

# A Symbolic Treatment of Randomization in Concurrent Systems

Ling Cheung

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

Version of January 20, 2005

**Abstract.** First we define a new type of transition system in which states are *symbolic* (i.e., we should think of them as distributions on concrete states) and transitions are probabilistic. Apart from the intended meaning of states, this system type is equivalent to a special case of Segala's general PA. We propose a (non-deterministic) composition mechanism for this system type, as well as trace distribution and simulation semantics. (And pray very, very hard that these semantic notions are compositional.)

Then we describe how to derive a transition system of the above type from a simple PIOA, using notions of probabilistic transition bundles and fibers. These notions are based on *deterministic* schedulers, as opposed to the more conventional *randomized* schedulers.

## 1 Introduction

The concept of schedulers for concurrent components is inherently *context-dependent*: a scheduler makes a decision regarding the next move based on some contextual information. The amount of information available to a scheduler varies depending on the particular framework. It could be

- state information of the composed system,
- history information of the composed system,
- or even information about the entire computation environment in which the composed system is supposed to operate.

All of these examples have one feature in common, namely, that the information used by the scheduler is *not* available in its entirety to any individual component. It is then not so surprising that compositionality often fails when scheduling is involved.

Our aim is to give a treatment of parallel composition that does not involve schedulers. To do so, we first distinguish between two views of a component:

- **intensional**: a description of how the component implements a particular causal dependence (e.g. an internal coin toss takes place and action  $a$  is performed in case of head while action  $b$  is performed in case of tail);
- **extensional**: a description of the causal dependence implemented by the component (e.g. action  $a$  is performed with probability  $\frac{1}{2}$  and so is action  $b$ ).

We stress this fact: it is *not* the amount of detail that distinguishes these views, but rather the perspective in which the component is viewed. Our thesis is that, if a compositionality theorem is desirable, then (1) composition should be defined in terms of the extensional view of components, and (2) the composition mechanism itself should be context-independent.

Below we provide an example in which a very weak scheduler (history-independent and deterministic) produces undesirable effects.

*Example 1.* Automaton  $A$  is  $a$  followed by  $\langle b, \frac{1}{2} \rangle$  and  $\langle c, \frac{1}{2} \rangle$ . Think of  $a$  as a coin toss,  $b$  is  $\text{send}(0)$ , and  $c$  is  $\text{send}(1)$ . Automaton  $B$  is  $a$  (as an input) followed by  $d$ . We should *not* have  $\langle abd \mapsto \frac{1}{2}, adc \mapsto \frac{1}{2} \rangle$ , because  $b$  and  $c$  represent the same move, only with different parameters. Argue this is a typical use of randomization in security setting.

**Draft remark:** Further argue that compositionality is achieved in the synchronous model of [dAHJ01], PIOA with exponential delay [WSS94] and Switched PIOA [CLSV04], all of which avoids scheduling among components. **:kramer tfarD**

**Draft remark:** Break. Older materials below. **:kramer tfarD**

In modeling a system or a protocol, we distinguish between two types of choices:

- probabilistic, e.g. a party invokes a random source to obtain a randomly chosen bit;
- non-deterministic, e.g. which protocol party makes a move first.

Our definitions are motivated by the desire to separate probabilistic concerns away from analysis of concurrent behavior, which is modeled by non-deterministic choices among concurrent components. We think of *basic* computational units as probabilistic algorithms, in which the only type of uncertainty comes from internal coin flips.

We argue that the resulting notion of behavior retains the linear flavor of trace semantics. Moreover, we define composition in terms of ordered pairs of state distributions, as opposed to distributions on ordered pairs of states. We hope that holds the key to compositionality of trace distribution semantics.

**Draft remark:** Basic components are purely probabilistic. In a composition, either component makes a locally controlled move, but not both. That is, a transition bundle of the composite contains locally controlled actions of *at most* one component. This avoids correlation between:

- outcome of an internal coin toss in one component and
- interleavings of actions between different components.

See Example 1. **:kramer tfarD**

## 2 Symbolic System Type

Fix a (countable) action alphabet  $Act$ . Consider the following functor.

$$F(S) := \mathcal{P}(Act \rightarrow ([0, 1] \times S))$$

For our purposes, a probabilistic automaton is an  $F$ -coalgebra  $\langle S, \Delta \rangle$  satisfying an additional axiom.

For all  $\mu \in S$  and  $f \in \Delta(\mu)$ ,  $\pi_1(f)$  is a discrete distribution on  $Act$ .

Notice we use  $\mu$  to denote a typical state in  $S$ . We should think of  $\mu$  as a discrete distribution on concrete states of some underlying automaton. We will explain the notion of underlying automaton in Section 11.

This system type is a variant of (*general*) *probabilistic automata* as introduced by Segala and Lynch [Seg95]. There the functor is

$$G(S) := \mathcal{P}(\text{Disc}(Act \times S)).$$

Our systems differ from  $G$ -coalgebras in the following aspect: if both  $\langle a, \mu_1 \rangle$  and  $\langle a, \mu_2 \rangle$  appear in the support of some  $f \in \Delta(\mu)$ , it must be the case that  $\mu_1 = \mu_2$  (thus our systems are deterministic inside  $\text{Disc}$ ).

We give an explicit definition.

**Definition 1.** A symbolic probabilistic automaton (PA) is a triple  $A = \langle S_A, \mu_A^0, \Delta_A \rangle$  where

- $S_A$  is the set of symbolic states,
- $\mu_A^0 \in S_A$  is the initial state, and
- $\Delta_A : S_A \rightarrow \mathcal{P}(Act \rightarrow ([0, 1] \times S_A))$  is the transition function satisfying: for all  $\mu \in S_A$  and  $f \in \Delta_A(\mu)$ ,  $\pi_1(f)$  is a discrete distribution on  $Act$ .

We say that  $f$  is a transition bundle from  $\mu$  if  $f \in \Delta_A(\mu)$ . Also,  $\nu$  is reachable from  $\mu$  via the  $a$ -fiber of  $f$ , written  $\mu \xrightarrow{f,a} \nu$ , just in case (i)  $f$  is a transition bundle from  $\mu$  and (ii)  $f(a) = \langle p, \nu \rangle$  for some  $p > 0$ . The fiber probability of  $a$  in  $f$  is simply  $\pi_1(f(a))$ .

Intuitively, a symbolic state is a distribution on concrete states of an underlying system, while a transition bundle represents a one-step evolution of the system. During such a one-step evolution, we may observe different actions. We interpret  $\mu \xrightarrow{f,a} \nu$  as follows, if

1. the system starts from distribution  $\mu$  and follows transition bundle  $f$  (here the choice of  $f$  is *non-deterministic*), and
2. an  $a$  action is observed during the execution of  $f$  (here the choice of  $a$  is *probabilistic*),

then the resulting distribution is  $\nu$ . We stress the fact that the probabilistic distribution over actions (i.e.,  $\pi_1(f)$ ) results from uncertainty in the source distribution  $\mu$ , and *not* from any form of randomized schedulers. This will become apparent as we describe concrete constructions of transition bundles in Section 12.

### 3 Probabilistic Executions and Trace Distributions

In the non-probabilistic case, an execution (or path) is obtained by resolving all nondeterministic choices in a deterministic fashion. We attempt to do the same for symbolic PA, using transition bundles.

Roughly speaking, we define a probabilistic execution to be a purely probabilistic tree in which each node is a symbolic state  $\mu$ , enabling at most one probabilistic transition bundle  $f$  from  $\mu$ . The tree-like structure results from the fact that  $f$  leads to various end distributions, depending on the action observed. We begin by defining a branch in such a tree.

**Definition 2.** Let  $\mu \in S_A$  be given. We use joint recursion to define the set of probabilistic branches from  $\mu$ , denoted  $\text{PBran}(\mu)$ , together with two functions  $\text{last} : \text{PBran}(\mu) \rightarrow S_A$ :

- the length-1 sequence  $\mu$  is in  $\text{PBran}(\mu)$ , with  $\text{last}(\mu) := \mu$ ;
- if  $r$  is in  $\text{PBran}(\mu)$ , then so is  $r.f.a.\nu$ , provided:
  - $f$  is a transition bundle from  $\text{last}(r)$ ;
  - $a \in \text{Act}$  and  $\pi_1(f(a)) > 0$ ;
  - $\nu = \pi_2(f(a))$ ;
 moreover,  $\text{last}(r.f.a.\nu) := \nu$ .

The probability of following a branch  $r$  (given the initial symbolic state  $\mu$  and the choices of transition bundles along  $r$ ) is the value  $\Pi[r]$  defined as follows:

- $\Pi[\mu] = 1$ ;
- $\Pi[r.f.a.\nu] = \Pi[r] \cdot \pi_1(f(a))$ .

(Keep in mind, though,  $\Pi$  is not a probability measure over  $\text{PBran}(\mu)$ , because there are no distributions specified for the choices of  $f$ 's.) The trace of  $r$ , denoted  $\text{tr}(r)$ , is defined in the obvious way:

- $\text{tr}(\mu) := \epsilon$ ;
- $\text{tr}(r.f.a.\nu) := \text{tr}(r)a$ .

Finally, the length of  $r$ , denoted  $|r|$  is given by:

- $|\mu| := 0$ ;
- $|r.f.a.\nu| := |r| + 1$ .

A branch  $r$  is called rooted if the initial symbolic state is  $\mu^0$ . We write  $\text{PBran}(A)$  for  $\text{PBran}(\mu^0)$ .

**Definition 3.** Let  $\mu \in S_A$  be given. A probabilistic execution from  $\mu$  is a partial function  $Q : \text{PBran}(\mu) \rightarrow \bigcup \Delta_A(S_A) \cup \{\top\}$  satisfying the following two conditions.

1. For all  $r \in \text{dom}(Q)$ ,  $Q(r) \in \Delta_A(\text{last}(r)) \cup \{\top\}$ .
2. The domain of  $Q$  is generated by the following closure rules:
  - (a)  $\mu \in \text{dom}(Q)$ ;

(b) for all  $r \in \text{dom}(Q)$  with  $Q(r) \neq \top$  and for all  $a \in \text{Act}$  with  $\pi_1(Q(r)(a)) > 0$ , the one-step extension of  $r$

$$r' = r.Q(r).a.\pi_2(Q(r)(a))$$

is in  $\text{dom}(Q)$ . (Notice, by Condition (1),  $r'$  is in fact a probabilistic branch.)

A probabilistic branch  $r$  (and its end distribution  $\text{last}(r)$ ) is reachable from  $\mu$  via  $Q$  if  $r \in \text{dom}(Q)$ .

We write  $\text{ProbExec}(\mu)$  for the set of all probabilistic executions from  $\mu$ . If  $\mu = \mu^0$ , then these probabilistic executions are said to be rooted and we write  $\text{ProbExec}(A)$  for  $\text{ProbExec}(\mu^0)$ .

**Draft remark:** Here  $\top$  is a symbol for termination. :kramer tfaRD

**Definition 4.** Fix a probabilistic execution  $Q$  in  $\text{ProbExec}(\mu)$ . The trace distribution induced by  $Q$ , denoted  $\text{td}(Q)$ , is the function from  $\text{Act}^{<\omega}$  to  $[0, 1]$  defined by:

$$\text{td}(Q)(\alpha) := \sum_{r \in \text{dom}(Q) \cap \text{tr}^{-1}(\alpha)} \Pi[r].$$

We write  $\text{TrDist}(A)$  for the set  $\text{td}(\text{ProbExec}(A))$  and define trace distribution inclusion ( $\leq_{\text{td}}$ ) as usual.

**Lemma 1.** Let  $Q$  be a probabilistic execution from  $\mu$  and let  $r \in \text{dom}(Q)$  be given.

1. Every probabilistic branch from  $\mu$  that is a prefix of  $r$  is in  $\text{dom}(Q)$ .
2. For every action  $a$ , there is at most one extension  $r'$  of  $r$  such that  $r' \in \text{dom}(Q)$  and  $\text{tr}(r') = \text{tr}(r)a$ .
3. For all  $r, r' \in \text{dom}(Q)$ ,  $\text{tr}(r) = \text{tr}(r')$  implies  $r = r'$ .

*Proof.* For Item (1), simply note that  $\text{dom}(Q)$  is the smallest subset of  $\text{PBran}(\mu)$  closed under the two rules in Definition 3.

For Item (2), suppose  $r' = r.f.a.\nu'$  and  $r'' = r.g.a.\nu''$  are both extensions of  $r$  reachable via  $Q$ . By minimality of  $\text{dom}(Q)$ , we know that  $f = Q(r) = g$ . By definition of probabilistic branches, it must be the case that  $\nu' = \pi_2(Q(r)(a)) = \nu''$ .

Item (3) follows from induction on the length of traces, using Items (1) and (2).  $\square$

**Corollary 1.** For each probabilistic execution  $Q$  in  $\text{ProbExec}(\mu)$ ,

- $\text{td}(Q)(\alpha) = 0$ , if no probabilistic branch  $r \in \text{dom}(Q)$  has trace  $\alpha$ ;
- otherwise,  $\text{td}(Q)(\alpha) := \Pi[r]$ , where  $r$  is the unique probabilistic branch in  $\text{dom}(Q)$  with trace  $\alpha$ .

*Proof.* By Lemma 1 (Item (3)).

**Draft remark:** I have not bothered with theorems proving that these new notions are in fact probability measures. I claim what I have here are special cases of Roberto's more general definitions. Nonetheless, I should check it at some point.

## 4 Finite Approximations

Fix a symbolic PA  $A$ .

**Definition 5.** We define an ordering  $\leq$  on  $(\bigcup \Delta_A(S_A)) \cup \{\top\}$  as follows:

$$\leq := \mathcal{I}d_{\bigcup \Delta_A(S_A)} \cup \{\langle \top, f \rangle \mid f \in \bigcup \Delta_A(S_A)\},$$

where  $\mathcal{I}d_{\bigcup \Delta_A(S_A)}$  is the identity relation on  $\bigcup \Delta_A(S_A)$ .

This induces an ordering on  $\text{ProbExec}(\mu)$ :  $Q \leq Q'$  if and only if

- $\text{dom}(Q) \subseteq \text{dom}(Q')$ ;
- for all  $r \in \text{dom}(Q)$ ,  $Q(r) \leq Q'(r)$ .

**Draft remark:** Hmm...,  $\top$  is bottom. Actually, if we have  $\perp$  for undefined, it would be the real bottom. **:kramer tfarD**

Essentially,  $\leq$  on  $\text{ProbExec}(\mu)$  is the subset ordering on graphs of partial functions, with a minor modification for  $\top$ . In fact, we could have defined probabilistic executions in such a way that  $\leq$  coincides precisely with subset ordering, but we would lose the nice property that  $\text{dom}(Q)$  is precisely the set of reachable branches (i.e., from  $\text{dom}(Q)$  we need to go one step further to get all reachable branches).

**Lemma 2.** The poset  $(\text{ProbExec}(\mu), \leq)$  is closed under limits of chains.

*Proof.* Let  $\{Q_k \mid k \in \mathbb{N}\} \text{ProbExec}(\mu)$  be a chain of probabilistic executions (with respect to  $\leq$ ). Clearly,  $(\text{PBran}(\mu), \leq)$  is closed under arbitrary joins. We use that to define a partial function  $Q : \text{PBran}(\mu) \rightarrow \bigcup \Delta_A(S_A) \cup \{\top\}$ : for all  $r \in \text{PBran}(\mu)$

- $Q$  is undefined on  $r$  if, for all  $k$ ,  $Q_k$  is undefined on  $r$ ;
- otherwise,  $Q(r) := \bigvee_{k \in \mathbb{N}} Q_k(r)$ .

Clearly,  $Q_k \leq Q$  for all  $k \in \mathbb{N}$ . It remains to show  $Q \in \text{ProbExec}(\mu)$ .

Condition (1) in Definition 3, is satisfied because for all  $r \in \text{dom}(Q)$ , there exists  $k$  such that  $Q(r) = Q_k(r)$ . Moreover, note that  $\text{dom}(Q) = \bigcup_{k \in \mathbb{N}} \text{dom}(Q_k)$ , hence  $\text{dom}(Q)$  is closed under the two rules in Condition (2). We need to show it is the smallest such set.

Let  $\mathcal{B} \subseteq \text{PBran}(\mu)$  be closed under those two rules. We prove, for all  $r \in \text{PBran}(\mu)$ ,  $r \in \text{dom}(Q)$  implies  $r \in \mathcal{B}$  by induction on  $|\text{tr}(r)|$ .

- Clearly  $\mu \in \mathcal{B}$ .
- Consider  $r'$  of the form  $r.f.a.\nu$  and assume  $r' \in \text{dom}(Q)$ . By definition of  $Q$ , choose  $k \in \mathbb{N}$  such that  $r \in \text{dom}(Q_k)$ . By Lemma 1, we know  $r \in \text{dom}(Q_k) \subseteq \text{dom}(Q)$ . Therefore we may apply the induction hypothesis to conclude that  $r \in \mathcal{B}$ . On the other hand, we know
  - $Q(r) = Q_k(r) = f \neq \top$ ;
  - $\pi_1(Q(r)(a)) = \pi_1(f(a)) > 0$ ;
  - $\pi_2(Q(r)(a)) = \pi_2(f(a)) = \nu$ .

By the closure assumption on  $\mathcal{B}$ , it must also contain  $r'$ .

**Definition 6.** Let  $\mu \in S_A$  and  $Q \in \text{ProbExec}(\mu)$  be given. For each  $n \in \mathbb{N}$ , let  $\text{PBran}^n(\mu)$  denote the set of probabilistic branches  $r \in \text{PBran}(\mu)$  with  $|r| = n$ . Then  $\text{dom}^n(Q) := \text{dom}(Q) \cap \text{PBran}^n(\mu)$ . Similarly for  $\text{PBran}^{\leq n}(\mu)$  and  $\text{dom}^{\leq n}(Q)$ . Then the  $n$ -step restriction of  $Q$ , denoted  $Q \upharpoonright_n$ , is the partial function with domain  $\text{dom}^{\leq n}(Q)$  such that:

- for all  $m < n$  and  $r \in \text{dom}^m(Q)$ ,  $Q \upharpoonright_n(r) = Q(r)$ ;
- for all  $r \in \text{dom}^n(Q)$ ,  $Q \upharpoonright_n(r) = \top$ .

**Lemma 3.** For every  $Q \in \text{ProbExec}(\mu)$ , the sequence  $\{Q \upharpoonright_n \mid n \in \mathbb{N}\}$  forms a chain with respect to  $\leq$ . Moreover,  $Q = \bigvee_{n \in \mathbb{N}} Q \upharpoonright_n$ .

**Draft remark:** Algebraic CPO? Needs to consider branching degree. Also for trance distributions? :kramer tfarD

## 5 Relation Lifting

For this section, we fix symbolic PAs  $A$  and  $B$  and a relation  $R \subseteq S_A \times S_B$ .

**Definition 7.** We define correspondence under  $R$  on three levels.

(i) **transition bundle:** Let  $\mu_A \in S_A$  and  $\mu_B \in S_B$  be given. A transition bundle  $f$  from  $\mu_A$  in  $A$  is said to be  $R$ -related to a transition bundle  $g$  from  $\mu_B$  in  $B$  just in case:

- (a)  $\langle \mu_A, \mu_B \rangle \in R$ ,
- (b) for all  $a \in \text{Act}$ ,  $\pi_1(f(a)) = \pi_1(g(a))$ , and
- (c) for all  $a \in \text{Act}$ ,  $\langle \pi_2(f(a)), \pi_2(g(a)) \rangle \in R$ .

(ii) **probabilistic branch:** Let  $\mu_A \in S_A$  and  $\mu_B \in S_B$  be given. We lift  $R$  to a relation  $\bar{R} \subseteq \text{PBran}(\mu_A) \times \text{PBran}(\mu_B)$  as follows:

- (a)  $\langle \mu_A, \mu_B \rangle \in \bar{R}$  (as a pair of length-1 sequences) if and only if  $\langle \mu_A, \mu_B \rangle \in R$  (as a pair of symbolic states);
- (b)  $\langle r_A.f.a.\pi_2(f(a)), r_B.g.b.\pi_2(g(a)) \rangle \in \bar{R}$  if and only if (i)  $\langle r_A, r_B \rangle \in \bar{R}$ , (ii)  $f$  is  $R$ -related to  $g$ , and (iii)  $a = b$ .

Then  $r_A$  is said to be  $R$ -related to  $r_B$  if  $\langle r_A, r_B \rangle \in \bar{R}$ .

(iii) **probabilistic execution:** Let  $\mu_A \in S_A$  and  $\mu_B \in S_B$  be given. A probabilistic execution  $Q_A \in \text{ProbExec}(\mu_A)$  is said to be  $R$ -related to a probabilistic execution  $Q_B \in \text{ProbExec}(\mu_B)$  just in case:

- (a)  $\langle \mu_A, \mu_B \rangle \in R$ ;
- (b) for all probabilistic branches  $r_A \in \text{dom}(Q_A)$  and  $r_B \in \text{dom}(Q_B)$ ,  $r_A$  is  $R$ -related to  $r_B$  implies
  - i. either  $Q_A(r_A) = Q_B(r_B) = \top$ ,
  - ii. or  $Q_A(r_A) \neq \top$ ,  $Q_B(r_B) \neq \top$ , and  $Q_A(r_A)$  is  $R$ -related to  $Q_B(r_B)$ .

For the following lemmas, let  $\mu_A \in S_A$  and  $\mu_B \in S_B$  be given and assume  $\langle \mu_A, \mu_B \rangle \in R$ .

**Lemma 4.** *Let  $r_A \in \text{P Bran}(\mu_A)$  and  $r_B \in \text{P Bran}(\mu_B)$  be  $R$ -related. Then*

1.  $r_A$  and  $r_B$  are of the same length;
2.  $\langle \text{last}(r_A), \text{last}(r_B) \rangle \in R$ .
3.  $\text{tr}(r_A) = \text{tr}(r_B)$  and  $\Pi[r_A] = \Pi[r_B]$ .

*Proof.* Item (1) follows from a trivial induction. Item (2) is also by induction, using the definition of  $R$ -relatedness for transition bundles at the inductive step.

We prove Item (3) by (nested) induction on the lengths of  $r_A$  and  $r_B$ .

- $r_A = \mu_A$  and  $r_B = \mu_B$ . Clearly  $\text{tr}(r_A) = \text{tr}(r_B) = \epsilon$  and  $\Pi[r_A] = \Pi[r_B] = 1$ .
- $r_A = \mu_A$  and consider  $r'_B$  of the form  $r_B.g.a.\nu_B$ . Then  $r_A$  is not  $R$ -related to  $r'_B$  because they are of different lengths (Item (1)).
- $r'_A$  is of the form  $r_A.f.a.\nu_A$  and  $r'_B$  is of the form  $r_B.g.b.\nu_B$ . By definition of  $R$ -relatedness for branches, we know that  $r_A$  is  $R$ -related to  $r_B$ ,  $f$  is  $R$ -related to  $g$  and  $a = b$ . By the induction hypothesis, we have  $\text{tr}(r_A) = \text{tr}(r_B)$  and  $\Pi[r_A] = \Pi[r_B]$ . It is immediate that  $\text{tr}(r'_A) = \text{tr}(r'_B)$ . Furthermore, by definition of  $R$ -relatedness for transition bundles, we have  $\pi_1(f(a)) = \pi_1(g(a))$  for all  $a \in \text{Act}$ . Hence,

$$\Pi[r'_A] = \Pi[r_A] \cdot \pi_1(f(a)) = \Pi[r_B] \cdot \pi_1(g(a)) = \Pi[r'_B].$$

□

**Lemma 5.** *Let  $Q_A \in \text{ProbExec}(\mu_A)$  and  $Q_B \in \text{ProbExec}(\mu_B)$  be given. Let  $\bar{R} \subseteq \text{P Bran}(\mu_A) \times \text{P Bran}(\mu_B)$  be as in Definition 7. If  $Q_A$  is  $R$ -related to  $Q_B$ , then  $\bar{R}$  is a bijection between  $\text{dom}(Q_A)$  and  $\text{dom}(Q_B)$ .*

*Proof.* It suffices to show that  $\bar{R}$  is a bijection between  $\text{dom}^n(Q_A)$  and  $\text{dom}^n(Q_B)$  for all  $n \in \mathbb{N}$ . We proceed by induction on  $n$  (what else).

- By assumption,  $\langle \mu_A, \mu_B \rangle$  (as a pair of symbolic states) is in  $R$ , thus  $\langle \mu_A, \mu_B \rangle$  (as a pair of probabilistic branches) is in  $\bar{R}$ . Moreover,  $\text{dom}^0(Q_A) = \{\mu_A\}$  and  $\text{dom}^0(Q_B) = \{\mu_B\}$ , thus  $\bar{R}$  is in fact a bijection.
- Let  $r'_A \in \text{dom}^{n+1}(Q_A)$  be given. Then  $r'_A$  is of the form

$$r_A.Q_A(r_A).a.\pi_2(Q_A(r_A)(a))$$

where  $r_A \in \text{dom}^n(Q_A)$ . By the induction hypothesis, there is unique  $r_B \in \text{dom}^n(Q_B)$  with  $\langle r_A, r_B \rangle \in \bar{R}$ . By definition of  $R$ -relatedness for probabilistic executions, we know that  $Q_B(r_B) \neq \top$  and  $Q_A(r_A)$  is  $R$ -related to  $Q_B(r_B)$ . Therefore,  $\pi_1(Q_B(r_B)(a)) = \pi_1(Q_A(r_A)(a)) > 0$ , so

$$r'_B := r_B.Q_B(r_B).a.\pi_2(Q_B(r_B)(a))$$

is in fact a probabilistic branch in  $\text{dom}^{n+1}(Q_B)$ . Clearly,  $\langle r'_A, r'_B \rangle \in \bar{R}$ . This shows every  $r'_A \in \text{dom}^{n+1}(Q_A)$  is related to some  $r'_B \in \text{dom}^{n+1}(Q_B)$  via  $\bar{R}$ . Now suppose there is some  $r''_B \in \text{dom}^{n+1}(Q_B)$  such that  $\langle r'_A, r''_B \rangle$  is also in  $\bar{R}$ . By Lemma 4 (Item (3)), we have  $\text{tr}(r''_B) = \text{tr}(r'_A) = \text{tr}(r'_B)$ . Applying Lemma 1, we conclude that  $r''_B = r'_B$ . Thus, every  $r'_A \in \text{dom}^{n+1}(Q_A)$  is related to a unique  $r'_B \in \text{dom}^{n+1}(Q_B)$  via  $\bar{R}$ .

By symmetry, every  $r'_B \in \text{dom}^{n+1}(Q_B)$  is related to a unique  $r'_A \in \text{dom}^{n+1}(Q_A)$  via  $\bar{R}^{-1}$ .

□

**Corollary 2.** *Let  $Q_A \in \text{ProbExec}(\mu_A)$  and  $Q_B \in \text{ProbExec}(\mu_B)$  be given. If  $Q_A$  is  $R$ -related to  $Q_B$ , then  $\text{td}(Q_A) = \text{td}(Q_B)$ .*

*Proof.* By Lemmas 4 and 5,  $\bar{R}$  is a bijection between  $\text{dom}(Q_A)$  and  $\text{dom}(Q_B)$  satisfying: if  $\langle r_A, r_B \rangle \in \bar{R}$ , then  $\text{tr}(r_A) = \text{tr}(r_B)$  and  $\Pi[r_A] = \Pi[r_B]$ . Therefore, by definition of trace distribution,  $\text{td}(Q_A) = \text{td}(Q_B)$ . □

**Lemma 6.** *Let  $\{Q_{A,n} \mid n \in \mathbb{N}\} \subseteq \text{ProbExec}(\mu_A)$  and  $\{Q_{B,n} \mid n \in \mathbb{N}\} \subseteq \text{ProbExec}(\mu_B)$  be two chains (with respect to  $\leq$ ) satisfying: for all  $n \in \mathbb{N}$ ,  $Q_{A,n}$  is  $R$ -related to  $Q_{B,n}$ . Let  $Q_A := \bigvee_{n \in \mathbb{N}} Q_{A,n}$  and similarly for  $Q_B$ . Then  $Q_A$  is  $R$ -related to  $Q_B$ .*

*Proof.* By assumption  $\langle \mu_A, \mu_B \rangle \in R$ . Let  $r_A \in \text{dom}(Q_A)$  and  $r_B \in \text{dom}(Q_B)$  be given and assume  $r_A$  is  $R$ -related to  $r_B$ . Choose  $n$  such that  $r_A \in \text{dom}(Q_{A,n})$  and  $Q_A(r_A) = Q_{A,n}(r_A)$ . Similarly, choose such  $m$  for  $r_B$ . Define  $N := \max(n, m)$ . Then  $r_A \in \text{dom}(Q_{A,N})$  and  $r_B \in \text{dom}(Q_{B,N})$ . Since  $Q_{A,N}$  is  $R$ -related to  $Q_{B,N}$ , it must be the case that

- either  $Q_{A,N}(r_A) = Q_{B,N}(r_B) = \top$ ,
- or  $Q_{A,N}(r_A) \neq \top$ ,  $Q_{B,N}(r_B) \neq \top$ , and  $Q_{A,N}(r_A)$  is  $R$ -related to  $Q_{B,N}(r_B)$ .

On the other hand, we have  $Q_A(r_A) = Q_{A,N}(r_A) = Q_{A,n}(r_A)$  and  $Q_B(r_B) = Q_{B,N}(r_B) = Q_{B,m}(r_B)$ , therefore the desired conclusion follows.

## 6 Simulation and Soundness

**Definition 8.** *Let  $A$  and  $B$  be symbolic PAs. A super-duper simulation from  $A$  to  $B$  is a relation  $R \subseteq S_A \times S_B$  such that:*

1.  $\langle \mu_A^0, \mu_B^0 \rangle \in R$ ;
2. if  $\langle \mu_A, \mu_B \rangle \in R$  and there exists a transition bundle  $f$  from  $\mu_B$ , then there exists a transition bundle  $g$  from  $\mu_A$  such that for all action  $a \in \text{Act}$ ,
  - (a)  $\pi_1(f(a)) = \pi_1(g(a))$  and
  - (b)  $\langle \pi_2(f(a)), \pi_2(g(a)) \rangle \in R$ .
 (Equivalently, there exists  $g$  from  $\mu_B$  such that  $f$  is  $R$ -related to  $g$ .)

Notice this definition has exactly the form suggested by the functor  $F(S) = \mathcal{P}(\text{Act} \rightarrow ([0, 1] \times S))$ . Now we prove an execution correspondence lemma.

**Lemma 7.** *Let  $A$  and  $B$  be symbolic PAs and let  $R$  be a simulation from  $A$  to  $B$  in the sense of Definition 8. Let  $\mu_A \in S_A$  and  $\mu_B \in S_B$  be given and assume that  $\langle \mu_A, \mu_B \rangle \in R$ . For every  $Q_A \in \text{ProbExec}(\mu_A)$ , there exists  $Q_B \in \text{ProbExec}(\mu_B)$  such that  $Q_A$  is  $R$ -related to  $Q_B$ .*

*Proof.* Consider the chain  $\{Q_A \upharpoonright_n \mid n \in \mathbb{N}\}$ . We construct recursively a chain  $\{Q_{B,n} \mid n \in \mathbb{N}\} \subseteq \text{ProbExec}(\mu_B)$  such that  $Q_A \upharpoonright_n$  is  $R$ -related to  $Q_{B,n}$  for all  $n \in \mathbb{N}$ . Then, by Lemmas 3 and 6, we may conclude that  $Q_A$  is  $R$ -related to  $\bigvee_{n \in \mathbb{N}} Q_{B,n} \in \text{ProbExec}(\mu_B)$ .

1.  $\text{dom}(Q_{B,0}) := \{\mu_B\}$  and  $Q_{B,0} := \top$ . Clearly,  $Q_A \upharpoonright_0$  is  $R$ -related to  $Q_{B,0}$ .
2. Suppose we have defined a chain  $\{Q_{B,i} \mid 0 \leq i \leq n\} \subseteq \text{ProbExec}(\mu_B)$  such that  $Q_A \upharpoonright_i$  is  $R$ -related to  $Q_{B,i}$  for all  $0 \leq i \leq n$ . By Lemma 5,  $\bar{R}$  (as in Definition 7) is a bijection between  $\text{dom}(Q_A \upharpoonright_n)$  and  $\text{dom}(Q_{B,n})$ . Therefore  $\bar{R}^{-1}(r_B)$  is well-defined for each  $r_B \in \text{dom}(Q_{B,n})$ . Define  $Q_{B,n+1}$  as follows.
  - (a) For each  $r_B \in \text{dom}(Q_{B,n})$  with  $|r_B| < n$ ,  $Q_{B,n+1}(r_B) := Q_{B,n}(r_B)$ .
  - (b) For each  $r_B \in \text{dom}(Q_{B,n})$  with  $|r_B| = n$ ,  $Q_{B,n+1}(r_B) := \top$  if  $Q_A(\bar{R}^{-1}(r_B)) = \top$ . Otherwise, by Definition 8, we may choose a transition bundle  $g$  from  $\text{last}(r_B)$  such that  $Q_A(\bar{R}^{-1}(r_B))$  is  $R$ -related to  $g$ . Define  $Q_{B,n+1}(r_B) := g$ .
  - (c) For all  $r'_B \in \text{PBran}(\mu_B)$  of the form  $r_B.g.a.v$ , where  $r_B$  and  $g$  are as in Item (2b),  $Q_{B,n+1}(r'_B) := \top$ .
  - (d) For all other  $r_B \in \text{PBran}(\mu_B)$ ,  $Q_{B,n+1}(r_B)$  is undefined.
 Clearly,  $Q_{B,n} \leq Q_{B,n+1}$  and  $Q_A \upharpoonright_{n+1}$  is  $R$ -related to  $Q_{B,n+1}$ .

**Theorem 1.** *Let  $A$  and  $B$  be symbolic PAs and let  $R$  be a simulation from  $A$  to  $B$  in the sense of Definition 8. Then  $A \leq_{\text{td}} B$ .*

*Proof.* By Corollary 2 and Lemma 7.

## 7 I/O Distinction

**Draft remark:** Argue here that I/O distinction is essential for a reasonable definition of composition, because it removes unrealistic possibilities of deadlock and thus avoids the need to normalize probabilities. For example, consider a case in which component  $A$  sends a message to  $B$  with probability 1, but  $B$  accepts the message with some probability  $p < 1$ , while wanting to do something else with probability  $1 - p$ . Then a deadlock occurs with probability  $1 - p$ . This is not sensible, because  $B$  should not have control over whether a message is sent to  $B$  from some other party. **:kramer tfarD**

**Draft remark:** The definitions below are rather restrictive, although we believe they are sufficient for OT, which involves randomization of message contents only. In particular, every hidden bundle carries a unique action label. This eliminates the need to flatten target symbolic states in the definition of weak transition bundles. Notice that flattening doesn't really make sense unless we require that  $S_A$  is closed under convex combinations. See also Lemma 8 for further implications of this restriction. **:kramer tfarD**

**Definition 9.** *A symbolic probabilistic I/O automaton (PIOA) is a tuple  $A = \langle S_A, \mu_A^0, \text{Act}_A, \Delta_A \rangle$  where*

- the triple  $\langle S_A, \mu_A^0, \Delta_A \rangle$  is a symbolic PA;
- $\text{Act}_A \subseteq \text{Act}$  is partitioned into  $\{I_A, O_A, H_A\}$  (input, output, and hidden actions, respectively);
- $\Delta_A$  satisfies two axioms:
  1. i/o: for all  $\mu \in S_A$  and  $f \in \Delta_A(\mu)$ ,

- $\text{Supp}(\pi_1(f)) = \{a\}$  for some  $a \in I_A$  ( $f$  is then an input transition bundle with label  $a$ );
  - $\text{Supp}(\pi_1(f)) \subseteq O_A$  (we say in this case  $f$  is an output transition bundle), or
  - $\text{Supp}(\pi_1(f)) = \{a\}$  for some  $a \in H_A$  ( $f$  is then a hidden transition bundle with label  $a$ );
2. input enabling: for all  $\mu \in S_A$  and  $a \in I_A$ , there exists an input transition bundle  $f \in \Delta_A(\mu)$  with label  $a$ ;

The i/o axiom divides transition bundles into three classes: input, output and hidden. As usual, the first two are referred to as *visible* and the last two *locally controlled*. In addition, the axiom requires that every input is received with probability 1 (otherwise a deadlock may occur). **Draft remark:** See last draft remark for justifications of the hidden clause. **:kramer tfarD** As in the model of I/O Automata, the input enabling axiom requires that the automaton is willing to accept any input at any state.

Traces and trace distributions are defined as usual.

**Definition 10.** Let  $A$  be a symbolic PIOA and let  $\mu \in S_A$  be given.

- (i) The (visible) trace of a probabilistic branch  $r$  in  $\text{PBran}(\mu)$ , denoted  $\text{tr}(r)$ , is defined recursively as follows:
- $\text{tr}(\mu) := \epsilon$ ;
  - $\text{tr}(r.f.a.\nu) := \text{tr}(r)$  if  $a \in H_A$ ;
  - $\text{tr}(r.f.a.\nu) := \text{tr}(r)a$  if  $a \in I_A \cup O_A$ .
- (ii) A probabilistic branch  $r'$  is said to be minimal if, for every prefix  $r$  of  $r'$ ,  $\text{tr}(r) = \text{tr}(r')$  implies  $r = r'$ . (Equivalently,  $r'$  is either empty or of the form  $r.f.a.\nu$  where  $a \notin H_A$ .)
- (iii) Let  $Q \in \text{ProbExec}(\mu)$  be given. The trace distribution induced by  $Q$ , denoted  $\text{td}(Q)$ , is the function from  $(I_A \cup O_A)^{<\omega}$  to  $[0, 1]$  defined by:

$$\text{td}(Q)(\alpha) := \sum_{\{r \in \text{dom}(Q) \mid r \text{ minimal and } \text{tr}(r) = \alpha\}} \Pi[r].$$

- (iv) As before, we write  $\text{TrDist}(A)$  for the set  $\text{td}(\text{ProbExec}(A))$  and define trace distribution inclusion ( $\leq_{\text{td}}$ ) accordingly.

Below we define hidden executions and examine some of their properties.

**Definition 11.** Let  $A$  be a symbolic PIOA and let  $\mu \in S_A$  be given. A probabilistic execution  $Q \in \text{ProbExec}(\mu)$  is said to be hidden if, for all  $r \in \text{dom}(Q)$ ,  $Q(r)$  is a hidden transition bundle.

**Lemma 8.** Let  $Q$  be a probabilistic execution from  $\mu$ .

- (1) For all  $r \in \text{dom}(Q)$  and  $n \in \mathbb{N}$ , there is at most one  $n$ -step extension  $r'$  of  $r$  such that  $r' \in \text{dom}(Q)$  and  $\text{tr}(r') = \text{tr}(r)$ .
- (2) If  $Q$  is hidden, then  $\text{dom}(Q)$  is linearly ordered by prefix.

(3) If  $Q$  is hidden and  $\text{dom}(Q)$  is finite, then there exists a unique maximal  $r \in \text{dom}(Q)$ .

*Proof.* By the i/o axiom, every hidden transition bundle carries a unique label, thus leads to a unique target symbolic state. Hence, there is at most one  $r'$  such that  $r' = r.Q(r).a.\nu$  where  $a \in H_A$ . Following this observation, a simple inductive argument proves Item (1).

For Item (2), we prove by induction on  $n$  that the set  $\text{dom}^n(Q) = \{r \in \text{dom}(Q) \mid |r| = n\}$  is either empty or a singleton. Notice  $\text{dom}^0(Q) = \{\mu\}$  by minimality of  $\text{dom}(Q)$ , therefore the base case holds. The inductive step follows from Item (1).

Item (3) is an immediate consequence of Item (2).

By virtue of Lemma 8, every finite hidden execution  $Q$  can be identified with the maximal element  $r \in \text{dom}(Q)$ . We write  $r_Q$  for this element and  $\text{last}(Q)$  for  $\text{last}(r_Q)$ .

## 8 Weak Transition Bundles

Here we introduce the notion of weak transition bundles, following the style of Milner's weak transitions. In Section 9, we will define weak simulation based on these weak bundles.

**Definition 12.** Let  $A$  be a symbolic PIOA and let  $\mu \in S_A$  be given. A weak transition bundle from  $\mu$  is a probabilistic execution  $Q' \in \text{ProbExec}(\mu)$  for which there exists a hidden execution  $Q \in \text{ProbExec}(\mu)$  such that

- $Q \leq Q'$  (in the sense of Definition 5);
- $Q'(r_Q)$  is a visible bundle;
- for all  $r \in \text{dom}(Q') \setminus \text{dom}(Q)$ ,  $Q'(r) = \top$ .

We write  $\text{Fiber}(Q', a)$  for the fiber probability  $\pi_1(Q'(r_Q)(a))$  and  $\text{new}(Q', a)$  for the target symbolic state  $\pi_2(Q'(r_Q)(a))$ .

Essentially, a weak transition bundle is a visible transition bundle preceded by a possibly empty sequence of hidden bundles. For simplicity, we do not consider executions in which hidden actions are performed *after* the visible actions. Clearly, both options lead to the same notion of weak simulation, which is our primary interest.

**Lemma 9.** Let  $A$  be a symbolic PIOA and let  $\mu \in S_A$  be given. For each weak transition bundle  $Q'$  from  $\mu$ , the function  $\text{Fiber}(Q', -) : (\text{Act}_A \setminus H_A) \rightarrow [0, 1]$  is either a discrete distribution on  $O_A$ , or the dirac distribution on  $a$  for some  $a \in I_A$ .

*Proof.* Choose hidden execution  $Q$  as in Definition 12. Notice that  $\text{Fiber}(Q', -) = \pi_1(Q'(r_Q))$ . Therefore, the desired condition follows from the definition of (strong) transition bundles and the i/o axiom.

Next, we defined in the familiar manner a new transition structure based on weak bundles.

**Definition 13.** *Let  $A$  be a symbolic PIOA. The weak transition structure on  $A$ , denoted  $\Omega_A$ , is the function from  $S_A$  to  $\mathcal{P}((Act \setminus H_A) \rightarrow ([0, 1] \times S_A))$  given by: for all  $\mu \in S_A$ ,*

$$\Omega_A(\mu) := \{\langle \text{Fiber}(Q', -), \text{new}(Q', -) \rangle \mid Q' \text{ is a weak transition bundle from } \mu.\}$$

By Lemma 9, the tuple  $\bar{A} = \langle S_A, \mu_A^0, \langle I_A, O_A \rangle, \Omega_A \rangle$  is a symbolic PA satisfying the  $i/o$  axiom. Notice that  $\Omega_A$  satisfies input enabling whenever  $\Delta_A$  does. Therefore  $\bar{A}$  is in fact a symbolic PIOA.

## 9 Weak Simulation and Soundness

**Definition 14.** *Let  $A$  and  $B$  be symbolic PAs. A weak simulation from  $A$  to  $B$  is a relation  $R \subseteq S_A \times S_B$  such that:*

1.  $\langle \mu_A^0, \mu_B^0 \rangle \in R$ ;
2. if  $\langle \mu_A, \mu_B \rangle \in R$  and there exists a transition bundle  $f$  from  $\mu_B$ , then there exists a transition bundle  $g$  from  $\mu_A$  such that for all action  $a \in Act$ ,
  - (a)  $\pi_1(f(a)) = \pi_1(g(a))$  and
  - (b)  $\langle \pi_2(f(a)), \pi_2(g(a)) \rangle \in R$ .
 (Equivalently, there exists  $g$  from  $\mu_B$  such that  $f$  is  $R$ -related to  $g$ .)

**Draft remark:** Materials from this point on do not concern OT. :kramer tfarD

## 10 Parallel Composition

Symbolic PIOAs  $A$  and  $B$  are said to be *compatible* if  $O_A \cap O_B = \emptyset$ . Let  $\{A_i \mid 1 \leq i \leq n\}$  be pairwise compatible symbolic PIOAs. We define the composite  $\parallel^n \{A_i \mid 1 \leq i \leq n\}$  to be the following symbolic PIOA  $P$ .

1.  $S_P := \prod_{i=1}^n S_i$  and the start state is  $\langle \mu_1^0, \dots, \mu_n^0 \rangle$ ;
2.  $I_P := (\bigcup_{i=1}^n I_i) \setminus (\bigcup_{i=1}^n O_i)$ ,  $O_P := (\bigcup_{i=1}^n O_i)$ ;
3. given a state  $\langle \mu_1, \dots, \mu_n \rangle$ ,  $\Delta_P(\langle \mu_1, \dots, \mu_n \rangle)$  contains precisely those functions  $f : Act_P \rightarrow [0, 1] \times S_P$  satisfying one of the following conditions.
  - (a) There exists  $1 \leq i \leq n$  and locally controlled transition bundle  $g_i$  from  $\mu_i$  such that:
    - for all  $a \in O_i$ ,  $f(a) = \langle \pi_1(g_i(a)), \langle \nu_1, \dots, \nu_n \rangle \rangle$ , where
      - $\nu_i = \pi_2(g_i(a))$ ;
      - for all  $j \neq i$  such that  $a \notin I_j$ ,  $\nu_j = \mu_j$ ;
      - for all  $j \neq i$  such that  $a \in I_j$ ,  $\nu_j = \pi_2(h_j(a))$  for some input transition bundle  $h_j$  from  $\mu_j$  with label  $a$ ;
    - for all other  $a \in Act_P$ ,  $f(a) = \langle 0, \langle \mu_1, \dots, \mu_n \rangle \rangle$ .
  - (b) There exists action  $a \in I_P$  such that:
    - $f(a) = \langle 1, \langle \nu_1, \dots, \nu_n \rangle \rangle$ , where

- for all  $i$  such that  $a \in I_i$ ,  $\nu_i = \pi_2(h_i(a))$  for some input transition bundle  $h_i$  from  $\mu_i$  carrying label  $a$ ;
- for all  $i$  such that  $a \notin I_i$ ,  $\nu_i = \mu_i$ ;
- for all other  $b \in Act_P$ ,  $f(b) = \langle 0, \langle \mu_1, \dots, \mu_n \rangle \rangle$ .

This composite is *closed* if  $I_P = \emptyset$ . For  $n = 2$ , we write  $\parallel$  and use infix notation.

**Lemma 10.** *The automaton  $\parallel^n \{A_i \mid 1 \leq i \leq n\}$  is in fact a symbolic PIOA.*

*Proof.* Let  $\mu = \langle \mu_1, \dots, \mu_n \rangle$  and  $f \in \Delta_{\parallel^n \{A_i \mid 1 \leq i \leq n\}}(\langle \mu_1, \dots, \mu_n \rangle)$  be given. If  $f$  arises from Clause (3b) in Definition 9, then clearly  $\pi_1(f)$  is the dirac measure on  $a$ . Otherwise, assume that  $f$  arises from Clause (3a). Choose locally controlled transition bundle  $g_i$  from  $\mu_i$  such that  $f$  is induced by  $g_i$ . Notice that  $\pi_1(f)$  coincides with  $\pi_1(g_i)$  on  $O_i$  and is 0 elsewhere. Since  $\pi_1(g_i)$  is a discrete distribution, so is  $\pi_1(f)$ .

The argument above shows that  $f$  satisfies both the axiom for PA in Definition 1 and the i/o axiom in Definition 9. Input enabling follows from Clause (3b) of Definition 9 and the fact that every  $A_i$  satisfies input enabling.

**Lemma 11.** *The composition operator  $\parallel$  is commutative and associative.*

*Proof.* Commutativity is trivial. For associativity, it is easy to see that  $(A \parallel B) \parallel C$  is isomorphic to  $\parallel^3 \{A, B, C\}$ .

## 11 Probabilistic Automata

Our underlying theory is that of (*simple*) *probabilistic automata* as introduced by Segala and Lynch [Seg95]. This extends the usual non-deterministic automata model by allowing probabilistic information at the target of each transition. More precisely, every transition in a probabilistic automaton leads to a probability distribution over possible next states, rather than a single state.

**Draft remark:** As a starting point, we consider systems with no internal actions and no input/output distinction. Though, we keep in mind that internal actions are important in modeling OT. Also, i/o distinction is important in defining composition.

Fix a countable action alphabet  $Act$ . The set of finite (resp. infinite) traces is denoted  $Act^{<\omega}$  (resp.  $Act^\omega$ ), while the set of all traces is  $Act^{\leq\omega}$ . Also, we write  $\epsilon$  for the empty trace.

**Definition 15.** *A probabilistic automaton (PA) is a triple  $A = (S, s^0, \Delta)$  where*

- $S$  is the set of states,
- $s^0 \in S$  is the initial state, and
- $\Delta \subseteq S \times Act \times \text{Disc}(S)$  is the transition relation.

We write  $s \xrightarrow{a} \mu$  for  $(s, a, \mu) \in \Delta$ . Also, we write  $s \xrightarrow{a, \mu} t$  whenever  $s \xrightarrow{a} \mu$  and  $\mu(t) > 0$ . To avoid confusion, we sometimes refer to the components of  $A$  as  $S_A$ ,  $s_A^0$  and  $\Delta_A$ .

Intuitively, we can view target distributions in the transition relation  $\Delta$  as a form of probabilistic branching; i.e., we think of  $s \xrightarrow{a, \mu} t$  as a non-deterministic transition  $s \xrightarrow{a} \mu$  followed by a probabilistic transition  $\mu \xrightarrow{\mu(t)} t$ . In this way, we obtain an informal notion of the *underlying non-deterministic automaton* of  $A$ , where we “forget” probabilistic information (i.e.,  $\mu(t)$ ) at each probabilistic transition.

As in the case of non-deterministic automata, we are interested in certain finiteness properties in branching structure.

**Definition 16.** *A PA  $A$  is finitely (resp. countably) branching if, for each state  $s$ , the set  $\{\langle a, \mu \rangle \mid s \xrightarrow{a} \mu\}$  is finite (resp. countable). It is image finite if for each state  $s$  and action  $a$ , the set  $\{\mu \mid s \xrightarrow{a} \mu\}$  is finite.*

Thus, each state in a finitely branching PA has finitely many outgoing transitions, while a state in an image finite PA may have infinitely many. In both cases, the set  $\{t \mid s \xrightarrow{a, \mu} t \text{ for some } a, \mu\}$  maybe infinite, since a target distribution  $\mu$  may have infinite support. As a result, given a finite trace  $\beta \in \text{Act}^{<\omega}$ , a finitely branching (or image finite) PA may have infinitely many paths with trace  $\beta$ . This is different from the case of non-deterministic automata.

## 12 Transition Bundles and Fibers

**Definition 17.** *Let  $\mu$  be a discrete distribution on the states of  $A$ . A transition bundle from  $\mu$  is a function  $f : \text{Supp}(\mu) \rightarrow (\text{Act} \times \text{Disc}(S_A)) \cup \{\perp\}$  such that, for all  $s, a$  and  $\nu$ ,*

- $f(s) = \langle a, \nu \rangle$  implies  $s \xrightarrow{a} \nu$ ;
- $f(s) = \perp$  implies  $s$  does not enable any transitions.

*We say that a transition  $s \xrightarrow{a} \nu$  is in  $f$  just in case  $f(s) = \langle a, \nu \rangle$ . We use  $\text{Bun}(\mu)$  to denote the set of all transition bundles from  $\mu$  and  $\text{Bun}(A)$  to denote  $\bigcup_{\mu \in \text{Disc}(S_A)} \text{Bun}(\mu)$ .*

The  $\perp$  element represents the default choice in case an execution reaches some terminal state with non-zero probability. Notice we don’t allow  $f$  to return  $\perp$  if the execution may continue. This prevents  $f$  from “revealing” the fact that a coin has been tossed by halting in case of head and not halting in case of tail.

**Draft remark:** When we move on to I/O setting, we need to replace “terminal” state with “quiescent” state while defining an output transition bundle. Input transition bundles should carry a unique input label, because all inputs should be enabled with probability 1.

Given such a distribution  $\mu$  and a transition bundle  $f$  from  $\mu$ , we can compute the probability of observing a particular action  $a$ . To do so, we first introduce the notion of fibers within a bundle.

**Definition 18.** Let  $a$  be an action in  $Act$ . The  $a$ -fiber of a transition bundle  $f$  (from distribution  $\mu$ ), denoted  $\text{Fiber}(f, a)$ , is defined to be

$$\{\langle s, a, \nu \rangle \mid s \in \text{Supp}(\mu) \text{ and } f(s) = \langle a, \nu \rangle\}.$$

In other words,  $\text{Fiber}(f, a)$  is the set of transitions in  $f$  carrying label  $a$ . With slight abuse of notation, we say that a state  $s$  is in the  $a$ -fiber of  $f$ , written  $s \in \text{Fiber}(f, a)$  if  $f(s) = \langle a, \nu \rangle$  for some  $\nu$ . (Intuitively, this says  $f$  chooses an  $a$ -transition from state  $s$ .)

To each fiber we associate a fiber probability, denoted  $\mathbf{P}[a|\mu, f]$ , as follows:

$$\mathbf{P}[a|\mu, f] := \sum_{s \in \text{Fiber}(f, a)} \mu(s).$$

(Note that, by definition,  $\mathbf{P}[a|\mu, f] = 0$  whenever  $\text{Fiber}(f, a)$  is empty.)

Moreover, we can compute the resulting distribution on states, given that a particular action  $a$  is observed.

**Definition 19.** Let  $\mu, f, a$  be given as above with  $\text{Fiber}(f, a)$  non-empty. Define a distribution on states  $\text{new}(\mu, f, a)$  by

$$\sum_{s \in \text{Fiber}(f, a)} \frac{\mu(s)}{\mathbf{P}[a|\mu, f]} \cdot \nu_s,$$

where  $\nu_s$  is the unique  $\nu$  such that  $f(s) = \langle a, \nu \rangle$  (i.e.,  $\nu_s = \pi_2(f(s))$ ). Intuitively,  $\text{new}(\mu, f, a)$  is the target distribution reached given that (i) the initial distribution is  $\mu$ ; (ii) the transition bundle  $f$  is chosen; and (iii) the action  $a$  is observed.

Thus, each transition bundle  $f$  from  $\mu$  induces a mapping  $\bar{f} : Act \rightarrow [0, 1] \times \text{Disc}(S_A)$  where  $\bar{f}(a) := \langle \mathbf{P}[a|\mu, f], \text{new}(\mu, f, a) \rangle$ . Notice that  $\pi_1(\bar{f})$  is a discrete distribution on  $Act$  (because  $\mu$  is a discrete distribution on states), therefore we can view  $\bar{f}$  as a probabilistic transition from  $\mu$ . However, every  $\bar{f}$ , thus obtained, has the property that every action leads to a unique target distribution. This is because, in the definition of  $\text{new}(\mu, f, a)$ , we “flatten” the set of target distributions reached by performing the same action  $a$ .

*Example 2.* Suppose the initial distribution  $\mu$  is  $\{\langle s_1, \frac{1}{4} \rangle, \langle s_2, \frac{1}{4} \rangle, \langle s_3, \frac{1}{2} \rangle\}$  and the transition bundle  $f$  is  $\{\langle s_1, a, \nu_1 \rangle, \langle s_2, a, \nu_2 \rangle, \langle s_3, b, \nu_3 \rangle\}$ . Then the fiber probability for  $a$  is  $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$  and that for  $b$  is  $\frac{1}{2}$ . Moreover,  $\text{new}(\mu, f, a)$  is  $\frac{1}{2}\nu_1 + \frac{1}{2}\nu_2$  and  $\text{new}(\mu, f, b)$  is  $\nu_3$ .

## 13 Internal Actions

## 14 $\varepsilon$ -Bisimulation and Oblivious Transfer

## 15 Compositionality

## References

- [Agg94] 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.
- [Can01] R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computing*, pages 136–145, 2001.
- [CLSV04] 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.
- [dAHJ01] 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.
- [LMMS98] P. Lincoln, J.C. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic poly-time framework for protocol analysis. In *ACM Conference on Computer and Communications Security*, pages 112–121, 1998.
- [LSS94] 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.
- [LSV03] 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.
- [LT89] N.A. Lynch and M.R. Tuttle. An introduction to input/output automata. *CWI Quarterly*, 2(3):219–246, September 1989.
- [PSL00] A. Pogonyants, 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.
- [Seg95] 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.
- [Sto02] M.I.A. Stoelinga. An introduction to probabilistic automata. *Bulletin of the European Association for Theoretical Computer Science*, 78:176–198, October 2002.
- [SV99] 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.

- [SV04] 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*, volume 2925 of *Lecture Notes in Computer Science*, pages 1–43. Springer-Verlag, 2004.
- [WSS94] S.-H. Wu, S. A. Smolka, and E. W. Stark. Composition and behaviors of probabilistic i/o automata. In B. Jonsson and J. Parrow, editors, *Proceedings CONCUR 94*, Uppsala, Sweden, volume 836 of *Lecture Notes in Computer Science*, pages 513–528. Springer-Verlag, 1994.