# Probabilistic Extensions of Process Algebras*

*Bengt Jonsson*[1] *Kim G Larsen*[2] *Wang Yi*[1]

[1] Department of Computer Systems, Uppsala University, Sweden, E-mail:
{bengt,yi}@docs.uu.se

[2] Department of Computer Science, Aalborg University, Denmark, E-mail:
kgl@cs.auc.dk

In this chapter, we adopt Probabilistic Transition Systems as a basic model for probabilistic processes, in which probabilistic and nondeterministic choices are independent concepts. The model is essentially a nondeterministic version of Markov decision processes or probabilistic automata of Rabin. We develop a general framework to define probabilistic process languages to describe probabilistic transition systems. In particular, we show how operators for nonprobabilistic process algebras can be lifted to probabilistic process algebras in a uniform way similar to de Simone format. To establish a notion of refinement, we present a family of preorders including probabilistic bisimulation and simulation, and probabilistic testing preorders as well as their logical or denotational characterization. These preorders are shown to be precongruences with respect to the algebraic operators that can be defined in our general framework. Finally, we give a short account of the important work on extending the succesfull field of model checking to probabilistic settings and a brief discusion on current research in the area.

**Keywords:** *Probabilistic Transition Systems, Probabilistic Process Algebras, Bisimulation, Simulation, Testing, Testing Preorders, Probabilistic Temporal Logics and Model-Checking*

## 1 INTRODUCTION

Classic process, algebras such as CCS, CSP and ACP, are well-established techniques for modelling and reasoning about functional aspects of concurrent processes. The motivation for studying probabilistic extensions of process algebras is to develop techniques dealing with non-functional aspects of process behavior, such as performance and reliability. We may want to investigate, e.g., the average response time of a system, or the

* This chapter is dedicated to the fond memory of Linda Christoff.

probability that a certain failure occurs. An analysis of these and similar properties requires that some form of information about the stochastic distribution over the occurrence of relevant events is put into the model. For instance, performance evaluation is often based on modeling a system as a continuous-time Markov process, in which distributions over delays between actions and over the choice between different actions are specified. Similarly, reliability can be analyzed quantitatively only if we know some probability of the occurrence of events related to a failure. Performance evaluation and reliability analysis are well-established topics, and it is not the aim to contribute in these areas. Rather, we should try to see what the process algebraic approach can offer to these fields. Process algebras has contributed to our understanding of

- how to describe (model) communicating systems compositionally,
- how to formulate correctness properties of systems,
- how properties of a system relate to properties of its components, and
- what it means for a description of a system component to be a correct implementation of another component description.

A solution to these problems would be very useful, e.g., in a stepwise development process. An abstract model can be analyzed by proving properties in some logic (for the non-probabilistic case, see e.g., Chapter 1.4 of this handbook). The abstract model can then be refined in a sequence of steps, where correctness is preserved in each step by establishing a preorder relation between the refined system and the refining one (techniques for the non-probabilistic case are described in Chapter 2.2).

In this chapter, we will study the above issues in the context of a simple yet general model of probabilistic processes. In the non-probabilistic setting, labeled transition systems are well-established as a basic semantic model for concurrent and distributed systems (e.g. [43,46]). In the literature, the model of transition systems has been extended to the probabilistic case by adding a mechanism for representing probabilistic choice (e.g. [54,29,31,41,44,47,48,50]). We will adopt a model of probabilistic transition systems, in which probabilistic and nondeterministic choice are independent concepts. Non-determinism can be used to represent underspecification, which can then be partly removed in refinement steps. For example, nondeterminism can be used to specify the allowed probabilities of failure of a medium, and a refinement can decrease the set of allowed failure rates [37]. Non-determinism can also represent incomplete information on the parameters of system behavior, such as Milner's "weather conditions" [43].

Our model is essentially a nondeterministic version of Markov decision processes [24] or the probabilistic automata of Rabin [50]. In the area of process algebra, the model has been put forward by Vardi under the term concurrent Markov chain [55], by Wang and Larsen [57] and by Hansson and Jonsson as the alternating model [31], and by Segala and Lynch [52, 51]. A deterministic version has been proposed as the reactive model by Larsen and Skou [41]. There are several other models of probabilistic transition systems proposed in the literature. A short summary will be provided in section 3.2.

The rest of the chapter will be organized as follows: Section 2 introduces notation and basic concepts from probability theory. Section 3 presents Probabilistic Transition Systems (PTS) as a basic model for probabilistic processes and summarizes variants of PTS proposed in the literature. Section 4 will consider how operators can be defined on probabilistic processes. We will, in particular, see how operators for nonprobabilistic process algebras can be lifted to the probabilistic case in a uniform way similar based on de Simone format. Probabilities are added only by means of a probabilistic choice construct. We will thereafter, in the following sections, consider different preorders between probabilistic processes, and how these preorders interact with the operators for constructing processes i.e. (pre)congruence results. Section 5 is devoted to the development of probabilistic versions of bisimulation and simulation. These are preorders based on relating states or distributions to each other i.e. relations on branching (or tree) structures. Section 6 presents testing preorders, another family of preorders that are defined in terms of an operational notion of testing, and characterize them in terms of simulations. Section 7 presents a basic probabilistic modal logic and show how it relates to and characterizes the various behavioural preorders. Also, we give a short account of the important work on extending the successful field of model checking to probabilistic settings. Section 8 conclude the chapter with a brief discussion on current research in the area.

## 2 Preliminaries

In this section, we introduce some notation and definitions from probability theory.

A *probability distribution* on a countable set $S$ is a function $\pi : S \rightarrow [0,1]$ such that $\sum_{s \in S} \pi(s) = 1$. More generally, a *weighting* on a set $S$ is a function $\pi : S \rightarrow \mathcal{R}_{\geq 0}$ from $S$ to nonnegative real numbers. Note that a *probability distribution* on a finite set $S$ is a weighting $\pi$ on $S$ such that $\pi(S) = 1$. The *support* of a distribution or weighting $\pi$ on $S$, denoted $\mathrm{support}(\pi)$ is the set of elements $s$ such that $\pi(s) > 0$. For a subset $S' \subseteq S$, we define $\pi(S') = \sum_{s \in S'} \pi(s)$. Let $\mathit{Dist}(S)$ denote the set of probability distributions on $S$. If $\pi$ is a probability distribution on $S$ and $\rho$ is a probability distribution on $T$, then their *product* $\sigma = \pi \times \rho$ is a probability distribution on $S \times T$, defined by $\sigma(\langle s,t \rangle) = \pi(s) * \rho(t)$. For simplicity, we shall write $\sigma(s,T)$ for $\sigma(\{s\} \times T)$ and $\sigma(S,t)$ for $\sigma(S \times \{t\})$.

We will next define a general way, proposed by Jonsson and Larsen [37], to lift a relation between two countable sets to a relation between distributions on these sets.

**Definition 1.** *Let $\approx \subseteq S \times T$ be a relation between the sets $S$ and $T$, $\pi$ be a probability distribution on $S$ and $\rho$ be a probability distribution on $T$. We define $\pi \approx^* \rho$ iff there is a distribution $\alpha \in \mathit{Dist}(S \times T)$ on $S \times T$ such that*

- *$\alpha(s,T) = \pi(s)$ for each $s \in S$,*
- *$\alpha(S,t) = \rho(t)$ for each $t \in T$ and*
- *$\alpha(s,t) = 0$ if $s \not\approx t$.* □

We shall write $\pi \approx \rho$ whenever $\pi \approx^* \rho$ and it is understood from the context.

Intuitively, $\pi \approx^* \rho$ means that there is a distribution on $S \times T$ whose projection onto $S$ is $\pi$, whose projection onto $T$ is $\rho$, and whose support is in $\approx$. The relation $\pi \approx^* \rho$ thus holds if for each $s \in S$, it is possible to distribute the probability $\pi(s)$ over elements of $T$ that are $\approx$-related to $s$, in such a way that the sum of these distributed probabilities, weighted by $\pi$, is the distribution $\rho$.

There is a simpler way of lifting equivalence relations on countable sets to the distributions on these sets.

**Theorem 1.** *Let $\approx$ be an equivalence relation over the set $S$. Then $\pi \approx^* \rho$ iff $\pi([s]) = \rho([s])$ for all equivalence classes $[s] \subseteq S$ of relation $\approx$.*

*Proof. If:* Assume that $\pi([s]) = \rho([s])$ for all $s \in S$. Let $\alpha(s,t) = \pi(s) * \rho(t)/\pi([s])$ if $s \approx t$ and $\pi(s) > 0$, and 0 otherwise. Then $\alpha(s,S) = \pi(s) *$

$\rho([s])/\pi([s]) = \pi(s)$ as $\pi([s]) = \rho([s])$. Similarly, $\alpha(S,t) = \rho(t)$. The third condition holds immediately from the definition of $\alpha$.

*Only If:* Assume that $\pi \approx^* \rho$ and further assume that $\alpha$ is as in Definition 1. Then $\pi([s]) = \sum_{s \in [s]} \alpha(s,S) = \alpha([s],S) = \alpha([s],[s])$ since $\alpha(s,t) = 0$ for $t \notin [s]$. Symmetrically, $\rho([s]) = \alpha([s],[s])$. Thus $\pi([s]) = \rho([s])$. □

The lifting operation on relations preserves the characteristic properties of preorders and equivalences.

**Theorem 2.** *Let $\approx$ be a relation on the set $S$. Then $\approx$ is a preorder implies that $\approx^*$ is a preorder on $Dist(S)$.*

*Proof.* We only show the transitivity of $\approx^*$. Assume that $\pi \approx \rho$ and $\rho \approx \varrho$. Then there are distributions $\alpha$ and $\beta$ on $S \times S$ satisfying the three conditions given in Definition 1. Now let $\gamma(s,t) = \sum_{s' \in S} \alpha(s,s')/\ \rho(s') * \beta(s',t)$. We check the three conditions in Definition 1. First $\gamma(s,S) = \sum_{s' \in S} \alpha(s,s')\ /\ \rho(s') * \beta(s',S) = \sum_{s' \in S}(\alpha(s,s')/\ \rho(s')) * \rho(s')) = \sum_{s' \in S} \alpha(s,s') = \pi(s)$. Second, $\gamma(S,t) = \sum_{s' \in S}(\alpha(S,s')/\ \rho(s') * \beta(s',t)) = \sum_{s' \in S}\ (\ \rho(s')/\ \rho(s') * \beta(s',t)) = \sum_{s' \in S} \beta(s',t) = \varrho(t)$. For the third condition, note that if $\gamma(s,t) > 0$ then there must be $s'$ such that $\alpha(s,s') * \beta(s',t) > 0$. This implies that $s \approx s'$ and $s' \approx t$. By the transitivity of $\approx$, $s \approx t$. □

The above result can be extended to equivalence relations.

**Theorem 3.** *Let $\approx$ be a relation on the set $S$. Then $\approx$ is an equivalence implies that $\approx^*$ is an equivalence on $Dist(S)$.*

*Proof.* Immediate from Theorem 1

Equivalences and preorders will be of particular interests in the rest of this chapter. We extend a relation $\approx$ on the set $S$ to the cartesian product $S \times S$ in the usual way. That is, $\langle s,t \rangle \approx \langle s',t' \rangle$ whenever $s \approx s'$ and $t \approx t'$. Then $\approx^*$ is preserved by the product operation on probability distributions.

**Theorem 4.** *Let $\approx$ be a preorder (or an equivalence relation) and $\pi, \rho, \pi'$ and $\rho'$ be probabilistic distributions on $S$. Then $\pi \approx^* \rho$ and $\pi' \approx^* \rho'$ imply that $\pi \times \pi' \approx^* \rho \times \rho'$.*

*Proof.* By the transitivity of $\approx^*$, we only need to establish that $\pi \approx \rho$ implies $\pi \times \varrho \approx^* \rho \times \varrho$.

Following definition 1, we construct a distribution $\beta$ over $(S \times S) \times (S \times S)$. First note that $\pi \approx^* \rho$. Thus, there exists a distribution $\alpha$ on $S \times S$ such that $\alpha(s, S) = \pi(s)$, $\alpha(S, s') = \varrho(t)$ and $\alpha(s, s') = 0$ for $s \not\approx s'$.

Now let $\beta(\langle s, t\rangle, \langle s', t'\rangle) = \alpha(s, s') * \varrho(t)$ whenever $t = t'$ and 0 otherwise. Then $\beta(\langle s, t\rangle, S \times S) = \sum_{s', t' \in S}(\alpha(s, s') * \varrho(t)) = \rho(t) * \sum_{s', t' \in S} \alpha(s, s') = \rho(t) * \sum_{s' \in S} \alpha(s, s') = \rho(t) * \pi(s)$. Thus $\beta$ holds for the first condition in definition 1. Similarly, we can establish the second condition for $\beta$. That is $\beta(S \times S, \langle s', t'\rangle) = \rho(s') * \varrho(t')$. The third condition is obvious. If $\langle\langle s, t\rangle, \langle s', t'\rangle\rangle \not\approx$ then $s \not\approx s'$. Thus $\alpha(s, t) = 0$ and therefore $\beta((s, t), (s', t')) = \alpha(s, s') * \varrho(t) = 0$. □

# 3 PROBABILISTIC MODELS

We consider a model of probabilistic transition systems, containing probabilistic and nondeterministic choices as independent concepts.

## 3.1 Probabilistic Transition Systems

We assume a set $\mathcal{A}ct$ of *actions*, ranged over by $a$ and $b$.

**Definition 2.** *A Probabilistic Transition System (PTS) is a tuple* $\langle S, \longrightarrow, \pi_0\rangle$*, where*

- $S$ *is a non-empty finite set of* states*,*
- $\longrightarrow \subseteq S \times \mathcal{A}ct \times Dist(S)$ *is a finite* transition relation*, and*
- $\pi_0 \in Dist(S)$ *is an* initial distribution *on* $S$*.*

*We shall use* $s \xrightarrow{a} \pi$ *to denote that* $\langle s, a, \pi\rangle \in \longrightarrow$*. We use* $s \xrightarrow{a}$ *to denote that there is a* $\pi$ *such that* $s \xrightarrow{a} \pi$*, and* $s \overset{a}{\not\longrightarrow}$ *to denote that there is no* $\pi$ *such that* $s \xrightarrow{a} \pi$*. We say that a state* $s$ *is* terminal *(written* $s \not\longrightarrow$*) if* $s \overset{a}{\not\longrightarrow}$ *for all* $a \in \mathcal{A}ct$*.* □

This definition occurs with minor variations in [55, 57, 31, 38, 52, 51, 41]. In each state, a probabilistic transition system can perform a number of possible actions. Each action leads to a distribution over successor states. In many cases when it is understood from the context, we will identify a state $s$ with the distribution that assigns probability 1 to the state $s$.

An *initial state* of a process $\mathcal{P} = \langle S, \longrightarrow, \pi_0 \rangle$ is a state $s \in S$ such that $\pi_0(s) > 0$. A state $s$ is *reachable* in $\mathcal{P}$ if there is a sequence $s_0 s_1 \ldots s_n$ where $s_0$ is initial, $s = s_n$, and for each $0 \leq i < n$ there is a distribution $\pi_{i+1}$ such that $s_i \xrightarrow{a_i} \pi_{i+1}$ and $\pi_{i+1}(s_{i+1}) > 0$. A distribution $\pi \in \mathit{Dist}(S)$ is *reachable* in $\mathcal{P}$ if it is either the initial distribution or if $s \xrightarrow{a} \pi$ for some $a$ and state $s$ which is reachable in $\mathcal{P}$.

We use $s \xrightarrow{a}\!\!\rightsquigarrow s'$ to denote that there is a $\pi$ such that $s \xrightarrow{a} \pi$ and $\pi(s') > 0$, and $s \longrightarrow \rightsquigarrow s'$ to denote that there is an $a$ such that $s \xrightarrow{a}\!\!\rightsquigarrow s'$. A *finite process* is a process $\langle S, \longrightarrow, \pi_0 \rangle$ with a finite number of states, in which the relation $\longrightarrow\!\!\rightsquigarrow$ is acyclic.

A scheduler will decide which action should be taken in each state and then makes a probabilistic choice. Thus each scheduler corresponds to a *probabilistic execution* of a process. In the following sections, we shall study how to compare processes in terms of such executions.

### 3.2 Variants of Probabilistic Transition Systems

**The Reactive Model** In [40, 41, 54] a simple class of probabilistic transition systems is identified as the *reactive models*. It is the class of probabilistic transition systems where all states $s$ are *deterministic* in the sense that for each action $a$, whenever $s \xrightarrow{a} \pi_1$ and $s \xrightarrow{a} \pi_2$, then $\pi_1 = \pi_2$. Systems in the reactive models have the same structure as Markov Decision Processes [24].

**The Generative Model** A definition of a generative probabilistic transition system differs from the one given above in that each transition is an element in $S \times \mathit{Dist}(\mathcal{A}ct \times S)$, i.e., the probability distribution also includes a distribution over the possible actions. If all actions in a distribution are identical, we get the above definition of a probabilistic transition system[1].

Several researchers (e.g., [30, 17] have noticed that it is not trivial to define a symmetric parallel composition operator in the generative model. One source of difficulty is that in a generative model, a probabilistic transition system defines in each of its states a probability distribution

[1] Note that the above notion of generative model still allows some nondeterminism in the sense that a state may have transitions to several distributions. In [54], states are allowed at most one transition.

over a set of enabled actions. This view makes sense if the set of enabled actions is offered by, e.g., the environment. If two probabilistic transition systems are composed in parallel, then each of them defines a separate probability distribution over a set of enabled actions. It is not clear how the set of ”enabled actions” is to be defined, nor how the two probability distributions should be composed. Approaches to this problem can be found in, e.g., [17,15].

Another interpretation of a distribution over different actions in the generative model is that the choice is under control of the process itself. This could be the case if the actions are “output” actions, which communicate with corresponding “input actions” in a process that does not constrain the choice of the “outputting” process. A natural resulting model is then a probabilistic version of I/O automata, defined as *Probabilistic I/O automata* [56]. Probabilistic I/O automata use continuous time and rates to define transition probabilities. An model in a discrete-time framework that captures an analogous distinction between input and could, but not the time-dependent behavior modeled by the rates of Probabilistic I/O-automata, can be obtained from the model in Definition 2, as follows. For each input action, there is an enabled transition from each state. In each state, at most one output action may be enabled. The choice between different output actions must have been made in the previous transitions.

## 4 OPERATORS OF PROBABILISTIC PROCESS ALGEBRAS

In this section, we consider how operators can be defined in probabilistic process algebras. We will present a general framework that allows essentially any non-probabilistic process algebra to be extended to a probabilistic process algebra by introduction of a probabilistic (internal) choice operator. The operators of the given non-probabilistic process algebra may be lifted to operators in the probabilistic process algebra in a completely uniform way, under certain assumptions. Later we will instantiate the framework to particular process algebras.

Terms in the resulting probabilistic algebra will denote probabilistic processes. A set of terms can be used to form a probabilistic transition systems as in Definition 2. Some of the terms will correspond to states, and the others will correspond to distributions over states.

Terms of the non-probabilistic process algebra are assumed to be formed in the usual way by the constant NIL, and a number of operators, each with a certain arity. NIL denotes a state which has no outgoing transitions.

Also, the meaning of an $n$-ary operator *op* of the non-probabilistic process algebra, is assumed to be defined by a finite set of rules in the so-called de Simone format [23] (see also Aceto, Fokking and Verhoef, Chapter 1.3 in this issue):

$$\frac{p_{i_1} \xrightarrow{a_1} q_{i_1} \qquad \cdots \qquad p_{i_k} \xrightarrow{a_k} q_{i_k}}{op(p_1, \cdots, p_n) \xrightarrow{a} t} \tag{1}$$

where $p_1, \ldots, p_n$ and $q_{i_1}, \ldots, q_{i_k}$ are all distinct process expression variables, $a$ and $a_1, \ldots, a_k$ are actions, $\{p_{i_1}, \ldots, p_{i_k}\}$ is a subset of $\{p_1, \ldots, p_n\}$, and $t$ is a linear term[2] over the process expression variables $\{p_1, \ldots, p_n\} \setminus \{p_{i_1}, \ldots, p_{i_k}\} \cup \{q_{i_1}, \ldots, q_{i_k}\}$.

In the above rule (1), we say that the $i$th argument is *initally active* if $i$ is among $i_1, \ldots, i_k$. For an $n$-ary operator *op*, the $i$th argument is said to be *initially active* if it is initially active in any of the defining rules for *op*.

The probabilistic extension of the process algebra is obtained by an extension of the syntax allowing for distributions to be expressed. We introduce a single additional binary operator, *internal probabilistic choice* $\oplus_p$, parameterized by a real number $0 \leq p \leq 1$. The term $E \oplus_p F$ denotes a distribution which assigns the probability $p \cdot E(s) + (1-p) \cdot F(s)$ to each state $s$ (note that we regard $E$ and $F$ as distributions and $E(s)$ and $F(s)$ are probabilities assigned by $E$ and $F$ on state $s$ respectively). The choice is *internal* in the sense that the term denotes a probability distribution over states.

Now, terms of the extended process algebra are build using the original operators *op* together with the new additional operator $\oplus_p$. Our guiding principle for separating these terms into those denoting states and those denoting distributions over states is, that all probabilistic choices should be resolved before any non-deterministic transition is taken. Thus, terms denoting distributions may be inductively defined as either terms of the form $E \oplus_p F$, or terms of the form $op(t_1, \ldots, t_n)$, where for some initially active position $i$ the $t_i$ denotes a distribution. Formally this becomes:

**Definition 3.** *The set of process expressions that denote states is defined as the smallest set that satisfies:*

[2] In a linear term each variable occurs at most once.

- *$op(t_1, \cdots, t_n)$ denotes a state if all terms $t_{i_1}, \ldots, t_{i_k}$ in initially active positions denote states.*

*An arbitrary process expression $op(t_1, \cdots, t_n)$, in which $i_1, \ldots, i_k$ are the initially active positions, and where each $t_{i_j}$ denotes a distribution $[\![t_{i_j}]\!]$ over states, denotes a distribution, which assigns the probability $[\![t_{i_1}]\!](s_{i_1}) * \cdots * [\![t_{i_k}]\!](s_{i_k})$ to the state $op(t_1, \ldots, s_{i_1}, \ldots, s_{i_k}, \ldots, t_n)$, where $op(t_1, \ldots, s_{i_1}, \ldots, s_{i_k}, \ldots, t_n)$ is obtained from $op(t_1, \cdots, t_n)$ by replacing the terms $t_{i_1}, \ldots, t_{i_k}$ in initially active positions by states $s_{i_1}, \ldots, s_{i_k}$.*

To understand our general framework it may be instructive to consider the following examples:

- NIL is a process expression, which denotes a state without outgoing transitions.
- The prefixing operator is defined by the rule

$$\frac{}{a.p \xrightarrow{a} p}$$

  Thus the term $a.E$ has no initially active positions, and hence it denotes a state.
- The nondeterministic choice operator is defined by the rules

$$\frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p + q \xrightarrow{a} p'}$$

  Thus, both positions are initially active. Figure 1 illustrates the behaviour of the term $(a \oplus_{0.3} b) + (c \oplus_{0.4} d)$. Note that this term (due to the presence of $\oplus$ in active positions) denotes a distribution. In the figure, we use dotted lines to indicate probabilistic choices from distributions, and filled lines to indicate action transitions from states. Also, we are omitting trailing occurrences of NIL, writing $a$ for $a$.NIL.
- The operator for synchronous parallel composition a'la CSP may be defined by the following rule[3]:

$$\frac{p \xrightarrow{a} p' \quad q \xrightarrow{a} q'}{p || q \xrightarrow{a} p' || p'}$$

  Clearly, both positions are initially active. Figure 2 illustrates the behaviour of the parallel term (denoting a distribution) $(a \oplus_{0.3} b) || (a + b)$.

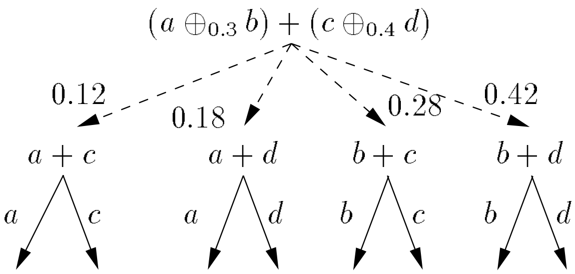

**Fig. 1.** Behaviour of $(a \oplus_{0.3} b) + (c \oplus_{0.4} d)$.

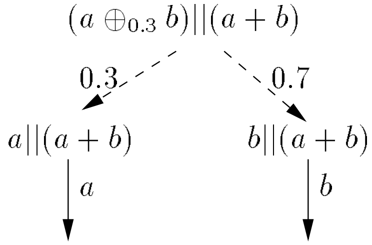

**Fig. 2.** Behaviour of $(a \oplus_{0.3} b)||(a + b)$.

As usual, process constants $P$ may be introduced by recursive definitions, $P \overset{\text{def}}{=} E$, with transitions of $P$ being inferred from its definition $E$.

Along similar lines, we can present a probabilistic version of SCCS, which is similar to the calculus PCCS presented by Giacalone et al [29]. As in SCCS, let $(Act, \times, 1, )$ be the Abelian group of *atomic actions*. Intuitively, actions of the form $\alpha \times \beta$ represent the simultaneous, atomic occurrence of the actions $\alpha$ and $\beta$. We will often write $\alpha\beta$ instead of $\alpha \times \beta$. The action 1 is the "idle action", and $\overline{\alpha}$ is the dual action of $\alpha$. The action $\alpha \times \overline{\alpha}$ represents a synchronized communication between complementary actions, and $\alpha \times \overline{\alpha} = 1$.

The syntax of PCCS is given by

$$E ::= \ \mathsf{nil} \mid \mathsf{X} \mid \alpha.\mathsf{E} \mid \mathsf{E} \oplus_{\mathsf{p}} \mathsf{F} \mid \mathsf{E} \times \mathsf{F} \mid \mathsf{E} \backslash \mathsf{A} \mid \mathsf{E}[\mathsf{f}] \mid \mathbf{rec}\mathsf{X}\ .\ \mathsf{E}$$

[3] We have chosen to illustrate a synchronous parallel operator, as we will need it later for introducing testing preorders. However, asynchronous parallel composition as in CCS may be extended in a similar manner.

| |
|---|
| $\dfrac{}{\alpha.P \xrightarrow{\alpha} P}$ |
| $\dfrac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\beta} Q'}{P \times Q \xrightarrow{\alpha\beta} P' \times Q'}$ |
| $\dfrac{P \xrightarrow{\alpha} P'}{P\backslash A \xrightarrow{\alpha} P'\backslash A} \quad (\alpha, \bar{\alpha} \notin A) \quad \dfrac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$ |

**Table 1.** Inference Rules for PCCS.

where $A$ is a subset of $Act$ such that $1 \in A$ and $f : Act \mapsto Act$ is a group morphism. Note the absence of nondeterministic choice in the above definition. The operational semantics of PCCS for action transitions is given in Table 1.

In [29], the presentation is slightly different in appearance. In the operational semantics, transitions are of the form $P \xrightarrow{\alpha[p]} P'$ meaning that "$P$ can perform the action $\alpha$ with probability $p$ and become process $P'$". In this representation, $P$ is a distribution over states, which assigns probability $p$ to the set of states that can perform $\alpha$ and become $P'$.

## 5 PROBABILISTIC BISIMULATION AND SIMULATION

In this section, we will present a series of bisimulation-preorders between probabilistic processes. These are intended to be generalizations of corresponding preorders on nonprobabilistic processes (see also Cleaveland and Sokolsky, Chapter 2.2 in this issue).

### 5.1 Probabilistic Bisimulation

A bisimulation can be understood as an equivalence relation, where two processes are equivalent if each process can mimick whatever the other process can do. We will first consider probabilistic bisimulation defined as an equivalence over states, not over distributions.

**Definition 4.** *An equivalence relation $R$ over a set of states $S$ is a* bisimulation *if $sRt$ implies that whenever $s \xrightarrow{a} \pi$ for some action $a$ and distribution $\pi$, then there is a distribution $\rho$ such that $t \xrightarrow{a} \rho$ and $\pi R \rho$.* □

In Figure 3 is an example of two bisimilar processes.

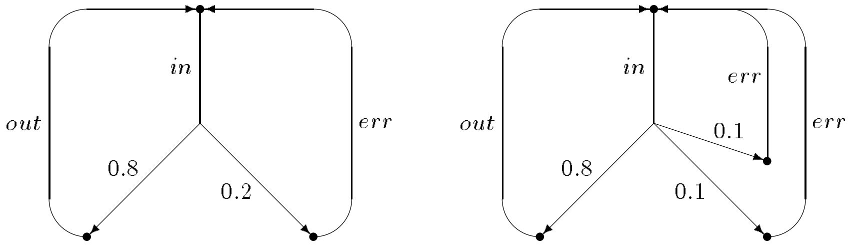

**Fig. 3.** Two bisimilar processes

In the process to the left, the action *in* leads to a distribution over two states. From the left state, the process can perform the action *out*, meaning that the action *in* has been performed normally. From the right state, only the action *err* can be performed, meaning that the previous action *in* was not successful. In the left process, the probability of an unsuccessful *in* action is 0.2. In the right process, the intention is to model two ways, represented by the states to the right, for the action *in* to be unsuccessful. Each of the states is reached with probability 0.1. In this process, both of these states are bisimilar, since they can perform equivalent actions, leading to equivalent states. Therefore, also the two start states (from which the action *in* can be performed) are equivalent, since they both lead to the successful equivalence class with probability 0.8 and to the unsuccessful equivalence class with probability 0.2).

The above definition of bisimulation was introduced by Larsen and Skou [41] for reactive systems and it is the most commonly used generalization of bisimulation to the probabilistic setting.

For models that embody both probabilistic and nondeterministic choice, in the way defined in this chapter, it can be argued that the above definition of probabilistic bisimulation is too restrictive. Namely, recall that we could let nondeterministic choices be resolved by schedulers, and that schedulers can be probabilistic. But the above definition of bisimulation considers only deterministic choices by schedulers. Consider for example the two processes of Figure 4.

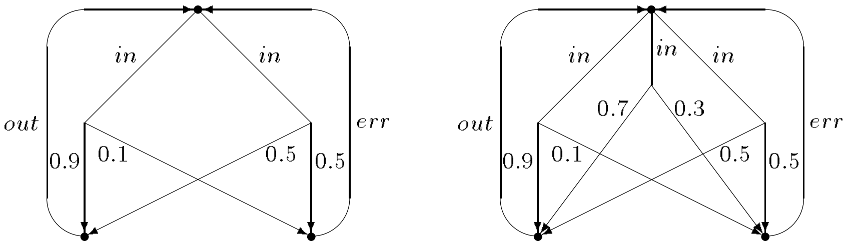

**Fig. 4.** Two probabilistic bisimilar processes

Note that ignoring the middle *in*-transition of the right process, these two processes are identical and the middle *in*-transition is a convex combination of the other two (see definition 5). Assuming that schedulers can be probabilistic, the extra transition can be generated as a probabilistic choice over the two other transitions. It therefore does not add anything to the process, meaning that the two processes ought to be equivalent. The definition of probabilistic bisimulation can thus be weakened by including combined transitions as follows:

**Definition 5.** *Let $t$ be a state of a probabilistic transition system and let $a$ be an action. We say that $t \xrightarrow{a} \rho$ is a* combined *transition of $t$ if $\rho$ is a convex combination of the set $\Pi = \{\rho_i | t \xrightarrow{a} \rho_i\}$ of distributions, that is, for each $\rho_i$, there is a nonnegative real number $\lambda_i$ such that $\sum_i \lambda_i = 1$ and $\rho = \sum_{\rho_i \in \Pi} \lambda_i * \rho_i$.* □

**Definition 6.** *An equivalence relation $R$ over $S$ is a* probabilistic bisimulation *if $sRt$ implies that whenever $s \xrightarrow{a} \pi$ for some action $a$ and distribution $\pi$, then there is a distribution $\rho$ such that $t \xrightarrow{a} \rho$ is a combined transition with $\pi R \rho$.*

*We use $\simeq$ to denote the largest probabilistic bisimulation i.e. the union of all probabilistic bisimulations over $S$ and $Dist(S)$.* □

Now we have a weaker equivalence relation than bisimulation. For example, consider the two processes shown in Figure 4. Note that the two processes are not bisimilar according to definition 4. But they are bisimilar according to definition 6 because the middle *in*-transition is a combined transition of the other two.

This definition was introduced by Segala and Lynch [52]. Following their terminology, we use the term *bisimulation* for the more restrictive definition, and the term *probabilistic bisimulation* for the definition which considers also combined transitions. With the definition of probabilistic bisimulation, we can use a combination of two or more transitions to denote a range of allowed probabilities for the outcome of an action, namely those that are in the convex closure of the outcomes of the actions actually described. In this way, it is similar to the use of intervals by Jonsson and Larsen [37].

## 5.2 Probabilistic Simulation

A simulation can be understood as a relation, which captures the idea that one of the processes can mimick whatever the other process can perform. In contrast to bisimulation, a simulation need not be symmetric, since the mimicking capability is required only of one of the processes.

**Definition 7.** *A preorder $R$ over $S$ is a simulation if $sRt$ implies that whenever $s \xrightarrow{a} \pi$ for some action $a$ and distribution $\pi$, then there is a distribution $\rho$ such that $t \xrightarrow{a} \rho$ and $\pi R^* \rho$.* □

As an example, consider again the two processes in Figure 4. The right process simulates the left one since it has more transitions. However, the left process does not simulate the right one . But as in the case of bisimulation, we can allow the matching transition $t \xrightarrow{a} \rho$ to be a combined transition. We can then use simulation to let a range of possible outcomes of an action be simulated by an even wider range of possible outcomes. Figure 5 shows a simple example of this. According to previous definition, the left process is not simulated by the right process. For example, its first *in*-transitions is not simulated by any of the *in* transitions of the right process. However, the *in*-transition is simulated by the convex combination of the two *in*-transitions of the right process, resulted by multiplying the two distributions by 3/4 and 1/4 respectively.

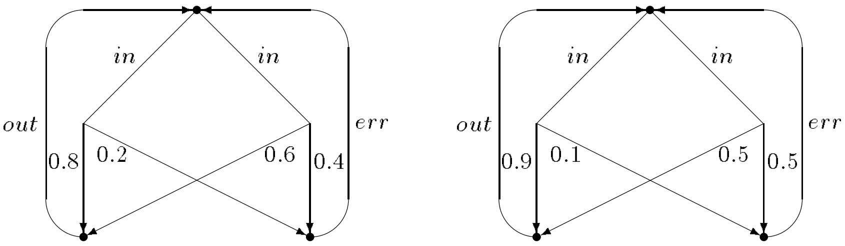

**Fig. 5.** Probabilistic simulation

It can thus be argued that a more appropriate definition of probabilistic simulation is the following:

**Definition 8.** *(Probabilistic Simulation) A preorder $R$ over $S$ is a probabilistic simulation if $sRt$ implies that whenever $s \xrightarrow{a} \pi$ for some action $a$ and distribution $\pi$, then there is a distribution $\rho$ such that $t \xrightarrow{a} \rho$ is a combined transition with $\pi R^* \rho$.*

*We use $\sim$ to denote the largest simulation i.e. the union of all probabilistic simulations over $S$ and $Dist(S)$.* □

For the nonprobabilistic case, we know that a bisimulation is a symmetric simulation and vice versa. This is also true in the probabilistic setting.

**Theorem 5.** *A probabilistic simulation is symmetric iff it is a probabilistic bisimulation.*

*Proof.* The result follows from Theorem 1.

### 5.3 Congruence Properties

The bisimulation and simulation relations that have been introduced in this section, can all be checked on finite-state processes by a standard iterative procedure that repeatedly refines the universal relation until it satisfies the appropriate definition of (bi-)simulation. At each step, the condition for (bi-)simulation may in general involve checking for inclusion between convex polytopes, spanned by different possible outcomes of a nondeterministic choice. Decision procedures for checking probabilistic (bi-)simulation can be found in e.g. [3, 4].

These relations may be used to reason about probabilistic processes by means of algebraic operators in a compositional manner.

A process expression may be generated by process constants, process variables and algebraic operators described in section 4. We define a process context to be a process expression containing free process variables. We use $\mathcal{C}$ to denote the set of process contexts. Let $C$ be a process context that contains free process variables $t_1 \ldots t_m$. Let $p_1 \ldots p_m$ be process expressions that denote states or distributions. We write $C[p_1/t_1 \ldots p_m/t_m]$ for the process expression $C$ in which all $t_i$ are substituted with $p_i$. As desired, the largest (bi-)simulation relations are all preserved by the algebraic operators described in section 4.

**Theorem 6.** *Let $C$ be a process context containing free process variables $t_1 \ldots t_m$. Let $p_1 \ldots p_m$ and $q_1 \ldots q_m$ be process expressions that may denote states or distributions.*

- *$p_i \simeq q_i$ for all $i$ implies that $C[p_1/t_i \cdots p_m/t_m] \simeq C[q_1/t_i \cdots q_m/t_m]$*
- *$p_i \sim q_i$ for all $i$ implies that $C[p_1/t_i \cdots p_m/t_m] \sim C[q_1/t_i \cdots q_m/t_m]$*

*Proof.* By the transitivity of $\sim$ and $\simeq$, the case when $m > 1$ can be transformed to the case of $m = 1$. For example, if $p[p_1/t_1, p_2/t_2] \sim p[q_1/t_1, p_2/t_2]$ and $p[q_1/t_1, p_2/t_2] \sim p[q_1/t_1, q_2/t_2]$, we have $p[p_1/t_1, p_2/t_2] \sim p[q_1/t_1, q_2/t_2]$. Thus, we consider only process contexts $C$ with a single variable $t$. Let $R = \{\langle C[p/t], C[q/t]\rangle \mid p \simeq q\}$. We will show that $R$ is a probabilistic bisimulation.

First we observe that, whenever $\pi \simeq \rho$, then $C[\pi/t] R C[\rho/t]$. Now $C$ will be of the form $op(\pi_1, \ldots, \pi_n, t, p_1, \ldots, p_k)$ for some derived operator $op$, with $\pi_1, \ldots, \pi_n$, and possibly $t$, denoting the initially active arguments. Assuming $t$ is initially active in $C$, $C[\pi/t]$ denotes the distribution, which assigns the probability $\pi(q) * \pi_1(q_1) * \cdots * \pi_n(q_n)$ to the state $op(q_1, \ldots, q_n, q, p_1, \ldots, p_k)$. Similarly, $C[\rho/t]$ assigns the probability $\rho(q) * \pi_1(q_1) * \cdots * \pi_n(q_n)$ to the same state. As $\pi \simeq \rho$ it follows that $C[\pi/t]$ and $C[\rho/t]$ assign the same probabilities to equivalence classes of

$R$, which are sets of the form $\{D[p/t] \,|\, p \simeq p_0\}$ for some state $p_0$. Thus, as claimed, $C[\pi/t]\ R\ C[\rho/t]$.

Now let $\langle C[p/t], C[q/t]\rangle \in R$ and assume that $C[p/t] \xrightarrow{b} \pi$. We must find a matching, combined transition for $C[q/t]$. The transition of $C[p/t]$ will be of one of the two following forms:

$i)$ $\pi = C'[p/t]$
$ii)$ $\pi = C'[\pi'/t]$ where $t$ is active in $C$ and $p \xrightarrow{a} \pi'$ for some $a$

In case $i)$, the "de Simone"-format allows us to infer that $C[q/t] \xrightarrow{b} C'[q/t]$. Since $p \simeq q$ (both as states and singleton distributions), it follows from our first observation that $C'[p/t]RC'[q/t]$.

In case $ii)$, it follows that $\pi' \simeq \rho'$ for some distribution $\rho'$, where $q \xrightarrow{a} \rho'$ is a combined transition (as $p \simeq q$). Now, due to the "de Simone"-format, it may be argued that also $C[q/t] \xrightarrow{b} C'[\rho'/t]$ is a combined transition. From our first observation, it follows that $C'[\pi'/t]\ R\ C'[\rho'/t]$.

As $R$ is symmetric, this suffices to demonstrate that $R$ is a probabilistic bisimulation. The case of simulation can be established by similar argument using Theorem 4. □

## 6 TESTING PREORDERS

In the previous section, we have defined preorders which are based on relating states or distributions to each other. For non-probabilistic systems, there is another rich family of preorders, which are based on the comparison between execution sequences. These preorders can be based on traces, refusals, ready-sets, behaviors, etc. In this section, we are going to give a partial answer to the question how these preorders generalize to a setting with probabilistic processes. However, we are not going to define first some analogue of e.g., traces. The reason is that such an analogue would not lead to a compositional preorder, as shown e.g., by Segala [51]. Instead, we will use another method for arriving at a weaker compositional preorder. We will use the framework of testing, as defined by de Nicola and Hennessy [22]. The idea of this framework is that processes are compared by their ability to pass a specified set of "tests".

We will generalize the framework of may-testing to probabilistic processes, and then characterize the resulting preorder. It will turn out that the resulting preorder will be rather different from the nonprobabilistic trace inclusion. In fact, it will be closer to the nonprobabilistic simulation preorder (in fact, it reduces to simulation in the nonprobabilistic case), for the reason that in the framework we will define, the probabilistic choices will induce an effect which is similar to "copying" of a process state. In this section, we give a simplified presentation by limiting tests to be finite processes.

### 6.1 Related Work

Testing-based preorders of probabilistic processes have also been studied by Christoff [10] and by Cleaveland, Smolka, and Zwarico [15] and by Yuen et al. [49,49,56]. These works consider a pure probabilistic model [54], and therefore their preorders do not capture the notion of refinement in the sense of being "less nondeterministic". On the other hand, they can be efficiently checked in the finite-state case, as demonstrated by Christoff and Christoff [11], using the polynomial-time algorithm for checking equivalence between probabilistic automata by Tzeng [53]. The work which is closest to the current one is by Segala [51], who define essentially the same testing preorders as in this work. However, Segala does not develop an explicit characterization of the testing preorder.

### 6.2 Tests, Testing Systems and Preorders

Following Wang and Larsen [57], we define tests as finite trees with a certain subset of the terminal states being "accepting states".

**Definition 9.** *A* (probabilistic) test *is a tuple* $\langle T, \longrightarrow, \rho_0, \mathcal{F}\rangle$*, where* $\langle\langle T, \longrightarrow\rangle, \rho_0\rangle$ *is a finite tree, and* $\mathcal{F} \subseteq T$ *is a set of* success-states*, each of which is terminal.* □

A test $\mathcal{T}$ is applied to a process $\mathcal{P}$ by putting the process $\mathcal{P}$ in parallel with the test $\mathcal{T}$ and observing whether the test reaches a success state.

We define a testing system as the parallel composition of a process and a test.

**Definition 10.** *Let* $\mathcal{P} = \langle S, \longrightarrow, \pi_0\rangle$ *be a process and* $\mathcal{T} = \langle T, \longrightarrow, \rho_0, \mathcal{F}\rangle$ *be a test. The composition of* $\mathcal{P}$ *and* $\mathcal{T}$*, denoted* $\mathcal{P}\|\mathcal{T}$ *is a so-called* testing system*, defined as the process* $\langle S, \longrightarrow, \pi_0\rangle \| \langle\langle T, \longrightarrow\rangle, \rho_0\rangle$ *with success states* $S \times \mathcal{F}$*.* □

Our intention is that a testing system defines a probability of reaching a success-state. However, since from each state there may be several outgoing transitions, such a probability is not uniquely defined. We will be interested in the maximal probabilities of success. These can be defined inductively on the structure of the testing system.

**Definition 11.** *Let $\mathcal{P}\|\mathcal{T}$ be a testing system, composed of the process $\mathcal{P} = \langle S, \longrightarrow, \pi_0\rangle$ and the test $\mathcal{T} = \langle T, \longrightarrow, \rho_0, \mathcal{F}\rangle$. For each state $s\|t$ of $\mathcal{P}\|\mathcal{T}$ we define its* maximal probability of success*, denoted $t\lceil s\rceil$ and its* minimal probability of success*, denoted $t\lfloor s\rfloor$, inductively by*

- *If $s\|t$ is terminal, then $t\lceil s\rceil = t\lfloor s\rfloor = 1$ if $t$ is a success-state, else $t\lceil s\rceil = t\lfloor s\rfloor = 0$.*
- *If $s\|t$ is not terminal, then*

$$t\lceil s\rceil = \underset{s\|t \xrightarrow{a} \pi\times\rho}{\mathit{max}} \left(\sum_{s'\|t'} (\pi\times\rho)(s'\|t') * t'\lceil s'\rceil\right)$$

*and*

$$t\lfloor s\rfloor = \underset{s\|t \xrightarrow{a} \pi\times\rho}{\mathit{min}} \left(\sum_{s'\|t'} (\pi\times\rho)(s'\|t') * t'\lfloor s'\rfloor\right)$$

*For a distribution $\pi$ on $S$ and a distribution $\rho$ on $T$, we define*

$$\rho\lceil\pi\rceil = \sum_{s\|t} (\pi\times\rho)(s\|t) * t\lceil s\rceil$$

*and*

$$\rho\lfloor\pi\rfloor = \sum_{s\|t} (\pi\times\rho)(s\|t) * t\lfloor s\rfloor$$

*We define $\mathcal{T}\lceil\mathcal{P}\rceil = \pi_0\lceil\rho_0\rceil$. and $\mathcal{T}\lfloor\mathcal{P}\rfloor = \pi_0\lfloor\rho_0\rfloor$.* □

We note that, using the definition of $\rho\lceil\pi\rceil$, we can make a simpler definition of $t\lceil s\rceil$ as

$$t\lceil s\rceil = \max_{s\|t \xrightarrow{a} \pi\times\rho} \rho\lceil\pi\rceil$$

We now define preorders of testing, which abstract from the set of possible expected outcomes when testing a process $\mathcal{P}$ by a test $\mathcal{T}$: *may* testing considers only maximal possible expected outcome of $\mathcal{P}\|\mathcal{T}$ and *must* testing considers only minimal possible outcome.

**Definition 12.** *Given two processes $\mathcal{P}$ and $\mathcal{Q}$, we define*

1. $\mathcal{P} \sqsubseteq_{may} \mathcal{Q}$ *if* $\forall \mathcal{T} : \mathcal{T}\lceil \mathcal{P} \rceil \leq \mathcal{T}\lceil \mathcal{Q} \rceil$
2. $\mathcal{P} \sqsubseteq_{must} \mathcal{Q}$ *if* $\forall \mathcal{T} : \mathcal{T}\lfloor \mathcal{P} \rfloor \leq \mathcal{T}\lfloor \mathcal{Q} \rfloor$

□

The intention, for example, behind the definition of $\sqsubseteq_{may}$ is that intuitively, $\mathcal{P} \sqsubseteq_{may} \mathcal{Q}$ should means that $\mathcal{P}$ refines $\mathcal{Q}$ with respect to "safety properties". We can regard the success-states of a test as states defining when the tester has observed some "bad" or "unacceptable" behavior. A process then refines another one if it has a smaller potential for "bad behavior" with respect to any test. In the definition of $\mathcal{P} \sqsubseteq_{may} \mathcal{Q}$, this means that the maximal probability of observing bad behavior of $\mathcal{P}$ should not exceed the maximal probability of observing bad behavior of $\mathcal{Q}$.

For example, consider process $P$ and $Q$ in Figure 6. The probability that $P$ may pass a test is always less or equal to the probability $Q$ may pass the same test; therefore $P \sqsubseteq_{may} Q$.

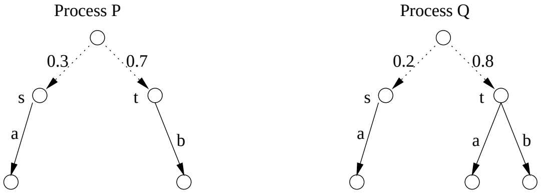

**Fig. 6.** $P \sqsubseteq_{may} Q$.

A useful property of $\sqsubseteq_{may}$ is that it is compositional in the sense that they are precongruences with respect to our parallel composition operator.

**Proposition 1.** *For arbitrary processes $P, Q, R$,*

1. $P \sqsubseteq_{may} Q$ *implies* $P||R \sqsubseteq_{may} Q||R$
2. $P \sqsubseteq_{must} Q$ *implies* $P||R \sqsubseteq_{must} Q||R$

□

### 6.3 Characterization of Testing Preorders

In the following, we show that testing preorders defined in Definition 12 can be characterized by variants of probabilistic simulation. When restricted to nonprobabilistic processes, probabilistic may-testing preorder coincides with ordinary simulation; whereas probabilistic must-testing preorder with refusal simulation [36]. We shall only consider may-testing. It may seem a little surprising that a preorder defined in terms of testing, which is a "linear-time" activity, is characterized by a simulation relation, which is a "branching-time" relation. The explanation is that the probabilistic choices of tests have the effect of "copying" the process under test into a number of copies, and that the testing of each copy is performed independently [1].

Recall that a probabilistic process is essentially a distribution over states. Such a distribution gives rise to a number of possibilities for choosing the next action and next distribution. We will capture these possibilities in a notion of *step* corresponding to a transition in the non-probabilistic setting.

**Definition 13.** *Let $\langle S, \longrightarrow \rangle$ be a probabilistic transition system. A* step *is a weighting on $\mathcal{A}ct \times Dist(S)$. We say that a step $\phi$ is a step of distribution $\pi$ on $S$ if there is a function $h : (\mathcal{A}ct \times Dist(S)) \mapsto S$ such that*

- *$s \xrightarrow{a}$ whenever $s$ is in the support of $h(\langle a, \pi' \rangle)$, and*
- *$(\sum_{h(\langle a,\pi' \rangle)=s} \phi(\langle a, \pi' \rangle)) \leq \pi(s)$ for each $s \in S$.*

*We say that a step $\phi$ is an $a$-*step *if $a' = a$ for all $\langle a', \pi' \rangle$ in the support of $\phi$.* □

Intuitively, a step represents a combination of next transitions that can be made by a process. A step from a distribution $\pi$ is a weighting over possible outgoing transitions, which is consistent with $\pi$ in the sense that it can be obtained by choosing for each state in the support of $\pi$ a subdistribution over outgoing transitions. Note that for a given distribution, there may be infinitely many steps possible. A step is *normal* if the function $h$ in Definition 13 can be chosen such that for each nonterminal $s$ in the support of $\pi$, there is a $\langle a, \pi' \rangle \in \phi$ such that $h(\langle a, \pi' \rangle) = s$ and $\pi(h(\langle a, \pi' \rangle)) = \phi(\langle a, \pi' \rangle)$.

That is, a normal step is obtained by choosing a unique transition from each state, satisfying the above condition. Since each state in a distribution in general has several outgoing transitions, there are many (but finitely many) normal steps from each distribution.

We define *post* on steps by

$$post(\phi) = \sum_{\langle a,\pi' \rangle} \phi(\langle a, \pi' \rangle) * \pi'$$

i.e., $post(\phi)$ is the weighting obtained by projecting a step onto the "next" distribution in its transitions. The notion of post weighting is analogous to the notions of next state in the non-probabilistic setting. We can now define the notion of step-simulation between weightings.

**Definition 14 (Probabilistic Step-Simulation).** *Let $\langle S, \longrightarrow \rangle$ and $\langle T, \longrightarrow \rangle$ be two probabilistic transition systems. A relation $\lhd \subseteq (Weight(S) \times Weight(T))$ between weightings on $S$ and weightings on $T$ is a* probabilistic step-simulation *if $\pi \lhd \rho$ implies that*

- *$\pi(S) \leq \rho(T)$, and*
- *for each normal step $\phi$ from $\pi$ there is a step $\psi$ from $\rho$ and a function $h : support(\phi) \mapsto Weight(Act \times Dist(R))$ from pairs $\langle a, \pi' \rangle$ in the support of $\phi$ to steps from $\rho$ such that*
  - *$h$ maps each $\langle a, \pi' \rangle$ to an $a$-step from $\psi$,*
  - *$h(\phi) \leq \psi$, i.e., the image of $\phi$ under $h$ is "covered" by $\psi$, and*
  - *for each pair $\langle a, \pi' \rangle$ in the support of $\phi$ we have*

$$\pi' \quad \lhd \quad post(h(\langle a, \pi' \rangle))$$

*For two probabilistic processes $\mathcal{P} = \langle S, \longrightarrow, \pi_0 \rangle$ and $\mathcal{Q} = \langle T, \longrightarrow, \rho_0 \rangle$, we say that $\mathcal{P}$* is simulated by *$\mathcal{Q}$ if there is a probabilistic step-simulation $\lhd$ between $\langle S, \longrightarrow \rangle$ and $\langle T, \longrightarrow \rangle$ such that $\pi_0 \lhd \rho_0$.* □

Intuitively, a weighting $\pi$ is simulated by a weighting $\rho$ if the total "mass" of $\pi$ is at most that of $\rho$ (first condition), and if each step $\phi$ from $\pi$ can be simulated by a step $\psi$ from $\rho$ in the sense that each "next transition" $\langle a, \pi' \rangle$ in the support of $\phi$ can be covered by an $a$-step from $\rho$, such that the weighted sum (weighted wrp. to $\phi$) of all the weightings $h(\langle a, \pi' \rangle)$ is covered by $\psi$, and such that $\pi'$ is simulated by the next-state distribution obtained from $h(\langle a, \pi' \rangle)$. In Figure 7, we illustrate why process $P$ is simulated by process $Q$. Note that $P \sqsubseteq_{may} Q$ as shown in Figure 6.

The following characterization theorem was stated and proven by Jonsson and Yi in [39].

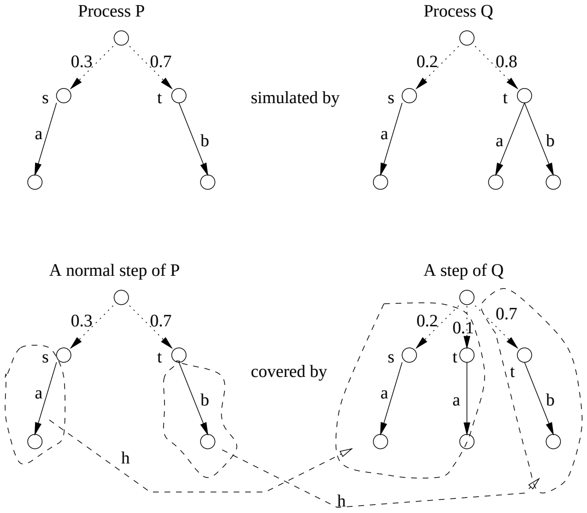

**Fig. 7.** A proof for $P \lhd Q$.

**Theorem 7.** *$\pi \sqsubseteq_{may} \rho$ if and only if there is a probabilistic step-simulation $\lhd$ such that $\pi \lhd \rho$.* □

# 7 PROBABILISTIC LOGICS

## 7.1 Characterizing Preorders

For non-probabilistic transition systems, several behavioural preorders have been characterized by simple modal logics. That is, the particular preorder has been shown to be completely captured by inclusion between the sets of logical properties satisfied by states (see also Bradfield and Stirling, Chapter 1.4 in this issue).

Bisimilarity between non-probabilistic transition systems is characterized by the so-called Hennessy-Milner Logic [33]. In [40–42], this characterization has been extended to probabilistic bisimulation for reactive probabilistic systems by identification of a suitable probabilistic extension of Hennessy-Milner Logic. In [9, 25, 26], a further generalization to non-discrete probabilistic systems (Labelled Markov Processes) is given, and in particular it is shown that probabilistic bisimulation can be characterized by a very weak, negation-free modal logic. In addition, and in contrast to [40–42], the characterization offered by [9, 25, 26] requires no finite branching assumptions.

Here, we adapt the probabilistic modal logic, PML, of [40–42] to the probabilistic model studied in this paper. As for states, the formulas of the logic come in two flavours: non-deterministic formulas (ranged over by $F, F_i, \ldots$) and probabilistic formulas (ranged over by $\varphi, \varphi_i, \ldots$), given by the following abstract syntax (where $p \in [0,1]$):

$$\begin{array}{rcl} F & ::= & tt \mid F_1 \wedge F_2 \mid \neg F \mid \langle a \rangle \varphi \\ \varphi & ::= & tt \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid \Diamond_p F \end{array}$$

The interpretation is relative to a probabilistic transition system, $\langle S, \longrightarrow, \pi_0 \rangle$. More precisely, the semantics of a formula $F$ (respectively $\varphi$) is a set of non-deterministic states $[\![F]\!]$ (respectively of probabilistic states $[\![\varphi]\!]$), defined inductively as follows:

$$\begin{array}{rlrl} i) & [\![tt]\!] = S & ii) & [\![F_1 \wedge F_2]\!] = [\![F_1]\!] \cap [\![F_2]\!] \\ iii) & [\![\neg F]\!] = S \backslash [\![F]\!] & iv) & [\![\langle a \rangle \varphi]\!] = \{ s \mid \exists a, \pi \in [\![\varphi]\!].\, s \xrightarrow{a} \pi \} \end{array}$$

$$\begin{array}{rlrl} i') & [\![tt]\!] = Dist(S) & ii') & [\![\varphi_1 \wedge \varphi_2]\!] = [\![\varphi_1]\!] \cap [\![\varphi_2]\!] \\ iii') & [\![\neg \varphi]\!] = Dist(S) \backslash [\![\varphi]\!] & iv') & [\![\Diamond_p F]\!] = \{ \pi \mid \pi([\![F]\!]) \geq p \} \end{array}$$

For $F$ a non-deterministic property, we write $[\![F]\!]_C$ to denote the set of non-deterministic states satisfying $F$, when we use the combined transition relation in *iv)*. For a state $s$ we denote by $Sat(s)$ and $Sat_C(s)$ the set of properties satisfied by $s$ with respect to $[\![]\!]$ and $[\![]\!]_C$. Bisimulation and probabilistic bisimulation as presented in Section 5 may now be characterized as follows:

**Theorem 8.** *Let $\langle S, \longrightarrow, \pi_0 \rangle$ be an image-finite[4] probabilistic transition system. Then two states $s, t$ are bisimilar respectively probabilistic bisimilar if and only if $Sat(s) = Sat(t)$ respectively $Sat_C(s) = Sat_C(t)$.*

Let $NSat(s)$ and $NSat_C(s)$ denote the set of negation-free properties satisfied by the state $s$ with respect to $[\![]\!]$ and $[\![]\!]_C$.. Then simulation and probabilistic simulation as presented in Section 5 are characterized as follows:

[4] A probabilistic transition system $\langle S, \longrightarrow, \pi_0 \rangle$ is said to be image-finite if for all reachable non-deterministic states $s$, the set $\{ \pi \mid \exists a, s.s \xrightarrow{a} \pi \}$ is finite, and for all reachable probabilistic states $\pi$, the set $\{ s \mid \pi(s) > 0 \}$ is finite.

**Theorem 9.** *Let $\langle S, \longrightarrow, \pi_0 \rangle$ be an image-finite probabilistic transition system. Then two states $s, t$ are in the simulation preorder respectively probabilistic simulation preorder if and only if $NSat(s) \subseteq NSat(t)$ respectively $NSat_C(s) \subseteq NSat_C(t)$.*

As an example reconsider Figure 4. Here the two processes are probabilistic bisimilar but not bisimilar. The lack of bisimilarity may, according to Theorem 8, be 'explained' by a distinguishing property, e.g.:

$$\langle in \rangle (\Diamond_{0.7} \langle out \rangle tt \wedge \Diamond_{0.3} \langle err \rangle tt)$$

which is satisfied by the right process but not by the left processes (with respect to $[\![\,]\!]$). The above property is clearly negation-free, thus according to Theorem 9, this also demonstrates that the right process is not even simulated by the left one.

Reconsidering Figure 5, the property

$$\langle in \rangle \Diamond_{0.9} \langle out \rangle tt$$

is satisfied by the right process but not by the left processes (with respect to $[\![\,]\!]_C$). Thus the right process is not probabilistically simulated by the left one.

### 7.2 Model Checking Probabilistic Temporal Logics

Several probabilistic extensions of temporal logics such as CTL and CTL$^*$ [13, 14] have been suggested for the formal specification of probabilistic properties of systems. In addition, associated model checking algorithms have in many cases been offered.

The first probabilistic extension of branching-time logics for expressing properties of probabilistic system was proposed by Hansson and Jonsson [31, 32]. Formulas of the resulting logic PCTL are obtained by adding subscripts and superscripts to CTL formulas, as in $\Diamond^{\leq 15}_{\geq 0.6} \varphi$, which expresses that the property $\varphi$ will hold within 15 transition-steps with probability at least 0.6. The presented model-checking algorithms rely on results on Markov chains and dynamic programming. Later, the logic PCTL was extended to systems including non-determinism by Hansson in [30] and Segala and Lynch in [52]. Christoff and Christoff [12] adapt a restricted form of the modal mu-calculus. A new probabilistic semantics for the mu-calculus has been developed by Narasimha et al. [45].

Aziz et al [2] introduces pCTL$^*$ a probabilistic extension of CTL$^*$. Here the model checking algorithm is based on early results due to Courcoubetis and Yannakakis [16]. The logic was later extended to systems with non-determinism by Bianco and de Alfaro [8, 21]. Symbolic model-checking algorithms of these logics was presented by Baier et al. in [6].

For more detailed information on the interesting topic of model checking probabilistic systems, we refer the reader to the excellent works by Alfaro [20] and Baier [4].

## 8 CONCLUSION and TRENDS

In this chapter, we have dealt with a number of classical process algebraic issues in a rather general setting allowing both discrete probabilistic choice as well as nondeterminism. In particular, we have

- shown how non-probabilistic process algebraic operators, in a uniform manner, may be extended to this probabilistic setting;
- offered a range of probabilistic extensions of well-known behavioural preorders such as simulation, bisimulation and testing;
- established congruence properties for these preorders;
- provided alternative characterizations of the probabilistic preorders, either in terms of "trace"- or "tree"-based denotational models, or in terms of probabilistic modal logics.

Current research considers further extensions of the process algebraic framework to settings with continuous-time [35, 34]. One goal is to combine the contributions of process-algebra, viz. compositionality and the use of logics to specify and analyze properties, with the work on efficient algorithms for analyzing performance of stochastic processes [5, 18]. Also, the basic notions of probabilistic bisimulation and probabilistic modal logics have been recast and analysed in settings with continuous-space probability distributions [9, 25, 19, 27, 28]. From this work it may be inferred that the negation-free version of the logic in section 7 suffices in order to characterize probabilistic bisimulation. In this chapter we have not considered the difficult problem of extending probabilistic bisimulation to allow for abstraction from internal computation (resulting in probablistic versions of weak bisimulation). For the most promising suggestions in this direction we refer the reader to [7].

## References

1. Samson Abramsky. Observation equivalence as a testing equivalence. *Theoretical Computer Science*, 53(2,3):225–241, 1987.
2. A. Aziz, V. Singhal, F. Balarin, R.K. Brayton, and A.L. Sangiovanni-Vincentelli. It usually works: the temporal logic of stochastic systems. *Lecture Notes in Computer Science*, 939, 1995. In proceedings of CAV'95.
3. C. Baier. Polynomial time algorithms for testing probabilistic bisimulation and simulation. In R. Alur and T. Henzinger, editors, *Proc. $8^{th}$ Int. Conf. on Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 38–49, New Brunswick, USA, 1996. Springer Verlag.
4. C. Baier. On algorithmic verification methods for probabilistic systems. Habilitation, thesis, University of Mannheim, 1999.
5. C. Baier, J.-P. Katoen, and H. Hermanns. Approximate symbolic model checking of continuous-time markov chains. In *Proc. CONCUR '99, $9^{th}$ Int. Conf. on Concurrency Theory*, volume 1664 of *Lecture Notes in Computer Science*, pages 146–162, 1999.
6. C. Baier, M. Kwiatkowska, M. Ryan, E. Clarke, and V. Hartonas Garmhausen. Symbolic model checking for probabilistic processes. *Lecture Notes in Computer Science*, 1256, 1997. In proceedings ICALP'97.
7. Christel Baier and Holger Hermanns. Weak bisimulation for fully probabilistic processes. *Lecture Notes in Computer Science*, 1254, 1997.
8. A. Bianco and L. De Alfaro. Model checking of probabilistic and nondeterministic systems. *Lecture Notes of Computer Science*, 1026, 1995. In proceedings of FSTTCS.
9. R. Blute, J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled markov processes. *Logic in Computer Science*, 1997. In Proceedins of the Twelfth IEEE Symposium on Logic in Computer Science.
10. I. Christoff. Testing equivalences and fully abstract models for probabilistic processes. In Baeten, editor, *Proc. CONCUR, Amsterdam*, volume 458 of *Lecture Notes in Computer Science*, pages 126–140. Springer Verlag, 1990.
11. L. Christoff and I. Christoff. Efficient algorithms for verification of equivalences for probabilistic processes. In Larsen and Skou, editors, *Proc. Workshop on Computer Aided Verification*, volume 575 of *Lecture Notes in Computer Science*. Springer Verlag, 1991.
12. L. Christoff and I. Christoff. Reasoning about safety and liveness properties for probabilistic properties. In R.K. Shyamasundar, editor, *Foundations of Software Technology and Theoretical Computer Science*, volume 652 of *Lecture Notes in Computer Science*, pages 342–355. Springer Verlag, 1992.
13. E. M. Clarke, E.A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logics specification: A practical approach. In *Proc. $10^{th}$ ACM Symp. on Principles of Programming Languages*, pages 117–126, 1983.
14. E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In D. Kozen, editor, *Proc. IBM workshop on*

*Logics of Programs*, volume 131 of *Lecture Notes in Computer Science*, 1982. also as Aiken Computation Lab TR-12-81, Harvard University 1981.

15. R. Cleaveland, S. Smolka, and A. Zwarico. Testing preorders for probabilistic processes. In *Proc. ICALP '92*, 1992.
16. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. In *Proc.* $29^{th}$ *Annual Symp. Foundations of Computer Science*, pages 338–345, 1988.
17. P. R. D'Argenio, H. Hermanns, and J-P. Katoen. On generative parallel composition. In *Proc. Workshop on Probabilistic Methods in Verification*, 1998.
18. P. R. D'Argenio, J-P. Katoen, and E. Brinksma. General purpose discrete-event simulation using SPADES. In *Proc. 6th Int. Workshop on Process Algebra and Performance Modelling.*, 1998.
19. P. d'Arginio, J. Katoen, and E. Brinksma. An algebraic approach to the specification of stochastic systems. In *PROCOMET'98*. Chapmann, Hall, 1998.
20. L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Dept. of Computer Sciences, Stanford University, Dec. 1997.
21. L. de Alfaro. Temporal logics for the specificaiton of performance and reliability. *Lecture Notes in Computer Science*, 1200, 1997. In proceedings of STACS'97.
22. R. de Nicola and M. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.
23. R de Simone. Higher-level synchronising devices in MEIJE-SCCS". *Theoretical Computer Science*, 37(3):245–267, 1985.
24. C. Derman. *Finite State Markovian Decision Processes*. Academic Press, 1970.
25. J. Desharnais, A. Edalat, and P. Panangaden. A logical characterization of bisimulation for labeled markov processes. *Logic in Computer Science*, 1998. In proceedings of the 13th IEEE Symposium on Logic in Computer Science.
26. J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled markov processes. In Proceedins of the 12th IEEE Symposium on Logic in Computer Science. To appear in Information and Computation, 1997, 2000.
27. Josee Desharnais, Abbas Edalat, and Prakash Panangaden. A logical characterization of bisimulation for labeled markov processes. *Logic in Computer Science*, 1998.
28. Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labeled markov systems. *Lecture Notes in Computer Science*, 1664, 1999.
29. A. Giacalone, C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proc. IFIP TC2 Working Conference on Programming Concepts and Methods*, Sea of Galilee, April 1990.
30. H. Hansson. *Time and Probabilities in Formal Design of Distributed Systems*. Real-Time Safety Critical Systems. Elsevier, 1994.
31. H. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *Proc.* $11^{th}$ *IEEE Real -Time Systems Symposium*, Orlando, Florida, 1990.
32. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
33. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.
34. H. Hermans. *Interactive Markov Chains*. PhD thesis, University of Erlangen-Nürnberg, 1998.

35. J. Hillston. *A Compositional Approach to Performance Modelling.* PhD thesis, University of Edinburgh, 1994. Published in Cambridge University Press (1996).
36. B. Jonsson. Simulations between specifications of distributed systems. In *Proc. CONCUR '91, Theories of Concurrency: Unification and Extension*, volume 527 of *Lecture Notes in Computer Science*, Amsterdam, Holland, 1991. Springer Verlag.
37. B. Jonsson and K. Larsen. Specification and refinement of probabilistic processes. In *Proc.* $6^{th}$ *IEEE Int. Symp. on Logic in Computer Science*, Amsterdam, Holland, July 1991.
38. B. Jonsson and W. Yi. Compositional testing preorders for probabilistic processes. In *Proc.* $10^{th}$ *IEEE Int. Symp. on Logic in Computer Science*, pages 431–441, 1995.
39. B. Jonsson and W. Yi. Testing preorders for probabilistic processes can be characterized by simulations. Technical report, DoCS, Uppsala University, January 2000. Submitted.
40. K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. In *Proc.* $16^{th}$ *ACM Symp. on Principles of Programming Languages*, 1989.
41. K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Control*, 94(1):1–28, 1991.
42. K.G. Larsen and A. Skou. Compositional verification of probabilistic processes. In Cleaveland, editor, *Proc. CONCUR '92, Theories of Concurrency: Unification and Extension*, volume 630 of *Lecture Notes in Computer Science*. Springer Verlag, 1992.
43. R. Milner. *Communication and Concurrency.* Prentice-Hall, 1989.
44. M.K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Trans. on Computers*, C-31(9):913–917, Sept. 1982.
45. M. Narasimha, R. Cleaveland, , and P. Iyer. Probabilistic temporal logics via the modal mu-calculus. In W. Thomas, editor, *Foundations of Software Science and Computation Structures*, volume 1578 of *Lecture Notes in Computer Science*. Springer Verlag, 1999.
46. G. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, Denmark, 1981.
47. A. Pnueli and L. Zuck. Verification of multiprocess probabilistic protocols. *Distributed Computing*, 1(1):53–72, 1986.
48. S. Purushothaman and P.A. Subrahmanyam. Reasoning about probabilistic behavior in concurrent systems. *IEEE Trans. on Software Engineering*, SE-13(6):740–745, June 1989.
49. S.A. Smolka R. Cleaveland, Z. Dayar and S. Yuen. Testing preorders for probabilistic processes. *Information and Computation*, 2(152):93–148, November 1999.
50. M.O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.
51. R. Segala. A compositional trace-based semantics for probabilistic automata. In *Proc. CONCUR '95,* $6^{th}$ *Int. Conf. on Concurrency Theory*, volume 962 of *Lecture Notes in Computer Science*, pages 234–248. Springer Verlag, 1995.
52. R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.
53. W-G. Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM J. Computing*, 21(2):216–227, Apr. 1992. To Appear.
54. R. van Glabbeek, S.A. Smolka, B. Steffen, and C. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Proc.* $5^{th}$ *IEEE Int. Symp. on Logic in Computer Science*, pages 130–141, 1990.
55. M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc.* $26^{th}$ *Annual Symp. Foundations of Computer Science*, pages 327–338, 1985.

56. S.-H. Wu, S.A. Smolka, and E.W. Stark. Composition and behaviors of probabilistic I/O-Automata. *Theoretical Computer Science*, 176(1-2):1–37, 1997.
57. Wang Yi and K. Larsen. Testing probabilistic and nondeterministic processes. In *Protocol Specification, Testing, and Verification XII*, 1992.