition systems, are a mixture of the generative and reactive transition systems of [9]. In a generative-reactive transition system (a restricted version of the general model of [15]), each transition is labeled with an action, which can be either a generative action a or a reactive action a_*, and a probability. Transitions leaving a state are grouped in several bundles. We have a single generative bundle composed of all the transitions labeled with a generative action and several reactive bundles, each one referring to a different action type a and composed of all the transitions labeled with a_*. A bundle of transitions expresses a probabilistic choice, hence the sum of the probability of the transitions composing a bundle must be 1. On the contrary the choice among bundles is performed nondeterministically. In a generative-reactive transition system we see the reactive actions as *incomplete* actions which must synchronize with generative actions of another system component in order to form a complete system. A system is considered to be *fully specified* only when it gives rise to a probabilistic transition system which is purely generative, in the sense that it does not include reactive bundles. Fully specified systems are therefore fully probabilistic systems (systems not including non-deterministic choices [15]), from which a Discrete Time Markov Chain can be trivially derived by discarding actions from transition labels, which can be easily analyzed to get performance measures of the systems. In other words the limited form of non-determinism that we consider is sufficient to have a clear notion of control over action synchronizations, but is harmless from the performance analysis viewpoint in the sense that for complete systems the underlying probabilistic transition system is fully probabilistic. Moreover the modeling experience [1, 2, 5] showed that considering, as in the generative-reactive approach, two kind of actions where one predominates over the other, leads to a master-slave discipline which is very suitable for describing real systems. As in [6], we consider an asynchronous CSP [11] like parallel composition operator $P \,||_S\, Q$ where processes P and Q are required to synchronize over actions of type in the set S and locally execute all the other actions. The CSP parallel operator expresses multi-way synchronizations by assuming that the result of the synchronization of two actions a is again a visible action a. In addition we impose that a synchronization between two actions of type a may occur only if either they are both reactive actions a_* (and the result is a reactive action a_*), or one of them is a generative action a and the other one is a reactive action a_* (and the result is a generative action a). As a consequence, an n-way synchronization is composed of all reactive actions except at most one generative action: the choice of a generative action a determines the button that is pushed (button a) and the other processes internally react by *independently* choosing one of their reactive actions a_*. The parallel composition operator $P \,||_S\, Q$ that we consider is asynchronous in the sense that it allows processes with different advancing speeds to be modeled. Obviously, a mechanism must be provided to choose which of two processes in parallel must make the next move in a system state. In order to be able to express the relative advancing speed of parallel processes and to avoid the introduction of unwanted non-determinism in generative-reactive transition systems (i.e. non-determinism between generative actions or between reactive actions of the same type), we adopt, as in [6], an approach similar to that of [3] where ACP [4] is extended with probability by means of probabilistic parallel composition operator. In particular, in the calculus we present here, this is realized, differently from [6], by parameterizing the parallel operator with a probability p, so that $P \,||_S^p\, Q$ represents a system where the relative advancing speed of process P with respect to process Q is determined by the probabilities p and $1-p$, respectively. This means that in a system state the probabilities p and $1-p$ are used to choose which process must make the next move. As an example, let us consider the system[2] $(a +^p b) \,||^q (c +^r d)$. According to the approach inspired by [3], we first choose which of the two processes must make the next move according to probabilities q and $1-q$. Then, if the lefthand process wins we locally choose between a and b according to probabilities p and $1-p$, otherwise if the righthand process wins we locally choose between c and d according to probabilities r and $1-r$. The approach of [6] integrates the generative-reactive approach inspired by [16] with the approach to probabilistic process choice inspired by [3] in that the selection of the action to be executed in a system state is conceptually carried out through two steps (generative choice and reactive choice) each employing probabilistic choice among processes. Note that extending a generative-reactive approach similar to that of [16] with probabilistic process choice allows us to obtain fully probabilistic models, nevertheless remaining in an untimed probabilistic setting. In [16], instead, each state of a probabilistic I/O automata must possess an exponentially

[2]We use $||^q$ instead of $||_S^q$ when $S = \emptyset$.

distributed sojourn time so that the relative advancing speeds of the probabilistic I/O automata involved in a parallel composition are determined by a timed race condition. The calculus we present here is a restricted variant of the stochastically timed process algebra EMPA_{gr} [6], an asynchronous calculus which is based on a timed extension of the approach just described, but expresses probabilistic choices and probabilistic parallel composition by attaching weights with actions instead of parameterizing operators with a probability. We show that, when restricting to an untimed probabilistic setting, the choice of expressing probabilities in the operators leads to a simplification of the operational semantics of the algebra. In particular, in the case of synchronization of two reactive actions, we do not have to introduce an ad hoc normalization factor that turns the probability that they are independently chosen into a weight as done in [6]. Expressing probabilities in the operators is particularly adequate in the case of the parallel operator, where the *fixed* probability p, that parameterizes $P \,||^p_S\, Q$, expresses the probabilistic relative advancing speed of P with respect to Q. In this way the modeler can first specify the behaviors of P and Q in isolation and then establish, independently on how they are specified, their relative advancing speed when composing them.

Finally we show that employing probabilistic operators instead of weighted actions has a great impact on the structure of the equational characterization of strong probabilistic bisimulation. Differently from [6], a new auxiliary operator $\langle P \rangle_K$ must be introduced, so that when a term $\langle P \rangle_K$ is produced by the axiom system, K denotes which bundles are executable by P. As a result we present an axiomatization for the calculus which is sound and complete for finite processes.

2 The Generative-Reactive Model

Generative-reactive transition systems, are a mixture of the generative and reactive transition systems of [9]. In a generative-reactive transition system (a restricted version of the general model of [15]), each transition is labeled with an action, which can be either a generative action a or a reactive action a_*, and a probability. Formally, we denote the set of action types by *AType*, ranged over by $a, b, \ldots$. As usual *AType* includes the special type τ denoting internal actions. We denote the set of reactive actions by $RAct = \{a_* \mid a \in AType\}$ and the set of generative actions by $GAct = AType$. The set of actions is denoted by $Act = RAct \cup GAct$, ranged over by $\pi, \pi', \ldots$.

Transitions leaving a state are grouped in several bundles. We have a single generative bundle composed of all the transitions labeled with a generative action and several reactive bundles, each one referring to a different action type a and composed of all the transitions labeled with a_*. A bundle of transitions expresses a probabilistic choice. On the contrary the choice among bundles is performed non-deterministically.

Definition 1 *A generative-reactive probabilistic transition system GRTS is a qua-*

druple $(S, AType, T, s_i)$ *with* S *a set of states,* $AType$ *a set of action types,* $T \in \mathcal{M}(S \times Act \times]0,1] \times S)$ *a multiset*[3] *of probabilistic transitions, such that*

1. $\forall s \in S, \forall a_* \in RAct,\ \sum\{|\, p \mid \exists t \in S : (s, a_*, p, t) \in T \,|\} \in \{0,1\}$
2. $\forall s \in S,\ \sum\{|\, p \mid \exists a \in GAct, t \in S : (s, a, p, t) \in T \,|\} \in \{0,1\}$

and $s_i \in S$ *the initial state.* ■

The first requirement defines the reactive bundles (one for each action type), and the second requirement defines the unique generative bundle. Both the requirements say that for each state the probabilities of the transitions composing a bundle, if there are any, sum up to 1 (otherwise the summation over empty multisets is defined equal to 0).

3 The Calculus: Syntax and Semantics

3.1 Syntax

The syntax of the generative-reactive process algebra is defined as follows.

Let *Const* be a set of constants, ranged over by $A, B, \ldots$.

Definition 2 *The set* $\mathcal{L}$ *of process terms is generated by the syntax:*

$$P ::= \underline{0} \mid \pi.P \mid P +^p P \mid P \|^p_S P \mid P[a \to b]^p \mid A$$

where $a \in AType - \{\tau\}, b \in AType, S \subseteq AType - \{\tau\}$, $p \in]0,1[$. *The set* $\mathcal{L}$ *is ranged over by* $P, Q, \ldots$. *We denote by* $\mathcal{G}$ *the set of guarded and closed terms of* $\mathcal{L}$. ■

$\underline{0}$ represents a terminated or deadlocked term having no transitions. The prefix operator $\pi.P$ performs the action π with probability 1 and then behaves like P.

The alternative composition operator $P +^p Q$ represents a probabilistic choice between the generative actions of P and Q and between the reactive actions of P and Q of the same type. As far as generative actions are concerned, $P +^p Q$ executes a generative action of P with probability p and a generative action of Q with probability $1 - p$. In the case one process P or Q cannot execute generative actions, $P +^p Q$ chooses a generative action of the other process with probability 1 (similarly as in [3]). Similarly, $P +^p Q$ chooses between the reactive actions a_* of P and Q according to probability p, following the same mechanism.

Example 1 *As an example,* [4] $a +^p b$ *and* $a_* +^p a_*$ *represent purely probabilistic choices made according to* p. $a +^p b_*$ *and* $a_* +^p b_*$ *represent purely nondeterministic choices, where* p *is not considered.* $(a +^{p'} b_*) +^p (b +^{p''} b_*)$ *represent a mixed probabilistic and non-deterministic choice where* p' *and* p'' *are*

[3] We use "$\{|$" and "$|\}$" as brackets for multisets and $\mathcal{M}(S)$ to denote the collection of multisets over set S.

[4] We abbreviate terms $\pi.\underline{0}$ by omitting the final $\underline{0}$.

not considered. Hence we have a generative bundle made up of two transitions a and b, with probabilities p and $1-p$ respectively, and a reactive bundle of type b made up of two transitions b_*, with probabilities p and $1-p$, respectively. ■

The parallel composition operator $P\,\|_S^p\,Q$ is based on a CSP like synchronization policy, where processes P and Q are required to synchronize over actions of type in the set S, and locally execute all the other actions. Moreover, we impose that a synchronization between two actions of type a may occur only if either they are both reactive actions a_* (and the result is a reactive action a_*), or one of them is a generative action a and the other one is a reactive action a_* (and the result is a generative action a). The generative actions of P (Q) executable by $P\,\|_S^p\,Q$ are such that either their type a is not in S, or a is in S and Q (P) can perform some reactive action a_*. The choice among the generative actions of P and Q executable by $P\,\|_S^p\,Q$ is made according to probability p, by following the same probabilistic mechanism seen for alternative composition. In the case of synchronizing generative actions a of P (Q), their probability is further redistributed among the reactive actions a_* executable by Q (P), according to the probability they are chosen in Q (P). As far as reactive actions of a given type $a \notin S$ are concerned, $P\,\|_S^p\,Q$ may perform all the actions a_* executable by P or Q and the choice among them is made according to probability p, by following the same probabilistic mechanism seen for alternative composition. As far as reactive actions of a given type $a \in S$ are concerned, if both P and Q may execute some action a_*, the choice of the two actions a_* of P and Q forming the actions a_* executable by $P\,\|_S^p\,Q$ is made according to the probability they are *independently* chosen by P and Q.

The relabeling operator $P[a \to b]^p$ turns actions of type a into actions of type b. The parameter p expresses the probability that actions b_* obtained by relabeling actions a_* of P are executed with respect to the actions b_* previously performable by term P. In particular such a probabilistic choice is made by following the same mechanism seen for alternative composition. This is necessary because the choice between actions a_* and b_* in P is just non-deterministic. Parameter p is, instead, not used when relabeling generative actions because the choice between generative actions of type a and b in P is already probabilistic. Since b may be the internal action τ, $P[a \to b]^p$ may behave as an hiding operator.

Finally, constants A are used to specify recursive systems. In general, when defining an algebraic specification, we assume a set of constant defining equations of the form $A \stackrel{\Delta}{=} P$ to be given.

Note that we employ a unique alternative composition operator, instead of a purely probabilistic one and a purely non-deterministic one for two main reasons. First, it permits to circumscribe all the non-determinism to bundles of different kinds, so that fully specified systems are performance closed. Second, a unique probabilistic choice operator is the natural counterpart of the parallel composition operator $P\,\|_S^p\,Q$. An important consequence is that, as we will see in Sect. 5, we can simply characterize parallel composition by means of alternative composition.

3.2 Semantics

The formal semantics of our calculus maps terms onto generative-reactive probabilistic transition systems, where each label is composed of an action and a probability.

We assume the following abbreviations that will make the definition of the semantic rules easier. We use $P \xrightarrow{\pi}$ to stand for $\exists\, p, P' : P \xrightarrow{\pi, p} P'$, meaning that P can execute action π and $P \xrightarrow{G}$ to stand for $\exists a \in G : P \xrightarrow{a}$, $G \subseteq$ *AType*, meaning that P can execute a generative action of type belonging to set G.

The generative-reactive transition system deriving from a term $\mathcal{G}$ is defined by the operational rules in Tables 1 and 2, where in addition to the rules (r2_l) $-$ (r5_l) and (g2_l) $-$ (g7_l) referring to a local move of the lefthand process P, we consider also the symmetrical rules (r2_r) $-$ (r5_r) and (g2_r) $-$ (g7_r) taking into account the local moves of the righthand process Q, obtained by exchanging the roles of terms P and Q in the premises and, for rules (r2_r), (r4_r), (g2_r), (g4_r) and (g6_r), by replacing p with $1-p$ in the label of the derived transitions. Like in [10], we consider the operational rules as generating a multiset of transitions (consistently with the definition of a GRTS), where a transition has arity n if and only if it can be derived in n possible ways from the operational rules. We only point out that in rules (g4) $-$ (g7) we assume the set $G_{S,P} \subseteq AType$, with $S \subseteq AType - \{\tau\}$ and $P \in \mathcal{G}$, to be defined as follows: $G_{S,P} = \{a \in AType \,|\, a \notin S \vee (a \in S \wedge P \xrightarrow{a_*})\}$.

In this way $G_{S,Q}$ ($G_{S,P}$) is the set of types of the generative transitions of P (Q) executable by $P \,||^p_S\, Q$. As we consider a restricted set of executable actions, we normalize the probabilities of the generative transitions of P (Q) executable by $P \,||^p_S\, Q$ so that their overall probability sums up to 1 [9]. To this aim in semantics rules we employ the function $\nu_P(G) : \mathcal{P}(AType) \longrightarrow]0,1]$, with $P \in \mathcal{G}$, defined as follows: $\nu_P(G) = \sum\{|\, p \,|\, \exists P', a \in G : P \xrightarrow{a,p} P' \,|\}$ which computes the sum of the probabilities of the generative transitions executable by P whose type belongs to the set G. In this way $\nu_P(G_{S,Q})$ ($\nu_Q(G_{S,P})$) computes the overall probability of the generative transitions of P (Q) executable by $P \,||^p_S\, Q$ and can be used to normalize the probabilities of the generative transitions of P (Q).

Definition 3 *The operational semantics of $P \in \mathcal{G}$ is the GRTS $[\![P]\!]$ composed of the terms reachable from P according to the operational semantics. We say that $P \in \mathcal{G}$ is performance closed if and only if $[\![P]\!]$ does not include reactive transitions.* ■

It is easy to see that if P and Q are two processes of $\mathcal{G}$ and $[\![P]\!]$ and $[\![Q]\!]$ are *GRTS*, then $[\![\pi.P]\!]$, $[\![P[a \to b]^p]\!]$, $[\![P +^p Q]\!]$ and $[\![P \,||^p_S\, Q]\!]$ are *GRTS*. Moreover, recursive definitions preserve the property of being a *GRTS*.

Theorem 1 *If P is a process of $\mathcal{G}$, then $[\![P]\!]$ is a GRTS.* ■

$$(r1) \qquad a_*.P \xrightarrow{a_*,1} P$$

$$(r2_l) \qquad \frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}}{P +^p Q \xrightarrow{a_*,p\cdot q} P'} \qquad\qquad (r3_l) \qquad \frac{P \xrightarrow{a_*,q} P' \quad Q \not\xrightarrow{a_*}}{P +^p Q \xrightarrow{a_*,q} P'}$$

$$(r4_l) \qquad \frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}}{P \|^p_S Q \xrightarrow{a_*,p\cdot q} P' \|^p_S Q}\ a \notin S \qquad (r5_l) \qquad \frac{P \xrightarrow{a_*,q} P' \quad Q \not\xrightarrow{a_*}}{P \|^p_S Q \xrightarrow{a_*,q} P' \|^p_S Q}\ a \notin S$$

$$(r6) \qquad \frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*,q'} Q'}{P \|^p_S Q \xrightarrow{a_*,q\cdot q'} P' \|^p_S Q'}\ a \in S$$

$$(r7) \qquad \frac{P \xrightarrow{a_*,q} P' \quad P \xrightarrow{b_*}}{P[a \to b]^p \xrightarrow{b_*,p\cdot q} P'} \qquad\qquad (r8) \qquad \frac{P \xrightarrow{a_*,q} P' \quad P \not\xrightarrow{b_*}}{P[a \to b]^p \xrightarrow{b_*,q} P'}$$

$$(r9) \qquad \frac{P \xrightarrow{b_*,q} P' \quad P \xrightarrow{a_*}}{P[a \to b]^p \xrightarrow{b_*,(1-p)\cdot q} P'} \qquad\qquad (r10) \qquad \frac{P \xrightarrow{b_*,q} P' \quad P \not\xrightarrow{a_*}}{P[a \to b]^p \xrightarrow{b_*,q} P'}$$

$$(r11) \qquad \frac{P \xrightarrow{c_*,q} P'}{P[a \to b]^p \xrightarrow{c_*,q} P'}\ c \notin \{a,b\}$$

$$(r12) \qquad \frac{P \xrightarrow{a_*,q} P'}{A \xrightarrow{a_*,q} P'}\ \text{if } A \triangleq P$$

Table 1: Semantic rules for Reactive Transitions

3.3 Equivalence

We now equip the algebra with a probabilistic bisimulation equivalence (along the lines of [12]), which relates systems having the same functional and probabilistic behavior. We first introduce a function capturing the total probability with which a term reaches a given class of terms by executing a given action of *Act*.

Definition 4 *We define function* $Prob : (\mathcal{G} \times Act \times \mathcal{P}(\mathcal{G})) \longrightarrow [0,1]$ *by*

$$Prob(P,\pi,C) = \sum \{\!|\, p \mid P \xrightarrow{\pi,p} P' \wedge P' \in C \,|\!\}$$

An equivalence relation $\mathcal{B} \subseteq \mathcal{G} \times \mathcal{G}$ *is a strong probabilistic bisimulation if and only if, whenever* $(P,Q) \in \mathcal{B}$*, then for all* $\pi \in Act$ *and equivalence classes* $C \in$

$$(g1) \quad a.P \xrightarrow{a,1} P$$

$$(g2_l) \quad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{GAct}}{P +^p Q \xrightarrow{a,p\cdot q} P'} \qquad (g3_l) \quad \frac{P \xrightarrow{a,q} P' \quad Q \not\xrightarrow{GAct}}{P +^p Q \xrightarrow{a,q} P'}$$

$$a \notin S$$

$$(g4_l) \quad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{G_{S,P}}}{P \|_S^p Q \xrightarrow{a,p\cdot q/\nu_P(G_{S,Q})} P' \|_S^p Q} \qquad (g5_l) \quad \frac{P \xrightarrow{a,q} P' \quad Q \not\xrightarrow{G_{S,P}}}{P \|_S^p Q \xrightarrow{a,q/\nu_P(G_{S,Q})} P' \|_S^p Q}$$

$$a \in S$$

$$(g6_l) \quad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{a_*,q'} Q' \quad Q \xrightarrow{G_{S,P}}}{P \|_S^p Q \xrightarrow{a,p\cdot q'\cdot q/\nu_P(G_{S,Q})} P' \|_S^p Q'} \qquad (g7_l) \quad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{a_*,q'} Q' \quad Q \not\xrightarrow{G_{S,P}}}{P \|_S^p Q \xrightarrow{a,q'\cdot q/\nu_P(G_{S,Q})} P' \|_S^p Q'}$$

$$(g8) \quad \frac{P \xrightarrow{a,q} P'}{P[a \to b]^p \xrightarrow{b,q} P'} \qquad (g9) \quad \frac{P \xrightarrow{c,q} P'}{P[a \to b]^p \xrightarrow{c,q} P'} \; a \neq c$$

$$(g10) \quad \frac{P \xrightarrow{a,q} P'}{A \xrightarrow{a,q} P'} \; \text{if } A \triangleq P$$

Table 2: Semantic Rules for Generative Transitions

$\mathcal{G}/\mathcal{B}$: $Prob(P,\pi,C) = Prob(Q,\pi,C)$. The strong probabilistic bisimulation equivalence, denoted by $\sim_{PB}$, is the union of all strong probabilistic bisimulations. ■

Theorem 2 *$\sim_{PB}$ is a congruence w.r.t. all operators and recursive definitions.* ■

4 Axiomatization

In this section we develop an equational characterization of $\sim_{PB}$ for the set $\mathcal{G}_{fin}$ of nonrecursive terms of $\mathcal{G}$, i.e. terms not including constants A. In order to produce an axiom system which is complete over $\mathcal{G}_{fin}$ processes we need to introduce an auxiliary operator $\langle P \rangle_K$, so that when a term $\langle P \rangle_K$ is produced by the axiom system, K denotes the bundles executable by P.

Formally, we define bundle *kinds* as follows. Given a term P, "$\bullet$" is the kind of the generative bundle executable by P, while a is the kind of the reactive bundle of type a executable by P. We denote with $BKind = \{\bullet\} \cup AType$ the set of bundle kinds, ranged over by $\kappa, \kappa', \ldots$ We denote by $\mathcal{G}_{ext}$ the set of terms obtained by extending the syntax of $\mathcal{G}_{fin}$ terms with the new operator $\langle P\rangle_K$, where K is a finite set of bundle kinds, i.e. a subset of *BKind*. Intuitively we have to introduce the new operator $\langle P\rangle_K$ in order to understand the role of parameter p in a choice $P +^p Q$, which can be probabilistic or non-deterministic depending on which bundles are executable by P and Q. This is, instead, not necessary in [6] where probabilistic choices are determined by weights attached to actions. In particular we have that the choice $\langle P\rangle_K +^p \langle Q\rangle_{K'}$ is purely probabilistic when there exists κ such that $K = K' = \{\kappa\}$, while it is purely non-deterministic whenever K and K' are disjoint.

In order to make it possible to produce an axiom system which, besides being complete, is also sound, we define an operational semantics for the new operator $\langle P\rangle_K$. Since the role of the operator $\langle P\rangle_K$ in our axiom system is just to denote that the bundles executable by P are exactly the bundles stated by K, $\langle P\rangle_K$ must have no effect on the behavior of processes P satisfying this requirement. However, even if terms $\langle P\rangle_K$ such that P does not satisfy the requirement above cannot be produced by our axiom system, we have to define $\langle P\rangle_K$ over arbitrary terms P. The operational semantics of $\langle P\rangle_K$ is defined by the following semantic rules:

$$(e1)\frac{P \xrightarrow{\pi,q} P'}{\langle P\rangle_K \xrightarrow{\pi,q} P'} \text{ if } \kappa(\pi) \in K \quad (e2)\frac{P \stackrel{a_*}{\not\longrightarrow}}{\langle P\rangle_K \xrightarrow{a_*,1} \underline{0}} \text{ if } a \in K \quad (e3)\frac{P \stackrel{GAct}{\not\longrightarrow}}{\langle P\rangle_K \xrightarrow{a,1} \underline{0}} \text{ if } \bullet \in K$$

We employ the function $\kappa : Act \longrightarrow BKind$ which determines the kind of bundle a transition belongs to. Formally $\kappa(\pi)$ is defined as follows:

$$\begin{array}{lrcl} \forall a \in GAct. & \kappa(a) & = & \bullet \\ \forall a_* \in RAct. & \kappa(a_*) & = & a \end{array}$$

Rule (e1) eliminates all the bundles executable by P whose kind is not in K. Rule (e2) creates a reactive bundle of type a in the case P cannot execute such a bundle and $a \in K$. Rule (e3) creates a generative bundle in the case P cannot execute such a bundle and $\bullet \in K$. The action a used in (e_3) is any *fixed* generative action a and can be considered as a parameter of the operator $\langle P\rangle_K$.

Clearly if P is a process of $\mathcal{G}_{ext}$ and $[\![P]\!]$ is a *GRTS* then $[\![\langle P\rangle_K]\!]$ is a *GRTS*.

Theorem 3 *If P is a process of $\mathcal{G}_{ext}$, then $[\![P]\!]$ is a GRTS.* ■

Theorem 4 *$\sim_{PB}$ is a congruence w.r.t. the new operator $\langle P\rangle_K$.* ■

The equational characterization of $\sim_{PB}$ is composed of the set $\mathcal{A}$ of axioms shown in Table 3.

Axioms $\mathcal{A}_1 - \mathcal{A}_8$ express the basic properties of the choice operator. In particular $\mathcal{A}_3$ expresses a form of commutativity which complements p ([3]). Axioms $\mathcal{A}_4 - \mathcal{A}_6$ generate the brackets $\langle P\rangle_K$ denoting the bundles executable by P. Axiom $\mathcal{A}_7$ expresses associativity for pure probabilistic choices, i.e. choices among terms enabling a single common bundle. It defines the adjustment of the probabilistic parameters so that the probability of executing a given member of the sum is preserved (as in [3]). Thanks to the properties of commutativity and associativity of pure probabilistic choices we can just use the following notation:

$$
\begin{array}{llllll}
(\mathcal{A}_1) & P +^p \underline{0} = P & (\mathcal{A}_2) & P +^p P = P & (\mathcal{A}_3) & P +^p Q = Q +^{1-p} P \\
(\mathcal{A}_4) & a.P = \langle a.P \rangle_{\{\bullet\}} & (\mathcal{A}_5) & a_*.P = \langle a_*.P \rangle_{\{a\}} & (\mathcal{A}_6) & \langle P \rangle_K +^p \langle Q \rangle_{K'} = \langle \langle P \rangle_K +^p \langle Q \rangle_{K'} \rangle_{K \cup K'}
\end{array}
$$

$$
\begin{array}{ll}
(\mathcal{A}_7) & (\langle P \rangle_{\{\kappa\}} +^p \langle Q \rangle_{\{\kappa\}}) +^q \langle R \rangle_{\{\kappa\}} = \langle P \rangle_{\{\kappa\}} +^{pq} (\langle Q \rangle_{\{\kappa\}} +^{(1-p)\cdot q/(1-p\cdot q)} \langle R \rangle_{\{\kappa\}}) \\
(\mathcal{A}_8) & (\langle P \rangle_K +^p \langle Q \rangle_{K'}) +^q R = \langle P \rangle_K +^q (\langle Q \rangle_{K'} +^q R) \qquad \textit{if } K \cap K' = \emptyset
\end{array}
$$

$$
\begin{array}{ll}
(\mathcal{A}_9) & ((\sum_{i \in I} [p_i] c_i.P_i) +^{ND} (\sum_{c \in AType} [ND] \sum_{i \in I^c} [p_i] c_*.P_i))[a \to b]^p = \\
& \quad (\sum_{j \in I, c_j \neq a} [p_j] c_j.(P_j[a \to b]^p) + \sum_{j \in I, c_j = a} [p_j] b.(P_j[a \to b]^p)) \\
& \quad +^{ND} \\
& \quad ((\sum_{c \in AType, c \notin \{a,b\}} [ND] \sum_{j \in I^c} [p_j] c_*.(P_j[a \to b]^p)) \\
& \quad\ +^{ND} \\
& \quad\ ((\sum_{j \in I^a} [p_j] b_*.(P_j[a \to b]^p)) +^p (\sum_{j \in I^b} [p_j] b_*.(P_j[a \to b]^p))))
\end{array}
$$

$$
\begin{array}{ll}
(\mathcal{A}_{10}) & P^0 \parallel_S^p P^1 = \\
& \quad (((\sum_{j \in I_0, a_j \notin S} [p_j/\nu_0(G_{S,1})] a_j.(P_j \parallel_S^p P^1) + \\
& \qquad \sum_{k \in I_0, a_k \in S, I_1^{a_k} \neq \emptyset} \sum_{h \in I_1^{a_k}} [p_h \cdot p_k/\nu_0(G_{S,1})] a_k.(P_k \parallel_S^p P_h)) +^{ND} \\
& \qquad \sum_{a \in AType, a \notin S} [ND] \sum_{j \in I_0^a} [p_j] a_*.(P_j \parallel_S^p P^1)) \\
& \quad\ +^p \\
& \quad\ ((\sum_{j \in I_1, a_j \notin S} [p_j/\nu_1(G_{S,0})] a_j.(P^0 \parallel_S^p P_j) + \\
& \qquad \sum_{k \in I_1, a_k \in S, I_0^{a_k} \neq \emptyset} \sum_{h \in I_0^{a_k}} [p_h \cdot p_k/\nu_1(G_{S,0})] a_k.(P_h \parallel_S^p P_k)) +^{ND} \\
& \qquad \sum_{a \in AType, a \notin S} [ND] \sum_{j \in I_1^a} [p_j] a_*.(P^0 \parallel_S^p P_j))) \\
& \quad +^{ND} \\
& \quad \sum_{a \in AType, a \in S, I_0^a \neq \emptyset, I_1^a \neq \emptyset} [ND] (\sum_{k \in I_0^a} \sum_{h \in I_1^a} [p_k \cdot p_h] a_*.(P_k \parallel_S^p P_h)) \\
& \quad \text{where :} \qquad P^l \equiv (\sum_{i \in I_l} [p_i] a_i.P_i) +^{ND} (\sum_{a \in AType} [ND] \sum_{i \in I_l^a} [p_i] a_*.P_i) \qquad \text{for } l \in \{0,1\}
\end{array}
$$

Table 3: Axiomatization

$$
\sum_{i \in \{1,\ldots,n\}} [p_i] \langle P_i \rangle_{\{\kappa\}} \equiv \sum_{i \in \{1,\ldots,k\}} [p_i] \langle P_i \rangle_{\{\kappa\}} + \sum_{i \in \{k,\ldots,n\}} [p_i] \langle P_i \rangle_{\{\kappa\}} \equiv [p_1] \langle P_1 \rangle_{\{\kappa\}} + \ldots + [p_n] \langle P_n \rangle_{\{\kappa\}}
$$

to stand for any term obtained by choosing a particular way of grouping the members of the sum. [5] According to this notation $[p_i]\langle P_i \rangle_{\{\kappa\}}$ denotes that the member of the sum $\langle P_i \rangle_{\{\kappa\}}$ is executed with probability p_i. Axiom $\mathcal{A}_8$ expresses associativity for choices where two of the terms involved enable disjoint sets of bundle kinds. By taking $R \equiv \underline{0}$ in $\mathcal{A}_8$ and applying axiom $\mathcal{A}_1$ we obtain

$$
\langle P \rangle_K +^p \langle Q \rangle_{K'} = \langle P \rangle_K +^q \langle Q \rangle_{K'} \text{ if } K \cap K' = \emptyset
$$

This equation expresses the fact that whenever we have a choice between two

[5] We assume $\sum_{i \in I} [p_i] \langle P_i \rangle_{\{\kappa\}} \equiv \underline{0}$ whenever $I = \emptyset$.

terms enabling different bundles, i.e. a pure non-deterministic choice, the probabilistic parameter of the choice is not important. Thanks to this property, supposed $K \cap K' = \emptyset$, we can just use the notation $\langle P \rangle_K +^{ND} \langle Q \rangle_{K'}$, to stand for any term $\langle P \rangle_K +^p \langle Q \rangle_{K'}$ obtained for a particular choice of p. [6]

By taking $R \equiv \langle R' \rangle_{K''}$, such that $K'' \cap K = \emptyset$ and $K'' \cap K' = \emptyset$, in $\mathcal{A}_8$ and by taking into account the previous equation we obtain:

$$(\langle P \rangle_K +^{ND} \langle Q \rangle_{K'}) +^{ND} \langle R \rangle''_K \; = \langle P \rangle_K +^{ND} (\langle Q \rangle_{K'} +^{ND} \langle R \rangle''_K) \quad \text{if } K, K', K'' \text{ are pairwise disjoint}$$

This equation expresses associativity for pure non-deterministic choices among terms enabling disjoint sets of bundles. Thanks to the properties of commutativity and associativity of pure non-deterministic choices, supposed that K_i with $i \in \{1, \dots, n\}$ are pairwise disjoint, we can just use the notation

$$\sum_{i \in \{1,\dots,n\}} [ND] \langle P_i \rangle_{K_i} \equiv \sum_{i \in \{1,\dots,k\}} [ND] \langle P_i \rangle_{K_i} + \sum_{i \in \{k,\dots,n\}} [ND] \langle P_i \rangle_{K_i} \equiv [ND] \langle P_1 \rangle_{K_1} + \dots + [ND] \langle P_n \rangle_{K_n}$$

to stand for any term obtained by choosing a particular way of grouping the members of the sum.

Moreover, axiom $\mathcal{A}_8$ allows to turn any sequential term (a term made up of $\underline{0}$, prefix and choice) into *sum normal form*. In particular, $P \in \mathcal{G}_{ext}$ is in *sum normal form* (*snf*) if and only if $P \equiv \sum_{i \in I} [p_i] a_i . P_i +^{ND} \sum_{a \in AType} [ND] \sum_{i \in I^a} [p_i] a_* . P_i$ where every P_i is itself in *snf*. [7] We assume the sets of indexes I and I^a for each $a \in AType$ to be pairwise disjoint, i.e. $\forall a \in AType. I_a \cap I = \emptyset$ and $\forall a, b \in AType, a \neq b. I_a \cap I_b = \emptyset$.

Axioms $\mathcal{A}_9$ and $\mathcal{A}_{10}$ refer to the relabeling and parallel composition operators, respectively. Axiom $\mathcal{A}_9$, where we assume the sets of indexes I and I^a for each $a \in AType$ to be pairwise disjoint, determines the term in snf obtained by applying relabeling to a term in snf. Similarly the axiom $\mathcal{A}_{10}$, where we assume the sets of indexes I_0, I_0^a for each $a \in AType$, I_1 and I_1^a for each $a \in AType$ to be pairwise disjoint, determines the term in snf obtained by composing in parallel two terms in snf. In axiom $\mathcal{A}_{10}$ we use 0 and 1 when we consider information concerning the terms P_0 and P_1, respectively. In particular, as in the operational semantics, we use the following auxiliary definitions. The set $G_{S,i} \subseteq AType$, with $S \subseteq AType - \{\tau\}$ and $i \in \{0, 1\}$ is defined by: $G_{S,i} = \{a \in AType \mid a \notin S \vee (a \in S \wedge I_i^a \neq \emptyset)\}$. The function $\nu_i : \mathcal{P}(AType) \longrightarrow]0, 1]$, with $i \in \{0, 1\}$ is defined by: $\nu_i(G) = \sum \{\pi_j \mid j \in I_i \wedge a_j \in G\}$.

Theorem 5 *The axiom system $\mathcal{A}$ is sound for $\sim_{PB}$ over processes of $\mathcal{G}_{ext}$.* ■

Lemma 1 *For any $P \in \mathcal{G}_{fin}$ there exists $P' \in \mathcal{G}_{fin}$ in snf such that $\mathcal{A} \vdash P = P'$.* ■

Theorem 6 *The axiom system $\mathcal{A}$ is complete for $\sim_{PB}$ over processes of $\mathcal{G}_{fin}$.* ■

[6] This notation is employed just to improve the readability of axioms: whenever $P +^{ND} Q$ occurs in some equation, it could just be replaced by any choice of a particular parameter p.

[7] Note that the use of $\sum$ and ND is correct since we have $a_i . P_i = \langle a_i . P_i \rangle_{\{\bullet\}}$ (axiom $\mathcal{A}_4$) and $a_* . P_i = \langle a_* . P_i \rangle_{\{a\}}$ (axiom $\mathcal{A}_5$).

References

[1] A. Aldini, M. Bernardo, R. Gorrieri, *"An Algebraic Model for Evaluating the Performance of an ATM Switch with Explicit Rate Marking"*, in Proc. of the *7th Int. Workshop on Process Algebras and Performance Modeling (PAPM '99)*, Prensas Universitarias de Zaragoza, pp. 119-138, 1999

[2] A. Aldini, M. Bernardo, R. Gorrieri, M. Roccetti, *"A Simulative Analysis of Internet Audio Mechanisms Using Formal Methods"*, to appear in Proc. of the *11th European Simulation Symp. (ESS '99)*, SCSI, 1999

[3] J.C.M. Baeten, J.A. Bergstra, S.A. Smolka, *"Axiomatizing Probabilistic Processes: ACP with Generative Probabilities"*, in Information and Computation 121:234-255, 1995

[4] J.C.M. Baeten, W.P. Weijland, *"Process Algebra"*, Cambridge University Press, 1990

[5] M. Bernardo, *"Theory and Application of Extended Markovian Process Algebra"*, Ph.D. Thesis, University of Bologna (Italy), 1999

[6] M. Bravetti, M. Bernardo, *"Compositional Asymmetric Cooperations for Process Algebras with Probabilities, Priorities, and Time"*, Tech. Rep. UBLCS-2000-01, University of Bologna (Italy), 2000

[7] R. Cleaveland, G. Luttgen, V. Natarajan, *"Priority in Process Algebras"*, to appear in Handbook of Process Algebra, Elsevier, 2000

[8] P. R. D'Argenio, H. Hermanns, and J. P. Katoen, *"On Generative Parallel Composition"*, Electronic Notes in Theoretical Computer Science, Vol. 22, 1999

[9] R.J. van Glabbeek, S.A. Smolka, B. Steffen, *"Reactive, Generative and Stratified Models of Probabilistic Processes"*, in Information and Computation 121:59-80, 1995

[10] J. Hillston, *"A Compositional Approach to Performance Modelling"*, Cambridge University Press, 1996

[11] C.A.R. Hoare, *"Communicating Sequential Processes"*, Prentice Hall, 1985

[12] K. Larsen, A. Skou, *"Bisimulation through Probabilistic Testing"*, in Information and Computation 94:1-28, 1991

[13] N.A. Lynch, M.R. Tuttle, *"Hierarchical Correctness Proofs for Distributed Algorithms"*, in Proc. of the *6th ACM Symp. on Principles of Distributed Computing (PODC '87)*, ACM Press, pp. 137-151, Vancouver (Canada), 1987

[14] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989

[15] R. Segala, *"Modeling and Verification of Randomized Distributed Real-Time Systems", Ph.D. Thesis, MIT, Boston (MA), 1995*

[16] S. H. Wu, S.A. Smolka, E.W. Stark, *"Composition and Behaviors of Probabilistic I/O Automata"*, in Theoretical Computer Science 176:1-38, 1997

Alessandro Aldini and Mario Bravetti are with the Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy. E-mail: {aldini, bravetti}@cs.unibo.it