Compositionality for Probabilistic Automata*

Nancy Lynch†
Computer Science and Artificial Intelligence Laboratory
MIT, Cambridge, MA 02139, USA
lynch@theory.csail.mit.edu

Roberto Segala‡
Dipartimento di Informatica, Università di Verona, Italy
roberto.segala@univr.it

Frits Vaandrager§
Institute for Computing and Information Sciences
University of Nijmegen, P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
fvaan@cs.kun.nl

November 2, 2004

Abstract

We establish that on the domain of probabilistic automata, the trace distribution precongruence coincides with the simulation preorder.

1 Introduction

Probabilistic automata [13, 15, 18] constitute a mathematical framework for modeling and analyzing probabilistic systems, specifically, systems of asynchronously interacting components that may make nondeterministic and probabilistic choices. They have been applied successfully to distributed algorithms [7, 11, 1] and practical communication protocols [19].

An important part of a system modeling framework is a notion of *external behavior* of system components. Such a notion can be used to define implementation and equivalence relationships between components. For example, the external behavior of a nondeterministic automaton can be defined as its set of *traces*—the sequences of external actions that arise during its executions [9]. Implementation and equivalence of nondeterministic automata can be defined in terms of inclusion and equality of sets of traces. By analogy, Segala [13] has proposed defining the external behavior of a probabilistic automaton as its set of *trace distributions*, and defining implementation

*A preliminary version of this paper appeared as [8].

†Supported by AFOSR contract #F49620-00-1-0097, AFRL Award #FA9550-04-1-0121, NSF grants #CCR-0121277 and #CCR-0326277, and DARPA/AFOSR MURI #F49620-02-1-0325.

‡Supported by MURST projects MEFISTO and CoVer.

§Supported by PROGRESS project TES4999: Verification of Hard and Softly Timed Systems (HaaST) and DFG/NWO bilateral cooperation project 600.050.011.01 Validation of Stochastic Systems (VOSS).

and equivalence in terms of inclusion and equality of sets of trace distributions. Stoelinga and Vaandrager have proposed a simple testing scenario for probabilistic automata, and have proved that the equivalence notion induced by their scenario coincides with Segala's trace distribution equivalence [20]. Another equivalent testing scenario was proposed by Segala [14].

However, a problem with these notions is that trace distribution inclusion and equivalence are not compositional. To address this problem, Segala [13] defined more refined notions of implementation and equivalence. In particular, he defined the *trace distribution precongruence*, $\leq_{DC}$, as the coarsest precongruence included in the trace distribution inclusion relation. This yields compositionality by construction, but does not provide insight into the nature of the $\leq_{DC}$ relation. Segala also provided a characterization of $\leq_{DC}$ in terms of the set of trace distributions observable in a certain *principal context*—a rudimentary probabilistic automaton that makes very limited nondeterministic and probabilistic choices. However, this indirect characterization still does not provide much insight into the structure of $\leq_{DC}$, for example, it does not explain its branching structure.

In this paper, we provide an explicit characterization of the trace distribution precongruence, $\leq_{DC}$, for probabilistic automata, which completely explains its branching structure. Namely, we show that $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$ if and only if there exists a *weak probabilistic (forward) simulation relation* from $\mathcal{P}_1$ to $\mathcal{P}_2$. Moreover, we provide a similar characterization of $\leq_{DC}$ for nondeterministic automata in terms of the existence of a weak (non-probabilistic) simulation relation. It was previously known that simulation relations are sound for $\leq_{DC}$ [13], for both nondeterministic and probabilistic automata; we show the surprising fact that they are also *complete*. That is, we show that, for both nondeterministic and probabilistic automata, probabilistic contexts can observe all the distinctions that can be expressed using simulation relations.

Another approach to achieving compositionality for behaviors of probabilistic automata is to define implementation as trace distribution inclusion, but to restrict parallel composition so that the nondeterminism of each component is resolved based only on externally-visible behavior of the other components. This approach was investigated by De Alfaro, Henzinger, and Jhala [4] in a synchronous model; however, it is still an open problem to find appropriate restrictions for parallel composition in a model with asynchronous computation. Some initial steps toward this goal appear in [3].

Sections 2 and 3 contain basic definitions and results for nondeterministic and probabilistic automata, respectively, and for the preorders we consider. These sections contain no new material, but recall definitions and theorems from the literature. For a more leisurely introduction see [9, 10, 18, 16]. The last two references also contain an extensive discussion of the relationships of our probabilistic automata with other modelling frameworks for probabilistic systems. The proofs of our completeness results rely on a special context for a probabilistic automaton, the *dual probabilistic automaton*, which is introduced in Section 4. Sections 5 and 6 contain our characterization results for nondeterministic and probabilistic automata. Since the proof of the characterization result for the general case of probabilistic automata with internal actions is highly complex, we first present a proof for the special case of nondeterministic automata without internal actions (Section 5.1). Then we successively show how we can also handle internal actions (Section 5.2) and probabilistic choice (Section 6.1) before dealing with the general case of probabilistic automata with internal actions (Section 6.2). Section 7 contains our conclusions.

2 Definitions and Basic Results for Nondeterministic Automata

2.1 Nondeterministic Automata, Executions, and Traces

A *(nondeterministic) automaton* is a tuple $\mathcal{A} = (Q, \bar{q}, E, H, D)$, where

- Q is a set of *states*,
- $\bar{q} \in Q$ is a *start state*,
- E is a set of *external actions*,
- H is a set of *internal (hidden) actions* with $E \cap H = \emptyset$, and
- $D \subseteq Q \times (E \cup H) \times Q$ is a *transition relation*.

We denote $E \cup H$ by A and we refer to it as the set of *actions*. We denote a transition (q, a, q') of D by $q \xrightarrow{a} q'$. We write $q \rightarrow q'$ if $q \xrightarrow{a} q'$ for some a, and we write $q \rightarrow$ if $q \rightarrow q'$ for some q'.

We assume finite branching[1]: for each state q the number of pairs (a, q') such that $q \xrightarrow{a} q'$ is finite. We denote the elements of an automaton $\mathcal{A}$ by $Q_{\mathcal{A}}, \bar{q}_{\mathcal{A}}, E_{\mathcal{A}}, H_{\mathcal{A}}, D_{\mathcal{A}}, A_{\mathcal{A}}, \xrightarrow{a}_{\mathcal{A}}$. Often we use the name $\mathcal{A}$ for a generic automaton; in this case, we usually omit the subscripts, writing simply Q, $\bar{q}$, E, H, D, A, and $\xrightarrow{a}$. We extend this convention to allow indices and primes as well; thus, the set of states of automaton $\mathcal{A}'_i$ is denoted by Q'_i.

An *execution fragment* of an automaton $\mathcal{A}$ is a finite or infinite sequence $\alpha = q_0 a_1 q_1 a_2 q_2 \cdots$ of alternating states and actions, starting with a state and, if the sequence is finite, ending in a state, where each $(q_i, a_{i+1}, q_{i+1}) \in D$. State q_0, the first state of α, is denoted by $fstate(\alpha)$. If α is a finite sequence, then the last state of α is denoted by $lstate(\alpha)$. An *execution* of $\mathcal{A}$ is an execution fragment whose first state is the start state $\bar{q}$. We let $frags(\mathcal{A})$ denote the set of execution fragments of $\mathcal{A}$ and $frags^*(\mathcal{A})$ the set of finite execution fragments. Similarly, we let $execs(\mathcal{A})$ denote the set of executions of $\mathcal{A}$ and $execs^*(\mathcal{A})$ the set of finite executions.

Execution fragment α is a *prefix* of execution fragment α', denoted by $\alpha \leq \alpha'$, if sequence α is a prefix of sequence α'. Finite execution fragment $\alpha_1 = q_0 a_1 q_1 \cdots a_k q_k$ and execution fragment α_2 can be concatenated if $fstate(\alpha_2) = q_k$. In this case the *concatenation* of α_1 and α_2, $\alpha_1 \frown \alpha_2$, is the execution fragment $q_0 a_1 q_1 \cdots a_k \alpha_2$. Given an execution fragment α and a finite prefix α', $\alpha \triangleright \alpha'$ (read as "α after α'") is defined to be the unique execution fragment α'' such that $\alpha = \alpha' \frown \alpha''$.

The *trace* of an execution fragment α of an automaton $\mathcal{A}$, written $trace_{\mathcal{A}}(\alpha)$, or just $trace(\alpha)$ when $\mathcal{A}$ is clear from context, is the sequence obtained by restricting α to the set of external actions of $\mathcal{A}$. For a set S of executions of an automaton $\mathcal{A}$, $traces_{\mathcal{A}}(S)$, or just $traces(S)$ when $\mathcal{A}$ is clear from context, is the set of traces of the executions in S. We say that β is a trace of an automaton $\mathcal{A}$ if there is an execution α of $\mathcal{A}$ with $trace(\alpha) = \beta$. Let $traces(\mathcal{A})$ denote the set of traces of $\mathcal{A}$. We define the *trace preorder* relation on automata as follows: $\mathcal{A}_1 \leq_T \mathcal{A}_2$ iff $E_1 = E_2$ and $traces(\mathcal{A}_1) \subseteq traces(\mathcal{A}_2)$. We use $\equiv_T$ to denote the kernel of $\leq_T$.

If $a \in A$, then $q \stackrel{a}{\Longrightarrow} q'$ iff there exists an execution fragment α such that $fstate(\alpha) = q$, $lstate(\alpha) = q'$, and $trace(\alpha) = trace(a)$. (Here and elsewhere, we abuse notation slightly by extending the *trace* function to arbitrary sequences.) We call $q \stackrel{a}{\Longrightarrow} q'$ a *weak transition*.

We let tr range over either transitions or weak transitions. For a transition $tr = (q, a, q')$, we denote q by $source(tr)$ and q' by $target(tr)$.

[1]This restriction is given for technical reasons. The results generalize to countable branching at the cost of adding complexity to the proofs.

2.2 Composition

Automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are *compatible* if $H_1 \cap A_2 = A_1 \cap H_2 = \emptyset$. The *composition* of compatible automata $\mathcal{A}_1$ and $\mathcal{A}_2$, denoted by $\mathcal{A}_1 \| \mathcal{A}_2$, is the automaton $\mathcal{A} \triangleq (Q_1 \times Q_2, (\bar{q}_1, \bar{q}_2), E_1 \cup E_2, H_1 \cup H_2, D)$ where D is the set of triples (q, a, q') such that, for $i \in \{1, 2\}$:

$$a \in A_i \Rightarrow (\pi_i(q), a, \pi_i(q')) \in D_i \text{ and } a \notin A_i \Rightarrow \pi_i(q) = \pi_i(q').$$

Let α be an execution fragment of $\mathcal{A}_1 \| \mathcal{A}_2$, $i \in \{1, 2\}$. Then $\pi_i(\alpha)$, the i^{th} projection of α, is the sequence obtained from α by projecting each state onto its i^{th} component, and removing each action not in A_i together with its following state. Sometimes we denote this projection by $\alpha \lceil \mathcal{A}_i$.

Proposition 2.1 *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be automata, with $\mathcal{A}_1 \leq_T \mathcal{A}_2$. Then, for each automaton $\mathcal{C}$ compatible with both $\mathcal{A}_1$ and $\mathcal{A}_2$, $\mathcal{A}_1 \| \mathcal{C} \leq_T \mathcal{A}_2 \| \mathcal{C}$.*

2.3 Simulation Relations

We define two kinds of simulation relations: forward simulations, which provide a step-by-step correspondence, and weak forward simulations, which are insensitive to the occurrence of internal steps. Namely, relation $R \subseteq Q_1 \times Q_2$ is a *forward simulation* (resp., *weak forward simulation*) from $\mathcal{A}_1$ to $\mathcal{A}_2$ iff $E_1 = E_2$ and both of the following hold:

1. $\bar{q}_1 \; R \; \bar{q}_2$.
2. If $q_1 \; R \; q_2$ and $q_1 \xrightarrow{a} q_1'$, then there exists q_2' such that $q_2 \xrightarrow{a} q_2'$ (resp., $q_2 \stackrel{a}{\Longrightarrow} q_2'$) and $q_1' \; R \; q_2'$.

We write $\mathcal{A}_1 \leq_F \mathcal{A}_2$ (resp., $\mathcal{A}_1 \leq_{wF} \mathcal{A}_2$) when there is a forward simulation (resp., a weak forward simulation) from $\mathcal{A}_1$ to $\mathcal{A}_2$. It is easy to prove that both $\leq_F$ and $\leq_{wF}$ are preorders, that is, reflexive and transitive. Since all simulation relations in this paper are forward simulations, we often omit the word "forward".

Proposition 2.2 *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be automata. Then:*

1. *If $\mathcal{A}_1 \leq_F \mathcal{A}_2$ then $\mathcal{A}_1 \leq_{wF} \mathcal{A}_2$.*
2. *If $H_1 = H_2 = \emptyset$, then $\mathcal{A}_1 \leq_F \mathcal{A}_2$ iff $\mathcal{A}_1 \leq_{wF} \mathcal{A}_2$.*
3. *If $\mathcal{A}_1 \leq_{wF} \mathcal{A}_2$ then $\mathcal{A}_1 \leq_T \mathcal{A}_2$.*

Proof. Standard; for instance, see [10]. □

2.4 Tree-Structured Nondeterministic Automata

An automaton is *tree-structured* if each state is reached via a unique execution.

The *unfolding* of automaton $\mathcal{A}$, denoted by $\mathit{Unfold}(\mathcal{A})$, is the tree-structured automaton $\mathcal{B}$ obtained from $\mathcal{A}$ by unfolding its transition graph into a tree. Formally,

- $Q_{\mathcal{B}} = \mathit{execs}^*(\mathcal{A})$,
- $\bar{q}_{\mathcal{B}} = \bar{q}_{\mathcal{A}}$,

- $E_{\mathcal{B}} = E_{\mathcal{A}}$,
- $H_{\mathcal{B}} = H_{\mathcal{A}}$, and
- $D_{\mathcal{B}} = \{(\alpha, a, \alpha a q) \mid (lstate(\alpha), a, q) \in D_{\mathcal{A}}\}$.

Proposition 2.3 *$\mathcal{A} \equiv_F Unfold(\mathcal{A})$.*

Proof. See [10]. It is easy to check that the relation R, where $\alpha \; R \; q$ iff $lstate(\alpha) = q$, is a forward simulation from $Unfold(\mathcal{A})$ to $\mathcal{A}$ and that the inverse relation of R is a forward simulation from $\mathcal{A}$ to $Unfold(\mathcal{A})$. □

Proposition 2.4 *$\mathcal{A} \equiv_T Unfold(\mathcal{A})$.*

Proof. By Proposition 2.3 and Proposition 2.2, Parts 1 and 3. □

3 Definitions and Basic Results for Probabilistic Automata

3.1 Preliminaries and Notation on Measure Theory

We recall a few basic definitions and results from measure theory that can be retrieved from any standard book on the subject [5].

A *σ-field* over a set X is a set $\mathcal{F} \subseteq 2^X$ that contains the empty set and is closed under complement and countable union. A pair $(X, \mathcal{F})$ where $\mathcal{F}$ is a σ-field over X, is called a *measurable space*. A measure on a measurable space (X, F) is a function $\mu : \mathcal{F} \to [0, \infty]$ that is countably additive: for each countable family $\{X_i\}_i$ of pairwise disjoint elements of $\mathcal{F}$, $\mu(\cup_i X_i) = \sum_i \mu(X_i)$. A *probability measure* on $(X, \mathcal{F})$ is a measure μ on $(X, \mathcal{F})$ such that $\mu(X) = 1$. A sub-probability measure on $(X, \mathcal{F})$ is a measure μ on $(X, \mathcal{F})$ such that $\mu(X) \leq 1$. A *discrete probability measure* on a set X is a probability measure μ on $(X, 2^X)$. A *discrete sub-probability measure* on X is a sub-probability measure μ on $(X, 2^X)$. We denote the set of discrete probability measures and discrete sub-probability measures on X by $Disc(X)$ and $SubDisc(X)$, respectively. We denote the support of a discrete measure μ, that is, the set of elements that have non-zero measure, by $supp(\mu)$. We let $\delta(q)$ denote the *Dirac measure* for q, the discrete probability measure that assigns probability 1 to $\{q\}$. Finally, if X is nonempty and finite, then $\mathcal{U}(X)$ denotes the *uniform distribution* over X, the measure that assigns probability $1/|X|$ to each element of X. Given two discrete probability measures μ_1, μ_2 on $(X, 2^X)$ and $(Y, 2^Y)$, respectively, we denote by $\mu_1 \times \mu_2$ the *product measure*, that is, the measure on $(X \times Y, 2^{(X \times Y)})$ such that $\mu_1 \times \mu_2((x, y)) = \mu_1(x)\mu_2(y)$ for each $x \in X, y \in Y$.

A function $f : X \to Y$ is said to be *measurable* from $(X, \mathcal{F}_X)$ to $(Y, \mathcal{F}_Y)$ if the inverse image of each element of $\mathcal{F}_Y$ is an element of $\mathcal{F}_X$, that is, for each $C \in F_Y$, $f^{-1}(C) \in \mathcal{F}_X$. In such a case, given a measure μ on $(X, \mathcal{F}_X)$, the function $f(\mu)$ defined on $\mathcal{F}_Y$ by $f(\mu)(C) = \mu(f^{-1}(C))$ for each $C \in \mathcal{F}_Y$ is a measure on $(Y, \mathcal{F}_Y)$ and is called the *image measure* of μ under f.

Given a countable collection of measures $\{\mu_i\}_i$ on $(X, \mathcal{F}_X)$ and a countable collection $\{p_i\}_i$ of real numbers in $[0, \infty)$, denote by $\sum_i p_i \mu_i$ a new function μ such that, for each element $C \in \mathcal{F}_X$, $\mu(C) = \sum_i p_i \mu_i(C)$. We state a few standard properties.

Proposition 3.1 *The following hold.*

1. *$\sum_i \mu_i$ is a measure on $(X, \mathcal{F}_X)$.*

2. *If each μ_i is a (sub)-probability measure and $\sum_i p_i = 1$, then $\sum_i p_i\mu_i$ is a (sub)-probability measure.*

3. *If f is a measurable function from $(X, \mathcal{F}_X)$ to $(Y, \mathcal{F}_Y)$, then $f(\sum_i p_i\mu_i) = \sum_i p_i f(\mu_i)$.*

3.2 Probabilistic Automata, Executions, and Traces

A *probabilistic automaton (PA)* is a tuple $\mathcal{P} = (Q, \bar{q}, E, H, D)$, where all components are exactly as for nondeterministic automata, except that:

- D, the *transition relation*, is a subset of $Q \times (E \cup H) \times Disc(Q)$.

We define A as before. We denote transition (q, a, μ) by $q \xrightarrow{a} \mu$. We assume finite branching: for each state q the number of pairs (a, μ) such that $q \xrightarrow{a} \mu$ is finite. Given a transition $tr = (q, a, \mu)$ we denote q by $source(tr)$ and μ either by $target(tr)$ or by μ_{tr}.

Thus, a probabilistic automaton differs from a nondeterministic automaton in that a transition leads to a probability measure over states rather than to a single state. A nondeterministic automaton is a special case of a probabilistic automaton, where the last component of each transition is a Dirac measure. Conversely, we can associate a nondeterministic automaton with each probabilistic automaton by replacing transition relation D by the relation D' given by

$$(q, a, q') \in D' \quad \Leftrightarrow \quad (\exists \mu)[(q, a, \mu) \in D \wedge \mu(q') > 0].$$

Using this correspondence, notions such as execution fragments and traces carry over from nondeterministic automata to probabilistic automata.[2]

A *scheduler* for a PA $\mathcal{P}$ is a function $\sigma : frags^*(\mathcal{P}) \to SubDisc(D)$ such that $tr \in supp(\sigma(\alpha))$ implies $source(tr) = lstate(\alpha)$. A scheduler σ is said to be *deterministic* if for each finite execution fragment α, either $\sigma(\alpha)(D) = 0$ or else $\sigma(\alpha) = \delta(tr)$ (the Dirac measure for tr) for some $tr \in D$.

A scheduler σ and a state q induce a measure ϵ on the σ-field generated by cones of execution fragments as follows. If α is a finite execution fragment, then the *cone* of α is defined by $C_\alpha = \{\alpha' \in frags(\mathcal{P}) \mid \alpha \leq \alpha'\}$. The measure ϵ of a cone C_α is defined to be 1 if $\alpha = q$, 0 if $\alpha = q' \neq q$, and, if α is of the form $\alpha' a q'$, it is defined by the recursive equation

$$\epsilon(C_\alpha) \quad = \quad \epsilon(C_{\alpha'}) \sum_{tr \in D(a')} \sigma(\alpha')(tr)\mu_{tr}(q'), \tag{1}$$

where $D(a')$ denotes the set of transitions of D that are labeled by a'. Standard measure theoretical arguments ensure that ϵ is well defined. We call the measure ϵ a *probabilistic execution fragment* of $\mathcal{P}$ and we say that ϵ is *generated* by σ and q_0. We call state q_0 the *first state* of ϵ and denote it by $fstate(\epsilon)$. If $fstate(\epsilon)$ is the start state $\bar{q}$, then ϵ is called a *probabilistic execution*.

The trace function is a measurable function from the σ-field generated by cones of execution fragments to the σ-field generated by cones of traces, where the cone of a finite trace β is defined by $C_\beta = \{\beta' \in E^* \cup E^\omega \mid \beta \leq \beta'\}$. Here $\leq$ denotes the prefix ordering on sequences. Given a probabilistic execution fragment ϵ, we define the *trace distribution* of ϵ, $tdist(\epsilon)$, to be the image measure of ϵ under *trace*. We denote the set of trace distributions of probabilistic executions of a

[2]The correspondence between nondeterministic automata and probabilistic automata is worked out in great detail in [2].

PA $\mathcal{P}$ by $tdists(\mathcal{P})$. We define the *trace distribution preorder* relation on probabilistic automata by: $\mathcal{P}_1 \leq_D \mathcal{P}_2$ iff $E_1 = E_2$ and $tdists(\mathcal{P}_1) \subseteq tdists(\mathcal{P}_2)$.

An example of a measurable set of traces that is used extensively throughout the paper is the set of traces in which a specific action a occurs. We denote this set by $\diamond a$. The inverse image under trace of $\diamond a$ can be expressed as a disjoint union of cones of executions. Thus, we have the following proposition.

Proposition 3.2 *Let η be the trace distribution of a probabilistic execution ϵ of a probabilistic automaton $\mathcal{P}$, and let Θ_a be the set of finite executions of $\mathcal{P}$ with a single occurrence of action a whose last transition is labeled by a. Then,*

$$\eta(\diamond a) = \sum_{\alpha \in \Theta_a} \epsilon(C_\alpha). \tag{2}$$

3.3 Combined Transitions and Weak Transitions

Let $\{q \xrightarrow{a} \mu_i\}_{i \in I}$ be a collection of transitions of a PA $\mathcal{P}$, 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 $\mathcal{P}$.

Consider a probabilistic execution fragment ϵ 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 induced by a deterministic scheduler, then $fstate(\epsilon) \stackrel{a}{\Longrightarrow} \mu$ is a *weak transition*. We refer to ϵ as a *representation* of $fstate(\epsilon) \stackrel{a}{\Longrightarrow} \mu$. Observe that the measure μ can be seen alternatively as the image measure of ϵ under *lstate*. This is an abuse of notation because *lstate* is not defined for infinite executions; however, since ϵ assigns measure 1 to the set of finite executions, we can extend arbitrarily and safely the definition of *lstate* to infinite executions for this purpose.

Proposition 3.3 *Let $\{tr_i\}_{i \in I}$ be a collection of weak combined transitions of a PA $\mathcal{P}$, all starting in the same state q, and all labeled by the same action a, and let $\{p_i\}_{i \in I}$ be probabilities such that $\sum_{i \in I} p_i = 1$. Then $\sum_{i \in I} p_i tr_i$ is a weak combined transition of $\mathcal{P}$ labeled by a.*

Proof. For each $i \in I$, let ϵ_i be a representation of tr_i, and σ_i be a scheduler that, together with state q, induces ϵ_i. We omit the index set I in the rest of the proof. Define a new scheduler σ as follows.

$$\sigma(\alpha) = \begin{cases} \displaystyle\sum_i \frac{p_i \epsilon_i(C_\alpha)}{\sum_i p_i \epsilon_i(C_\alpha)} \sigma_i(\alpha) & \text{if } \exists_i p_i \epsilon_i(C_\alpha) > 0 \\ \text{arbitrarily} & \text{otherwise.} \end{cases}$$

Let ϵ be the probabilistic execution fragment induced by σ and q. Let α be a finite execution fragment of $\mathcal{P}$. We prove by induction on the length of α that $\epsilon(C_\alpha) = \sum_i p_i \epsilon_i(C_\alpha)$. The base case is trivial since $\epsilon(C_q) = 1$ and for each i, $\epsilon_i(C_q) = 1$, which implies $\sum_i \epsilon_i(C_q) = 1$; similarly, for each state $q' \neq q$, $\epsilon(C_{q'}) = 0$ and for each i, $\epsilon_i(C_{q'}) = 0$. For the inductive step, let $\alpha = \alpha' a' q'$. If $\epsilon(C_{\alpha'}) = 0$, then, by induction, $\sum_i p_i \epsilon_i(C_{\alpha'}) = 0$, which implies that for each i, $p_i \epsilon_i(C_{\alpha'}) = 0$. By definition of measure of a cone, Equation (1), $\epsilon(C_\alpha) = 0$. Furthermore, for each i, if $p_i = 0$ then $p_i \epsilon_i(C_\alpha) = 0$ trivially, and if $p_i > 0$, then $\epsilon_i(C_{\alpha'}) = 0$ and by definition of measure of a

cone, Equation (1), $\epsilon_i(C_\alpha) = 0$, which implies $p_i\epsilon_i(C_\alpha) = 0$. Thus, $\sum_i p_i\epsilon_i(C_\alpha) = 0$ as needed. If $\epsilon(C_{\alpha'}) > 0$, then, by definition of measure of a cone, Equation (1),

$$\epsilon(C_\alpha) = \epsilon(C_{\alpha'}) \sum_{tr \in D(a')} \sigma(\alpha')(tr)\mu_{tr}(q').$$

By expanding $\sigma(\alpha')(tr)$ with the definition of σ we obtain

$$\epsilon(C_\alpha) = \epsilon(C_{\alpha'}) \sum_{tr \in D(a')} \left(\sum_i \frac{p_i\epsilon_i(C_{\alpha'})}{\sum_i p_i\epsilon_i(C_{\alpha'})} \sigma_i(\alpha')(tr) \right) \mu_{tr}(q'),$$

where we know that the denominator is strictly positive by hypothesis. By standard algebraic manipulations (exchanges of sums and rearrangements of constants) we obtain

$$\epsilon(C_\alpha) = \frac{\epsilon(C_{\alpha'})}{\sum_i p_i\epsilon_i(C_{\alpha'})} \sum_i \sum_{tr \in D(a')} p_i\epsilon_i(C_{\alpha'})\sigma_i(\alpha')(tr)\mu_{tr}(q').$$

By induction, $\epsilon(C_{\alpha'}) = \sum_i p_i\epsilon_i(C_{\alpha'})$. Thus, by simplifying (removing) the leftmost term and rearranging constants we obtain

$$\epsilon(C_\alpha) = \sum_i \left(p_i\epsilon_i(C_{\alpha'}) \sum_{tr \in D(a')} \sigma_i(\alpha')(tr)\mu_{tr}(q') \right).$$

Finally, by definition of measure of a cone, Equation (1), we get the desired equation

$$\epsilon(C_\alpha) = \sum_i p_i\epsilon_i(C_\alpha).$$

Thus, $\epsilon = \sum_i p_i\epsilon_i$, which implies that the probability of termination in ϵ is 1. Furthermore, by Proposition 3.1, Item 3, $lstate(\epsilon) = \sum_i p_i\, lstate(\epsilon_i)$. That is, ϵ is a representation of $\sum_i p_i tr_i$. □

3.4 Composition

Two PAs, $\mathcal{P}_1$ and $\mathcal{P}_2$, are *compatible* if $H_1 \cap A_2 = A_1 \cap H_2 = \emptyset$. The *composition* of two compatible PAs $\mathcal{P}_1$ and $\mathcal{P}_2$, denoted by $\mathcal{P}_1 \| \mathcal{P}_2$, is the PA $\mathcal{P} = (Q_1 \times Q_2, (\bar{q}_1, \bar{q}_2), E_1 \cup E_2, H_1 \cup H_2, D)$ where D is the set of triples $(q, a, \mu_1 \times \mu_2)$ such that, for $i \in \{1, 2\}$:

$$a \in A_i \Rightarrow (\pi_i(q), a, \mu_i) \in D_i \text{ and } a \notin A_i \Rightarrow \mu_i = \delta(\pi_i(q)).$$

Let ϵ be a probabilistic execution (fragment) of $\mathcal{P}_1 \| \mathcal{P}_2$ and let $i \in \{1, 2\}$. Define $\pi_i(\epsilon)$, the i^{th} projection of ϵ, to be the image measure under π_i of ϵ. It is easy to verify that the projection function is measurable. When convenient, we denote a projection by $\epsilon \lceil \mathcal{P}_i$, where $\mathcal{P}_i$ is the PA that appears in the i^{th} position.

Proposition 3.4 *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be compatible PAs and let ϵ be a probabilistic execution (fragment) of $\mathcal{P}_1 \| \mathcal{P}_2$. Then for each $i \in \{1, 2\}$, $\pi_i(\epsilon)$ is a probabilistic execution (fragment) of $\mathcal{P}_i$.*

Proof. By Propositions 4.3.4 and 4.3.5 of [13]. □

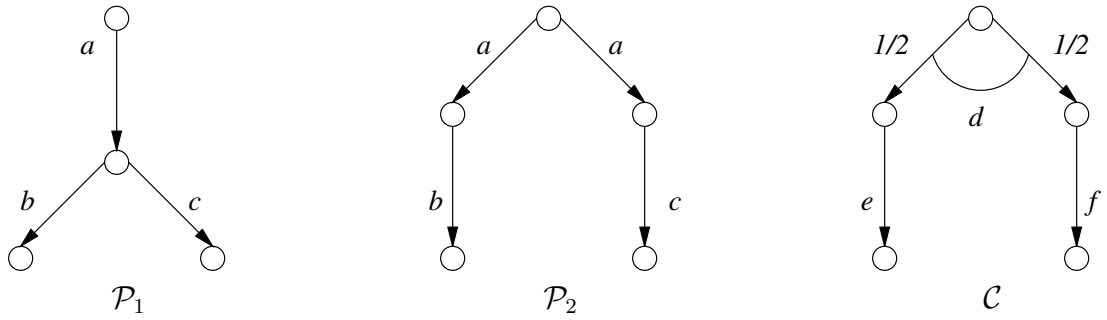

Figure 1: Trace distribution inclusion is not preserved by composition (without communication).

The trace distribution preorder is not preserved by composition [15, 17] as is shown by the following example.

Example 3.1 Failure of compositionality

Consider the two automata $\mathcal{P}_1$ and $\mathcal{P}_2$ of Figure 1. The two automata are trace equivalent, and it is easy to see that they are also trace distribution equivalent. Now consider the compositions $\mathcal{P}_1 \| \mathcal{C}$ and $\mathcal{P}_2 \| \mathcal{C}$, where $\mathcal{C}$ is the probabilistic automaton of Figure 1 and we assume that the actions of $\mathcal{C}$ are not shared with $\mathcal{P}_1$ and $\mathcal{P}_2$. It is possible to build a probabilistic execution of $\mathcal{P}_1 \| \mathcal{C}$ as follows: first a is scheduled followed by d; then e or f is scheduled depending on the outcome state of the transition labeled by d; finally, b or c is scheduled depending on whether e or f was scheduled. Thus, in the resulting trace distribution there is a total correlation between e, b and f, c, respectively. The same trace distribution cannot be obtained from $\mathcal{P}_2 \| \mathcal{C}$ because after scheduling the transition labeled by a we are already bound to b or c, and thus the occurrence of b or c cannot be correlated to e or f in this case.

Example 3.1 may appear pathological since, in the probabilistic execution of $\mathcal{P}_1 \| \mathcal{C}$ that correlates the choices between e and f and between b and c, a nondeterministic choice of $\mathcal{P}_1$ is resolved based on information that is not available to $\mathcal{P}_1$. This may lead us to propose a naive solution to the non-preservation of trace distribution inclusion by parallel composition where we require that each probabilistic automaton in a parallel composition can resolve its nondeterministic choices based on local knowledge only. However, a more elaborate example shows that this naive idea also does not work.

Example 3.2 Failure of compositionality

Consider the two automata $\mathcal{P}_1$ and $\mathcal{P}_2$ of Figure 2, which are essentially the automata of Example 3.1 where self-loop transitions labeled by e and f are added to each state. In this case the context $\mathcal{C}$ synchronizes with $\mathcal{P}_1$ and $\mathcal{P}_2$ on actions e and f, and $\mathcal{P}_1$ is able to learn which of e or f occurs, thus determining the correlation with b and c based on local knowledge only.

The solution of resolving nondeterminism based on local knowledge is adopted in [4] for a probabilistic extension of reactive modules; however the idea of [4] cannot be extended easily to probabilistic automata because of key structural differences in the models: in probabilistic automata

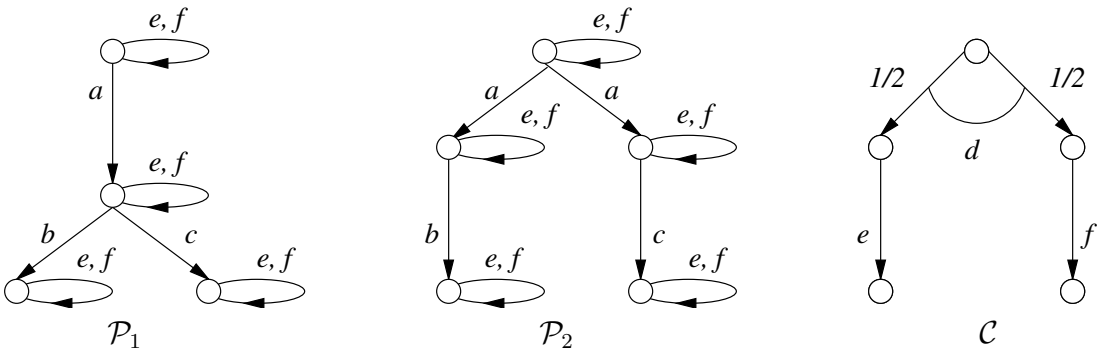

Figure 2: Trace distribution inclusion is not preserved by composition (with communication).

there is a total interleaving of the transitions taken by different automata in a parallel composition, while in probabilistic reactive modules there are several independent *atoms* that are not forced to interleave. A direct adaptation of the idea of [4] to probabilistic automata would require drastic modifications of the model that go beyond the scope of this paper: transitions should be labeled by sets of actions and should be structured in such a way that each action affects different parts of the state.

An alternative approach, followed in [13] and adopted in this paper, consists of defining a new *trace distribution precongruence* relation, denoted by $\leq_{DC}$, as the coarsest precongruence that is included in the trace distribution preorder $\leq_D$, and finding alternative characterizations of $\leq_{DC}$. It is known from [13] that there exists a simple context, called the *principal context*, that is sufficiently powerful to distinguish all probabilistic automata that are not in the trace distribution precongruence relation; alternatively, a testing scenarios is proposed in [14].

In this paper we characterize $\leq_{DC}$ in terms of probabilistic simulation relations. Another simple alternative characterization of $\leq_{DC}$ that is useful for our study is given by the following proposition.

Proposition 3.5 *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be PAs. Then $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$ iff for every PA $\mathcal{C}$ that is compatible with both $\mathcal{P}_1$ and $\mathcal{P}_2$, $\mathcal{P}_1 \| \mathcal{C} \leq_D \mathcal{P}_2 \| \mathcal{C}$.*

Proof. Define relation $\sqsubseteq$ such that $\mathcal{P}_1 \sqsubseteq \mathcal{P}_2$ iff $\mathcal{P}_1$ and $\mathcal{P}_2$ have the same external actions and for every PA $\mathcal{C}$ that is compatible with both $\mathcal{P}_1$ and $\mathcal{P}_2$, $\mathcal{P}_1 \| \mathcal{C} \leq_D \mathcal{P}_2 \| \mathcal{C}$.

Let $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$ and let $\mathcal{C}$ be a PA compatible with both $\mathcal{P}_1$ and $\mathcal{P}_2$. Since $\leq_{DC}$ is a precongruence by definition, then $\mathcal{P}_1 \| \mathcal{C} \leq_{DC} \mathcal{P}_2 \| \mathcal{C}$. Since, again by definition, $\leq_{DC}$ is included in $\leq_D$, then $\mathcal{P}_1 \| \mathcal{C} \leq_D \mathcal{P}_2 \| \mathcal{C}$. Thus, $\mathcal{P}_1 \sqsubseteq \mathcal{P}_2$, which implies that $\leq_{DC}$ is included in $\sqsubseteq$.

Conversely, observe that $\sqsubseteq$ is reflexive and transitive, and thus a preorder relation. Observe also that, by using a trivial context $\mathcal{C}$ with no external actions and no transitions, $\sqsubseteq$ is included in $\leq_D$. Finally, using the associativity of parallel composition, observe that $\sqsubseteq$ is preserved by parallel composition, and thus is a precongruence. This means that $\sqsubseteq$ is a precongruence included in $\leq_D$. Since $\leq_{DC}$ is the coarsest precongruence included in $\leq_D$, we get that $\sqsubseteq$ is included in $\leq_{DC}$. $\square$

3.5 Simulation Relations

The definitions of forward simulation and weak forward simulation in Section 2 can be extended naturally to PAs [15]. However, Segala has shown [12] that the resulting simulations are not complete for $\leq_{DC}$, and has defined new candidate simulations. These new simulations relate states to probability measures on states.

In order to define the new simulations formally, we need three new concepts. First we show how to *lift* a relation between sets to a relation between measures over sets [6]. 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.

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

Third and finally, we lift the notion of a transition to a *hyper-transition* [17] that begins and ends with a probability measure over states. Thus, let $\mathcal{P}$ be a PA and let $\mu \in \mathit{Disc}(Q)$. For each $q \in \mathit{supp}(\mu)$, let $q \xrightarrow{a} \mu_q$ be a combined transition of $\mathcal{P}$. Let μ' be $\sum_{q \in \mathit{supp}(\mu)} \mu(q)\mu_q$. Then $\mu \xrightarrow{a} \mu'$ 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 $\mathcal{P}$.

We now 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 PA $\mathcal{P}_1$ to PA $\mathcal{P}_2$ iff $E_1 = E_2$ 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 $q_1 \xrightarrow{a} \mu_1'$, there exists a measure $\mu_2' \in \mathit{Disc}(\mathit{Disc}(Q_2))$ such that $\mu_1' \; R \; \mu_2'$ and such that $\mu_2 \xrightarrow{a} \mathit{flatten}(\mu_2')$ (resp., $\mu_2 \stackrel{a}{\Longrightarrow} \mathit{flatten}(\mu_2')$) is a hyper-transition (resp., a weak hyper-transition) of $\mathcal{P}_2$.

We write $\mathcal{P}_1 \leq_{PF} \mathcal{P}_2$ (resp., $\mathcal{P}_1 \leq_{wPF} \mathcal{P}_2$) whenever there is a probabilistic forward simulation (resp., a weak probabilistic forward simulation) from $\mathcal{P}_1$ to $\mathcal{P}_2$. Note that a forward simulation between nondeterministic automata is a probabilistic forward simulation between the two automata viewed as PAs:

Proposition 3.6 *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be nondeterministic automata. Then:*

1. *$\mathcal{A}_1 \leq_F \mathcal{A}_2$ iff $\mathcal{A}_1 \leq_{PF} \mathcal{A}_2$, and*
2. *$\mathcal{A}_1 \leq_{wF} \mathcal{A}_2$ iff $\mathcal{A}_1 \leq_{wPF} \mathcal{A}_2$.*

Proof. The left-to-right inclusions are easy since, given a (weak) forward simulation R from $\mathcal{A}_1$ to $\mathcal{A}_2$, it is immediate to observe that the relation $R' \stackrel{\Delta}{=} \{(q_1, \delta(q_2)) \mid q_1 \; R \; q_2\}$ is a (weak) probabilistic forward simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$.

For the converse implication, let R be a (weak) forward simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$. Define a relation $R' \triangleq \{(q_1, q_2) \mid \exists_\mu q_1 \ R \ \mu, q_2 \in \mathit{supp}(\mu)\}$. We show that R' is a (weak) forward simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$.

The start condition is trivial since $\bar{q}_1 \ R \ \delta(\bar{q}_2)$, and thus $q_1 \ R' \ q_2$. For the step condition, let $q_1 \ R' \ q_2$, and let $q_1 \xrightarrow{a} q_1'$. By definition of R', there exists a measure μ such that $q_1 \ R \ \mu$ and $q_2 \in \mathit{supp}(\mu)$. Since R is a (weak) forward simulation, there exists a hyper-transition $\mu \xrightarrow{a} \mu'$ (a weak hyper-transition $\mu \stackrel{a}{\Longrightarrow} \mu'$) where μ' is the flattening of some measure μ'' such that $\delta(q_1') \ R \ \mu''$. Observe that, since $\mu' = \mathit{flatten}(\mu'')$, each element $q_2' \in \mathit{supp}(\mu')$ is also in the support of some measure $\rho \in \mathit{supp}(\mu'')$. Thus, $q_1' \ R \ \rho$, and, by definition of R', $q_1' \ R' \ q_2'$. Observe also that, by definition of hyper-transition, there is a combined transition $q_2 \xrightarrow{a} \mu_2$ (a weak combined transition $q_2 \stackrel{a}{\Longrightarrow} \mu_2$) such that $\mathit{supp}(\mu_2) \subseteq \mathit{supp}(\mu')$. Thus, by choosing any of the transitions that are combined in $q_2 \xrightarrow{a} \mu_2$, we obtain a transition $q_2 \xrightarrow{a} q_2'$ such that $q_1' \ R' \ q_2'$ as needed. Similarly, for the weak case, it is enough to consider a scheduler σ that generates $q_2 \stackrel{a}{\Longrightarrow} \mu_2$ and replace it by a new scheduler $\sigma'(\alpha)$ that stops (does not return any transition) if $\sigma(\alpha)$ stops with some non-zero probability, and chooses any transition in $\mathit{supp}(\alpha)$ that reduces the distance from a stopping point otherwise. This leads to a weak transition $q_2 \stackrel{a}{\Longrightarrow} q_2'$ such that $q_1' \ R' \ q_2'$ as needed. □

Proposition 3.7 *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be PAs. Then:*

1. *If $\mathcal{P}_1 \leq_{PF} \mathcal{P}_2$ then $\mathcal{P}_1 \leq_{wPF} \mathcal{P}_2$.*
2. *If $H_1 = H_2 = \emptyset$ then $\mathcal{P}_1 \leq_{PF} \mathcal{P}_2$ iff $\mathcal{P}_1 \leq_{wPF} \mathcal{P}_2$.*
3. *If $\mathcal{P}_1 \leq_{wPF} \mathcal{P}_2$ then $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$.*

Proof. The first item follows from the fact that a combined transition is a special case of a weak combined transition; the second item follows from the fact that in the absence of internal actions a weak combined transition is a combined transition. For the third item see Proposition 8.7.1 of [13]. □

3.6 Tree-Structured Probabilistic Automata

A *path* of a PA $\mathcal{P}$ is a finite sequence $\gamma = q_0 a_1 \mu_1 q_1 a_2 \mu_2 q_2 \ldots q_n$ of alternating states, actions and distribution over states, starting with the start state of $\mathcal{P}$ such that for each non-final i, $q_i \xrightarrow{a_{i+1}} \mu_{i+1}$ and $q_{i+1} \in \mathit{supp}(\mu_{i+1})$. We write $\mathit{lstate}(\gamma)$ to denote q_n and $\mathit{paths}(\mathcal{P})$ for the set of all path of $\mathcal{P}$. We say that $\mathcal{P}$ is *tree-structured* if each state is reached via a unique path. Tree-structured probabilistic automata are characterized uniquely by the property that all states are reachable, the start state does not occur in the target of any transition, and each of the other states occurs in the target of exactly one transition. Also tree-structured nondeterministic automata are characterized uniquely by this property, albeit for a different notion of transition.

If a probabilistic automaton is tree-structured then its underlying automaton is also tree-structured. The following example shows that the converse does not hold.

Example 3.3 Non-tree-structured probabilistic automata

Figure 3 shows a probabilistic automaton that is not tree-structured, as state q' can be reached via two different paths. The underlying automaton is tree-structured, however, since the only way to reach state q' is via the execution qaq'.

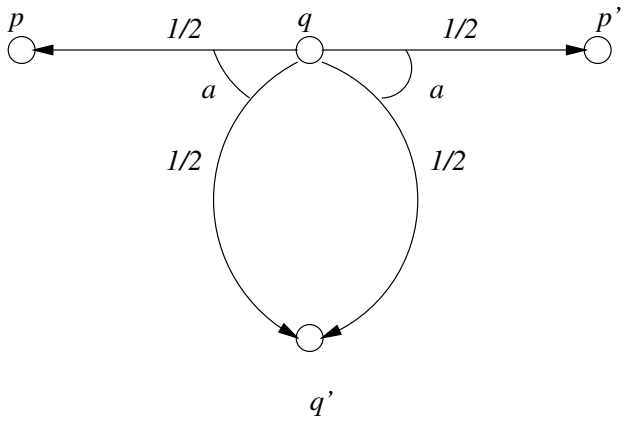

Figure 3: A PA that not tree-structured even though its underlying automaton is.

The *unfolding* of a probabilistic automaton $\mathcal{P}$, denoted by $Unfold(\mathcal{P})$, is the tree-structured probabilistic automaton $\mathcal{Q}$ obtained from $\mathcal{P}$ by unfolding its transition graph into a tree. Formally,

- $Q_{\mathcal{Q}} = paths(\mathcal{P})$,
- $\bar{q}_{\mathcal{Q}} = \bar{q}_{\mathcal{P}}$,
- $E_{\mathcal{Q}} = E_{\mathcal{P}}$,
- $H_{\mathcal{Q}} = H_{\mathcal{P}}$, and
- $D_{\mathcal{Q}} = \{(\gamma, a, \mu) \mid (\exists \mu')[(lstate(\gamma), a, \mu') \in D_{\mathcal{P}} \wedge (\forall q \in supp(\mu'))[\mu'(q) = \mu(\gamma a \mu' q)]]\}$.

Proposition 3.8 $\mathcal{P} \equiv_{PF} Unfold(\mathcal{P})$.

Proof. It is easy to check that the relation R where $\alpha \; R \; \delta(q)$ iff $lstate(\alpha) = q$ is a probabilistic forward simulation from $Unfold(\mathcal{P})$ to $\mathcal{P}$ and that the "inverse" of R, i.e., the relation R' such that $q \; R' \; \delta(\alpha)$ iff $\alpha \; R \; \delta(q)$, is a probabilistic forward simulation from $\mathcal{P}$ to $Unfold(\mathcal{P})$. □

Proposition 3.9 $\mathcal{P} \equiv_{DC} Unfold(\mathcal{P})$.

Proof. By Proposition 3.8, and Proposition 3.7, Parts 1 and 3. □

3.7 Truncations and Continuations

We now define two simple constructions on probabilistic execution fragments that will be useful for our proofs. Specifically, we define the truncation of a probabilistic execution fragment, which is the result of stopping the computation at some designated points, and the continuation of a probabilistic execution fragment, which represents the rest of a probabilistic execution fragment after some finite execution fragment has occurred.

Let ϵ be a probabilistic execution fragment of a PA $\mathcal{P}$, generated by some scheduler σ, and let Θ be a set of finite execution fragments of $\mathcal{P}$. Define the *truncation of* ϵ *at* Θ to be the same as ϵ except that no transition is scheduled from all the Θ places, that is, the probabilistic execution fragment ϵ', with the same start state as ϵ, generated by a new scheduler σ' such that $\sigma'(\alpha) = \sigma(\alpha)$ if $\alpha \notin \Theta$ and $\sigma'(\alpha)(D) = 0$ if $\alpha \in \Theta$.

Proposition 3.10 *The definition of truncation of a probabilistic execution fragment ϵ is independent of the choice of the inducing scheduler.*

Proof. Let q be the first state of ϵ and let σ_1, σ_2 be two schedulers that, together with q, induce ϵ. Let Θ be a set of finite execution fragments of $\mathcal{P}$, and let σ_1', σ_2' be the schedulers built from σ_1, σ_2, respectively, according to the definition of truncation. Let ϵ_1, ϵ_2 be the induced probabilistic execution fragments, and suppose by contradiction that $\epsilon_1 \neq \epsilon_2$. Then there exists a finite execution α such that $\epsilon_1(C_\alpha) \neq \epsilon_2(C_\alpha)$. Consider such a finite execution α of minimum length. Observe that $|\alpha| > 0$ since $\epsilon(C_q) = \epsilon_1(C_q) = \epsilon_2(C_q) = 1$ and, for each state $q' \neq q$, $\epsilon(C_{q'}) = \epsilon_1(C_{q'}) = \epsilon_2(C_{q'}) = 0$. Thus, $\alpha = \alpha' a' q'$ for some α', a', q', where $\epsilon_1(C_{\alpha'}) = \epsilon_2(C_{\alpha'})$. We distinguish two cases.

If $\alpha' \in \Theta$, then, by definition of σ_1' and σ_2', $\sigma_1'(\alpha')(D) = \sigma_2'(\alpha')(D) = 0$. Thus, $\epsilon_1(C_\alpha) = \epsilon_2(C_\alpha) = 0$, a contradiction.

If $\alpha' \notin \Theta$, then, by definition of σ_1' and σ_2', $\sigma_1'(\alpha') = \sigma_1(\alpha')$ and $\sigma_2'(\alpha') = \sigma_2(\alpha')$. Since σ_1 and σ_2 induce the same probabilistic execution fragment ϵ, by definition of measure of a cone, Equation (1), $\sum_{tr \in D(a')} \sigma_1(\alpha')(tr)\mu_{tr}(q') = \sum_{tr \in D(a')} \sigma_2(\alpha')(tr)\mu_{tr}(q')$. Thus, it is also the case that $\sum_{tr \in D(a')} \sigma_1'(\alpha')(tr)\mu_{tr}(q') = \sum_{tr \in D(a')} \sigma_2'(\alpha')(tr)\mu_{tr}(q')$. By definition of measure of a cone, Equation (1), $\epsilon_1(C_\alpha) = \epsilon_2(C_\alpha)$, again a contradiction. □

Proposition 3.11 *The truncation of ϵ at Θ is a probabilistic execution fragment of $\mathcal{P}$.*

Proof. Trivial since the definition of truncation provides the generating scheduler. □

Let ϵ be a probabilistic execution fragment of a PA $\mathcal{P}$, generated by a scheduler σ, and let α be a finite execution fragment with the same start state as ϵ. Define $\epsilon \triangleright \alpha$, the *rest of ϵ after prefix α*, to be the probabilistic execution fragment generated by the following scheduler σ' from $lstate(\alpha)$:

$$\sigma'(\alpha') = \begin{cases} \sigma(\alpha \frown \alpha') & \text{if } fstate(\alpha') = lstate(\alpha) \\ \sigma(\alpha') & \text{otherwise} \end{cases}$$

Observe that the second line in the definition of σ' is irrelevant, and thus can be replaced by any arbitrary expression, since the execution fragment generated by σ' from $lstate(\alpha)$ depends only on σ' applied to execution fragments that start from $lstate(\alpha)$.

Proposition 3.12 *The definition of $\epsilon \triangleright \alpha$ is independent of the choice of the inducing scheduler.*

Proof. Let q be the first state of ϵ and let σ_1, σ_2 be two schedulers that, together with q, induce ϵ. Let q' be $lstate(\alpha)$. Let σ_1', σ_2' be the schedulers built from σ_1, σ_2, respectively, according to the definition of $\epsilon \triangleright \alpha$. Let ϵ_1, ϵ_2 be the induced probabilistic execution fragments from q', and suppose by contradiction that $\epsilon_1 \neq \epsilon_2$. Then there exists a finite execution α' such that $\epsilon_1(C_{\alpha'}) \neq \epsilon_2(C_{\alpha'})$. Consider such a finite execution α' of minimum length. Observe that $|\alpha'| > 0$ since $\epsilon(C_{q'}) = \epsilon_1(C_{q'}) = \epsilon_2(C_{q'}) = 1$ and, for each state $q'' \neq q'$, $\epsilon(C_{q''}) = \epsilon_1(C_{q''}) = \epsilon_2(C_{q''}) = 0$. Thus, $\alpha' = \alpha'' a'' q''$ for some α'', a'', q'', where $\epsilon_1(C_{\alpha''}) = \epsilon_2(C_{\alpha''})$. We distinguish two cases.

If $fstate(\alpha'') \neq q'$, then, by definition of ϵ_1 and ϵ_2, $\epsilon_1(C_{\alpha'}) = \epsilon_2(C_{\alpha'}) = 0$, a contradiction.

If $fstate(\alpha'') = q'$, then, by definition of σ_1' and σ_2', $\sigma_1'(\alpha'') = \sigma_1(\alpha \frown \alpha'')$ and $\sigma_2'(\alpha'') = \sigma_2(\alpha \frown \alpha'')$. Since σ_1 and σ_2 induce the same probabilistic execution fragment ϵ, by definition of measure of a cone, Equation (1), $\sum_{tr \in D(a'')} \sigma_1(\alpha \frown \alpha'')(tr)\mu_{tr}(q'') = \sum_{tr \in D(a'')} \sigma_2(\alpha \frown \alpha'')(tr)\mu_{tr}(q'')$. Thus, we derive $\sum_{tr \in D(a'')} \sigma_1'(\alpha'')(tr)\mu_{tr}(q'') = \sum_{tr \in D(a'')} \sigma_2'(\alpha'')(tr)\mu_{tr}(q'')$. By definition of measure of a cone, Equation (1), $\epsilon_1(C_{\alpha'}) = \epsilon_2(C_{\alpha'})$, again a contradiction. □

Proposition 3.13 *The following properties are valid.*

1. $\epsilon \triangleright \alpha$ *is a probabilistic execution fragment of* $\mathcal{P}$.
2. *For each finite execution* α' *with* $lstate(\alpha) = fstate(\alpha')$, $\epsilon(C_{\alpha^\frown\alpha'}) = \epsilon(C_\alpha) \cdot (\epsilon \triangleright \alpha)(C_{\alpha'})$.

Proof. The first item is trivial since the definition of $\triangleright$ provides the generating scheduler. The second item follows directly from the definition of the probability of a cone. □

4 Dual Probabilistic Automata

The proofs of our completeness results rely on a special context for a probabilistic automaton, which we call its *dual probabilistic automaton*. The dual automaton, $dual(\mathcal{P})$, of a PA $\mathcal{P}$ can observe the states $\mathcal{P}$ goes through and the transitions that are scheduled during a probabilistic execution. This information is revealed by means of externally visible transitions of $dual(\mathcal{P})$ with the help of a specific scheduler that synchronizes $\mathcal{P}$ with its dual.

In this section we introduce the construction of a dual probabilistic automaton, we introduce the scheduler that synchronizes a probabilistic automaton with its dual, and we prove some results about the resulting trace distributions.

Informally, the dual of a probabilistic automaton $\mathcal{P}$ is a probabilistic automaton $\mathcal{C}$ whose states include a distinguished start state, all the states of $\mathcal{P}$, and all the transitions of $\mathcal{P}$. Automaton $\mathcal{C}$ has a special transition from its own start state, $\bar{q}_\mathcal{C}$, to the start state of $\mathcal{P}$, $\bar{q}_\mathcal{P}$, labeled by $\bar{q}_\mathcal{P}$. Also, from every state q of $\mathcal{P}$, $\mathcal{C}$ has a uniform transition labeled by ch to the set of transitions of $\mathcal{P}$ that begin in state q. Finally, for every transition tr of $\mathcal{P}$, and every state q in the support of μ_{tr}, $\mathcal{C}$ has a transition labeled by q from tr to q.

Definition 4.1 *The* dual probabilistic automaton *of a PA* $\mathcal{P}$ *is a PA* $\mathcal{C}$ *such that*

- $Q_\mathcal{C} = \{\bar{q}_\mathcal{C}\} \cup Q_\mathcal{P} \cup D_\mathcal{P}$,
- $E_\mathcal{C} = Q_\mathcal{P} \cup \{ch\}$,
- $H_\mathcal{C} = \emptyset$, *and*
- $D_\mathcal{C} = \{(\bar{q}_\mathcal{C}, \bar{q}_\mathcal{P}, \bar{q}_\mathcal{P})\} \cup$
  $\{(q, ch, \mathcal{U}(\{tr \in D_\mathcal{P} \mid source(tr) = q\})) \mid q \in Q_\mathcal{P} \wedge q \rightarrow\} \cup$
  $\{(tr, q, q) \mid tr \in D_\mathcal{P}, q \in supp(\mu_{tr})\}$.

Observe that the dual of an ordinary nondeterministic automaton enables at most one transition from each state. Indeed, the only states that may enable more than one transition are the states of the form $tr \in D_\mathcal{P}$, which enable one transition for each state in $supp(\mu_{tr})$. However, the size of $supp(\mu_{tr})$ is 1 in an ordinary automaton.

We assume without loss of generality that a probabilistic automaton $\mathcal{P}$ and its dual do not have any actions in common (otherwise we can simply rename states of $\mathcal{P}$ to achieve our goal), and thus $\mathcal{P}$ and its dual are compatible.

Since $\mathcal{C}$ and $\mathcal{P}$ share no actions, merely composing $\mathcal{C}$ with $\mathcal{P}$ does not ensure that $\mathcal{C}$ faithfully emulates the behavior of $\mathcal{P}$. However, an appropriate scheduler can synchronize the two automata and ensure such an emulation, which will be sufficient for our purposes. Given a probabilistic

automaton $\mathcal{P}$ and its dual $\mathcal{C}$, we define a scheduler σ for $\mathcal{P}\|\mathcal{C}$, called the *observer* of $\mathcal{P}$, that synchronizes the two automata so that the internal structure of $\mathcal{P}$ is visible in the trace. Specifically, the scheduler σ starts by scheduling the transition of $\mathcal{C}$ from the start state of $\mathcal{C}$ to the start state of $\mathcal{P}$, leading to state $(\bar{q}, \bar{q})$, which is of the form (q, q). Then σ repeats the following as long as $q \rightarrow$:

1. Schedule the ch transition of $\mathcal{C}$, thus choosing a transition tr of $\mathcal{P}$.
2. Schedule transition tr of $\mathcal{P}$, leading $\mathcal{P}$ to a new state q'.
3. Schedule the transition of $\mathcal{C}$ labeled by the state q', resulting in the state (q', q'), which is again of the form (q, q).

Scheduler σ induces a trace distribution η for $\mathcal{P}\|\mathcal{C}$ where all states and external actions of $\mathcal{P}$ appear explicitly.

We state and prove some properties of η. The first property, Equation (3), says that the cone of traces beginning with the start state of $\mathcal{P}$ has probability 1. The second property, Equation (4), says that for any state q of $\mathcal{P}$ from which some transition is enabled and for each finite trace β of $\mathcal{P}\|\mathcal{C}$, the probability of the cone of traces beginning with βq is the same as the probability of the cone beginning with $\beta q\, ch$, that is, once βq occurs, the probability that ch follows is 1. The third property, Equation (5), says that for any state q of $\mathcal{P}$ and for each finite trace β of $\mathcal{P}\|\mathcal{C}$, the probability of the cone of traces beginning with $\beta q\, ch$ is the same as the sum of the probabilities of the cones beginning with $\beta q\, ch\, \beta'$ where β' represents one single step of $\mathcal{P}$ from q, that is, once ch occurs, one of the transitions of $\mathcal{A}$ that are enabled from q is exposed. The right-hand side of Equation (5) consists of two parts dealing with external and internal transitions, respectively.

Proposition 4.2 *The trace distribution η induced by the observer of a probabilistic automaton $\mathcal{P}$ satisfies the following three properties, for all finite traces β of $\mathcal{P}\|\mathcal{C}$ and for all states q of $\mathcal{P}$:*

$$\eta(C_{\bar{q}}) = 1 \tag{3}$$

$$q \rightarrow \quad \Longrightarrow \quad \eta(C_{\beta q}) = \eta(C_{\beta q\, ch}) \tag{4}$$

$$\begin{aligned} \eta(C_{\beta q\, ch}) \quad = \quad & \sum_{(a,q')\mid a\in E, (\exists \mu)[(q,a,\mu)\in D, q'\in supp(\mu)]} \eta(C_{\beta q\, ch\, aq'}) + \\ & \sum_{q'\mid (\exists (a,\mu))[a\in H, (q,a,\mu)\in D, q'\in supp(\mu)]} \eta(C_{\beta q\, ch\, q'}) \end{aligned} \tag{5}$$

Proof. Equation (3) follows from the fact that σ schedules action $\bar{q}$ immediately. Equation (4) follows from the fact that, after scheduling action q, thus leading to a state of the form (q, q), σ immediately schedules action ch if q enables at least one transition. Equation (5) follows from the fact that, after scheduling ch, σ schedules one of the transitions of $\mathcal{P}$ that are enabled from q, say $q \xrightarrow{a} \mu$, followed by a transition of $\mathcal{C}$ labeled by a state in $supp(\mu)$. □

Observe that Equation (5) has a simpler formulation in case $\mathcal{P}$ is an ordinary nondeterministic automaton.

Proposition 4.3 *The trace distribution η induced by the observer of an automaton $\mathcal{A}$ satisfies the following property, for all finite traces β of $\mathcal{A}\|\mathcal{C}$ and for all states q of $\mathcal{A}$:*

$$\eta(C_{\beta q\ ch}) = \sum_{(a,q')|a\in E, q\xrightarrow{a}q'} \eta(C_{\beta q\ ch\ aq'}) + \sum_{q'|(\exists a)a\in H, q\xrightarrow{a}q'} \eta(C_{\beta q\ ch\ q'}) \tag{6}$$

The following properties are not needed for the proof of completeness for nondeterministic automata, so the reader may skip them for the moment and return here when reading the proofs of Section 6.

Proposition 4.4 *Let η be the trace distribution induced by the observer of a tree-structured probabilistic automaton $\mathcal{P}$, and let $tr = (q, a, \mu)$ be a transition of $\mathcal{P}$. Let k be the number of transitions that are enabled from q in $\mathcal{P}$, and let q' be a state in $supp(\mu)$. Then the following properties hold:*

$$\eta(\diamond q') = \frac{\eta(\diamond q)}{k}\mu(q') \tag{7}$$

$$\eta(\diamond q') = \left(\sum_{q''\in supp(\mu)} \eta(\diamond q'')\right)\mu(q') \tag{8}$$

Proof. Let σ be the observer of $\mathcal{P}$, and let ϵ_σ be the probabilistic execution induced by σ. Since $\mathcal{P}$ is tree-structured, the set Θ_q contains a single execution α. Indeed, by definition of tree-structured, there is only one execution in $\mathcal{P}$ ending with state q, and σ simply interleaves this execution with transitions labeled by ch, by the names of the transitions of $\mathcal{P}$ that are needed to reach q, and by the names of the states that are reached. Similarly, $\Theta_{q'}$ contains a single execution α'.

Once state q is reached, σ schedules action ch, reaching state tr of C with probability $1/k$. Then, σ schedules transition tr, reaching state q' in $\mathcal{P}$ with probability $\mu(q')$, and finally σ schedules the transition of $\mathcal{C}$ labeled by q'. Thus, $\epsilon_\sigma(C_{\alpha'}) = \epsilon_\sigma(C_\alpha)(1/k)\mu(q')$. Then Equation (7) follows by Equation (2).

By summing over $supp(\mu)$ in Equation (7), we get

$$\sum_{q''\in supp(\mu)} \eta(\diamond q'') = \frac{\eta(\diamond q)}{k} \sum_{q''\in supp(\mu)} \mu(q'') \tag{9}$$

Observe that $\sum_{q''\in supp(\mu)} \mu(q'') = 1$. Thus, deriving $\eta(\diamond q)$ from Equation (9), replacing it in Equation (7), and cancelling k from numerator and denominator, we get Equation (8) as needed. □

5 Characterizations of the Trace Distribution Precongruence Relation for Nondeterministic Automata

In this section, we present our characterization theorems for $\leq_{DC}$ for nondeterministic automata: Theorem 5.2 characterizes $\leq_{DC}$ in terms of $\leq_F$, for automata without internal actions, and Theorem 5.4 characterizes $\leq_{DC}$ in terms of $\leq_{wF}$, for arbitrary nondeterministic automata. In each case, we prove the result first for tree-structured automata and then extend it to the non-tree-structured

case via unfolding. The interesting direction for each of these results is the completeness direction, showing that $\mathcal{A}_1 \leq_{DC} \mathcal{A}_2$ implies the existence of a simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$.

Our proofs of completeness for nondeterministic automata use the simple characterization in Proposition 3.5, applied with $\mathcal{C}$ equal to the dual probabilistic automaton of $\mathcal{A}_1$.

5.1 Nondeterministic Automata Without Internal Actions

We begin by considering nondeterministic automata without internal actions. We first consider tree-structured automata.

Proposition 5.1 *Let $\mathcal{A}_1$, $\mathcal{A}_2$ be nondeterministic automata without internal actions such that $\mathcal{A}_1$ tree-structured. Then $\mathcal{A}_1 \leq_{DC} \mathcal{A}_2$ implies $\mathcal{A}_1 \leq_F \mathcal{A}_2$.*

Proof. Assume that $\mathcal{A}_1 \leq_{DC} \mathcal{A}_2$. Let $\mathcal{C}$ be the dual probabilistic automaton of $\mathcal{A}_1$ and consider the observer σ_1 of $\mathcal{A}_1$ as defined in Section 4. Let η be the trace distribution induced by σ_1.

Since $\mathcal{A}_1 \leq_{DC} \mathcal{A}_2$, Proposition 3.5 implies that η is also a trace distribution of $\mathcal{A}_2 \| \mathcal{C}$. That is, there exists a probabilistic execution ϵ of $\mathcal{A}_2 \| \mathcal{C}$, induced by some scheduler σ_2, such that $tdist(\epsilon) = \eta$.

For each state q_1 in Q_1, let Θ_{q_1} be the set of finite executions of $\mathcal{A}_2 \| \mathcal{C}$ whose last transition is labeled by q_1. For each state q_2 of $\mathcal{A}_2$, let Θ_{q_1,q_2} be the set of executions in Θ_{q_1} whose last state is the pair (q_2, q_1).

Define a relation R as follows: $q_1 \; R \; q_2$ if and only if there exists a finite execution α in Θ_{q_1,q_2} such that $\epsilon(C_\alpha) > 0$. We claim that R is a forward simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$.

For the start condition, we must show that $\bar{q}_1 \; R \; \bar{q}_2$. Consider the start state $(\bar{q}_2, \bar{q}_\mathcal{C})$ of $\mathcal{A}_2 \| \mathcal{C}$. Since there are no internal actions in $\mathcal{A}_2$ or $\mathcal{C}$, and since, by Equation (3) from Proposition 4.2, $\eta(C_{\bar{q}_1}) = 1$, the only action that is scheduled initially by σ_2 is $\bar{q}_1$, leading to state $(\bar{q}_2, \bar{q}_1)$. Thus, the finite execution $\alpha = (\bar{q}_2, \bar{q}_\mathcal{C})\bar{q}_1(\bar{q}_2, \bar{q}_1)$ is an element of $\Theta_{\bar{q}_1,\bar{q}_2}$ such that $\epsilon(C_\alpha) > 0$, as needed.

For the step condition, assume $q_1 \; R \; q_2$ and let $q_1 \xrightarrow{a}_1 q_1'$ be a transition of $\mathcal{A}_1$, which we denote by tr for convenience. We exhibit a matching transition $q_2 \xrightarrow{a}_2 q_2'$.

By definition of R, there exists a finite execution α in Θ_{q_1,q_2}, such that $\epsilon(C_\alpha) > 0$. Since Θ_{q_1,q_2} is a subset of Θ_{q_1}, by definition of Θ_{q_1}, $trace(\alpha) = \beta q_1$ for some finite trace β. Therefore, $\eta(C_{\beta q_1}) > 0$. Since q_1 enables at least one transition in $\mathcal{A}_1$, specifically transition tr, Equation (4) from Proposition 4.2 implies that $\eta(C_{\beta q_1\, ch}) = \eta(C_{\beta q_1})$. Then, since $\mathcal{A}_2$ and $\mathcal{C}$ have no internal actions, σ_2 schedules action ch from α with probability 1.

By definition of $dual(\mathcal{A}_1)$, the transition labeled by ch that leaves from state q_1 of $\mathcal{C}$ leads to state tr with non-zero probability. Therefore, $\epsilon(C_{\alpha\, ch\, (q_2,tr)}) > 0$. By Equation (6) from Proposition 4.3, where only the first term of the right-hand side is used due to the absence of internal actions, σ_2 must extend $\alpha\, ch\, (q_2, tr)$ with two steps labeled by an action and a state of $\mathcal{A}_1$, respectively, where the action and the state are compatible with one of the transitions of $\mathcal{A}_1$ that are enabled from q_1. Since state tr of $\mathcal{C}$ enables only action q_1', and since, by the tree-structure of $\mathcal{A}_1$, a is uniquely determined by q_1', the action and state scheduled by σ_2 are a and q_1'. Therefore, there exists a state q_2' of $\mathcal{A}_2$ such that the execution $\alpha' = \alpha\, ch\, (q_2, tr) a (q_2', tr) q_1' (q_2', q_1')$ is an execution in $\Theta_{q_1',q_2'}$ such that $\epsilon(C_{\alpha'}) > 0$. Then $q_1' \; R \; q_2'$ and $q_2 \xrightarrow{a} q_2'$ as needed. □

Now we present our result for general (non-tree-structured) nondeterministic automata without internal actions.

Theorem 5.2 *Let $\mathcal{A}_1$, $\mathcal{A}_2$ be nondeterministic automata without internal actions. Then $\mathcal{A}_1 \leq_{DC} \mathcal{A}_2$ if and only if $\mathcal{A}_1 \leq_F \mathcal{A}_2$.*

Proof. First we prove soundness of forward simulations:

$$\begin{array}{lcl}
\mathcal{A}_1 \leq_F \mathcal{A}_2 & \Rightarrow & \text{(Proposition 3.6, Part 1)} \\
\mathcal{A}_1 \leq_{PF} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 3.7, Part 1)} \\
\mathcal{A}_1 \leq_{wPF} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 3.7, Part 3)} \\
\mathcal{A}_1 \leq_{DC} \mathcal{A}_2 \ . & &
\end{array}$$

Next we prove completeness:

$$\begin{array}{lcl}
\mathcal{A}_1 \leq_{DC} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 2.3)} \\
\mathit{Unfold}(\mathcal{A}_1) \leq_F \mathcal{A}_1 \leq_{DC} \mathcal{A}_2 & \Rightarrow & \text{(as in soundness proof)} \\
\mathit{Unfold}(\mathcal{A}_1) \leq_{DC} \mathcal{A}_1 \leq_{DC} \mathcal{A}_2 & \Rightarrow & (\leq_{DC} \text{ is transitive}) \\
\mathit{Unfold}(\mathcal{A}_1) \leq_{DC} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 5.1)} \\
\mathit{Unfold}(\mathcal{A}_1) \leq_F \mathcal{A}_2 & \Rightarrow & \text{(Proposition 2.3)} \\
\mathcal{A}_1 \leq_F \mathit{Unfold}(\mathcal{A}_1) \leq_F \mathcal{A}_2 & \Rightarrow & (\leq_F \text{ is transitive}) \\
\mathcal{A}_1 \leq_F \mathcal{A}_2 \ . & &
\end{array}$$

□

5.2 Nondeterministic Automata With Internal Actions

Next we extend the results of Section 5.1 to automata that may include internal actions. The proofs are analogous to those in Section 5.1. The difference is that, in several places in the proof of Proposition 5.3, we need to reason about multi-step extensions of executions instead of single-step extensions. Again, we begin with tree-structured automata.

Proposition 5.3 *Let $\mathcal{A}_1$, $\mathcal{A}_2$ be nondeterministic automata such that $\mathcal{A}_1$ is tree-structured. Then $\mathcal{A}_1 \leq_{DC} \mathcal{A}_2$ implies $\mathcal{A}_1 \leq_{wF} \mathcal{A}_2$.*

Proof. Assume that $\mathcal{A}_1 \leq_{DC} \mathcal{A}_2$. Define the dual probabilistic automaton $\mathcal{C}$ of $\mathcal{A}_1$, the observer σ_1, the trace distribution η, the scheduler σ_2, and the probabilistic execution ϵ as in the proof of Proposition 5.1. Without loss of generality we assume that σ_2 schedules action $\bar{q}_1$ with probability 1 from the start state of $\mathcal{A}_2 \| \mathcal{C}$ (essentially, since internal transitions are only transitions of $\mathcal{A}_2$ and the transition labeled by $\bar{q}_1$ is only a transition of $\mathcal{C}$, we can exchange any internal transitions of $\mathcal{A}_2$ that occur before the transition labeled by $\bar{q}_1$ with the transition labeled by $\bar{q}_1$ and reach exactly the same states with the same probabilities).

Define the Θ sets and the relation R as in the proof of Proposition 5.1. Now we claim that R is a weak forward simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$.

For the start condition, we must show that $\bar{q}_1 \ R \ \bar{q}_2$. Consider the start state $(\bar{q}_2, \bar{q}_{\mathcal{C}})$ of $\mathcal{A}_2 \| \mathcal{C}$. Since, by assumption, σ_2 schedules action $\bar{q}_1$ with probability 1 from the start state of $\mathcal{A}_1 \| \mathcal{C}$, the finite execution $\alpha = (\bar{q}_2, \bar{q}_{\mathcal{C}}) \bar{q}_1 (\bar{q}_2, \bar{q}_1)$ is an element of $\Theta_{\bar{q}_1, \bar{q}_2}$ such that $\epsilon(C_\alpha) > 0$, as needed.

For the step condition, assume $q_1 \ R \ q_2$ and let $q_1 \xrightarrow{a}_1 q_1'$ be a transition of $\mathcal{A}_1$, which we denote by tr. We exhibit a matching weak transition $q_2 \stackrel{a}{\Longrightarrow}_2 q_2'$.

By definition of R, there exists a finite execution α in Θ_{q_1,q_2} such that $\epsilon(C_\alpha) > 0$. Since Θ_{q_1,q_2} is a subset of Θ_{q_1}, by definition of Θ_{q_1}, $\mathit{trace}(\alpha) = \beta q_1$ for some finite trace β. Therefore,

$\eta(C_{\beta q_1}) > 0$. Since q_1 enables at least one transition in $\mathcal{A}_1$, specifically transition tr, Equation (4) from Proposition 4.2 implies that $\eta(C_{\beta q_1\ ch}) = \eta(C_{\beta q_1})$. Thus, there exists an execution fragment α' of $\mathcal{A}_2 \| \mathcal{C}$ with trace ch such that $\epsilon(C_{\alpha \frown \alpha'}) > 0$. Furthermore, since, by definition of $dual(\mathcal{A}_1)$, the transition of $\mathcal{C}$ labeled by ch that leaves from state q_1 leads to state tr with non-zero probability, we can assume that the last state of α' is of the form (q', tr) for some state q' of $\mathcal{A}_2$.

Since $\eta(C_{\beta q_1\ ch}) > 0$, by Equation (6) from Proposition 4.3, σ_2 must extend $\alpha \frown \alpha'$ in such a way that the first or the first two external actions are compatible with one of the transitions of $\mathcal{A}_1$ that are enabled from q_1. (The number of external actions depends on whether the compatible transition of $\mathcal{A}_1$ is labeled by an internal or external action.) Since state tr of $\mathcal{C}$ enables only action q_1', and since, by the tree-structure of $\mathcal{A}_1$, a is uniquely determined by q_1', the first or first two external actions of $\mathcal{A}_2 \| \mathcal{C}$ scheduled by σ_2 are either q_1' or aq_1' depending on whether a is internal or external. Thus, there exists an execution fragment α'' of $\mathcal{A}_2 \| \mathcal{C}$, with trace $trace(aq_1')$, such that $\epsilon(C_{\alpha \frown \alpha' \frown \alpha''}) > 0$. Furthermore, we can assume that the last transition of α'' is labeled by q_1' (simply truncate α'' otherwise).

Let (q_2', q_1') be the last state of α''. Then, $\alpha \frown \alpha' \frown \alpha'' \in \Theta_{q_1', q_2'}$, thus showing that $q_1'\ R\ q_2'$. It remains to show that $q_2 \stackrel{a}{\Longrightarrow} q_2'$. For this, it suffices to observe that the execution fragment $(\alpha' \frown \alpha'') \lceil \mathcal{A}_2$ has trace a, first state q_2, and last state q_2'. □

Theorem 5.4 *Let $\mathcal{A}_1$, $\mathcal{A}_2$ be nondeterministic automata. Then $\mathcal{A}_1 \leq_{DC} \mathcal{A}_2$ if and only if $\mathcal{A}_1 \leq_{wF} \mathcal{A}_2$.*

Proof. Analogous to the proof of Theorem 5.2. First we prove soundness of weak forward simulations:

$$\begin{array}{lcl} \mathcal{A}_1 \leq_{wF} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 3.6, Part 2)} \\ \mathcal{A}_1 \leq_{wPF} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 3.7, Part 3)} \\ \mathcal{A}_1 \leq_{DC} \mathcal{A}_2\ . & & \end{array}$$

Now we prove completeness:

$$\begin{array}{lcl} \mathcal{A}_1 \leq_{DC} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 2.3)} \\ \mathit{Unfold}(\mathcal{A}_1) \leq_F \mathcal{A}_1 \leq_{DC} \mathcal{A}_2 & \Rightarrow & \text{(as in proof Theorem 5.2)} \\ \mathit{Unfold}(\mathcal{A}_1) \leq_{DC} \mathcal{A}_1 \leq_{DC} \mathcal{A}_2 & \Rightarrow & (\leq_{DC} \text{ is transitive}) \\ \mathit{Unfold}(\mathcal{A}_1) \leq_{DC} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 5.3)} \\ \mathit{Unfold}(\mathcal{A}_1) \leq_{wF} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 2.3)} \\ \mathcal{A}_1 \leq_F \mathit{Unfold}(\mathcal{A}_1) \leq_{wF} \mathcal{A}_2 & \Rightarrow & \text{(Proposition 2.2, Part 1)} \\ \mathcal{A}_1 \leq_{wF} \mathit{Unfold}(\mathcal{A}_1) \leq_{wF} \mathcal{A}_2 & \Rightarrow & (\leq_{wF} \text{ is transitive}) \\ \mathcal{A}_1 \leq_{wF} \mathcal{A}_2\ . & & \end{array}$$

□

6 Characterizations of the Trace Distribution Precongruence Relation for Probabilistic Automata

Now we present our characterization theorems for $\leq_{DC}$ for probabilistic automata: Theorem 6.3 characterizes $\leq_{DC}$ in terms of $\leq_{PF}$, for PAs without internal actions, and Theorem 6.5 characterizes $\leq_{DC}$ in terms of $\leq_{wPF}$, for arbitrary probabilistic automata. Again, we give the results first for tree-structured automata and extend them by unfolding. Again, the interesting direction is the

completeness direction, showing that $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$ implies the existence of a simulation relation from $\mathcal{P}_1$ to $\mathcal{P}_2$. Our proofs of completeness for PAs are analogous to those for nondeterministic automata.

6.1 Probabilistic Automata Without Internal Actions

We first consider tree-structured automata.

Proposition 6.1 *Let $\mathcal{P}_1$, $\mathcal{P}_2$ be probabilistic automata without internal actions such that $\mathcal{P}_1$ is tree-structured. Then $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$ implies $\mathcal{P}_1 \leq_{PF} \mathcal{P}_2$.*

Proof. Assume that $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$. Define the dual probabilistic automaton $\mathcal{C}$ of $\mathcal{A}_1$, the observer σ_1, the trace distribution η, the scheduler σ_2, and the probabilistic execution ϵ as in the proof of Proposition 5.1. Define the Θ sets as in the proof of Proposition 5.1.

Define a relation R as follows: $q_1 \; R \; \mu_2$ if and only if $\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha) > 0$ and for each state $q_2 \in Q_2$,

$$\mu_2(q_2) = \frac{\sum_{\alpha \in \Theta_{q_1,q_2}} \epsilon(C_\alpha)}{\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha)}. \tag{10}$$

That is, the measure μ_2 describes probabilities of the various Θ_{q_1,q_2}'s relative to Θ_{q_1}. Note that the equation above is well defined since, by the tree-structure of $\mathcal{P}_1$, all the cones represented by Θ_{q_1} are disjoint, and thus $\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha) \leq 1$. We claim that R is a probabilistic forward simulation from $\mathcal{P}_1$ to $\mathcal{P}_2$.

Before proving that R is a probabilistic forward simulation we make several observations.

1. Relation R is a function from Q_1 to $Disc(Q_2)$.

   Indeed, if $\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha) > 0$, then there exists exactly one measure that satisfies Equation (10). Furthermore, given the construction of η, every state q_1 of Q_1 occurs with some positive probability in η, thus, $\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha) > 0$ for all states q_1 of Q_1.

2. If $q_1 \; R \; \mu_2$, then, for each state $q_2 \in Q_2$ and each execution $\alpha \in \Theta_{q_1,q_2}$,

   $$\epsilon(C_\alpha) > 0 \Rightarrow q_2 \in supp(\mu_2). \tag{11}$$

   That is, the execution α occurs with non-zero probability in ϵ only if μ_2 assigns non-zero probability to q_2. This property is a direct consequence of Equation (10).

3. For each transition $q_1 \xrightarrow{a} \mu_1'$ of $\mathcal{P}_1$, the following equation holds:

   $$\mu_1'(q_1') = \frac{\sum_{\alpha \in \Theta_{q_1'}} \epsilon(C_\alpha)}{\sum_{q \in supp(\mu_1'),\, \alpha \in \Theta_q} \epsilon(C_\alpha)}. \tag{12}$$

   That is, the relative probabilities of the states of $supp(\mu_1')$ in ϵ are given by μ_1'. This result follows by instantiating Equation (8) from Proposition 4.4 with $q_1 \xrightarrow{a} \mu_1'$ to derive the probability of a state q_1' in the support of μ_1', and by replacing the diamond expressions according to Equation (2) from Proposition 3.2.

4. For each transition $q_1 \xrightarrow{a} \mu_1'$ of $\mathcal{P}_1$, the following equation holds:

$$\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha) = k \sum_{q \in supp(\mu_1')\,,\, \alpha \in \Theta_q} \epsilon(C_\alpha), \tag{13}$$

where k is the number of transitions of $\mathcal{P}_1$ enabled from q_1. That is, the probability of reaching q_1 in ϵ is k times the probability of reaching q_1 and scheduling tr, where tr denotes the transition $q_1 \xrightarrow{a} \mu_1'$. Informally, transition tr is scheduled only if state q_1 is reached and the outcome of the following transition labeled by ch is tr, which happens with probability $1/k$. The reason why $\sum_{q \in supp(\mu_1')\,,\, \alpha \in \Theta_q} \epsilon(C_\alpha)$ is the probability of reaching q_1 and scheduling tr is that states from $supp(\mu_1')$ can occur only after q_1 has occurred and tr is reached (see the definition of dual automaton and of observer of a dual automaton) and furthermore states from $supp(\mu_1')$ occur with probability 1 once tr is reached (see Equation (5) from Proposition 4.2).

This result follows by instantiating Equation (7) from Proposition 4.4 with $q_1 \xrightarrow{a} \mu_1'$ to derive the probability of a state q_1' in the support of μ_1', replacing the diamond expressions according to Equation (2) from Proposition 3.2, summing over $supp(\mu_1')$, observing that $\sum_{q_1' \in supp(\mu_1')} \mu_1'(q_1') = 1$, and deriving $\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha)$ from the resulting equation.

We are now ready to show that R is a probabilistic forward simulation. For the start condition, we must show that $\bar{q}_1 \; R \; \delta(\bar{q}_2)$.

Consider the start state $(\bar{q}_2, \bar{q}_\mathcal{C})$ of $\mathcal{A}_2 \| \mathcal{C}$. Since there are no internal actions in $\mathcal{A}_2$ or $\mathcal{C}$, and since, by Equation (3) from Proposition 4.2, $\eta(C_{\bar{q}_1}) = 1$, the only action that is scheduled initially by σ_2 is $\bar{q}_1$, leading to state $(\bar{q}_2, \bar{q}_1)$ with probability 1. Thus, the finite execution $\alpha = (\bar{q}_2, \bar{q}_\mathcal{C})\bar{q}_1(\bar{q}_2, \bar{q}_1)$ is an element of $\Theta_{\bar{q}_1, \bar{q}_2}$ such that $\epsilon(C_\alpha) = 1$, and, by definition of R, $\bar{q}_1 \; R \; \delta(\bar{q}_2)$ as needed.

For the step condition, assume that $q_1 \; R \; \mu_2$ and let $q_1 \xrightarrow{a}_1 \mu_1'$ be a transition of $\mathcal{P}_1$, which we denote by tr. We must exhibit a probability measure $\mu_2' \in Disc(Disc(Q_2))$ and a hyper-transition $\mu_2 \xrightarrow{a}_2 \mu_2''$, matching the given transition, where $\mu_2'' = flatten(\mu_2')$ and $\mu_1' \; R \; \mu_2'$. We do this by deriving a transition tr_α for each execution α of Θ_{q_1} and by combining the tr_α's appropriately into transitions tr_q, for each state $q \in supp(\mu_2)$, that are the basis for the required hyper-transition. The tr_α transitions are derived from η; the construction considers only those α's for which $\epsilon(C_\alpha) > 0$. The other α's can be treated arbitrarily.

Consider an execution α of Θ_{q_1} such that $\epsilon(C_\alpha) > 0$. By Property (11), $\alpha \in \Theta_{q_1, q_2}$ for some state q_2 in $supp(\mu_2)$. Since Θ_{q_1,q_2} is a subset of Θ_{q_1}, by definition of Θ_{q_1}, $trace(\alpha) = \beta q_1$ for some finite trace β. Therefore, $\eta(C_{\beta q_1}) > 0$. Since q_1 enables at least one transition in $\mathcal{P}_1$, specifically transition tr, Equation (4) from Proposition 4.2 implies that $\eta(C_{\beta q_1\, ch}) = \eta(C_{\beta q_1})$. Then, since $\mathcal{A}_2$ and $\mathcal{C}$ have no internal actions, σ_2 schedules action ch from α with probability 1.

By definition of $dual(\mathcal{A}_1)$, the transition labeled by ch that leaves from state q_1 of $\mathcal{C}$ leads to state tr with non-zero probability. Therefore, $\epsilon(C_{\alpha\, ch\, (q_2, tr)}) > 0$. By Equation (5) from Proposition 4.2, where only the first term of the right-hand side is used due to the absence of internal actions, σ_2 must extend $\alpha\, ch\, (q_2, tr)$ with two steps labeled by an action and a state of $\mathcal{A}_1$, respectively, where the action and the state are compatible with one of the transitions of $\mathcal{A}_1$ that are enabled from q_1. Since state tr of $\mathcal{C}$ enables only actions in $supp(\mu_1')$, and since, by the tree-structure of $\mathcal{A}_1$, a is uniquely determined by μ_1', the action that is scheduled is a and the state that is scheduled is a state in $supp(\mu_1')$. Thus, $\sigma_2(\alpha\, ch\, (q_2, tr))$ returns a probability measure over transitions labeled by a. This measure identifies a combined transition of $\mathcal{A}_2$ labeled by a that leaves from q_2, which we denote by tr_α.

Now, using the tr_α transitions, we define a combined transition from each state in the support of μ_2. Namely, for each state $q \in supp(\mu_2)$, let tr_q be the combined transition of $\mathcal{P}_2$ defined by:

$$tr_q \stackrel{\triangle}{=} \sum_{\alpha \in \Theta_{q_1,q}} \frac{\epsilon(C_\alpha)}{\sum_{\alpha' \in \Theta_{q_1,q}} \epsilon(C_{\alpha'})} tr_\alpha. \tag{14}$$

Informally, each element of $\Theta_{q_1,q}$ is an execution in are ϵ that contributes to the emulation of transition $q_1 \stackrel{a}{\rightarrow}_1 \mu_1'$ from q. Equation (14) computes tr_q, the overall contribution to the emulation from q, by averaging over all elements of $\Theta_{q_1,q}$. We could prove that $\Theta_{q_1,q}$ contains only one element α' such that $\epsilon(C_{\alpha'}) > 0$ and simplify Equation (14) accordingly. However, this simplification is not necessary for the proof. Now we define the measure $\mu_2'' \in Disc(Q_2)$:

$$\mu_2'' \stackrel{\triangle}{=} \sum_{q \in supp(\mu_2)} \mu_2(q) \mu_{tr_q}. \tag{15}$$

Then, by construction, $\mu_2 \stackrel{a}{\rightarrow} \mu_2''$ is a hyper-transition of $\mathcal{P}_2$.

It remains to define a probability measure $\mu_2' \in Disc(Disc(Q_2))$ such that $\mu_2'' = flatten(\mu_2')$ and $\mu_1' \; R \; \mu_2'$.

For each $q \in supp(\mu_1')$, let μ_q be the unique measure such that $q \; R \; \mu_q$. We can identify μ_q because R is a function. Define $\mu_2' \in Disc(Disc(Q_2))$ such that, for each $q \in supp(\mu_1')$, $\mu_2'(\mu_q) = \sum_{q' \in supp(\mu_1') | \mu_{q'} = \mu_q} \mu_1'(q')$. Then $\mu_1' \; R \; \mu_2'$ by definition of μ_2'.

It remains to show that $\mu_2'' = flatten(\mu_2')$, that is, that $\mu_2'' = \sum_{\rho \in supp(\mu_2')} \mu_2'(\rho)\rho$. From the definition of μ_2' and of the flatten operator, it suffices to show that for every $q_2 \in Q_2$,

$$\mu_2''(q_2) = \sum_{q \in supp(\mu_1')} \mu_1'(q) \mu_q(q_2). \tag{16}$$

To prove Equation (16) we first claim that the following equation is valid for each pair of states q_1, q_2 of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, if k denotes the number of transitions of $\mathcal{P}_1$ that are enabled from q_1:

$$\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha) \mu_{tr_\alpha}(q_2) = k \sum_{q \in supp(\mu_1') \,, \alpha \in \Theta_{q,q_2}} \epsilon(C_\alpha). \tag{17}$$

Informally, the left-hand side of Equation (17) represents the probability of scheduling q_1 and then reaching q_2 according to the transition tr_α, without considering the outcome of the transition labeled by ch. The right-hand side, on the other hand, computes the probability of scheduling q_1, scheduling ch and reaching μ_1', and then scheduling tr_α and reaching q_2. State μ_1' is reached by ch with probability $1/k$, which justifies the k factor in the right-hand side.

To prove Equation (17), consider an execution $\alpha \in \Theta_{q,q_2}$ where $q \in supp(\mu_1')$. Since q occurs always after q_1, execution α can be split into $\alpha' \frown \alpha''$ where $\alpha' \in \Theta_{q_1}$. Furthermore, $trace(\alpha'') = ch\, a q$, and since there are no internal actions in $\mathcal{P}_2$ and $\mathcal{C}$, α is the unique extension of α' that is in Θ_{q,q_2}. In particular, $\alpha'' = (q', q_1)\, ch\, (q', tr) a (q_2, tr) q (q_2, q)$ for some state q' of $\mathcal{A}_2$, and $\epsilon(C_\alpha) = \epsilon(C_{\alpha'})(1/k)\mu_{tr_{\alpha'}}(q_2)$. Thus, each summand in the right-hand side of Equation (17) has a corresponding summand in the left-hand side that differs by a factor of k, and the correspondence relation is an injection. If the correspondence is not a bijection, then the α terms that are left out on the left-hand side are such that $\mu_{tr_\alpha}(q_2) = 0$ (otherwise an extension in Θ_{q,q_2} for some q exists). This suffices.

We now consider the left-hand side of Equation (16). Consider the definition of μ_2'' given by Equation (15). By expanding $\mu_2(q)$ according to the definition of μ_2 given by Equation (10), and expanding $\mu_{tr}(q_2)$ according to the definition of μ_{tr} given by Equation (14), we obtain

$$\mu_2''(q_2) = \sum_{q \in supp(\mu_2)} \frac{\sum_{\alpha \in \Theta_{q_1,q}} \epsilon(C_\alpha)}{\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha)} \frac{\sum_{\alpha \in \Theta_{q_1,q}} \epsilon(C_\alpha) \mu_{tr_\alpha}(q_2)}{\sum_{\alpha \in \Theta_{q_1,q}} \epsilon(C_\alpha)}.$$

By cross simplifying the top leftmost and bottom rightmost factors, and by factoring the left denominator out of the sum, we obtain

$$\mu_2''(q_2) = \frac{\sum_{q \in supp(\mu_2)} \sum_{\alpha \in \Theta_{q_1,q}} \epsilon(C_\alpha) \mu_{tr_\alpha}(q_2)}{\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha)}.$$

By Property (11), we can rewrite the numerator as follows:

$$\mu_2''(q_2) = \frac{\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha) \mu_{tr_\alpha}(q_2)}{\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha)}.$$

By multiplying numerator and denominator by k, applying Equation (17) to the numerator, and applying Equation (13) to the denominator, we obtain

$$\mu_2''(q_2) = \frac{\sum_{q \in supp(\mu_1'),\, \alpha \in \Theta_{q,q_2}} \epsilon(C_\alpha)}{\sum_{q \in supp(\mu_1'),\, \alpha \in \Theta_q} \epsilon(C_\alpha)}. \tag{18}$$

We now consider the right-hand side of Equation (16). By applying Equations (12) and (10) to the two factors of the right-hand side of Equation (16), and by simplifying common factors algebraically, we obtain

$$\sum_{q \in supp(\mu_1')} \mu_1'(q) \mu_q(q_2) = \frac{\sum_{q \in supp(\mu_1'),\, \alpha \in \Theta_{q,q_2}} \epsilon(C_\alpha)}{\sum_{q \in supp(\mu_1'),\, \alpha \in \Theta_q} \epsilon(C_\alpha)}. \tag{19}$$

Now Equation (16) follows by direct combination of Equations (18) and (19). □

Interestingly, the probabilistic forward simulation that we constructed in the above proof is functional. Functional simulations are usually called *refinement mappings*. Write $\mathcal{P}_1 \leq_{PR} \mathcal{P}_2$ if there exists a functional probabilistic forward simulations from $\mathcal{P}_1$ to $\mathcal{P}_2$. Then we can state the following corollary of Proposition 6.1, which is a probabilistic version of Proposition 3.12 in [10]:

Corollary 6.2 *Let $\mathcal{P}_1$, $\mathcal{P}_2$ be probabilistic automata without internal actions such that $\mathcal{P}_1$ is tree-structured. Then $\mathcal{P}_1 \leq_{PF} \mathcal{P}_2$ iff $\mathcal{P}_1 \leq_{PR} \mathcal{P}_2$.*

Proof. It is enough to observe that each state q_1 of $\mathcal{P}_1$ occurs with some positive probability in the trace distribution η of the proof of Proposition 6.1. □

Theorem 6.3 *Let $\mathcal{P}_1$, $\mathcal{P}_2$ be probabilistic automata without internal actions. Then $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$ if and only if $\mathcal{P}_1 \leq_{PF} \mathcal{P}_2$.*

Proof. First we prove soundness of probabilistic forward simulations:

$$\begin{array}{lcl} \mathcal{P}_1 \leq_{PF} \mathcal{P}_2 & \Rightarrow & \text{(Proposition 3.7, Part 1)} \\ \mathcal{P}_1 \leq_{wPF} \mathcal{P}_2 & \Rightarrow & \text{(Proposition 3.7, Part 3)} \\ \mathcal{P}_1 \leq_{DC} \mathcal{P}_2 \ . & & \end{array}$$

Now we prove completeness:

$$\begin{array}{lcl} \mathcal{P}_1 \leq_{DC} \mathcal{P}_2 & \Rightarrow & \text{(Proposition 3.9)} \\ \mathit{Unfold}(\mathcal{P}_1) \leq_{DC} \mathcal{P}_1 \leq_{DC} \mathcal{P}_2 & \Rightarrow & (\leq_{DC} \text{ is transitive}) \\ \mathit{Unfold}(\mathcal{P}_1) \leq_{DC} \mathcal{P}_2 & \Rightarrow & \text{(Proposition 6.1)} \\ \mathit{Unfold}(\mathcal{P}_1) \leq_{PF} \mathcal{P}_2 & \Rightarrow & \text{(Proposition 2.3)} \\ \mathcal{P}_1 \leq_{PF} \mathit{Unfold}(\mathcal{P}_1) \leq_{PF} \mathcal{P}_2 & \Rightarrow & (\leq_{PF} \text{ is transitive}) \\ \mathcal{P}_1 \leq_{PF} \mathcal{P}_2 \ . & & \end{array}$$

□

6.2 Probabilistic Automata With Internal Actions

Again, we start with tree-structured PAs.

Proposition 6.4 *Let $\mathcal{P}_1$, $\mathcal{P}_2$ be probabilistic automata with $\mathcal{P}_1$ tree-structured. Then $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$ implies $\mathcal{P}_1 \leq_{wPF} \mathcal{P}_2$.*

Proof. Assume that $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$. Define the dual probabilistic automaton $\mathcal{C}$ of $\mathcal{A}_1$, the observer σ_1, the trace distribution η, the scheduler σ_2, and the probabilistic execution ϵ as in the proof of Proposition 5.1. Without loss of generality we assume that σ_2 schedules action $\bar{q}_1$ with probability 1 from the start state of $\mathcal{A}_2 \| \mathcal{C}$ (essentially we can exchange the internal transitions of $\mathcal{A}_2$ that occur before the transition labeled by $\bar{q}_1$ with the transition labeled by $\bar{q}_1$).

Define the Θ sets as in the proof of Proposition 5.1, and define relation R according to Equation (10) as in the proof of Proposition 6.1. Observe that Property (11) and Equations (12) and (13) hold for the same reasons as before.

The proof that R is a weak probabilistic forward simulation is exactly as before except for the definition of the tr_α transitions. Thus, in the rest of the proof we construct the tr_α's and prove that Equation (17) still holds.

Assume that $q_1 \ R \ \mu_2$ and let $q_1 \xrightarrow{a}_1 \mu_1'$ be a transition of $\mathcal{P}_1$, which we denote by tr.

We introduce a special *conditional* construction that is needed for the definition of the tr_α's. Let $\mathcal{C}_{tr}$ be the same as $\mathcal{C}$ except that the transition $q_1 \xrightarrow{ch} \mu$, where μ is uniquely determined by q_1, is replaced by $q_1 \xrightarrow{ch} \delta(tr)$. Given a scheduler σ for $\mathcal{A}_2 \| \mathcal{C}$, define the scheduler $\sigma \mid tr$ for $\mathcal{A}_2 \| \mathcal{C}_{tr}$ that is the same as σ except that transition $q_1 \xrightarrow{ch} \delta(tr)$ of $\mathcal{C}_{tr}$ is chosen whenever σ chooses $q_1 \xrightarrow{ch} \mu$. Given a probabilistic execution fragment ϵ' of $\mathcal{A}_2 \| \mathcal{C}$, generated by some scheduler σ, define $\epsilon' \mid tr$ to be the result of $\sigma \mid tr$ applied to $\mathcal{A} \| \mathcal{C}_{tr}$ from the start state of ϵ'. The intuition behind $\epsilon' \mid tr$ is that we study ϵ' under the condition that tr is the outcoming state of $\mathcal{C}$ whenever $q_1 \xrightarrow{ch} \mu$ is scheduled. Then, the following two properties are valid.

1. $(\epsilon' \mid tr)\lceil \mathcal{A}_2$ is a probabilistic execution fragment of $\mathcal{A}_2$.
2. For each finite execution fragment α of $\mathcal{A}_2 \| \mathcal{C}$ where state tr occurs and such that $\mathit{fstate}(\alpha)$ is not of the form $(\cdot, tr)$, $(\epsilon' \mid tr)(C_\alpha) = k\epsilon(C_\alpha)$, where k is the size of $\mathit{supp}(\mu)$.

The first item follows immediately from Proposition 3.4 given that $\epsilon' \mid tr$ is a probabilistic execution fragment of $\mathcal{A}_2 \| \mathcal{C}_{tr}$. The second item follows directly from the definition of probability of a cone since in ϵ' the probability associated with the edge $q\ ch\ (\cdot, tr)$ is $1/k$ while in $\epsilon' \mid tr$ the probability of the same edge is 1.

We now define the tr_α's. Consider an execution α of Θ_{q_1} such that $\epsilon(C_\alpha) > 0$. Let ϵ^1 be the truncation of ϵ at all the points in $\cup_{q \in supp(\mu_1')} \Theta_q$, which is a probabilistic execution of $\mathcal{A}_2 \| \mathcal{C}$ by Proposition 3.11. Let ϵ^1_α be $\epsilon^1 \triangleright \alpha$, which is a probabilistic execution fragment of $\mathcal{A}_2 \| \mathcal{C}$ by Item 1 of Proposition 3.13. Finally, let ϵ^2_α be $(\epsilon^1_\alpha \mid tr) \lceil \mathcal{A}_2$, which is a probabilistic execution fragment of $\mathcal{A}_2$ by Property 1.

By definition of Θ_{q_1}, $trace(\alpha) = \beta q_1$ for some finite trace β. Therefore, $\eta(C_{\beta q_1}) > 0$. Since q_1 enables at least one transition in $\mathcal{P}_1$, specifically transition tr, Equation (4) from Proposition 4.2 implies that $\eta(C_{\beta q_1\ ch}) = \eta(C_{\beta q_1})$. Thus, action ch occurs as the first external action with probability 1 in μ^1_α.

By Equation (6) from Proposition 4.3, if the occurrence of action ch leads $\mathcal{C}$ to state tr, then an action in $supp(\mu_1')$ occurs eventually in ϵ with probability 1, leading $\mathcal{C}$ to a state in $supp(\mu_1')$, which is a truncation point according to the definition of ϵ^1. Thus, the probability of termination in $\epsilon^1_\alpha \mid tr$ is 1, as well as the probability of termination in ϵ^2_α, i.e., ϵ^2_α assigns probability 1 to the set of finite executions. Furthermore, given that action a is uniquely determined by μ_1' ($\mathcal{A}_1$ is tree-structured), again by Equation (6) from Proposition 4.3 all finite executions α' with $\epsilon^2_\alpha(\alpha') > 0$ have trace a (empty trace if a is internal). Thus, ϵ^2_α denotes a weak combined transition labeled by a (no action if a is internal) from $lstate(\alpha) \lceil \mathcal{A}_2$. Denote such transition by tr_α.

We are left to show that Equation (17) still holds. That is,

$$\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha) \mu_{tr_\alpha}(q_2) = k \sum_{q \in supp(\mu_1')\,,\, \alpha \in \Theta_{q,q_2}} \epsilon(C_\alpha).$$

We consider first the term $\mu_{tr_\alpha}(q_2)$. From the definition of tr_α and of weak combined transition we get

$$\mu_{tr_\alpha}(q_2) = \sum_{\alpha' \mid lstate(\alpha') = q_2} \epsilon^2_\alpha(\alpha').$$

By applying the definition of projection, and using the fact that $\epsilon^1_\alpha \mid tr$ assigns probability 1 to the set of finite executions, we get

$$\mu_{tr_\alpha}(q_2) = \sum_{\alpha' \mid lstate(\alpha' \lceil \mathcal{A}_2) = q_2} (\epsilon^1_\alpha \mid tr)(\alpha').$$

Given that the truncation points of ϵ^1 are all at the $\cup_{q \in supp(\mu_1')} \Theta_q$ points, the only finite executions α' that have non-zero probability are such that $\alpha \frown \alpha'$ is in some set Θ_q. Furthermore, given that no execution in $\cup_{q \in supp(\mu_1')} \Theta_q$ is a prefix of another (our PAs are tree-structured and all actions in $supp(\mu_1')$ occur in different branches), the probabilities of the finite executions can be replaced by the probabilities of their cones, thus getting

$$\mu_{tr_\alpha}(q_2) = \sum_{q \in supp(\mu_1')} \sum_{\alpha' \mid \alpha \frown \alpha' \in \Theta_{q,q_2}} (\epsilon^1_\alpha \mid tr)(C_{\alpha'}).$$

By Property 2 we can get rid of the conditional on tr by introducing a k factor, thus getting

$$\mu_{tr_\alpha}(q_2) = \sum_{q \in supp(\mu_1')} \sum_{\alpha' \mid \alpha \frown \alpha' \in \Theta_{q,q_2}} k \epsilon^1_\alpha(C_{\alpha'}). \tag{20}$$

By replacing $\mu_{tr_\alpha}(q_2)$ according to Equation (20) in the left-hand side of Equation (17), and by rearranging terms algebraically, we obtain

$$\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha)\mu_{tr_\alpha}(q_2) = k \sum_{q \in supp(\mu_1')} \sum_{\alpha \in \Theta_{q_1}} \sum_{\alpha' | \alpha^\frown \alpha' \in \Theta_{q,q_2}} \epsilon(C_\alpha)\epsilon^1_\alpha(C_{\alpha'}).$$

By using the definition of ϵ^1_α and Item 2 of Proposition 3.13, the two probabilities in the equation above can be grouped into $\epsilon(C_{\alpha^\frown\alpha'})$. By observing that all elements in Θ_{q,q_2}, with $q \in supp(\mu_1')$, have a prefix in Θ_{q_1}, the intermediate sum can be removed, thus getting

$$\sum_{\alpha \in \Theta_{q_1}} \epsilon(C_\alpha)\mu_{tr_\alpha}(q_2) = k \sum_{q \in supp(\mu_1')} \sum_{\alpha \in \Theta_{q,q_2}} \epsilon(C_\alpha),$$

which is Equation (17) as needed. □

Theorem 6.5 *Let $\mathcal{P}_1$, $\mathcal{P}_2$ be probabilistic automata. Then $\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$ if and only if $\mathcal{P}_1 \leq_{wPF} \mathcal{P}_2$.*

Proof. Soundness of weak probabilistic forward simulations follows immediately from Proposition 3.7. Completeness is established by:

$\mathcal{P}_1 \leq_{DC} \mathcal{P}_2$	$\Rightarrow$	(Proposition 3.9)
$\mathit{Unfold}(\mathcal{P}_1) \leq_{DC} \mathcal{P}_1 \leq_{DC} \mathcal{P}_2$	$\Rightarrow$	($\leq_{DC}$ is transitive)
$\mathit{Unfold}(\mathcal{P}_1) \leq_{DC} \mathcal{P}_2$	$\Rightarrow$	(Proposition 6.4)
$\mathit{Unfold}(\mathcal{P}_1) \leq_{wPF} \mathcal{P}_2$	$\Rightarrow$	(Proposition 2.3)
$\mathcal{P}_1 \leq_{PF} \mathit{Unfold}(\mathcal{P}_1) \leq_{wPF} \mathcal{P}_2$	$\Rightarrow$	(Proposition 3.7)
$\mathcal{P}_1 \leq_{wPF} \mathit{Unfold}(\mathcal{P}_1) \leq_{wPF} \mathcal{P}_2$	$\Rightarrow$	($\leq_{wPF}$ is transitive)
$\mathcal{P}_1 \leq_{wPF} \mathcal{P}_2$.		

□

7 Concluding Remarks

We have characterized the trace distribution precongruence for nondeterministic and probabilistic automata, with and without internal actions, in terms of four kinds of simulation relations, $\leq_F$, $\leq_{wF}$, $\leq_{PF}$, and $\leq_{wPF}$. In particular, this shows that probabilistic contexts are capable of observing all the distinctions that can be expressed using these simulation relations.

Some technical improvements are possible. For example, our finite branching restriction can be relaxed to countable branching, simply by replacing uniform distributions in the dual automata by other distributions such as exponential distributions. Calculations become more complicated, however.

For future work, it would be interesting to try another approach to achieving compositionality for PA behaviors: define implementation as trace distribution inclusion, but restrict parallel composition so that the nondeterminism of each component is resolved based only on externally-visible behavior of the other components. This approach also requires some ways of resolving the nondeterminism of scheduling different components. Some initial steps towards this goal appear in our recent work on *switched automata* [3].

References

[1] S. Aggarwal. Time optimal self-stabilizing spanning tree algorithms. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1994. Available as Technical Report MIT/LCS/TR-632.

[2] F. Bartels, A. Sokolova, and E.P. de Vink. A hierarchy of probabilistic system types. *Electronic Notes in Theoretical Computer Science*, 82(1), 2003.

[3] L. Cheung, N.A. Lynch, R. Segala, and F.W. Vaandrager. Switched probabilistic I/O automata. In *Proceedings First International Colloquium on Theoretical Aspects of Computing (ICTAC2004),* Guiyang, China, 20-24 September 2004, Lecture Notes in Computer Science. Springer-Verlag, 2004. To appear.

[4] L de Alfaro, T.A. Henzinger, and R. Jhala. Compositional methods for probabilistic systems. In K.G. Larsen and M. Nielsen, editors, *Proceedings CONCUR 01,* Aalborg, Denmark, August 20-25, 2001, volume 2154 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2001.

[5] W. Feller. An Introduction to Probability Theory and its Applications. Volume 1. Jokn Wiley & Sons, Inc., 1950.

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

[7] N.A. Lynch, I. Saias, and R. Segala. Proving time bounds for randomized distributed algorithms. In *Proceedings of the 13th Annual ACM Symposium on the Principles of Distributed Computing*, pages 314–323, Los Angeles, CA, August 1994.

[8] N.A. Lynch, R. Segala, and F.W. Vaandrager. Compositionality for probabilistic automata. In R. Amadio and D. Lugiez, editors, *Proceedings 14th International Conference on Concurrency Theory (CONCUR 2003),* Marseille, France, volume 2761 of *Lecture Notes in Computer Science*, pages 208–221. Springer-Verlag, September 2003.

[9] N.A. Lynch and M.R. Tuttle. An introduction to input/output automata. *CWI Quarterly*, 2(3):219–246, September 1989.

[10] N.A. Lynch and F.W. Vaandrager. Forward and backward simulations, I: Untimed systems. *Information and Computation*, 121(2):214–233, September 1995.

[11] A. Pogosyants, R. Segala, and N.A. Lynch. Verification of the randomized consensus algorithm of Aspnes and Herlihy: a case study. *Distributed Computing*, 13(3):155–186, 2000.

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

[13] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1995. Available as Technical Report MIT/LCS/TR-676.

[14] R. Segala. Testing probabilistic automata. In U. Montanari and V. Sassone, editors, *Proceedings CONCUR 96,* Pisa, Italy, August 26-29, 1996, volume 1119 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 1996.

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

[16] A. Sokolova and E.P. de Vink. Probabilistic automata: system types, parallel composition and comparison. In C. Baier et al., editor, *Validation of Stochastic Systems*, pages 1–43. LNCS 2925, 2004.

[17] M.I.A. Stoelinga. *Alea Jacta Est: Verification of Probabilistic, Real-Time and Parametric Systems*. PhD thesis, University of Nijmegen, April 2002.

[18] M.I.A. Stoelinga. An introduction to probabilistic automata. *Bulletin of the European Association for Theoretical Computer Science*, 78:176–198, October 2002.

[19] M.I.A. Stoelinga and F.W. Vaandrager. Root contention in IEEE 1394. In J.-P. Katoen, editor, *Proceedings 5th International AMAST Workshop on Formal Methods for Real-Time and Probabilistic Systems,* Bamberg, Germany, volume 1601 of *Lecture Notes in Computer Science*, pages 53–74. Springer-Verlag, 1999.

[20] M.I.A. Stoelinga and F.W. Vaandrager. A testing scenario for probabilistic automata. In J.C.M. Baeten, J.K. Lenstra, J. Parrow, and G.J. Woeginger, editors, *Proceedings* 30^{th} *ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 407–418. Springer-Verlag, 2003.