Axiomatization of Trace Semantics for Stochastic Nondeterministic Processes*

Augusto Parma and Roberto Segala
Dipartimento di Informatica - Università di Verona

Abstract

We give a complete axiomatization of trace distribution precongruence for probabilistic nondeterministic processes based on a process algebra that includes internal behavior and recursion. The axiomatization is given for two different semantics of the process algebra that are consistent with the alternating model of Hansson and the non-alternating model of Segala, respectively. It is shown that the two semantics coincide up to trace distribution precongruence.

1. Introduction

Randomness is used extensively in computer science, both from the algorithmic point of view, where it is used to solve problems that would otherwise be unfeasible, and from the analytical point of view, where the performance of a complex system depends on phenomena that are governed by stochastic laws. The pervasive use of randomness requires adequate formal models; in this paper we are interested in the special case of concurrent probabilistic systems.

When embedded into a concurrent system, randomness interferes with a phenomenon called *nondeterminism* that arises from the unknown relative speeds of processes running in parallel. The choice of the next process that takes a computational step is not governed by any known law, but rather is under the control of an entity called either arbiter, or scheduler, or adversary. The objective is then to study the performance of a system under any scheduling policy.

Among the many formalisms proposed in the literature for this purpose [1, 7, 9, 10, 11, 12, 13, 16, 20, 23, 24, 29, 26, 33] we concentrate on two operational models: the Labeled Concurrent Markov Chains of Hansson [10], also known as the *alternating model* of concurrent probabilistic systems, and the Probabilistic Automata of Segala [26], also known as the *non-alternating model* of concurrent probabilistic systems. Both models are conservative extensions of Labeled Transition Systems [15], and thus are amenable to the extension of related concepts and verification techniques. In particular, the two models are equipped with simulation and bisimulation relations [10, 26, 28, 22] as well as trace and testing semantics [25, 32]. These extended concepts satisfy most of the properties that are valid within Labeled Transition Systems, but also satisfy several other properties that are specific to stochastic behavior. In this paper we study how these properties relate to nondeterminism from an algebraic point of view.

From the definitional point of view the alternating and non-alternating models are very similar. Formally, in a probabilistic automaton a state enables several transitions that lead to discrete probability measures over states, while in a labeled concurrent Markov chain states are partitioned into nondeterministic and probabilistic states, where a nondeterministic state enables several transitions that lead to single probabilistic states, and a probabilistic state enables a single transition that leads to a discrete probability measure over nondeterministic states. Thus, in the alternating model a strict alternation between nondeterministic and probabilistic states is imposed. There are also folklore constructions to convert one model to the other, where a transition of a probabilistic automaton is transformed into two consecutive transitions of a labeled concurrent Markov chain and vice versa.

Yet, the two models are different, and the differences are not completely clear. It is known that checking weak bisimulation equivalence is polynomial for the alternating model [22] and exponential for the non-alternating model [4], and that bisimulation distinguishes too much in the alternating model when considering randomized schedulers [3]. Furthermore, weak bisimulation is incomparable in the two models [3]. The alternating model can be seen as a special case of the non-alternating model, which justifies several of the observations above. In this paper we contribute further to

* Supported by MURST project CoVer.

the understanding of the relations between the models.

Our study is concentrated on a natural stochastic extension of language inclusion: we are interested in studying the properties of stochastic language inclusion and the potential differences that exist between the alternating and non-alternating models in the context of language inclusion. The problem here is particularly interesting because the naive extension of language inclusion to the stochastic case is not preserved by parallel composition, and thus a new *trace distribution precongruence* relation is defined as the coarsest precongruence that refines language inclusion [25]. It is already known that trace distribution precongruence coincides with probabilistic forward simulations [17]; here we complete the picture by addressing the problem via process algebras and axiomatizations. Indeed, process algebras allow us to describe nondeterministic stochastic systems via very few basic operators, and complete axiomatizations give us clear ideas of the algebraic structure of the models, especially when the axiom systems are simple like in our case.

We consider a process algebra with nondeterministic choice, probabilistic choice, and recursion, that is, an algebra that extends classical process algebras with just one new operator that models probability. Then we give two operational semantics that reflect the alternating and non-alternating models and that respect the folklore conversions between the two models. Finally, we study complete axiomatizations for the trace distribution precongruence of [25]. The axioms we obtain are the classical axioms for nondeterministic systems [8, 18] plus a few axioms that are specific to the probabilistic choice operator. In this way we also achieve a complete algebraic separation between probability and nondeterminism. Besides obtaining simple axiom systems, we discover that the alternating and non-alternating models are equivalent up to trace semantics.

The process algebra considered in this paper does not include parallel composition nor action renaming. We have chosen not to consider these operators because their axiomatization follows the standard rules that are known for CCS. In particular action renaming can be removed by moving it to the atomic expressions of a process, while parallel composition can be removed by means of an expansion rule like the expansion rule of CCS [19]. Thus, the analysis of parallel composition and action renaming does not add any new insight to our study. Indeed, we do not run into the faithfulness problems of [16] nor into the associativity problems of [26] because the discussions in [16, 26] refer to a generative model of concurrent probabilistic systems according to [9], while the models studied in this paper are reactive. As shown in [26], most of the problems related to parallel composition disappear in a reactive model.

Similar studies of axiomatizations exist already for bisimulation relations. In particular the recursion free version of our algebra is used in [3] to study axiomatizations for weak and strong probabilistic bisimulations in the alternating and non-alternating models. The axiomatization of strong bisimulation was studied already in [10]. Other axiomatizations of probabilistic process calculi with different underlying models, usually without nondeterminism, are studied in [2, 14, 21, 30].

The rest of the paper is structured as follows. Section 2 introduces the alternating and non-alternating models and related concepts that are relevant for our treatment; Section 3 introduces some of the simulation relations of [28] that are used in the paper; Section 4 introduces the trace distribution precongruence relation and recalls some of its properties; Section 5 introduces our probabilistic process algebra and gives it an alternating and non-alternating semantics following the approach of [3]; Section 6 describes our axiomatization; Section 7 gives some concluding remarks.

2. Probabilistic Model

In this section we define the non-alternating and alternating models together with related basic concepts. We also illustrate the folklore transformation between the two models. We start with some preliminary mathematical concepts and notation.

2.1. Probability Measures and Notation

A *σ-field* on a set Ω, denoted by $\mathcal{F}$, is a family of subsets of Ω that contains Ω and that is closed under complement and countable union. The elements of a σ-field are called *measurable sets*. The σ-field generated by a family of sets $\mathcal{C}$, denoted by $\sigma(\mathcal{C})$ is the smallest σ-field that contains $\mathcal{C}$. A pair $(\Omega, \mathcal{F})$, where $\mathcal{F}$ is a σ-field on Ω, is called a *measurable space*.

A measure on a measurable space $(\Omega, \mathcal{F})$ is a function $\mu : \mathcal{F} \to R^{\geq 0}$ such that, for each countable family $\{X_i\}_{i \in I}$ of pairwise disjoint elements of $\mathcal{F}$, $\mu(\cup_{i \in I} X_i) = \sum_{i \in I} \mu(X_i)$. Function μ is said to be σ-additive. A *probability measure* on $(\Omega, \mathcal{F})$ is a measure μ on $(\Omega, \mathcal{F})$ such that $\mu(\Omega) = 1$, and a *sub-probability measure* on $(\Omega, \mathcal{F})$ is a measure μ on $(\Omega, \mathcal{F})$ such that $\mu(\Omega) < 1$. A *discrete probability measure* over a set Ω is a probability measure on $(\Omega, 2^{\Omega})$. Denote by $Disc(Q)$ the set of discrete probability measures over Q and by $SubDisc(Q)$ the set of discrete sub-probability measures; given an element $q \in Q$, denote by $\delta(q)$ the prob-

ability measure that assigns probability 1 to $\{q\}$. This is called the *Dirac measure* on q.

Given a collection of measures $\{\mu_i\}_{i\in I}$ over $\mathcal{F}$ and a collection of non-negative numbers $\{p_i\}_{i\in I}$ such that $\sum_{i\in I} p_i \leq 1$, define the *convex combination* $\sum_{i\in I} p_i\mu_i$ of the μ_i's to be the measure μ such that, for each set $X \in \mathcal{F}$, $\mu(X) = \sum_{i\in I} p_i\mu_i(X)$.

A function $f : \Omega_1 \to \Omega_2$ is *measurable* from $(\Omega_1, \mathcal{F}_1)$ to $(\Omega_2, \mathcal{F}_2)$ if the pre-image of each element of $\mathcal{F}_2$ is an element of $\mathcal{F}_1$. If f is a measurable function from $(\Omega_1, \mathcal{F}_1)$ to $(\Omega_2, \mathcal{F}_2)$ and μ is a measure on $(\Omega_1, \mathcal{F}_1)$, then it is possible to define the image measure of μ under f on $(\Omega_2, \mathcal{F}_2)$, denoted by $f(\mu)$, as follows: for each $X \in \mathcal{F}_2$, $f(\mu)(X) = \mu(f^{-1}(X))$.

2.2. Probabilistic Automata

A *probabilistic automaton* is a tuple $\mathcal{P} = (Q, \overline{q}, Act, \mathcal{D})$, where Q is the set of *states*, $\overline{q}$ is the *start state*, Act is the set of *actions*, and $\mathcal{D}$ is the *transition relation*, where $\mathcal{D} \subseteq Q \times Act \times Disc(Q)$. The set Act is partitioned into two sets H and $Act - H$ of *internal* and *external* actions, respectively.

Probabilistic automata were defined in [26] as a conservative extension of Labeled Transition Systems [15], also called automata. Indeed, an ordinary automaton can be seen as a probabilistic automaton where each transition leads to a Dirac measure. Probabilistic automata can also be seen as nondeterministic Markov Decision Processes [6], that is, Markov Decision Processes where more than one measure over states can be associated with a state and a label.

An *alternating probabilistic automaton* is a tuple $\mathcal{P} = (N, P, \overline{q}, Act, \mathcal{D}_n, \mathcal{D}_p)$, where N and P are two disjoint sets of *nondeterministic states*, and *probabilistic states*, respectively, $\overline{q} \in N$ is the *start state*, Act is the set of *actions*, and $\mathcal{D}$ is the *transition relation*, where $\mathcal{D} = \mathcal{D}_n \cup \mathcal{D}_p$, $\mathcal{D}_n \subseteq N \times Act \times P$, and $\mathcal{D}_p \subseteq P \times Disc(N)$. The set Act is partitioned into two sets H and $Act - H$ of *internal* and *external* actions, respectively.

Alternating probabilistic automata were defined in [10] as an extension of the (unlabeled) Concurrent Markov Chains of [33], and were used as a semantical model for a probabilistic process algebra and as an underlying model for probabilistic model checking.

For notational convenience we denote the elements of a probabilistic automaton $\mathcal{P}$ by $Q, \overline{q}, Act, \mathcal{D}$, and we propagate the notation to primes and indices as well. Thus, the elements of a probabilistic automaton $\mathcal{P}'_i$ are $Q'_i, \overline{q}'_i, Act'_i, \mathcal{D}'_i$. We adopt a similar notational convention for alternating probabilistic automata.

Remark 1 *The alternating model, that is, alternating probabilistic automata, can be seen as a special case of the non-alternating model, that is, probabilistic automata. Indeed, an alternating probabilistic automaton $\mathcal{P}$ can be seen as a probabilistic automaton $\mathcal{P}'$ where*

- $Q' = N \cup P$,
- $\overline{q}' = \overline{q}$,
- $Act' = Act \cup \{\tau\}$, *and*
- $\mathcal{D}' = \mathcal{D}'_n \cup \mathcal{D}'_p$,

where $\mathcal{D}'_n$ is obtained from $\mathcal{D}_n$ by replacing the third element of each triplet by a Dirac measure, and $\mathcal{D}'_p$ is obtained from $\mathcal{D}_p$ by adding a label τ to each transition. Action τ is meant to be internal.

Based on Remark 1, in the rest of this section we give definitions for probabilistic automata only. The same definitions apply to alternating probabilistic automata as well.

An *execution fragment* of a probabilistic automaton $\mathcal{P}$ is a finite or infinite sequence $\alpha = q_0a_1q_1a_2q_2\cdots$ of alternating states and actions, starting with a state and, if the sequence finite, ending in a state, where for each i, there exists a measure μ such that $(q_i, a_{i+1}, \mu) \in D$ and $\mu(q_{i+1}) > 0$. State q_0 is called the first state of α and is denoted by $fstate(\alpha)$. If α is a finite sequence, then the last state of α is denoted by $lstate(\alpha)$. Denote by $frags(\mathcal{P})$ the set of execution fragments of $\mathcal{P}$ and by $frags^*(\mathcal{P})$ the set of finite execution fragments of $\mathcal{P}$. An *execution* is an execution fragment whose first state is a start state. Denote by $execs(\mathcal{P})$ the set of executions of $\mathcal{P}$ and by $execs^*(\mathcal{P})$ the set of finite executions of $\mathcal{P}$.

An execution fragment α is a *prefix* of an execution fragment α', denoted by $\alpha \leq \alpha'$ if the sequence α is a prefix of the sequence α'. A finite execution fragment $\alpha_1 = q_0a_1q_1\cdots a_kq_k$ and an execution fragment α_2 can be concatenated if $fstate(\alpha_2) = q_k$. In such case the *concatenation* of α_1 and α_2, denoted by $\alpha_1 \frown \alpha_2$, is the execution fragment $q_0a_1q_1\cdots a_k\alpha_2$.

Execution fragments and executions are the result of the resolution of both probabilistic and nondeterministic choices. If we resolve nondeterministic choices only, then we obtain a structure on which we can study probability measures over executions. We use the notion of a scheduler to resolve the nondeterministic choices.

A *scheduler* for a probabilistic automaton $\mathcal{P}$ is a function $\sigma\colon frags^*(\mathcal{P}) \to SubDisc(\mathcal{D})$ such that $\sigma(\alpha)(q, a, \mu) > 0$ implies $q = lstate(\alpha)$. A scheduler induces a probability measure over executions on a σ-field whose construction is standard. Specifically, we consider the σ-field generated by *cones*, where the cone of

a finite execution α, denoted by C_α, is the set of executions that have α as a prefix, that is, $C_\alpha = \{\alpha' \in frags(\mathcal{P}) \mid \alpha \leq \alpha'\}$. Fixed a starting state q_0 and a finite execution fragment $\alpha = q_0 a_1 q_1 \cdots q_k$, the measure $\mu(C_\alpha)$ of the cone C_α is defined as

$$\prod_{i \in \{0,\ldots,k-1\}} \left(\sum_{(q_i, a_{i+1}, \mu') \in \mathcal{D}} \sigma(\alpha^{\leq i})(q_i, a_{i+1}, \mu') \mu'(q_{i+1}) \right),$$

where $\alpha^{\leq i}$ denotes the prefix $q_0 a_1 \cdots a_i q_i$ of α.

Standard measure theoretical arguments ensure that the measure defined on cones extends uniquely to a measure defined on the generated σ-field. The measure μ is called a *probabilistic execution fragment* of $\mathcal{P}$, and state q_0 is said to be the *first state* of μ. If q_0 is the start state of $\mathcal{P}$, then μ is called a *probabilistic execution* of $\mathcal{P}$.

2.3. Relation between the models

The alternating and non-alternating models are related by folklore transformations that convert one model to the other. In this section we define the transformations formally.

The conversion of a probabilistic automaton to an alternating probabilistic automaton consists of splitting each transition into two parts. Formally, given a probabilistic automaton $\mathcal{P}$, define the alternating version of $\mathcal{P}$, denoted by $Ap(\mathcal{P})$, to be the following alternating probabilistic automaton $\mathcal{P}'$:

- $N' = Q$;
- $P' = \{\mu \mid \exists_{q,a}(q, a, \mu) \in \mathcal{D}\}$;
- $\bar{q}' = \bar{q}$;
- $Act' = Act$;
- $\mathcal{D}'_n = \{(q, a, \mu) \mid (q, a, \mu) \in \mathcal{D}\}$;
- $\mathcal{D}'_p = \{(\mu, \mu) \mid \mu \in P'\}$.

Note the double use of μ as the name of a probabilistic state and as a probability measure over nondeterministic states. For this reason we have not written $\mathcal{D}'_n = \mathcal{D}$ in the definition above.

The conversion of an alternating probabilistic automaton to a probabilistic automaton consists of merging transitions from nondeterministic states with the following transitions from probabilistic states. Formally, given an alternating probabilistic automaton $\mathcal{P}$, define the non-alternating version of $\mathcal{P}$, denoted by $Np(\mathcal{P})$, to be the following probabilistic automaton $\mathcal{P}'$:

- $Q' = N$;
- $\bar{q}' = \bar{q}$;
- $Act' = Act$;
- $\mathcal{D}' = \{(q, a, \mu) \mid \exists_s (q, a, s) \in \mathcal{D}_n, (s, \mu) \in \mathcal{D}_p\}$.

2.4. Composition

In this section we define parallel composition for probabilistic automata. Although composition is not used explicitly in this paper, we need to be aware of this operator because it is the main cause of difficulties in the extension of language inclusion to probabilistic automata.

Two probabilistic automata $\mathcal{P}_1$ and $\mathcal{P}_2$ are *compatible* if the set of internal actions of $\mathcal{P}_1$ and the set of external actions of $\mathcal{P}_2$ are disjoint, and vice versa. The *parallel composition* of two compatible probabilistic automata $\mathcal{P}_1, \mathcal{P}_2$ is a probabilistic automaton $\mathcal{P}$, denoted by $\mathcal{P}_1 \parallel \mathcal{P}_2$, such that

- $Q = Q_1 \times Q_2$,
- $\bar{q} = (\bar{q}_1, \bar{q}_2)$,
- $Act = Act_1 \cup Act_2$,
- The transition relation $\mathcal{D}$ is the set of triplets $((q_1, q_2), a, \mu_1 \times \mu_2)$ such that, for each $i \in \{1, 2\}$, $q_i \in Q_i$, and either
  - $(q_i, a, \mu_i) \in \mathcal{D}_i$, or
  - $a \notin Act_i$ and $\mu_i = \delta(q_i)$.

The expression $\mu_1 \times \mu_2$ denotes the independent product of μ_1 and μ_2, that is, for each pair of states q_1, q_2, $(\mu_1 \times \mu_2)(q_1, q_2) = \mu_1(q_1)\mu_2(q_2)$.

A similar definition of composition can be given for alternating probabilistic automata. In this case we consider only pairs consisting of nondeterministic states only or probabilistic states only.

3. Simulation Relations

In this section we introduce some of the simulation relations of [28] that are used in the paper. We use a notation along the lines of [17], and we start with some preliminary concepts.

First we define combined transitions, weak transitions, and hyper-transitions [31]. Let $\{q \xrightarrow{a} \mu_i\}_{i \in I}$ be a collection of transitions of a probabilistic automaton $\mathcal{P}$, and let $\{p_i\}_{i \in I}$ be a collection of probabilities such that $\sum_{i \in I} p_i = 1$. Then the triple $(q, a, \sum_{i \in I} p_i \mu_i)$ is called a *combined transition* of $\mathcal{P}$.

Let ϵ be a probabilistic execution fragment that assigns probability 1 to the set of all finite execution fragments with trace a. Let μ be the measure defined by $\mu(q) = \epsilon(\{\alpha \mid lstate(\alpha) = q\})$. Then $fstate(\epsilon) \stackrel{a}{\Longrightarrow} \mu$

is a *weak combined transition* of $\mathcal{P}$. If ϵ can be generated by a deterministic scheduler, then $\mathit{fstate}(\epsilon) \stackrel{a}{\Longrightarrow} \mu$ is a *weak transition*.

Let $\mu \in \mathit{Disc}(Q)$, and for each $q \in \mathit{supp}(\mu)$ let $q \stackrel{a}{\longrightarrow} \mu_q$ be a combined transition of $\mathcal{P}$. Let μ' be $\sum_{q \in \mathit{supp}(\mu)} \mu(q)\mu_q$. Then $\mu \stackrel{a}{\longrightarrow} \mu'$ is called a *hyper-transition* of $\mathcal{P}$. Also, for each $q \in \mathit{supp}(\mu)$, let $q \stackrel{a}{\Longrightarrow} \mu_q$ be a weak combined transition of $\mathcal{P}$. Let μ' be $\sum_{q \in \mathit{supp}(\mu)} \mu(q)\mu_q$. Then $\mu \stackrel{a}{\Longrightarrow} \mu'$ is called a *weak hyper-transition* of $\mathcal{P}$.

Second, we show how to *lift* a relation over sets to a relation over discrete measures [13]. Let $R \subseteq X \times Y$. The *lifting* of R is a relation $R' \subseteq \mathit{Disc}(X) \times \mathit{Disc}(Y)$ such that $\mu_X \; R' \; \mu_Y$ iff there is a function $w : X \times Y \to [0, 1]$ that satisfies:

1. If $w(x, y) > 0$ then $x \; R \; y$.
2. For each $x \in X$, $\sum_{y \in Y} w(x, y) = \mu_X(x)$.
3. For each $y \in Y$, $\sum_{x \in X} w(x, y) = \mu_Y(y)$.

We abuse notation slightly and denote the lifting of a relation R by R as well.

Third, we define a *flattening* operation that converts a measure μ in $\mathit{Disc}(\mathit{Disc}(X))$ into a measure $\flat(\mu)$ in $\mathit{Disc}(X)$. Namely, we define $\flat(\mu) = \sum_{\rho \in \mathit{supp}(\mu)} \mu(\rho)\rho$.

We are now ready to define simulations for probabilistic automata. A relation $R \subseteq Q_1 \times \mathit{Disc}(Q_2)$ is a *probabilistic forward simulation* (resp., *weak probabilistic forward simulation*) from probabilistic automaton $\mathcal{P}_1$ to probabilistic automaton $\mathcal{P}_2$ iff $\mathcal{P}_1$ and $\mathcal{P}_2$ have the same external actions and both of the following hold:

1. $\bar{q}_1 \; R \; \delta(\bar{q}_2)$.
2. For each pair q_1, μ_2 such that $q_1 \; R \; \mu_2$ and each transition $q_1 \stackrel{a}{\longrightarrow} \mu_1'$, there exists a measure $\mu_2' \in \mathit{Disc}(\mathit{Disc}(Q_2))$ such that $\mu_1' \; R \; \mu_2'$ and such that $\mu_2 \stackrel{a}{\longrightarrow} \flat(\mu_2')$ (resp., $\mu_2 \stackrel{a}{\Longrightarrow} \flat(\mu_2')$) is a hyper-transition (resp., a weak hyper-transition) of $\mathcal{D}_2$.

We write $\mathcal{P}_1 \sqsubseteq_{pF} \mathcal{P}_2$ (resp., $\mathcal{P}_1 \sqsubseteq_{wpF} \mathcal{P}_2$) whenever there is a probabilistic forward simulation (resp., a weak probabilistic forward simulation) from $\mathcal{P}_1$ to $\mathcal{P}_2$. We remove the suffix p whenever there exists a probabilistic (weak) forward simulation that relates states to Dirac measures. In particular, if we consider ordinary automata, that is probabilistic automata whose transitions lead to Dirac measures, then $\sqsubseteq_{pF}$ coincides with $\sqsubseteq_F$ and $\sqsubseteq_{wpF}$ coincides with $\sqsubseteq_{wF}$.

4. Trace Distribution Precongruence

In this section we extend the trace semantics to probabilistic automata according to [25]. We start with the notion of trace, extend it to trace distributions, and define the trace distribution preorder and trace distribution precongruence. We also state some useful results about trace distribution precongruence.

The *trace* of an execution fragment α of a probabilistic automaton $\mathcal{P}$, written $\mathit{trace}_{\mathcal{P}}(\alpha)$, or just $\mathit{trace}(\alpha)$ when $\mathcal{P}$ is clear from context, is the list obtained by restricting α to the set of external actions of A. For a set S of executions of a probabilistic automaton $\mathcal{P}$, denote by $\mathit{traces}_{\mathcal{P}}(S)$, or just $\mathit{traces}(S)$ when $\mathcal{P}$ is clear from context, the set of traces of the executions in S. We say that β is a trace of a probabilistic automaton $\mathcal{P}$ if there is an execution α of $\mathcal{P}$ with $\mathit{trace}(\alpha) = \beta$. Denote by $\mathit{traces}(\mathcal{P})$ the set of traces of $\mathcal{P}$.

The trace function is measurable from the σ-field generated by cones of executions to the σ-field generated by cones of traces. Thus, given a probabilistic execution μ, the image measure under *trace* of μ, denoted by $\mathit{tdist}(\mu)$, is well defined and is called the *trace distribution* of μ. Denote the set of trace distributions of a probabilistic automaton $\mathcal{P}$ by $\mathit{tdists}(\mathcal{P})$. By analogy with the notion of language (trace) inclusion for ordinary automata, we define the *trace distribution preorder* as inclusion of trace distributions, that is, $\mathcal{P}_1 \sqsubseteq_D \mathcal{P}_2$ iff $\mathit{tdists}(\mathcal{P}_1) \subseteq \mathit{tdists}(\mathcal{P}_2)$.

While trace inclusion is a precongruence for ordinary automata (it is preserved by composition), the trace distribution preorder is not a precongruence (see [25] for counterexample and motivations). A solution proposed in [25] defines the *trace distribution precongruence*, denoted by $\sqsubseteq_{DC}$, as the coarsest precongruence that is contained in the trace distribution preorder. An alternative approach consists of modifying appropriately the notion of composition by restricting the power of schedulers [5].

Since the definition of trace distribution precongruence is not explicit, there have been several attempts to provide alternative characterizations of trace distribution precongruence. In [25] a *principal context* is identified, which is an elementary probabilistic automaton that can be used as a distinguishing context for any pair of probabilistic automata that are not in the trace distribution precongruence relation. Two alternative characterizations in terms of testing theory are given in [27, 32]. Finally, a characterization in terms of simulation relations and their probabilistic extensions [28] is given in [17]. In summary, trace distribution precongruence coincides with appropriate notions of simulation relations. The proof of completeness of our axiomatizations is based on the characterization of [17]; here we state the key results of [17] that are needed.

Proposition 1 *The following statements hold.*

1. *Two ordinary automata $\mathcal{P}_1$ and $\mathcal{P}_2$ with no internal actions are in the trace distribution precongruence relation iff there exists a forward simulation from $\mathcal{P}_1$ to $\mathcal{P}_2$. That is, $\mathcal{P}_1 \sqsubseteq_{DC} \mathcal{P}_2$ iff $\mathcal{P}_1 \sqsubseteq_F \mathcal{P}_2$.*
2. *Two probabilistic automata $\mathcal{P}_1$ and $\mathcal{P}_2$ with no internal actions are in the trace distribution precongruence relation iff there exists a probabilistic forward simulation from $\mathcal{P}_1$ to $\mathcal{P}_2$. That is, $\mathcal{P}_1 \sqsubseteq_{DC} \mathcal{P}_2$ iff $\mathcal{P}_1 \sqsubseteq_{pF} \mathcal{P}_2$.*
3. *Two probabilistic automata $\mathcal{P}_1$ and $\mathcal{P}_2$ are in the trace distribution precongruence relation iff there exists a weak probabilistic forward simulation from $\mathcal{P}_1$ to $\mathcal{P}_2$. That is, $\mathcal{P}_1 \sqsubseteq_{DC} \mathcal{P}_2$ iff $\mathcal{P}_1 \sqsubseteq_{wpF} \mathcal{P}_2$.*

Proposition 1 can be used to establish that the folklore conversions between the alternating and non-alternating models preserve the kernel $\equiv_{DC}$ of trace distribution precongruence.

Proposition 2 *The the following statements hold.*

1. *For every probabilistic automaton $\mathcal{P}$, $\mathcal{P} \equiv_{DC} Ap(\mathcal{P})$;*
2. *For every alternating probabilistic automaton $\mathcal{P}$, $Np(\mathcal{P}) \equiv_{DC} \mathcal{P}$.*

Proof outline. It is sufficient to exhibit appropriate simulation relation between the involved probabilistic automata after viewing an alternating probabilistic automaton as a special case of a probabilistic automaton according to Remark 1.

Specifically, for Item 1 the forward simulation from $\mathcal{P}$ to $Ap(\mathcal{P})$ is $R_1 = \{(q,q) \mid q \in Q\}$ and the probabilistic forward simulation from $Ap(\mathcal{P})$ to $\mathcal{P}$ is $R_2 = \{(q,\delta(q)) \mid q \in Q\} \cup \{(\mu,\mu) \mid \exists_{q,a}(q,a,\mu) \in \mathcal{D}\}$; for Item 2 the forward simulation from $Np(\mathcal{P})$ to $\mathcal{P}$ is $R_1 = \{(q,q) \mid q \in N\}$ and the probabilistic forward simulation from $\mathcal{P}$ to $Np(\mathcal{P})$ is $R_2 = \{(q,\delta(q)) \mid q \in N\} \cup \{(s,\mu) \mid (s,\mu) \in \mathcal{D}_p\}$. ■

5. Probabilistic Process Algebra

In this section we define our probabilistic process algebra (PPA) as an extension of the process algebra of [3] with recursion. We provide it with a non-alternating as well as an alternating semantics and show that the two semantics are related by the folklore transformation between the models.

Denote by $Act = \mathcal{L} \cup \{\tau\}$ the set of *actions*, where τ is the *silent action*. We let a range over Act.

Let *NProc* denote the set of *nondeterministic processes*, ranged over by E, and *PProc* denote the set of *probabilistic processes*, ranged over by P. Finally, let $Proc \triangleq NProc \cup PProc$ denote the set of *processes*. The syntax of our Probabilistic Process Algebra is given by the following rules:

$$\begin{array}{lcl} E & ::= & 0 \mid X \mid E + E \mid a.P \mid recX.E \\ P & ::= & \Delta(E) \mid P \oplus_p P \end{array}$$

Our probabilistic process algebra extends the algebra of [3] by adding the variable operator X and the recursion operator $recX.E$. Table 1 contains the operational semantics of PPA, where $E \xrightarrow{a} \mu$ describes a transition labeled by a that leaves from E and leads to a measure μ over processes, while $P \longmapsto \mu$ denotes the fact that the probability measure over nondeterministic processes associated with P is μ.

The alternating and non-alternating semantics differ only in the rule for prefixing: in the non-alternating semantics process $a.P$ moves to the measure denoted by P, while in the alternating semantics process $a.P$ moves to process P with probability 1. It is easy to show the following result.

Proposition 3 *Let E be a process, $\mathcal{P}$ be its non-alternating semantics, and $\mathcal{P}'$ be its alternating semantics. Then*

- *$\mathcal{P}$ and $Ap(\mathcal{P})$ are strongly bisimilar;*
- *$\mathcal{P}'$ and $Np(\mathcal{P}')$ are the same.*

Proof outline. The bisimulation relation for Item 1 is the symmetric closure of $NProc \times NProc \cup \{(P,\mu) \mid P \in PProc, P \mapsto \mu\}$, where the μ related to P is viewed as a state of $Ap(\mathcal{P})$. The equality of $\mathcal{P}'$ and $Np(\mathcal{P}')$ follows from the fact that rule **NA - prefix** collapses $a.P \xrightarrow{a} P \xrightarrow{\tau} \mu$ into $a.P \xrightarrow{a} \mu$ in the same way as the construction of $Np(\mathcal{P}')$. ■

The first item of Proposition 3 states that $\mathcal{P}$ and $Ap(\mathcal{P})$ are strongly bisimilar, and not the same, because the probabilistic states of $\mathcal{P}$ are probabilistic processes, while the corresponding states of $Ap(\mathcal{P})$ are the probability measures denoted by the probabilistic processes of $\mathcal{P}$: several syntactically different probabilistic processes denote the same probability measure.

6. Axiomatizations

In this section we give our complete axiomatization of the trace distribution precongruence relation. We provide several complete axiomatizations for different fragments of PPA. Specifically, we consider non-recursive τ-free nondeterministic processes, where we obtain a classical axiomatization for simulation relations, non-recursive τ-free probabilistic processes, where the axioms are enriched by

Nondeterministic

lchoice $\dfrac{E_1 \xrightarrow{a} \mu}{E_1 + E_2 \xrightarrow{a} \mu}$

rchoice $\dfrac{E_2 \xrightarrow{a} \mu}{E_1 + E_2 \xrightarrow{a} \mu}$

Probabilistic

idle $\dfrac{\cdot}{\Delta(E) \longmapsto \delta(E)}$

P - idle $\dfrac{P \longmapsto \mu}{P \xrightarrow{\tau} \mu}$

pchoice $\dfrac{P_1 \longmapsto \mu_1 \;\; P_2 \longmapsto \mu_2}{P_1 \oplus_p P_2 \longmapsto p\mu_1 + (1-p)\mu_2}$

Non-alternating

NA - prefix $\dfrac{P \longmapsto \mu}{a.P \xrightarrow{a} \mu}$

Alternating

A - prefix $\dfrac{\cdot}{a.P \xrightarrow{a} \delta(P)}$

Recursion

Rec $\dfrac{E\{recX.E/X\} \xrightarrow{a} \mu}{recX.E \xrightarrow{a} \mu}$

Table 1. Operational semantics of PPA

properties of $\oplus$, recursive τ-free probabilistic processes, where the axioms are enriched with classical axioms for simulation relations, and recursive probabilistic processes, where the axioms are enriched by three rules to deal with τ's. Our analysis is carried out in the non-alternating semantics only, but it applies to the alternating semantics as well given Theorems 2 and 3.

6.1. Non-Recursive τ-free Nondeterministic Processes

We consider the non-recursive fragment of PPA where no probabilistic choice is possible. This means that probabilistic processes can be only of the form $\Delta(E)$. Processes are derived according to the following grammar.

$$\begin{array}{lcl} E & ::= & 0 \mid E + E \mid a.P \\ P & ::= & \Delta(E) \end{array}$$

In this case the equivalence results of [17] (cf. Proposition 1) state that two processes are in the trace distribution precongruence relation iff there exists a simulation from one process to the other. Thus, it is sufficient to consider a complete axiomatization for simulation relations. We refer the reader to [8] for a treatment of complete axiomatizations for simulation relations.

A1	$E + F = F + E$
A2	$E + (F + G) = (E + F) + G$
A3	$E + E = E$
A4	$E + 0 = E$
DE	$E = \Delta(E)$
DC	$E \leq E + F$

Table 2. Axioms for non-recursive τ-free nondeterministic processes.

Table 2 contains the complete axioms for non-recursive nondeterministic processes. The first four axioms **A1-4** are the classical axioms for bisimulation equivalence, while Axiom **DC** is the key axiom for simulation relations stating that in a process that simulates there can be more options than in the process that is simulated. Axiom **DE** is the only axiom that relates nondeterministic and probabilistic processes. It states that a Dirac choice over a single nondeterministic process is equivalent to the nondeterministic process.

Theorem 1 *The axioms of Table 2 are sound and complete for trace distribution precongruence in the non-recursive τ-free nondeterministic fragment of PPA.*

Proof outline. The result follows from [8] after using Proposition 1 to reduce the problem to completeness for $\sqsubseteq_F$ and Axiom **DE** to treat $\Delta(E)$ as E. ■

Remark 2 *A derived rule from the axioms of Table 2 is*

$$\boldsymbol{A5} \qquad a.\Delta(E) + a.\Delta(F) \leq a.\Delta(E+F).$$

This follows by congruence and **A3** *after applying* ***DC*** *and* ***A1*** *to show that* $E \leq E+F$ *and* $F \leq E+F$. *Rule* ***A5*** *states the key property of forward simulations that nondeterministic choice can be moved forward when moving from a simulated process to a simulating process.*

6.2. Non-recursive τ-free Probabilistic Processes

We now consider the non-recursive τ-free fragment of PPA where also the probabilistic choice operator is allowed. In this case we simply need to add the commutativity, associativity and idempotence axioms for $\oplus$ (axioms **P1-3** of Table 3) and an axiom that relates

P1	$P \oplus_p Q = Q \oplus_{1-p} P$
P2	$P \oplus_{p_1} (Q \oplus_{\frac{p_2}{1-p_1}} R) = (P \oplus_{\frac{p_1}{p_1+p_2}} Q) \oplus_{(p_1+p_2)} R)$
P3	$P \oplus_p P = P$
P4	$a.(P \oplus_p Q) \leq \Delta(a.P) \oplus_p \Delta(a.Q)$

Table 3. Axioms for non-recursive τ-free processes.

probabilistic choice with prefixing. Axiom **P4** states that probabilistic choice can be moved backward when moving from a simulated process to a simulating process. This is the counterpart of derived rule **A5**, where nondeterministic choice can be moved forward.

The key observation about the axiomatization derived so far is that the rules for nondeterministic and probabilistic choices are completely separated, each axiom has a simple structure, and the number of axioms is small.

Theorem 2 *The axioms of Tables 2 and 3 are sound and complete for trace distribution precongruence in the non-recursive τ-free fragment of PPA.*

Proof outline. The proof follows the structure of the completeness proof for simulations [8]; however, since the target of a transition is a probability measure, and since the definition of probabilistic forward simulation requires to relate probability measures over states to probability measures over probability measures over states, Axioms **P1-3** are used heavily to rearrange terms so that an expression is decomposed into the correct measure over measures over states. Axiom **P4** is used to move probabilistic choices backward.

More precisely, by associativity, nondeterministic processes can be represented in the form $\sum_I a_i.P_i$ where the P_i's are probabilistic processes, and probabilistic processes can be represented in the form $\sum_J [p_j]E_j$ where the E_j's are nondeterministic processes and $\sum_J p_j = 1$. The proof shows by induction on the complexity of the processes that for every nondeterministic process E and probabilistic process Q such that $E \sqsubseteq_{wpF} Q$ (we use weak simulations because the first transition from Q is labeled by τ) it is the case that the axioms of Tables 2 and 3 prove $E \leq Q$. Since every nondeterministic process can be seen as the Δ of a probabilistic process, Axiom **DE** suffices to complete the proof.

Indeed, if $E \equiv \sum_I a_i.P_i$ and $Q \equiv \sum_J [p_j]F_j$, we prove that for each $i \in I$, $a_i.P_i \leq Q$ and then we use the congruence rules to combine all the summands of E. By definition of forward simulation, we know that for each j there is a combined transition of F_j labeled by a_i to a measure μ_j such that $\sum_J p_j\mu_j$ is the flattening of some measure μ such that $\rho \sqsubseteq_{wpF} \mu$, where ρ is the measure associated with P_i. We first use repeatedly Axiom **C** from Remark 3 below to add to each of the F_j's a term $a_i.Q_j$ such that $Q_j \mapsto \mu_j$; then we use Axiom **P4** to extract from Q a provably smaller term $a_i.\sum_J [p_j]Q_j$. Then we use the axioms for $\oplus$ to rearrange the terms of $\sum_J [p_j]Q_j$ to describe the measure μ. That is, $\sum_J [p_j]Q_j$ is rewritten into $\sum_K [r_k]R_k$, where each R_k denotes a measure whose μ probability is r_k. Finally we use induction to show that all the elements in the support of ρ are provably related to the appropriate processes among the R_k's, and we use the congruence rules to combine all pieces together. ■

Remark 3 *A derived rule from the axioms of Tables 2 and 3 is*

$$\boldsymbol{P5} \qquad a.P_1 \oplus_p a.P_2 \leq a.P_1 + a.P_2.$$

This follows by applying axiom ***DC*** *to* $a.P_1$, *axioms* ***DC*** *and* ***A1*** *to* $a.P_2$, *and then applying axiom* ***A3*** *to the resulting expression. Rule* ***P5*** *can be used together with axiom* ***DC*** *to derive*

$$\boldsymbol{C} \qquad a.P_1 + a.P_2 = a.P_1 + a.P_2 + a.(P_1 \oplus_p P_2)$$

stating that it is always possible to combine transitions arbitrarily within a process.

6.3. Recursive τ-free Processes

The addition of recursion to our fragments of PPA does not cause any problem. Indeed, the axioms for

recursion of [8], listed in Table 4, continue to work. The expression $F \not\rhd X$ in Table 4 means that variable X is guarded in process F, that is, it occurs always within the scope of some prefix operator, while the expression $F\{E/X\}$ is the result of replacing every occurrence of X by E in F after renaming some of the bound variables of F to avoid capturing of free variables. In summary, our axiomatization ends up to be the axiomatization for simulation relation of nondeterministic processes plus four axioms that deal with probabilistic choice and one axiom that relates probabilistic and nondeterministic processes. This is remarkably simple.

R1	$recX.E = E\{recX.E/X\}$
RL1	if $F \not\rhd X$ and $E \leq F\{E/X\}$ then $E \leq recX.F$
RL2	if $F\{E/X\} \leq E$ then $recX.F \leq E$

Table 4. Axioms for recursive processes

Theorem 3 *The axioms of Tables 2, 3, 4, and 5 are sound and complete for trace distribution precongruence in PPA.*

The proof of Theorem 3 follows the lines of the corresponding proof of [8]. Once again the main technical difficulty is the handling of probabilities on every transition. In particular we need to extend to the probabilistic case the notion of simulation up to.

6.4. Processes with τ actions

The final step to axiomatize PPA is the handling of internal actions. In this case we introduce a variation of the τ-laws of Milner [19] as proposed in [3], which allow us to saturate a process and then prove completeness following the same approach as for the τ-free case. Axioms **A5** and **A6** from Table 5 are taken directly from [3] and are the key axioms for the saturation process. Axiom **A5** is a stronger version than the corresponding axiom **A5** of [3]. Indeed, by applying the congruence rules and Axiom **DE** it is possible to derive $a.(\Delta(\tau.\Delta(E)) \oplus_p P) = a.(\Delta(E) \oplus_p P)$.

Theorem 4 *The axioms of Tables 2, 3, 4, and 5 are sound and complete for trace distribution precongruence in PPA.*

A5	$P = \tau.P$
A6	$\tau.\sum_{i\in I}[p_i](E_i + a.P_i) + a.\sum_{i\in I}[p_i]P_i = \tau.\sum_{i\in I}[p_i](E_i + a.P_i)$
A7	$a.\sum_{i\in I}[p_i](E_i + \tau.P_i) + a.\sum_{i\in I}[p_i]P_i = a.\sum_{i\in I}[p_i](E_i + \tau.P_i)$

Table 5. Axiom for τ-elimination.

7. Concluding Remarks

We have provided sound and complete axiomatizations for trace distribution precongruence on several fragments of a probabilistic process algebra for an alternating as well as a non-alternating semantic model. We have also shown that the alternating and non-alternating semantics of our process terms, which respect folklore transformations between the two models, are equivalent according to the kernel of trace distribution precongruence.

The axioms are remarkably simple in that they consist of complete axioms for the probability-free fragment of our process algebra plus four axioms that deal with probabilistic choice and one axiom that relates probabilistic and nondeterministic processes. These results support the fact that both the alternating and non-alternating models of concurrent probabilistic systems are conservative extensions of ordinary nondeterministic systems and continue to respect the well known rules of nondeterminism.

References

[1] L. de Alfaro. *Formal Verification of Probabilistic Systems.* PhD thesis, Stanford University, 1997. Available as Technical report STAN-CS-TR-98-1601.

[2] J.C.M. Baeten, J.A. Bergstra, and S.A. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities. *Information and Computation*, 122:234–255, 1995.

[3] E. Bandini and R. Segala. Axiomatizations for probabilistic bisimulation. In J. van Leeuwen F. Orejas, P.G. Spirakis, editor, *Proceedings* 28^{th} *ICALP,* Crete, Greece, volume 2076 of *Lecture Notes in Computer Science*, pages 370–381. Springer-Verlag, 2001.

[4] S. Cattani and R. Segala. Decision algorithms for probabilistic bisimulation. In P. Jankar and M. Kretinsky, editors, *Proceedings of CONCUR 2002,* Brno, Czech Republic, volume 2421 of *Lecture Notes in Computer Science*, pages 371–385. Springer-Verlag, 2002.

[5] L. de Alfaro, T.A. Henzinger, and R. Jhala. Compositional methods for probabilistic systems. In K.G. Larsen and M. Nielsen, editors, *Proceedings of CONCUR 2001,*

Aalborg, Denmark, volume 2154 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[6] C. Derman. *Finite State Markovian Decision Processes*. Acedemic Press, 1970.

[7] Y.A. Feldman and D. Harel. A probabilistic dynamic logic. *Journal of Computer and System Sciences*, 28(2):193–215, 1984.

[8] U. Frendrup and J.N. Jensen. A complete axiomatization of simulation for regular ccs expressions. Technical report, BRICS, Aalborg University, department of Computer science, Denmark, 2001.

[9] R.J. van Glabbeek, S.A. Smolka, B. Steffen, and C.M.N. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Proceedings* 5^{th} *Annual Symposium on Logic in Computer Science,* Philadelphia, USA, pages 130–141. IEEE Computer Society Press, 1990.

[10] H. Hansson. *Time and Probability in Formal Design of Distributed Systems*. PhD thesis, Department of Computer Science, Uppsala University, 1991.

[11] S. Hart, M. Sharir, and A. Pnueli. Termination of probabilistic concurrent programs. *ACM Transactions on Programming Languages and Systems*, 5(3):356–380, 1983.

[12] C. Jones and G. Plotkin. A probabilistic powerdomain of evaluations. In *Proceedings* 4^{th} *Annual Symposium on Logic in Computer Science,* Asilomar, California, pages 186–195. IEEE Computer Society Press, 1989.

[13] B. Jonsson and K.G. Larsen. Specification and refinement of probabilistic processes. In *Proceedings of the 6th IEEE Symposium on Logic in Computer Science*, pages 266–277, Amsterdam, July 1991.

[14] C.C. Jou and S.A. Smolka. Equivalences, congruences, and complete axiomatizations for probabilistic processes. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings of CONCUR 90,* Amsterdam, volume 458 of *Lecture Notes in Computer Science*, pages 367–383. Springer-Verlag, 1990.

[15] R. Keller. Formal verification of parallel programs. *Communications of the ACM*, 7(19):561–572, 1976.

[16] G. Lowe. Representing nondeterminism and probabilistic behavior in reactive proceses. Technical Report PRG-TR-11-93, Oxford University Computing Laboratory - Programming Research Group, 1993.

[17] N. Lynch, R. Segala, and F. Vaandrager. Compositionality for probabilistic automata. In D. Lugiez R. Amadio, editor, *Proceedings of CONCUR 2003,* Marseille, France, volume 2761 of *Lecture Notes in Computer Science*, pages 208–221. Springer-Verlag, 2003.

[18] R. Milner. A complete inference system for a class of regular behaviours. *Journal of Computer and System Sciences*, 28:439–466, 1984.

[19] R. Milner. *Communication and Concurrency*. Prentice-Hall International, Englewood Cliffs, 1989.

[20] Carroll Morgan, Annabelle McIver, and Karen Seidel. Probabilistic predicate transformers. *ACM Transactions on Programming Languages and Systems*, 18(3):325–353, May 1996.

[21] M. Nunez. An axiomatization of probabilistic testing. In *Proceedings of 5th AMAST Workshop on Real-Time and Probabilistic Systems*, Lecture Notes in Computer Science, pages 130–150, 1999.

[22] A. Philippou, I. Lee, and O. Sokolsky. Weak bisimulation for probabilistic systems. In C. Palamidessi, editor, *Proceedings of CONCUR 2000,* University Park, PA, USA, volume 1877 of *Lecture Notes in Computer Science*, pages 334–349. Springer-Verlag, 2000.

[23] A. Pnueli and L. Zuck. Verification of multiprocess probabilistic protocols. *Distributed Computing*, 1(1):53–72, 1986.

[24] J.R. Rao. Reasoning about probabilistic algorithms. In *Proceedings of the* 9^{th} *Annual ACM Symposium on Principles of Distributed Computing,* Quebec, Canada, August 1990.

[25] R. Segala. A compositional trace-based semantics for probabilistic automata. In I. Lee and S.A. Smolka, editors, *Proceedings of CONCUR 95,* Philadelphia, PA, USA, volume 962 of *Lecture Notes in Computer Science*, pages 234–248. Springer-Verlag, 1995.

[26] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, Dept. of Electrical Engineering and Computer Science, 1995. Also appears as technical report MIT/LCS/TR-676.

[27] R. Segala. Testing probabilistic automata. In U. Montanari and V. Sassone, editors, *Proceedings of CONCUR 95,* Pisa, Italy, volume 1119 of *Lecture Notes in Computer Science*, pages 299–314. Springer-Verlag, 1996.

[28] R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.

[29] K. Seidel. Probabilistic communicating processes. Technical Report PRG-102, Ph.D. Thesis, Programming Research Group, Oxford University Computing Laboratory, 1992.

[30] E.W. Stark and S.A. Smolka. A complete axiom system for finite-state probabilistic processes. In G. Plotkin, C.P. Stirling, and M. Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT Press, 1999.

[31] M.I.A. Stoelinga. *Alea jacta est: Verification of Probabilistic, Real-Time and Parametric Systems*. PhD thesis, University of Nijmegen, 2002.

[32] M.I.A. Stoelinga and F.W. Vaandrager. A testing scenario for probabilistic automata. In J. Parrow In J.C.M. Baeten, J.K. Lenstra and G.J. Woeginger, editors, *Proceedings* 30^{th} *ICALP,* Crete, Greece, volume 2719 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.

[33] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of 26th IEEE Symposium on Foundations of Computer Science*, pages 327–338, Portland, OR, 1985.