Composition and Behaviors of Probabilistic I/O Automata*

Sue-Hwey Wu, Scott A. Smolka, Eugene W. Stark

Department of Computer Science

State University of New York at Stony Brook

Stony Brook, NY 11794 USA

Abstract

We augment the I/O automaton model of Lynch and Tuttle with probability, as a step toward the ultimate goal of obtaining a useful tool for specifying and reasoning about asynchronous probabilistic systems. Our new model, called *probabilistic I/O automata*, preserves the fundamental properties of the I/O automaton model, such as the asymmetric treatment of input and output and the pleasant notion of asynchronous composition. For certain classes of probabilistic I/O automata, we show that *probabilistic behavior maps*, which are an abstract representation of I/O automaton behavior in terms of a certain expectation operator, are compositional and fully abstract with respect to a natural notion of probabilistic testing.

1 Introduction

I/O Automata are a kind of state machine that have been proposed by Lynch and Tuttle [LT87] as a tool for specifying and reasoning about asynchronous systems. The distinguishing features of the I/O automaton model are: (1) an asymmetric treatment of input and output actions, (2) a notion of asynchronous composition that takes a "compatible" collection of I/O automata and produces a new I/O automaton as a result, (3) a simple correspondence between computations of a composite I/O automaton and certain collections of computations of its component automata, (4) the treatment of liveness properties through the introduction of a "fairness partition" on the action set of an automaton, and (5) the use of simulation-based techniques for proving that the set of action sequences that

*A preliminary version of this paper appeared in the *Proceedings of CONCUR'94 – Fifth International Conference on Concurrence Theory,* Lecture Notes in Computer Science, volume 836. The research of the first and second authors was supported in part by NSF Grants CCR-9120995 and CCR-9208585, and AFOSR Grant F49620-93-1-0250. The research of the third author was supported in part by NSF Grant CCR-8902215. E-mail addresses: suewu@cs.sunysb.edu, sas@cs.sunysb.edu, stark@cs.sunysb.edu

can be produced by one I/O automaton is a subset of the set of action sequences that can be produced by another. In this paper, we consider the problem of augmenting I/O automata with probability information, as a step toward the ultimate goal of obtaining a useful tool for specifying and reasoning about asynchronous probabilistic systems. As much as possible, we would like to preserve the characteristic features of the I/O automaton model, especially the asymmetric treatment of input and output and the associated pleasant notion of composition.

There are some interesting issues that arise when one attempts to add probability to I/O automata. These issues derive from the input/output dichotomy and also from the asynchronous notion of composition for such automata, in which for any given state of a composite automaton there can be a number of component automata "competing" with each other to control the execution of the next action. It is inadequate simply to introduce, for each state q, a single probability distribution μ on the set of all transitions from state q, because intuitively there is no good reason why the choice between input transitions (which are "externally controlled" by the environment of the automaton) and output or internal transitions (which are "locally controlled" by the automaton itself) should admit a meaningful probabilistic description independent of any particular environment. So, instead of one probability distribution for *all* transitions for state q, we introduce several probability distributions: one distribution over all the locally controlled transitions from state q, and separate distributions for each input action e. Our model is thus a kind of hybrid between the "reactive" and "generative" approaches described in [vGSST90].

The introduction of multiple probability distributions on transitions still does not solve all problems, however. Although within a single automaton we do not wish to ascribe probabilities to choices between externally controlled and locally controlled transitions, when automata are composed we do wish to have a natural probabilistic description of the outcome of the competition between component automata for control of the next action. To this end we introduce the concept of the *delay parameter* $\delta(q)$ associated with each state q. The idea is as follows: when a component automaton in a composite system arrives in state q, it draws a random delay time from an exponential distribution with parameter $\delta(q)$. This time describes the length of time the automaton will remain in state q before executing its next locally controlled action. The competition between several component automata vying for control of the next locally controlled action is won by the automaton having the least amount of delay time left. If we assume that the delay time distributions of component automata are independent, we can assign a definite probability to the event that any given component automaton will win the competition in any given system state. The "memoryless" property of the exponential distribution makes it irrelevant whether the component automata draw one delay time when they first enter their local state, or whether each component draws a new delay time after each global transition. This last feature makes it possible to give a simple definition of composition for probabilistic I/O automata.

Having obtained definitions for probabilistic I/O automata and their composition, it becomes interesting to consider their "external behaviors." The external behavior of an ordinary I/O automaton is the set of all sequences of external actions that can be produced in the various executions of the automaton. Lynch and Tuttle show that the mapping from I/O automata to external behaviors respects composition, in that the external behavior of a composite automaton is determined in a natural way by the external behaviors of the component automata. Since ordinary I/O automata have sets of action sequences as their external behaviors, one might expect probabilistic I/O automata to have probability distributions on action sequences as their external behaviors. Although this intuition can be validated to a certain extent, if one wishes the mapping from probabilistic I/O automata to external behaviors to respect composition, then the situation requires a bit more finesse than simply using probability distributions on action sequences as external behaviors. The reason is this: to compute the probability distribution on action sequences determined by a composite automaton, it is necessary to have information about the internal delays of each of the component automata as well as the probability distribution they each induce on action sequences.

Our notion of external behavior for probabilistic I/O automata is obtained as follows: Let A be a probabilistic I/O automaton having no input or internal actions, and satisfying certain finite branching conditions. Then the automaton A induces a probability distribution on the set of all its executions, and, given an action sequence $\alpha = e_0 e_1 \ldots e_{n-1}$, a conditional distribution on the subset X_α of all executions whose action sequences extend α. We may view the sequences $d_0 d_1 \ldots d_n$ of delay parameters associated with the first $(n+1)$ states in such an execution as the values of an $(n+1)$-dimensional random variable $\mathbf{D}$ defined on X_α. We define $\mathcal{E}_\alpha^A$ to be the mapping that takes each real-valued function $g : \mathcal{R}^{n+1} \to \mathcal{R}$ to its expectation, weighted by the probability of the set X_α. Since the formal summation formula that defines $\mathcal{E}_\alpha^A$ makes sense even when A is allowed to have input actions, we can use the same formula to associate a functional $\mathcal{E}_\alpha^A$ with an arbitrary probabilistic I/O automaton A. We show that, for probabilistic I/O automata with no internal actions, a compositional notion of behavior is obtained if one takes the external behavior of an automaton A to be the mapping $\mathcal{E}^A$ that assigns to each action sequence α of length n the associated functional $\mathcal{E}_\alpha^A$ on $\mathcal{R}^{n+1} \to \mathcal{R}$.

Besides showing that our notion of behavior is compositional, we are also able to show that it is "fully abstract," in the sense that any two automata having distinct behaviors can be distinguished by a certain kind of probabilistic test. The key idea in the proof is that the success probability of tests in a certain class gives us the expectations of certain rational functions of the delay parameters. Using the uniqueness of partial fraction expansions of rational functions [Lan90], we can recover full information about the functionals $\mathcal{E}_\alpha^A$ from these expectations.

Finally, we extend the definition of $\mathcal{E}_\alpha^A$ to a class of probabilistic I/O automata with internal actions and again show that it is compositional and fully abstract with respect to probabilistic testing.

Related Work

The recent research literature contains a plethora of proposals for probabilistic models. Each of these proposals addresses different issues, and introduces probability in a different way. In *reactive processes* [Rab63, LS92], for each state q and action e, a separate probability distribution is associated with the set of e-labeled transitions leaving state q. In contrast, in *generative processes* [vGSST90], for each state q a single probability distribution is associated with the set of all transitions leaving state q. The *stratified processes* [vGSST90] model refines the generative model with a multi-level probabilistic choice mechanism. *Alternating processes* [HJ90, Han91] are a mixture of strictly alternating probabilistic and nondeterministic states. The *stochastic processes* of [Mol82, GHR92, Hil93] associate a stochastic delay, represented as a random variable, with the firing of transitions. In *probabilistic specifications* [JL91] transitions are labeled by *sets* of probabilities, rather than single probabilities. Finally, the model of *probabilistic communicating processes* [Sei92] contains both an external (non-deterministic) and internal (probabilistic) choice operator, and processes are defined as conditional probability measures.

The main contribution of our work is a compositional semantics for asynchronous probabilistic systems, which is fully abstract with respect to probabilistic testing. To our knowledge, we are the first to give such a result. The closest earlier work is that of Christoff [Chr90], although his approach differs from our own on a number of key aspects: Christoff considers only purely generative processes, whereas our model captures processes that are both generative and reactive. Christoff's tests are deterministic and there is no notion of success state or success action; instead, he introduces testing equivalences based on the probabilities induced by the interaction of a process and a test on L^*, where L is a set of observable events. In our model, tests are just probabilistic I/O automata with a distinguished "success action" ω, and testing equivalence is defined in terms of the probability of a process successfully passing a test. Christoff's denotational models, which he shows to be fully abstract with respect to testing, are defined in terms of "probability functions" that map $(2^L - \emptyset)^* \times L^*$ to $[0, 1]$. No composition operator on processes is defined in [Chr90] and thus the issue of compositionality of his denotational semantics is left untreated. On the other hand, our model is compositional, and to obtain this result we find it necessary to include information about internal delays in the abstract representation of a process.

In [HJ90, Han91], Hansson and Jonsson present a process algebra, *Timed Probabilistic Calculus of Communicating Systems* (TPCCS), which extends Milner's CCS [Mil89] with probabilities and time. Discrete time is used in TPCCS, and a timeout operator is introduced. The probabilistic extension is achieved through a *probabilistic choice operator*, which defines a probability distribution over a set of reachable successor states. For each state in this model, either a probabilistic choice (determined by the process only) or a nondeterministic choice (which can be affected by the environment) is made. For technical reasons, a strict alternation between probabilistic and nondeterministic choices is required. The motivation of the separation of probabilistic and nondeterministic transitions is to avoid making assumptions about the scheduling of processes and about the relative probabilities of internal actions versus external communications when composing processes. In our model, a natural notion of composition is achieved by introducing state delay parameters and different probability functions for each input action and the set of locally controlled actions. Hansson and Jonsson also equip TPCCS with a notion of strong bisimulation equivalence, for which they give a sound and complete axiomatization, while in our model, we define a testing equivalence based on probabilistic testing.

In [LS92], Larsen and Skou introduce *probabilistic bisimulation*, using reactive probabilistic transition systems as the underlying semantic model. They present a notion of testing and show that if two processes are probabilistically bisimilar then no test can distinguish them. In addition to the difference in underlying models, their notion of test is very different from ours. They consider a test as an algorithm describing the procedure of running experiments on a process and present a simple but powerful language for tests. Another language is introduced for writing down the possible observations during the execution of a test on a process. Each process can be viewed as defining a probabilistic distribution over the observations for a given test. They also introduce a new modal logic, Probabilistic Modal Logic (PML), and show that two processes are indistinguishable under PML if they give the same probability distribution on the observation set of any test (i.e. they are probabilistically bisimilar.)

Cleaveland *et al.* [CSZ92] present a testing preorder for probabilistic processes based on the notion of a process passing a test with a certain probability. They also exhibit strong links to the traditional testing theory of DeNicola and Hennessy [dNH83, Hen88]. Their work is similar to our own in its concept of testing and the homogeneity among processes and tests. However, there are a number of key differences: Processes in [CSZ92] are purely generative and there is no composition operator defined on processes. Thus the notion of an "interaction system" for a process and a test has to be defined explicitly. In followup work, Yuen *et al.* [YCDS94] present fully abstract characterizations of the testing preorders proposed in [CSZ92]. Since they are working in a generative model, probabilities in interaction systems must be normalized, and this leads to a more complicated characterization of the testing preorders.

Segala and Lynch [SL94] define several probabilistic simulation relations for probabilistic systems in a "probabilistic automaton model". Instead of associating probability distributions with transitions as in our model, the transition relation of a probabilistic automaton is a set of pairs $(s, (\Omega, \mathcal{F}, P))$, where s is a state and $(\Omega, \mathcal{F}, P)$ is a discrete probability distribution over (action, state) pairs and a symbol δ, representing deadlock. The model distinguishes between probabilistic choices and nondeterministic choices, which in some sense is similar to our modeling of generative outputs and reactive inputs. The nondeterministic choices are resolved by "adversaries" that schedule the next step of a probabilistic automaton A based on the steps A has previously performed. The simulation relations are evaluated according to two criteria: compositionality and preservation of properties expressible in PCTL, an untimed version of Hansson's Timed Probabilistic concurrent Computation Tree Logic (TPCTL) [Han91].

The rest of this paper is organized as follows: In Section 2, we review some basic definitions and results pertaining to ordinary I/O automata and their composition. In Section 3, we define our probabilistic version of I/O automata, and show how the notion of composition for ordinary I/O automata extends to the probabilistic case. In Section 4, we define our notion of probabilistic behavior, and we show that the map taking each automaton to its behavior respects composition. In Section 5, we show that our notion of behavior is fully abstract with respect to probabilistic testing. In Section 6, we extend the notion of probabilistic behavior maps to a restricted class of probabilistic I/O automata with internal actions, and we show that this map also respects composition and full abstraction. Finally, in Section 7, we summarize what we have accomplished and outline plans for future investigation.

2 I/O Automata

In this section, we review some basic definitions and results pertaining to ordinary I/O automata. For further details, the reader is referred to [Tut87].

An *I/O automaton* is a quadruple $A = (Q, q^{\mathrm{I}}, E, \Delta)$, where

- Q is a set of *states*.
- $q^{\mathrm{I}} \in Q$ is a distinguished *start state*.
- E is a set of *actions*, partitioned into disjoint sets of *input*, *output*, and *internal* actions, which are denoted by E^{in}, E^{out}, and E^{int}, respectively. The set $E^{\mathrm{loc}} = E^{\mathrm{out}} \cup E^{\mathrm{int}}$ of output and internal actions is called the set of *locally controlled* actions, and the set $E^{\mathrm{ext}} = E^{\mathrm{in}} \cup E^{\mathrm{out}}$ is called the set of *external* actions.
- $\Delta \subseteq Q \times E \times Q$ is the *transition relation*, which satisfies the following *input-enabledness* property: for any state $q \in Q$ and input action $e \in E^{\mathrm{in}}$, there exists a state $r \in Q$ such that $(q, e, r) \in \Delta$.

It will sometimes be convenient for us to use the notation $q \xrightarrow{e} r$ to assert that $(q, e, r) \in \Delta$.

The original definition of I/O automaton [Tut87] allowed several start states instead of just one distinguished start state. It also included an additional piece of data: a partition of the set of locally controlled actions. Such partitions are used to define a notion of *fair execution* for I/O automata, which is essential if one wishes to establish liveness properties for such automata. We do not treat liveness properties in this paper. However, we expect that in many cases, liveness can be treated in a probabilistic setting by dispensing with fairness and instead using probability information to define a notion of "satisfies a liveness property with probability one." We shall therefore ignore the partition component of I/O automata in our discussion.

Lynch and Tuttle define a *finite execution fragment* of an I/O automaton A to be an alternating sequence of states and actions of the form

$$q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \dots \xrightarrow{e_{n-1}} q_n,$$

such that $(q_k, e_k, q_{k+1}) \in \Delta$ for $0 \leq k < n$. In this paper, we find it convenient to use a slightly more liberal definition of execution fragment, to allow such fragments to contain actions not in E. To accommodate this, we assume the existence of a fixed, countable *universe of actions* U, and we require that the set E of actions of an I/O automaton be a subset of U. This is not really much of a restriction in practice, since in practical situations we have to be able to explicitly denote all actions by a finite sequence of symbols. We also need to be able to distinguish external actions from internal actions, and so we also assume that U is partitioned into disjoint sets U^{ext} and U^{int}, and we require that the partitioning of the set E of actions of an I/O automaton satisfy the condition $E^{\text{ext}} \subseteq U^{\text{ext}}$ and $E^{\text{int}} \subseteq U^{\text{int}}$.

For us, then, a *finite execution fragment* is an alternating sequence of states and actions as above such that $(q_k, e_k, q_{k+1}) \in \Delta$ whenever $e_k \in E$ and such that $q_{k+1} = q_k$ whenever $e_k \in U \setminus E$. An execution fragment with $q_0 = q^{\text{I}}$ (the distinguished start state) is called an *execution*. We use the term *native* to refer to an execution or execution fragment of A in which only actions from E appear.

If σ denotes an execution fragment as above, then we will use $\sigma(k)$ to denote the state q_k, for $0 \leq k \leq n$. We use the term *trace* to refer to a sequence of actions. If σ is an execution fragment as above, then the *trace* of σ, denoted $\text{tr}(\sigma)$, is the sequence of actions $e_0 e_1 \ldots e_{n-1}$ appearing in σ.

2.1 Composition

A collection $\{A_i : i \in I\}$ of I/O automata, where $A_i = (Q_i, q_i^{\text{I}}, E_i, \Delta_i)$, is called *compatible* if for all $i, j \in I$, $i \neq j$, we have $E_i^{\text{out}} \cap E_j^{\text{out}} = \emptyset$ and $E_i^{\text{int}} \cap E_j = \emptyset$. We define the *composition* $\prod_{i \in I} A_i$ of such a collection to be the I/O automaton $(Q, q^{\text{I}}, E, \Delta)$, defined as follows:

- $Q = \prod_{i \in I} Q_i$.
- $q^{\text{I}} = \langle q_i^{\text{I}} : i \in I \rangle$.
- $E = \bigcup_{i \in I} E_i$, where

$$E^{\text{out}} = \bigcup_{i \in I} E_i^{\text{out}} \qquad E^{\text{int}} = \bigcup_{i \in I} E_i^{\text{int}} \qquad E^{\text{in}} = (\bigcup_{i \in I} E_i^{\text{in}}) \setminus E^{\text{out}}.$$

- Δ is the set of all $(\langle q_i : i \in I \rangle, e, \langle r_i : i \in I \rangle)$ such that for all $i \in I$, if $e \in E_i$, then $(q_i, e, r_i) \in \Delta_i$, otherwise $r_i = q_i$.

With our more liberal definition of execution fragments, we have a simple correspondence between computations of a composite I/O automaton and the computations of its component automata. Suppose σ is an execution fragment for a composite automaton $\prod_{i \in I} A_i$, of the form

$$q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \ldots \xrightarrow{e_{n-1}} q_n,$$

where $q_k = \langle q_{k,i} : i \in I \rangle$. Then for each $i \in I$, the execution fragment σ projects in an obvious way to an execution fragment $\sigma | A_i$ of A_i by replacing each state q_k by its projection $q_{k,i}$. Suppose $\mathrm{tr}(\sigma) = \alpha$, then $\mathrm{tr}(\sigma|A_i) = \alpha$ for each $i \in I$. This mapping, taking each execution fragment σ of $\prod_{i\in I} A_i$ to indexed collections of execution fragments $\langle \sigma|A_i : i \in I\rangle$, is invertible, in the sense made precise by the following proposition.

Proposition 2.1 *Suppose $A = \prod_{i\in I} A_i$. Then for each action sequence α of the form $e_0e_1 \ldots e_{n-1}$, the map, that takes each execution fragment σ of $\prod_{i\in I} A_i$ with $\mathrm{tr}(\sigma) = \alpha$ to the collection of execution fragments $\{\sigma|A_i : i \in I\}$, is a bijection, from the set of execution fragments ρ of A having trace α to the set of indexed collections $\{\rho_i : i \in I\}$, where each ρ_i is an execution fragment of A_i with trace α.*

Proof – Suppose $\sigma = q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \ldots \xrightarrow{e_{n-1}} q_n$ is an execution fragment of A. Then $\sigma|A_i = q_{0,i} \xrightarrow{e_0} q_{1,i} \xrightarrow{e_1} \ldots \xrightarrow{e_{n-1}} q_{n,i}$ is an execution fragment of A_i since by the definition of composition, $(q_{k,i}, e_k, q_{k+1,i}) \in \Delta_i$ if $e_k \in E_i$ and $q_{k+1,i} = q_{k,i}$ if $e_k \notin E_i$.

Conversely, suppose $\sigma_i = q_{0,i} \xrightarrow{e_0} q_{1,i} \xrightarrow{e_1} \ldots \xrightarrow{e_{n-1}} q_{n,i}$ is an execution fragment of A_i, for all $i \in I$. Then $\sigma = q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \ldots \xrightarrow{e_{n-1}} q_n$ is an execution fragment of A, since by the definition of composition, $(q_k, e_k, q_{k+1}) \in \Delta$ if $e_k \in E_i$ for some $i \in I$, and

$$q_{k+1} = \langle q_{k+1,i} : i \in I\rangle = \langle q_{k,i} : i \in I\rangle = q_k$$

if $e_k \notin E_i$ for all $i \in I$; i.e. $e_k \notin E$. ∎

Besides composition, Lynch and Tuttle also defined "action hiding" on I/O automata [Tut87]. We do not treat action hiding in this paper.

3 Probabilistic I/O Automata

A *probabilistic I/O automaton* is a sextuple $A = (Q, q^{\mathrm{I}}, E, \Delta, \mu, \delta)$, where

- $(Q, q^{\mathrm{I}}, E, \Delta)$ is an I/O automaton, called the *underlying* I/O automaton. The transition relation Δ is required to satisfy the following properties:

  1. The *local finite-branching* property: for all $q \in Q$, the set

  $$\{(q, e, r) \in \Delta : e \in E^{\mathrm{loc}}\}$$

  is finite.

  2. The *input image-finiteness* property: for all $q \in Q$ and all $e \in E^{\mathrm{in}}$, the set

  $$\{r \in Q : (q, e, r) \in \Delta\}$$

  is finite.

- $\mu : (Q \times E \times Q) \to [0, 1]$ is the *transition probability* function, which is required to satisfy the following conditions:

1. $\mu(q,e,r) > 0$ iff $(q,e,r) \in \Delta$.
2. $\sum_{r \in Q} \mu(q,e,r) = 1$, for all $q \in Q$ and all $e \in E^{\text{in}}$.
3. For all $q \in Q$, if there exist $e \in E^{\text{loc}}$ and $r \in Q$ such that $(q,e,r) \in \Delta$, then $\sum_{r \in Q} \sum_{e \in E^{\text{loc}}} \mu(q,e,r) = 1$,

- $\delta : Q \to [0,\infty)$ is the *state delay* function, which is required to satisfy the following condition: for all $q \in Q$, we have $\delta(q) > 0$ if and only if there exist $e \in E^{\text{loc}}$ and $r \in Q$ such that $(q,e,r) \in \Delta$.

The local finite-branching condition on the transition relation Δ is imposed so that in Section 3.1 we can obtain a probability distribution on the set of all native executions of an automaton A with an empty set of input actions. This condition is also needed so that we can obtain discrete probability distributions in key situations in Section 4; thereby avoiding technical problems of measurability that would arise in a more general setting. Once we have imposed the local finite-branching condition, the input image-finiteness condition is required in order for the class of probabilistic I/O automata to be closed under the composition operation defined in Section 3.2. Though it might be possible to relax the finiteness to countability (or beyond) in these conditions, this is a purely measure-theoretic problem that would present no new computational issues, and so we do not attempt to solve it in this paper.

The transition probability function μ describes the probability, for each state q, of choosing one transition from state q as opposed to another. As discussed in the introduction, we do not ascribe any probability to the choice between an input transition and an output or internal transition, since any such probability will be determined by the environment. Similarly, the choice between a transition for one input action and a transition for a different input action is also under the control of the environment, so we do not attempt to assign probabilities in this case either. The stochastic conditions (2) and (3) on μ reflect this point of view: Condition (2) states that for each state q and input action e, the function μ determines a probability distribution on the set of states r such that $q \xrightarrow{e} r$. Condition (3) states that if there is some locally controlled action enabled in state q, then μ determines a probability distribution on the set of all pairs (e,r) such that e is locally controlled and $q \xrightarrow{e} r$.

The state delay function δ assigns to each state q a nonnegative real number $\delta(q)$. As discussed in the introduction, the intuitive interpretation of $\delta(q)$ is as the parameter of an exponential distribution describing the length of a random "delay period" from the time state q is entered by the automaton until the time it executes its next locally controlled action. The condition on δ corresponds to the intuition that if no locally controlled action is available in state q, then the delay period will be infinite.

Function δ can be extended to finite execution fragments as follows. Let σ be a finite execution fragment of the form

$$q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \dots \xrightarrow{e_{n-1}} q_n,$$

and d_i denote $\delta(q_i)$, the delay parameter associated with state q_i, for $0 \leq i \leq n$. Then let $\delta(\sigma)$ denote the sequence $d_0 d_1 \ldots d_n$. We call $\delta(\sigma)$ the *delay sequence* of σ. We use $\mathbf{d}$, $\mathbf{d}'$, possibly subscripted or superscripted, to denote delay sequences. Let $\mathbf{d} = d_0 d_1 \ldots d_n$ and $\mathbf{d}' = d'_0 d'_1 \ldots d'_n$ be two arbitrary delay sequences. Then $\mathbf{d} + \mathbf{d}'$ is the componentwise sum of $\mathbf{d}$ and $\mathbf{d}'$. This notation is used extensively in Sections 4, 5 and 6.

To simplify notation in the sequel, it will be convenient for us to adopt the following convention regarding the application of μ to triples (q, e, r) where e is not in the set E of actions of A: if $A = (Q, q^{\mathrm{I}}, E, \Delta, \mu, \delta)$, and $e \in U \setminus E$, then we define $\mu(q, e, q) = 1$ and $\mu(q, e, r) = 0$ for all other $r \in Q$.

3.1 Probability Distributions on Executions and Traces

Suppose $A = (Q, q^{\mathrm{I}}, E, \Delta, \mu, \delta)$ is a probabilistic I/O automaton. In this section, we consider the problem of assigning probabilities to sets of executions of A. If A is an arbitrary probabilistic I/O automaton, it does not make much sense to ask about the probability of sets of executions of A, since we lack any sort of probabilistic description of when input actions will occur and which ones they will be. There are two approaches that can be taken to this problem. The first approach, which we adopt, is to define the probability of sets of executions of automata A only in case $E^{\mathrm{in}} = \emptyset$; that is, A is a *closed* automaton, and only for sets of *native* executions of A, since actions outside E are under the control of the environment. In our approach, if we wish to make probabilistic statements about automata A that are not closed, we first specify a "test" T to be performed on A. For us, a test is a probabilistic I/O automaton that is "complementary" to A, in the sense that the composite $A|T$ makes sense and is a closed automaton. We then make probabilistic statements about the behavior of $A|T$. In a sense, in this approach there are no non-probabilistic choices, only choices for which probability information has not yet been specified. The second approach that can be taken is to accept the possibility of inherently non-probabilistic choices. This leads to the introduction of the notion of an external "adversary" that resolves all such choices. In this approach, a probabilistic automaton does not determine a single probability space, but rather a separate probability space for each possible adversary. Probabilistic statements are made about such automata by universally quantifying over adversaries. The adversary approach has been studied by Segala and Lynch [SL94].

In any discussion of probability, it is necessary to begin by describing the probability space. In our case, the set of basic outcomes is the set of all native executions of A (both finite and infinite). If

$$\sigma = q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \ldots \xrightarrow{e_{n-1}} q_n$$

is a finite native execution, then define the set $[\sigma]$ to be the set of all finite and infinite native executions ρ such that $\rho = \sigma$ or ρ is of the form

$$\rho = q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \ldots \xrightarrow{e_{n-1}} q_n \xrightarrow{e_n} q_{n+1} \cdots$$

In other words, for any finite native execution σ, the set $[\sigma]$ is the set of all finite and infinite native executions ρ that extend σ. Define a set of native executions of A to be *basic measurable* if it is the union (possibly empty) of a finite disjoint collection of sets of the form $[\sigma]$ and singleton sets of the form $\{\rho\}$, where ρ is a finite native execution with $[\rho] \neq \{\rho\}$. A basic measurable set of the form $\{\rho\}$ or of the form $[\sigma]$ is called *simple.*

Lemma 3.1 *The collection of basic measurable sets of executions of A is nonempty, and is closed under pairwise union, pairwise intersection, and complement. That is, it forms an algebra of sets.*

Proof – Let q^{I} denote the unique execution with no actions, then $[q^{\mathrm{I}}]$ is clearly basic measurable. Next, observe that the intersection of two sets $[\sigma]$ and $[\rho]$ is either $\emptyset$, $[\sigma]$, or $[\rho]$. From this, it is easy to see that the union of two basic measurable sets is basic measurable, and the intersection of two basic measurable sets is basic measurable. To show closure under complement, first observe that it follows from the local finite-branching property in the definition of probabilistic I/O automata that the complement of a set of the form $[\sigma]$ is the union of a finite disjoint collection of sets, consisting of the sets $\{\rho\}$, where ρ is a proper prefix of σ, and sets of the form $[\rho]$, where ρ is not a prefix of σ. The same characterization holds true for the complement of a set $\{\sigma\}$, where σ is finite. This observation, together with closure under pairwise intersection and DeMorgan's laws, implies that the complement of a basic measurable set is basic measurable. ∎

We now show how to assign probability to basic measurable sets of executions. Suppose

$$\sigma = q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \dots \xrightarrow{e_{n-1}} q_n$$

is a finite native execution. To a simple basic measurable set of the form $[\sigma]$ we assign probability as follows:

$$\Pr([\sigma]) = \prod_{k=0}^{n-1} \mu(q_k, e_k, q_{k+1}).$$

In particular $\Pr([q^{\mathrm{I}}]) = 1$. To a simple basic measurable set of the form $\{\rho\}$, where $[\rho] \neq \{\rho\}$ we assign probability 0. This corresponds to the idea that stopping in a state for which further transitions are enabled, occurs with probability zero. The definitions and results of this paper all generalize easily to the case of nonzero stopping probability.

To each basic measurable set $B = \bigcup_{i=1}^{n} B_i$ represented as a finite union of disjoint sets B_i of the form $[\sigma]$ or of the form $\{\rho\}$, where $[\rho] \neq \{\rho\}$, we assign probability as follows:

$$\Pr(B) = \sum_{i=1}^{n} \Pr(B_i).$$

We must show that this assignment of probability to the set B is independent of the choice of representation of B as $\bigcup_{i=1}^{n} B_i$. This follows from Lemma 3.3 below.

Lemma 3.2 *Suppose S is a simple basic measurable set and $S = \bigcup_{i=1}^{m} S_i$, where S_i are disjoint simple basic measurable sets. Then*

$$\Pr(S) = \sum_{i=1}^{m} \Pr(S_i).$$

Proof – Suppose S is of the form $\{\rho\}$, where $[\rho] \neq \{\rho\}$, or of the form $[\sigma]$, where $[\sigma] = \{\sigma\}$. Then there is precisely one way to represent S as a disjoint union of simple basic measurable sets, so the result holds in this case.

In case S is of the form $[\sigma]$, where $[\sigma] \neq \{\sigma\}$, the proof proceeds by induction on the *rank* k of a representation $S = \bigcup_{i=1}^{m} S_i$, which we define to be the number of sets S_i that are of the form $\{\rho\}$ with $[\rho] \neq \{\rho\}$.

For the basis case, suppose $k = 0$. Then each set S_i is of the form $[\sigma_i]$. In view of the assumption that $S = \bigcup_{i=1}^{m} S_i$ is a disjoint union, it must be the case that $m = 1$ and $\sigma_1 = \sigma$. Thus, $\Pr(S) = \sum_{i=1}^{1} \Pr(S_i)$ as required.

Now assume that the result has been shown for some $k \geq 0$, and suppose $S = \bigcup_{i=1}^{m} S_i$ is a representation of rank $k+1$. For $1 \leq i \leq m$, let σ_i be chosen such that either $S_i = [\sigma_i]$ or $S_i = \{\sigma_i\}$. Since $k+1 > 0$, the subset of $\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$, consisting of all those σ_i for which $S_i = \{\sigma_i\}$ with $[\sigma_i] \neq \{\sigma_i\}$, is nonempty. Let ρ be an execution of maximal length chosen arbitrarily from this subset. Suppose ρ is of length n, so that

$$\rho = q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \cdots \xrightarrow{e_{n-1}} q_n.$$

Let $C = \{\rho_1, \rho_2, \cdots, \rho_l\}$ be the set of all native executions of length $n+1$ that properly extend ρ. Then $[\rho] = (\bigcup_{j=1}^{l} [\rho_j]) \cup \{\rho\}$ by definition of $[\rho]$. Now, $\rho \in S = [\sigma]$, hence $\rho_j \in [\sigma]$ for $1 \leq j \leq l$. Since $S = \bigcup_{i=1}^{m} S_i$ is a disjoint union, it follows that for each j with $1 \leq j \leq l$ there exists a unique π_j with $1 \leq \pi_j \leq m$ and $\rho_j \in S_{\pi_j}$. Since ρ_j is a minimal proper extension of ρ, we must have $\sigma_{\pi_j} = \rho_j$, so that each S_{π_j} is either of the form $[\rho_j]$ or else of the form $\{\rho_j\}$ with $[\rho_j] \neq \{\rho_j\}$. Only the former case is possible, since if $S_{\pi_j} = \{\rho_j\}$ with $[\rho_j] \neq \{\rho_j\}$ we would have a contradiction with the assumption that ρ is of maximal length.

Thus, we may write

$$[\rho] = (\bigcup_{j=1}^{l} [\rho_j]) \cup \{\rho\} = (\bigcup_{j=1}^{l} S_{\pi_j}) \cup \{\rho\}$$

and

$$S = \bigcup_{i=1}^{m} S_i = (\bigcup_{\substack{1 \leq i \leq m \\ \sigma_i \notin [\rho]}} S_i) \cup [\rho].$$

Now, the last expression gives a representation of S of rank k, so by the induction hypothesis we have

$$\Pr(S) = \sum_{i=1}^{m} \Pr(S_i) - \sum_{j=1}^{l} \Pr(S_{\pi_j}) + \Pr([\rho]) \tag{1}$$

Since $[\rho] \neq \{\rho\}$, we have $\Pr(\{\rho\}) = 0$ by definition, hence

$$\Pr([\rho]) \quad = \quad \prod_{k=0}^{n-1} \mu(q_k, e_k, q_{k+1})$$

$$
\begin{aligned}
&= \left(\prod_{k=0}^{n-1} \mu(q_k, e_k, q_{k+1})\right) \cdot \left(\sum_{r \in Q} \sum_{e \in E^{\mathrm{loc}}} \mu(q_n, e, r)\right) \\
&= \sum_{i=1}^{l} \prod_{k=0}^{n} \mu(\rho_i(k), e_k, \rho_i(k+1)) \\
&= \sum_{i=1}^{l} \Pr([\rho_i]),
\end{aligned}
$$

where the second equality holds because $\sum_{r \in Q} \sum_{e \in E^{\mathrm{loc}}} \mu(q_n, e, r) = 1$ by the stochastic condition (3) in the definition of probabilistic I/O automata. Thus,

$$
\begin{aligned}
\Pr([\rho]) &= \sum_{i=1}^{l} \Pr([\rho_i]) + \Pr(\{\rho\}) \\
&= \sum_{j=1}^{l} \Pr(S_{\pi_j}) \qquad (2)
\end{aligned}
$$

From equations (1) and (2), we conclude that

$$
\Pr(S) = \sum_{i=1}^{m} \Pr(S_i).
$$

∎

Lemma 3.3 *Suppose $B = \bigcup_{i=1}^{m} S_i$ and $B = \bigcup_{j=1}^{n} S'_j$ are two representations of a basic measurable set B as a finite disjoint union of simple basic measurable sets. Then*

$$
\sum_{i=1}^{m} \Pr(S_i) = \sum_{j=1}^{n} \Pr(S'_j)
$$

Proof –

$$
\begin{aligned}
\sum_{i=1}^{m} \Pr(S_i) &= \sum_{i=1}^{m} \Pr(S_i \cap B) && (B = \textstyle\bigcup_{i=1}^{m} S_i) \\
&= \sum_{i=1}^{m} \Pr(\bigcup_{j=1}^{n} (S_i \cap S'_j)) && (B = \textstyle\bigcup_{j=1}^{n} S'_j) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} \Pr(S_i \cap S'_j) && (\text{Lemma 3.2}) \\
&= \sum_{j=1}^{n} \sum_{i=1}^{m} \Pr(S'_j \cap S_i) && (\text{Exchange order of summation}) \\
&= \sum_{j=1}^{n} \Pr(S'_j)
\end{aligned}
$$

∎

Lemma 3.4 Pr *is a measure (a countably additive set function) on the algebra of basic measurable sets.*

Proof – It is obvious from the definitions that Pr is finitely additive. We must show that Pr is countably additive: for any countable disjoint sequence $B_0, B_1, \ldots$ of basic measurable sets whose union B is basic measurable, we have

$$\Pr(B) = \sum_{i=0}^{\infty} \Pr(B_i).$$

But if $B = \bigcup_{i=0}^{\infty} B_i$, then B is in fact a countable union of simple basic measurable sets. A König's Lemma argument shows that if a basic measurable set B is a countable union of simple basic measurable sets S_j, then at most finitely many of the sets S_j can be nonempty. Hence for the class of basic measurable sets, countable additivity reduces to finite additivity. ∎

Proposition 3.5 Pr *extends to a complete measure (which we also denote by* Pr*) on a* σ*-algebra containing the algebra of basic measurable sets. Moreover, since* $\Pr([q^{\mathrm{I}}]) = 1$*, it follows that* Pr *is a probability measure.*

Proof – This is a corollary of the Extension Theorem for measures [Wei74, Theorem 2, p. 97] which states that any measure defined on a ring $\mathcal{R}$ extends to a complete measure on a σ-algebra containing $\mathcal{R}$. ∎

We can also assign probabilities to sets of traces. Still working with respect to a probabilistic I/O automaton A, we define a set V of traces of A to be *measurable* if $\mathrm{tr}_A^{-1}(V)$ is a measurable set of executions of A. To each such set we assign probability as follows:

$$\Pr(V) = \Pr(\mathrm{tr}^{-1}(V)).$$

It is easy to check that these definitions determine a probability space on the set of traces.

3.2 Composition

A collection $\{A_i : i \in I\}$ of probabilistic I/O automata, where

$$A_i = (Q_i, q_i^{\mathrm{I}}, E_i, \Delta_i, \mu_i, \delta_i),$$

is called *compatible* if the corresponding collection of underlying I/O automata is compatible. The *composition* $\prod_{i \in I} A_i$ of a *finite* compatible collection of probabilistic I/O automata is defined to be the sextuple $(Q, q^{\mathrm{I}}, E, \Delta, \mu, \delta)$, where

1. Q, q^{I}, E, and Δ are defined as for composition of ordinary I/O automata.
2. $\delta(\langle q_i : i \in I \rangle) = \sum_{i \in I} \delta_i(q_i)$.

3. If $e \in E^{\mathrm{in}}$, then

$$\mu(\langle q_i : i \in I\rangle, e, \langle r_i : i \in I\rangle) = \prod_{\{i \in I : e \in E_i\}} \mu_i(q_i, e, r_i).$$

If $e \in E_k^{\mathrm{loc}}$ for some k, then

$$\mu(\langle q_i : i \in I\rangle, e, \langle r_i : i \in I\rangle) = \frac{\delta_k(q_k)}{\sum_{i \in I} \delta_i(q_i)} \prod_{\{i \in I : e \in E_i\}} \mu_i(q_i, e, r_i).$$

In this paper, we restrict our attention to the composition of finite collections only. The finiteness assumption ensures that the sum in (2) converges, and that the products appearing in the definition of μ are nonzero.

The definition of composition can be motivated as follows: In any given state $q = \langle q_i : i \in I\rangle$ of a composite automaton $A = \prod_{i \in I} A_i$, the component automata A_i participate in a race to see which one will be the next to execute a locally controlled action. Conceptually, when each component automaton A_i enters its local state q_i, it chooses a random "delay period" from an exponential distribution with parameter $\delta_i(q_i)$. It then delays for this amount of time before executing its next locally controlled action. The winner of the race from state q will be that component automaton A_i with the least amount of time to wait. Because of the "memoryless" property of the exponential distribution, it is not necessary for us to keep track in the composite automaton of how long each component automaton A_i has already delayed in state q_i—the amount of time A_i has left to delay in state q_i is described by the same exponential distribution with parameter $\delta_i(q_i)$, regardless of how long A_i has already delayed. This fact simplifies the definition of composition considerably, and would also be important if we wished to construct a real-world implementation of the probabilistic behavior modeled by these automata.

Assuming that the random delay periods associated with the component automata A_i are independent, the probability that the winner of the race from state q will be a particular component A_k is the probability that the random delay period chosen by A_k is the minimum among all the delay periods chosen by the A_i. This probability is the ratio $\delta_k(q_k)/\sum_{i \in I} \delta_i(q_i)$.

The distribution of the time that composite automaton A delays in state q before executing its next locally controlled action is the distribution of the minimum of the delay times of each of the components. Here the situation is simplified by another property of the exponential distribution: the distribution of the minimum of a finite collection $\langle x_i : i \in I\rangle$ of independent random variables, where x_i is exponentially distributed with parameter $\delta_i(q_i)$, is again exponentially distributed with parameter $\sum_{i \in I} \delta_i(q_i)$ [Tri82]. This explains the definition of δ.

The definition of μ can now be explained as follows: If it has already been determined that the next action to be executed is a particular input action e, then the probability of choosing a particular transition (q, e, r), where $q = \langle q_i : i \in I\rangle$ and $r = \langle r_i : i \in I\rangle$ is simply the joint probability that component A_i executes (q_i, e, r_i), for all $i \in I$ such that $e \in E_i$. Assuming independence, this joint probability is just the product of the individual probabilities $\mu_i(q_i, e, r_i)$. On the other hand, if it has been determined that the next action to be executed is not an input action, but rather a locally controlled action, then which locally controlled action is actually executed depends on the outcome of the race for control between the component automata. The probability that the transition executed will be (q, e, r), where $q = \langle q_i : i \in I \rangle$ and $r = \langle r_i : i \in I \rangle$, and $e \in E_k^{\mathrm{loc}}$ is locally controlled by A_k, is the joint probability that each A_i will execute transition (q_i, e, r_i), times the probability that A_k will win the race. Assuming independence, the former is just the product of the individual probabilities $\mu_i(q_i, e, r_i)$. As already discussed, the latter probability is the ratio $\delta_k(q_k)/\sum_{i \in I} \delta_i(q_i)$.

Proposition 3.6 *If $\{A_i : i \in I\}$ is a finite compatible collection of probabilistic I/O automata, then $\prod_{i \in I} A_i$ is also a probabilistic I/O automaton.*

Proof – Suppose $A_i = (Q_i, q_i^{\mathrm{I}}, E_i, \Delta_i, \mu_i, \delta_i)$ and $A = \prod_{i \in I} A_i$.

We first verify the local finite-branching property and the input image-finiteness property. Since I is finite, and by input image-finiteness of the A_i, the sets $\{r_i : (q_i, e, r_i) \in \Delta_i\}$ are finite for all $q_i \in Q_i$ and $e \in E_i^{\mathrm{in}}$, it follows that the set $\{r : (q, e, r) \in \Delta\}$ is finite for all $q \in Q$ and $e \in E^{\mathrm{in}}$, so that A is input image-finite. By definition of composition, $e \in E^{\mathrm{loc}}$ exactly when there is a unique k such that $e \in E_k^{\mathrm{loc}}$ and such that whenever $i \neq k$ if $e \in E_i$ then $e \in E_i^{\mathrm{in}}$. By the local finite-branching property of A_k, the set $\{(q_k, e, r_k) \in \Delta_k : e \in E_k^{\mathrm{loc}}\}$ is finite, and by the input image-finiteness property of the A_i, the sets $\{(q_i, e, r_i) \in \Delta_i : e \in E_i^{\mathrm{in}}\}$ are finite for all $q_i \in Q_i$. Thus, the set $\{(q, e, r) \in \Delta : e \in E^{\mathrm{loc}}\}$ is also finite, so that A has local finite-branching.

We next verify that μ and δ have the required properties. First, consider δ. Clearly, since I is finite and $\delta_i(q_i) \in [0, \infty)$ for all $i \in I$ and $q_i \in Q_i$, it follows that $\delta(\langle q_i : i \in I \rangle) = \sum_{i \in I} \delta_i(q_i) \in [0, \infty)$ as well. If $\delta(q) = 0$, then for all $i \in I$ there exist no $e \in E_i^{\mathrm{loc}}$ and $r_i \in Q_i$ such that $(q_i, e, r_i) \in \Delta_i$, hence there exist no $e \in E^{\mathrm{loc}}$ and $r \in Q$ such that $(q, e, r) \in \Delta$. Conversely, if there exist no $e \in E^{\mathrm{loc}}$ and $r \in Q$ such that $(q, e, r) \in \Delta$, then there can be no $i \in I$, $e \in E_i^{\mathrm{loc}}$, and $r_i \in Q_i$ such that $(q_i, e, r_i) \in \Delta_i$, thus $\delta_i(q_i) = 0$ for all $i \in I$ and hence $\delta(q) = 0$.

Next, consider μ. We first show that $\mu(q, e, r) \in (0, 1]$ whenever $(q, e, r) \in \Delta$. If $e \in E^{\mathrm{in}}$, then

$$\mu(\langle q_i : i \in I \rangle, e, \langle r_i : i \in I \rangle) \quad = \prod_{\{i \in I : e \in E_i\}} \mu_i(q_i, e, r_i),$$

which is in $(0, 1]$ because I is finite and because $\mu_i(q_i, e, r_i) \in (0, 1]$ whenever $e \in E_i$. If $e \in E^{\mathrm{loc}}$, then $e \in E_k^{\mathrm{loc}}$ for some $k \in I$, and

$$\mu(\langle q_i : i \in I \rangle, e, \langle r_i : i \in I \rangle) = \frac{\delta_k(q_k)}{\sum_{i \in I} \delta_i(q_i)} \prod_{\{i \in I : e \in E_i\}} \mu_i(q_i, e, r_i).$$

Since $(q_k, e, r_k) \in \Delta_k$, we have $\delta_k(q_k) > 0$. Since I is finite, and since $\mu_i(q_i, e, r_i) > 0$ for each i such that $e \in E_i$, it follows that $\mu(q, e, r) > 0$.

Next, we prove that μ satisfies stochastic conditions (2) and (3) in the definition of probabilistic I/O automata. Suppose $q = \langle q_i : i \in I \rangle \in Q$ and $e \in E^{\mathrm{in}}$. Then

$$\begin{aligned} \sum_{r \in Q} \mu(q,e,r) &= \sum_{r \in Q} \prod_{\{i \in I : e \in E_i\}} \mu_i(q_i,e,r_i) \\ &= \prod_{\{i \in I : e \in E_i\}} \sum_{r_i \in Q_i} \mu_i(q_i,e,r_i) \\ &= 1, \end{aligned}$$

where the interchange of the sum and product is justified because Q is the Cartesian product of the Q_i. Finally, suppose $q = \langle q_i : i \in I \rangle \in Q$, and consider

$$\sum_{r \in Q} \sum_{e \in E^{\mathrm{loc}}} \mu(q,e,r) = \sum_{r \in Q} \sum_{k \in I} \sum_{e \in E_k^{\mathrm{loc}}} \frac{\delta_k(q_k)}{\delta(q)} \prod_{\{i \in I : e \in E_i\}} \mu_i(q_i,e,r_i).$$

If there is no $e \in E^{\mathrm{loc}}$ and $r \in Q$ such that $(q,e,r) \in \Delta$, then there can be no $k \in I$, $e \in E_k^{\mathrm{loc}}$, and $r_k \in Q_k$ such that $(q_k,e,r_k) \in \Delta_k$, hence $\delta_k(q_k) = 0$ for all k, and the above sum is zero. Otherwise, if $e \in E^{\mathrm{loc}}$ and $r \in Q$ are such that $(q,e,r) \in \Delta$, then $e \in E_k^{\mathrm{loc}}$ for some $k \in I$, and we have

$$\begin{aligned} \sum_{r \in Q} \sum_{k \in I} \sum_{e \in E_k^{\mathrm{loc}}} \frac{\delta_k(q_k)}{\delta(q)} \prod_{\{i \in I : e \in E_i\}} \mu_i(q_i,e,r_i) &= \sum_{k \in I} \sum_{e \in E_k^{\mathrm{loc}}} \frac{\delta_k(q_k)}{\delta(q)} \sum_{r \in Q} \prod_{\{i \in I : e \in E_i\}} \mu_i(q_i,e,r_i) \\ &= \sum_{k \in I} \frac{\delta_k(q_k)}{\delta(q)} \sum_{e \in E_k^{\mathrm{loc}}} \prod_{\{i \in I : e \in E_i\}} \sum_{r_i \in Q_i} \mu_i(q_i,e,r_i) \\ &= \sum_{k \in I} \frac{\delta_k(q_k)}{\delta(q)} \sum_{r_k \in Q_k} \sum_{e \in E_k^{\mathrm{loc}}} \mu_k(q_k,e,r_k) \\ &= 1, \end{aligned}$$

where the penultimate equality holds because $\sum_{r_i \in Q_i} \mu_i(q_i,e,r_i) = 1$ whenever $e \in E_i^{\mathrm{in}}$; that is, whenever $i \neq k$. ∎

4 Behaviors of Probabilistic I/O Automata

In this section and the next section, we consider the restricted class of probabilistic I/O automata $A = (Q, q^{\mathrm{I}}, E, \Delta, \mu, \delta)$ for which the set E^{int} of internal actions is empty. We wish to associate with such an automaton a more abstract representation in which we ignore the details of the particular state set and transition relation of the automaton, and focus instead on externally observable aspects of its probabilistic behavior.

Suppose A is a probabilistic I/O automaton without internal actions. Given a trace $\alpha = e_0 e_1 \ldots e_{n-1}$, for each delay sequence $\mathbf{d} = d_0 d_1 \ldots d_n$ define the quantity $p_\alpha^A(\mathbf{d})$ by:

$$p_\alpha^A(\mathbf{d}) = \sum_{\sigma} \prod_{k=0}^{n-1} \mu_A(\sigma(k), e_k, \sigma(k+1)),$$

where the summation is taken over all executions σ of A having trace α and delay sequence $\mathbf{d}$. Observe that convergence of the summation is automatic, since by the local finite-branching and input image-finiteness properties of A, the set $\{\sigma : tr(\sigma) = \alpha\}$ is finite. The same reasoning also shows that, for a fixed α, the set of all $\mathbf{d}$ for which $p^A_\alpha(\mathbf{d})$ is nonzero, is finite. In case the set of input actions of A is empty, and α contains only actions in E_A, the quantity $p^A_\alpha(\mathbf{d})$ is the probability of the set of all native executions of A having α as a prefix of their trace and $\mathbf{d}$ as a prefix of their delay sequence.

Now, if

$$g : \mathcal{R}^{n+1} \to \mathcal{R}$$

is a real-valued function, define

$$\mathcal{E}^A_\alpha[g(\mathbf{D})] = \sum_{\mathbf{d}} g(\mathbf{d}) p^A_\alpha(\mathbf{d}),$$

where the sum ranges over all $(n+1)$-tuples $\mathbf{d} = (d_0, d_1, \ldots, d_n)$ of nonnegative real numbers. We may view $\mathcal{E}^A_\alpha$ as a functional

$$\mathcal{E}^A_\alpha : (\mathcal{R}^{n+1} \to \mathcal{R}) \to \mathcal{R}.$$

In case the set of input actions of A is empty, we may regard the sequences $\mathbf{d}$ as the values of an $(n+1)$-dimensional random variable $\mathbf{D} = (D_0, D_1, \ldots, D_n)$ defined on the conditional probability space X_α of native executions of A whose traces extend α. In this case, the quantity $\mathcal{E}^A_\alpha[g(\mathbf{D})]$ is just the expectation of $g(\mathbf{D})$, times the probability p^A_α of the set X_α.

Our abstract representation for probabilistic I/O automata assigns, to each probabilistic I/O automaton A without internal actions, the mapping $\mathcal{E}^A$ that takes each trace $\alpha \in U^*$ of length n to the functional $\mathcal{E}^A_\alpha$ on $\mathcal{R}^{n+1} \to \mathcal{R}$. We call the mapping $\mathcal{E}^A$ the *probabilistic behavior map* associated with A. In the next section, we show that probabilistic behavior maps are fully abstract with respect to a natural notion of probabilistic testing.

The compositionality of the representation of automata by probabilistic behavior maps is established in Theorem 1 below. In this theorem, $A|B$ denote the composition of compatible automata A and B, and $\mathbf{D}^{A|B}$, $\mathbf{D}^A$, and $\mathbf{D}^B$ denote $(n+1)$-dimensional random variables representing the random sequences of delay parameters in an execution of $A|B$, A, and B, respectively. These symbols are used (as is conventional in probability theory) as dummies indicating the variables over which the summations are to be taken; thus the notation $\mathcal{E}^{A|B}$ denotes a summation over $\mathbf{D}^{A|B}$, the notation $\mathcal{E}^A$ denotes a summation over $\mathbf{D}^A$, and the notation $\mathcal{E}^B$ denotes a summation over $\mathbf{D}^B$.

Theorem 1 *Suppose A and B are compatible probabilistic I/O automata that have no internal actions and $\alpha = e_0 e_1 \ldots e_{n-1}$. Then*

$$\mathcal{E}^{A|B}_\alpha[g(\mathbf{D}^{A|B})] = \mathcal{E}^B_\alpha[\mathcal{E}^A_\alpha[g(\mathbf{D}^A + \mathbf{D}^B) \cdot h(\mathbf{D}^A, \mathbf{D}^B)]\,],$$

where

$$h(\mathbf{D}^A, \mathbf{D}^B) = \left(\prod_{k \in K^{\mathrm{out}}_A} \frac{D^A_k}{D^A_k + D^B_k} \right) \left(\prod_{k \in K^{\mathrm{out}}_B} \frac{D^B_k}{D^A_k + D^B_k} \right)$$

$$K_A^{\text{out}} = \{k : 0 \le k < n,\ \ e_k \in E_A^{\text{out}}\} \qquad \text{and} \qquad K_B^{\text{out}} = \{k : 0 \le k < n,\ \ e_k \in E_B^{\text{out}}\}$$

Proof – By definition,

$$\mathcal{E}_\alpha^{A|B}[g(\mathbf{D}^{A|B})] = \sum_{\mathbf{d}} g(\mathbf{d}) p_\alpha^{A|B}(\mathbf{d}),$$

in which the quantity $p_\alpha^{A|B}(\mathbf{d})$ is given by the following:

$$p_\alpha^{A|B}(\mathbf{d}) = \sum_\sigma \prod_{k=0}^{n-1} \mu_{A|B}(\sigma(k), e_k, \sigma(k+1)),$$

where the summation is taken over all executions σ of $A|B$ having trace α and delay sequence $\mathbf{d}$. Substituting, we have

$$\mathcal{E}_\alpha^{A|B}[g(\mathbf{D}^{A|B})] = \sum_{\mathbf{d}} g(\mathbf{d}) \sum_\sigma \prod_{k=0}^{n-1} \mu_{A|B}(\sigma(k), e_k, \sigma(k+1)).$$

By Proposition 2.1 and the definition of composition for probabilistic I/O automata, the executions σ of $A|B$ having trace α and delay sequence $\mathbf{d}$ are in bijective correspondence with pairs of executions (σ_A, σ_B), such that σ_A and σ_B both have trace α, and such that $\delta(\sigma_A) + \delta(\sigma_B) = \mathbf{d}$. Using this fact and the definition of $\mu_{A|B}$ we can rewrite the above expression as follows:

$$\begin{aligned}
\mathcal{E}_\alpha^{A|B}[g(\mathbf{D}^{A|B})] \quad = \quad & \sum_{\mathbf{d}^B} \sum_{\mathbf{d}^A} g(\mathbf{d}^A + \mathbf{d}^B) \\
& \sum_{\sigma_A} \sum_{\sigma_B} \left(\prod_{k=0}^{n-1} \mu_A(\sigma_A(k), e_k, \sigma_A(k+1)) \right) \left(\prod_{k \in K_A^{\text{out}}} \frac{d_k^A}{d_k^A + d_k^B} \right) \\
& \left(\prod_{k=0}^{n-1} \mu_B(\sigma_B(k), e_k, \sigma_B(k+1)) \right) \left(\prod_{k \in K_B^{\text{out}}} \frac{d_k^B}{d_k^A + d_k^B} \right),
\end{aligned}$$

where σ_A ranges over all executions of A having trace α and delay sequence $\mathbf{d}^A$, and σ_B ranges over all executions of B having trace α and delay sequence $\mathbf{d}^B$. Rearranging terms gives

$$\begin{aligned}
\mathcal{E}_\alpha^{A|B}[g(\mathbf{D}^{A|B})] \quad = \quad & \sum_{\mathbf{d}^B} \sum_{\mathbf{d}^A} g(\mathbf{d}^A + \mathbf{d}^B) \left(\prod_{k \in K_A^{\text{out}}} \frac{d_k^A}{d_k^A + d_k^B} \right) \left(\prod_{k \in K_B^{\text{out}}} \frac{d_k^B}{d_k^A + d_k^B} \right) \\
& \left(\sum_{\sigma_A} \prod_{k=0}^{n-1} \mu_A(\sigma_A(k), e_k, \sigma_A(k+1)) \right) \left(\sum_{\sigma_B} \prod_{k=0}^{n-1} \mu_B(\sigma_B(k), e_k, \sigma_B(k+1)) \right)
\end{aligned}$$

or more simply,

$$\sum_{\mathbf{d}^B} \sum_{\mathbf{d}^A} g(\mathbf{d}^A + \mathbf{d}^B) \cdot h(\mathbf{d}^A, \mathbf{d}^B) \cdot p_\alpha^A(\mathbf{d}^A) \cdot p_\alpha^B(\mathbf{d}^B).$$

But this is easily recognized as

$$\mathcal{E}_\alpha^B[\mathcal{E}_\alpha^A[g(\mathbf{D}^A + \mathbf{D}^B) \cdot h(\mathbf{D}^A, \mathbf{D}^B)]\ ],$$

as was to be shown. ∎

5 Testing Equivalence and Full Abstraction

In this section we show that probabilistic behavior maps are fully abstract with respect to a notion of probabilistic testing equivalence based on the classical testing theory of Hennessy and DeNicola [dNH83]. That is to say, probabilistic I/O automata A and B determine the same probabilistic behavior map if and only if in a certain sense they cannot be distinguished by any probabilistic test.

Formally, a *test* is simply a probabilistic I/O automaton T that has a distinguished output action ω. We interpret the occurrence of ω in a computation of T as an indication that the test has succeeded. A test is called *closed* if its set of input actions is empty. For closed tests T, it makes sense (see Section 3.1) to talk about the probability of sets of executions of T.

Lemma 5.1 *Suppose T is a closed test. Then the set of all successful native executions of T is measurable. The probability of this set is given by the formula*

$$\sum_{\alpha \in \bar{\Omega}} \mathcal{E}_{\alpha\omega}^T[1],$$

where $\bar{\Omega}$ is the set of all traces that do not contain ω.

Proof – The probability of success of T is the probability of the set of all native executions of T that contain an occurrence of ω. This set can be represented as a countable union of disjoint sets of the form $[\sigma]$, where σ ranges over all finite native executions

$$\sigma = q_0 \xrightarrow{e_0} q_1 \xrightarrow{e_1} \ldots \xrightarrow{e_{n-1}} q_n \xrightarrow{\omega} q_{n+1},$$

in which ω appears for the first time as the last action. Let $e_n = \omega$, then the probability of each set $[\sigma]$ is given by

$$\Pr([\sigma]) = \prod_{k=0}^{n} \mu_T(q_k, e_k, q_{k+1}).$$

Thus, the probability of success of T can be expressed as

$$\sum_{\alpha \in \bar{\Omega}} \sum_{\mathrm{tr}(\sigma) = \alpha\omega} \prod_{k=0}^{n} \mu_T(\sigma(k), e_k, \sigma(k+1)),$$

which is just

$$\sum_{\alpha\in\bar{\Omega}} \mathcal{E}^T_{\alpha\omega}[1],$$

as was to be shown. ∎

The probability of the set of all successful native executions of a closed test T is called the *success probability* of T.

Suppose $A = (Q_A, q^{\mathrm{I}}_A, E_A, \Delta_A, \mu_A, \delta_A)$ is a probabilistic I/O automaton. A *proper test* for A is a test $T = (Q_T, q^{\mathrm{I}}_T, E_T, \Delta_T, \mu_T, \delta_T)$ such that $E^{\mathrm{in}}_T \subseteq E^{\mathrm{out}}_A$, $E^{\mathrm{in}}_A \subseteq E^{\mathrm{out}}_T \setminus \{\omega\}$, and $E^{\mathrm{loc}}_A \cap E^{\mathrm{loc}}_T = \emptyset$. If T is a proper test for A, then the collection $\{A, T\}$ is compatible. Let $A|T$ denote its composition, then $A|T$ is a closed system.

If A and B are probabilistic I/O automata with the same set of actions, then we call A and B *testing equivalent* if for all proper tests T for A and B, the success probability of $A|T$ equals the success probability of $B|T$.

We now define a particular class of tests that will be useful for distinguishing probabilistic I/O automata. Let a set of actions $E = E_0 \cup E_1$ be fixed, with E_0 and E_1 disjoint. For each trace $\alpha = e_0 e_1 \ldots e_{n-1}$ with $e_k \in E$ for $0 \leq k < n$, and for each sequence $\mathbf{x} = x_0, x_1, \ldots, x_n$ of positive real numbers, we define a test $T_{\alpha,\mathbf{x}} = (Q, q^{\mathrm{I}}, E_T, \Delta, \mu, \delta)$ as follows:

- $Q = \{0, 1, 2, \ldots, n, n+1\}$.
- $q^{\mathrm{I}} = 0$.
- $E_T = E \cup \{\omega, *\}$, with $E^{\mathrm{in}}_T = E_1$ and $E^{\mathrm{out}}_T = E_0 \cup \{\omega, *\}$.
- Δ is the union of the following sets:
    1. $\{(k, e_k, k+1) : 0 \leq k < n\}$
    2. $\{(n, \omega, n+1)\}$
    3. $\{(k, *, n+1) : 0 \leq k < n\}$
    4. $\{(k, e, n+1) : 0 \leq k < n, e \in E^{\mathrm{in}}_T, e \neq e_k\}$
    5. $\{(n, e, n+1) : e \in E^{\mathrm{in}}_T\}$.
    6. $\{(n+1, e, n+1) : e \in E^{\mathrm{in}}_T\}$.

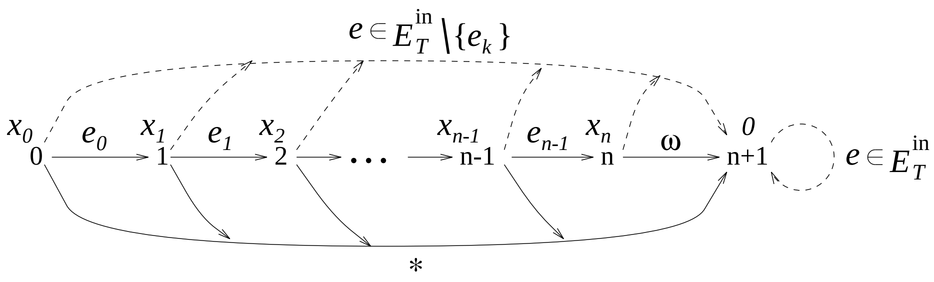

Figure 1: Test $T_{\alpha,\mathbf{x}}$

- μ is defined as follows:

  1. If $0 \le k < n$, then
  $$\mu(k, e_k, k+1) = \begin{cases} 1, & \text{if } e_k \in E_T^{\text{in}} \\ 1/2 & \text{otherwise.} \end{cases}$$
  2. $\mu(n, \omega, n+1) = 1$.
  3. If $0 \le k < n$, then
  $$\mu(k, *, n+1) = \begin{cases} 1, & \text{if } e_k \in E_T^{\text{in}} \\ 1/2 & \text{otherwise.} \end{cases}$$
  4. If $0 \le k < n$, $e \in E_T^{\text{in}}$, and $e \ne e_k$, then $\mu(k, e, n+1) = 1$.
  5. If $e \in E_T^{\text{in}}$, then $\mu(n, e, n+1) = 1$, and $\mu(n+1, e, n+1) = 1$.

- $\delta(k) = x_k$ for $0 \le k \le n$ and $\delta(n+1) = 0$.

Figure 1 depicts the structure of test $T_{\alpha,\mathbf{x}}$. Intuitively, $T_{\alpha,\mathbf{x}}$ succeeds when it manages to produce the trace $\alpha\omega$ by passing successively through states $0, 1, 2, \ldots, n$ and finally to $n+1$. For $0 \le k \le n$, the state k has delay parameter x_k, so the delay sequence $\delta(\sigma)$ associated with a successful execution σ of $T_{\alpha,\mathbf{x}}$ is the sequence $x_0 x_1 \ldots x_n 0$. This is the only way executions of $T_{\alpha,\mathbf{x}}$ can succeed; executions that deviate from α in the initial section cause $T_{\alpha,\mathbf{x}}$ to enter the state $n+1$ without performing the success action ω. In each state k, where $0 \le k < n$, the test $T_{\alpha,\mathbf{x}}$ has a nonzero chance of failing by performing the action $*$ and going directly to state $n+1$. This gives $T_{\alpha,\mathbf{x}}$ a certain sensitivity to the delays of its environment.

Lemma 5.2 *Suppose A is a probabilistic I/O automaton without internal actions. Then for each trace $\alpha = e_0 e_1 \ldots e_{n-1} \in E_A^*$, and for each sequence $\mathbf{x} = x_0, x_1, \ldots, x_n$ of positive real numbers, the test $T_{\alpha,\mathbf{x}}$ (with $E_0 = E_A^{\text{in}}$ and $E_1 = E_A^{\text{out}}$) is a proper test for A. Moreover, the success probability of $A|T_{\alpha,\mathbf{x}}$ is given by:*

$$2^{-c} \cdot \mathcal{E}_{\alpha\omega}^A[\prod_{k=0}^{n} \frac{y_k}{x_k + D_k^A}],$$

where for all $0 \leq k < n$ *we have*

$$y_k = \begin{cases} D^A_k, & \text{if } e_k \in E^{\text{out}}_A \\ x_k & \text{otherwise,} \end{cases}$$

$y_n = x_n$*, and* c *is the number of* $k \in \{0, 1, \ldots, n-1\}$ *for which* $e_k \in E^{\text{in}}_A$*.*

Proof – Fix α and $\mathbf{x}$, and let T abbreviate $T_{\alpha,\mathbf{x}}$. We first verify that T is a proper test for A. By the construction of T, it is obvious that T is a test (a probabilistic I/O automaton with distinguished action ω). Since $E^{\text{in}}_T = E^{\text{out}}_A$, and $E^{\text{in}}_A \subseteq E^{\text{out}}_T$, it follows that T is a proper test for A.

By Lemma 5.1, the success probability of $A|T$ is given by:

$$\sum_{\alpha' \in \bar{\Omega}} \mathcal{E}^{A|T}_{\alpha'\omega}[1],$$

where $\bar{\Omega}$ is the set of all traces that do not contain ω. By Theorem 1, this may be rewritten as follows:

$$\sum_{\alpha' \in \bar{\Omega}} \mathcal{E}^A_{\alpha'\omega}[\mathcal{E}^T_{\alpha'\omega}[h(\mathbf{D}^A, \mathbf{D}^T)]\,],$$

where

$$h(\mathbf{D}^A, \mathbf{D}^T) = \left(\prod_{k \in K^{\text{out}}_A} \frac{D^A_k}{D^A_k + D^T_k} \right) \left(\prod_{k \in K^{\text{out}}_T} \frac{D^T_k}{D^A_k + D^T_k} \right)$$

and index sets K^{out}_A and K^{out}_T contain the indices of actions in $\alpha'\omega$ that are output actions of A and T respectively.

Now, observe that T has just one execution that produces a trace of the form $\alpha'\omega$, namely

$$0 \xrightarrow{e_0} 1 \xrightarrow{e_1} 2 \xrightarrow{e_2} \ldots \xrightarrow{e_{n-1}} n \xrightarrow{\omega} n+1,$$

which produces trace $\alpha\omega$. From this it is easy to see that $\mathcal{E}^T_{\alpha'\omega}[h(D^A, D^T)] = 0$ unless $\alpha' = \alpha$, and that

$$\mathcal{E}^T_{\alpha\omega}[h(\mathbf{D}^A, \mathbf{D}^T)] = 2^{-c} \cdot \left(\prod_{k \in K^{\text{out}}_A} \frac{D^A_k}{D^A_k + x_k} \right) \left(\prod_{k \in K^{\text{out}}_T} \frac{x_k}{D^A_k + x_k} \right),$$

where c is the number of $k \in \{0, 1, \ldots, n-1\}$ such that $e_k \in E^{\text{in}}_A$. Using this information and the fact that each e_k is an output action of either A or T, the success probability of $A|T$ may be rewritten as:

$$2^{-c} \cdot \mathcal{E}^A_{\alpha\omega}[\prod_{k=0}^{n} \frac{y_k}{x_k + D^A_k}],$$

where for all $0 \leq k < n$ we have

$$y_k = \begin{cases} D^A_k, & \text{if } e_k \in E^{\text{out}}_A \\ x_k & \text{otherwise,} \end{cases}$$

and $y_n = x_n$. ∎

Lemma 5.3 below is a uniqueness theorem for partial fraction expansions of rational functions in several variables. It is a key component of the proof of full abstraction (Theorem 2). We state a somewhat more general version than our present needs dictate; the extra generality will be used in Section 6. See Appendix A for the proof of Lemma 5.3

Lemma 5.3 *Suppose f and f' are two rational functions of variables $x_0, x_1, \ldots, x_{n-1}$ ($n \geq 0$), defined as follows:*

$$f = \sum_{i \in I} \frac{a_i}{\prod_{k=0}^{n-1}(x_k + b_{i,k})^{r_{i,k}}} \qquad f' = \sum_{i \in I'} \frac{c_i}{\prod_{k=0}^{n-1}(x_k + d_{i,k})^{s_{i,k}}},$$

where I and I' are finite sets, for each $i \in I$ and $0 \leq k < n$, the exponent $r_{i,k}$ is a positive integer and $a_i \in (0, \infty)$, for each $i \in I'$ and $0 \leq k < n$, the exponent $s_{i,k}$ is a positive integer and $c_i \in (0, \infty)$, for each distinct $i, j \in I$ the sets $\{(k, b_{i,k}, r_{i,k}) : 0 \leq k < n\}$ and $\{(k, b_{j,k}, r_{j,k}) : 0 \leq k < n\}$ are distinct, and for each distinct $i, j \in I'$ the sets $\{(k, d_{i,k}, s_{i,k}) : 0 \leq k < n\}$ and $\{(k, d_{j,k}, s_{j,k}) : 0 \leq k < n\}$ are distinct. If $f = f'$, then there exists a bijection $(\text{ - })' : I \to I'$ such that $a_i = c_{i'}$, $b_{i,k} = d_{i',k}$ and $r_{i,k} = s_{i',k}$ for all $i \in I$ and $0 \leq k < n$.

Theorem 2 *Suppose A and B are probabilistic I/O automata with the same set of actions for which the set of internal actions is empty. Then A and B are testing equivalent if and only if the associated probabilistic behavior maps $\mathcal{E}^A$ and $\mathcal{E}^B$ are equal.*

Proof – Suppose $\mathcal{E}^A = \mathcal{E}^B$, and let T be an arbitrary proper test for A and B. By Lemma 5.1, the success probability of $A|T$ is

$$\sum_\alpha \mathcal{E}^{A|T}_{\alpha\omega}[1]$$

and the success probability of $B|T$ is

$$\sum_\alpha \mathcal{E}^{B|T}_{\alpha\omega}[1]\,,$$

where α ranges over all traces that do not contain ω. Applying Theorem 1 we can express the success probability of $A|T$ as

$$\sum_\alpha \mathcal{E}^A_{\alpha\omega}[\mathcal{E}^T_{\alpha\omega}[h(\mathbf{D}^A, \mathbf{D}^T)]\,]\,,$$

where

$$h(\mathbf{D}^A, \mathbf{D}^T) = \left(\prod_{k\in K^{\text{out}}_A} \frac{D^A_k}{D^A_k + D^T_k}\right)\left(\prod_{k\in K^{\text{out}}_T} \frac{D^T_k}{D^A_k + D^T_k}\right).$$

Similarly, we can express the success probability of $B|T$ as

$$\sum_\alpha \mathcal{E}^B_{\alpha\omega}[\mathcal{E}^T_{\alpha\omega}[h(\mathbf{D}^B, \mathbf{D}^T)]\,]\,.$$

Since $\mathcal{E}^B = \mathcal{E}^A$ by hypothesis, it follows that the two success probabilities are equal. Since for an arbitrary T, the success probability of $A|T$ equals the success probability of $B|T$, it follows that A and B are testing equivalent.

Conversely suppose A and B are testing equivalent. Then for all tests T, the success probability of $A|T$ equals the success probability of $B|T$. In particular, this is true for all tests of the form $T_{\alpha,\mathbf{x}}$ for any $\alpha = e_0e_1\ldots e_{n-1}$. By Lemma 5.2, the success probability of $A|T_{\alpha,\mathbf{x}}$ is given by

$$s_A(\mathbf{x}) = 2^{-c}\cdot \mathcal{E}^A_{\alpha\omega}[\prod_{k=0}^{n} \frac{y_k}{x_k + D^A_k}]$$

and the success probability of $B|T_{\alpha,\mathbf{x}}$ is given by

$$s_B(\mathbf{x}) = 2^{-c}\cdot \mathcal{E}^B_{\alpha\omega}[\prod_{k=0}^{n} \frac{y'_k}{x_k + D^B_k}]\,,$$

where c is the number of $k \in \{0, 1, \ldots, n-1\}$ for which $e_k \in E^{\text{in}}_A$ $(= E^{\text{in}}_B)$, $y_n = y'_n = x_n$, and for all $0 \le k < n$ we have

$$y_k = \begin{cases} D^A_k, & \text{if } e_k \in E^{\text{out}}_A \\ x_k & \text{otherwise,} \end{cases} \qquad \text{and} \qquad y'_k = \begin{cases} D^B_k, & \text{if } e_k \in E^{\text{out}}_B \\ x_k & \text{otherwise.} \end{cases}$$

Since $s_A(\mathbf{x}) = s_B(\mathbf{x})$ by the hypothesis that A and B are testing equivalent, and by the definition of $\mathcal{E}^A_{\alpha\omega}$ we have

$$\sum_{i\in I} \frac{p^A_{\alpha\omega}(\mathbf{d}_i)\cdot \prod_{k\in K^{\rm out}} d_{i,k}}{\prod_{k=0}^n (x_k + d_{i,k})} \;=\; \sum_{i'\in I'} \frac{p^B_{\alpha\omega}(\mathbf{d}'_{i'})\cdot \prod_{k\in K^{\rm out}} d'_{i',k}}{\prod_{k=0}^n (x_k + d'_{i',k})}$$

where

$$K^{\rm out} \;=\; \{k : 0 \le k \le n,\ e_n \in E^{\rm out}_A (= E^{\rm out}_B)\}\ ,$$

$\{\mathbf{d}_i : i \in I\}$ is the finite set of sequences $\mathbf{d}$ for which $p^A_{\alpha\omega}(\mathbf{d})$ is nonzero, and $\{\mathbf{d}'_{i'} : i' \in I'\}$ is the finite set of sequences $\mathbf{d}'$ for which $p^B_{\alpha\omega}(\mathbf{d}')$ is nonzero.

Thus, we have two rational functions in the variables $x_0, x_1, \ldots, x_n$ which are equal for all positive values of their arguments. From basic properties of rational functions, if two such functions are defined and equal for all values in some open interval, then they are equal at all points where either of them is defined. Then, applying Lemma 5.3, there exists a bijection $(\text{ - })' : I \to I'$ such that for all $i \in I$,

$$\mathbf{d}_i = \mathbf{d}'_{i'} \qquad \text{and} \qquad p^A_{\alpha\omega}(\mathbf{d}_i)\cdot \prod_{k\in K^{\rm out}} d_{i,k} \;=\; p^B_{\alpha\omega}(\mathbf{d}'_{i'})\cdot \prod_{k\in K^{\rm out}} d'_{i',k}$$

Since the products $\prod_{k\in K^{\rm out}} d_{i,k}$ and $\prod_{k\in K^{\rm out}} d'_{i',k}$ are positive by the assumption that $p^A_{\alpha\omega}(\mathbf{d}_i)$ and $p^B_{\alpha\omega}(\mathbf{d}'_i)$ are nonzero, and since $p^A_{\alpha\omega}(\mathbf{d}) = 0$ except when $\mathbf{d} = \mathbf{d}_i$ for some $i \in I$, and similarly for $p^B_{\alpha\omega}(\mathbf{d})$, we have shown that for all traces $\alpha = e_0e_1\ldots e_{n-1}$ in which ω does not appear, and for all sequences $\mathbf{d} = d_0d_1\ldots d_{n-1}d_nd_{n+1}$ we have

$$p^A_{\alpha\omega}(\mathbf{d}) \;=\; p^B_{\alpha\omega}(\mathbf{d}).$$

Now,

$$p^A_{\alpha\omega}(\mathbf{d}) = \sum_\sigma \left(\prod_{k=0}^{n-1} \mu_A(\sigma(k), e_k, \sigma(k+1))\right)\cdot \mu_A(\sigma(n), \omega, \sigma(n+1)),$$

where the summation is taken over all executions σ of A having trace $\alpha\omega$ and delay sequence $\mathbf{d} = d_0d_1\ldots d_{n-1}d_nd_{n+1}$. However, $\omega \notin E_A$, so the only such executions are those whose last transition is of the form (q, ω, q). It follows from this observation that

$$p^A_{\alpha\omega}(d_0d_1\ldots d_nd_{n+1}) = p^A_\alpha(d_0d_1\ldots d_n)$$

for all traces α and all sequences $d_0d_1\ldots d_nd_{n+1}$. Similar reasoning applies to B, so we may conclude that

$$p^A_\alpha(\mathbf{d}) = p^B_\alpha(\mathbf{d})$$

for all traces α in which ω does not appear, and all sequences $\mathbf{d}$. Moreover, this conclusion holds even without the restriction that ω not appear in α. For, given trace α containing occurrences of ω, it is easy to see that $p^A_\alpha = p^A_{\alpha'}$, where α' is obtained by changing all occurrences of ω in α to some other symbol ω' not in E_A. One may do the same thing for p^B_α, and then apply $p^A_{\alpha'} = p^B_{\alpha'}$.

We have thus shown that $p^A_\alpha(\mathbf{d}) = p^B_\alpha(\mathbf{d})$ for all traces α and all sequences $\mathbf{d}$. From this, it follows by the definition of $\mathcal{E}^A_\alpha$ and $\mathcal{E}^B_\alpha$ that $\mathcal{E}^A_\alpha = \mathcal{E}^B_\alpha$ for all traces α. In other words, $\mathcal{E}^A = \mathcal{E}^B$, as was to be shown. ∎

6 Probabilistic I/O Automata with Internal Actions

In this section, we consider the class of probabilistic I/O automata $A = (Q, q^{\mathrm{I}}, E, \Delta, \mu, \delta)$ satisfying the following conditions:

- For all $(q, e, r) \in \Delta$, if $e \in E^{\mathrm{int}}$ then $\delta(r) = \delta(q)$.
- The *divergence-free* condition: for all $q \in Q$, there exists no infinite sequence $q_0, q_1, \cdots$ with $q = q_0$ and $q_i \xrightarrow{e_i} q_{i+1}$ for actions $e_i \in E^{\mathrm{int}}$.

The first condition states that internal transitions do not change the state delay parameters, and the second condition is imposed so that at most finitely many states can be reached from any given state by internal executions. We call probabilistic I/O automata satisfying these conditions *delay-restricted* probabilistic I/O automata. It is easy to verify that the composition of two compatible delay-restricted probabilistic I/O automata is also a delay-restricted automaton. For any composite state $\langle q, r \rangle$, if an internal action e is enabled, then by the compatibility condition, either $q \xrightarrow{e} q'$ or $r \xrightarrow{e} r'$. Suppose $q \xrightarrow{e} q'$, then $\langle q, r \rangle \xrightarrow{e} \langle q', r \rangle$. Since $\delta(q) = \delta(q')$, it follows that $\delta(\langle q, r \rangle) = \delta(\langle q', r \rangle)$. The case $r \xrightarrow{e} r'$ is similar. The divergence-free condition on component states q and r also implies there exists no infinite sequence of internal steps starting from state $\langle q, r \rangle$.

By applying techniques similar to those used in Sections 4 and 5, we show how to associate with a delay-restricted automaton an abstract representation, similar to the probabilistic behavior maps for probabilistic I/O automata without internal actions, that is compositional and fully abstract with respect to probabilistic testing.

6.1 Behaviors

Suppose $A = (Q, q^{\mathrm{I}}, E, \Delta, \mu, \delta)$ is a delay-restricted probabilistic I/O automaton. For each sequence $\alpha = e_0 e_1 \ldots e_{m-1}$ of *external* actions, i.e. $\alpha \in (U^{\mathrm{ext}})^*$, define the set Ext^A_α to be the set of all executions of A having traces of the form:

$$\tau^* e_0 \tau^* e_1 \ldots \tau^* e_{m-1},$$

where τ^* denotes any finite number of internal actions of A. That is, the set Ext^A_α contains all executions σ of A having external trace α and ending with action e_{m-1}. Using a König's Lemma argument, the local finite-branching property, the input image-finiteness property, and the divergence-free property of A, one can show that for each sequence α of external actions, the set Ext^A_α is finite.

Suppose σ is an execution in Ext^A_α that has n_k internal steps immediately before doing action e_k, $0 \leq k < m$. Then we represent σ as:

$$\sigma(0,0)\xrightarrow{\tau_{0,0}}\sigma(0,1)\xrightarrow{\tau_{0,1}}\cdots\xrightarrow{\tau_{0,n_0-1}}\sigma(0,n_0)\xrightarrow{e_0}\sigma(1,0)\xrightarrow{\tau_{1,0}}$$
$$\cdots\xrightarrow{\tau_{m-1,n_{m-1}-1}}\sigma(m-1,n_{m-1})\xrightarrow{e_{m-1}}\sigma(m,0)$$

where for $0 \leq k < m$ and $0 \leq i < n_k$, $\tau_{k,i} \in E^{\text{int}}$. Usually it is not necessary to identify each internal action and we abbreviate the above sequence as:

$$\sigma(0,0)\xRightarrow{\tau^{n_0}}\sigma(0,n_0)\xrightarrow{e_0}\sigma(1,0)\xRightarrow{\tau^{n_1}}\cdots\xRightarrow{\tau^{n_{m-1}}}\sigma(m-1,n_{m-1})\xrightarrow{e_{m-1}}\sigma(m,0).$$

By the first condition of delay-restricted probabilistic I/O automata, the delay sequence of σ is simply of the form $(d_0)^{n_0+1}(d_1)^{n_1+1}\cdots(d_{m-1})^{n_{m-1}+1}(d_m)^1$ for some delay parameters $d_0, d_1, \cdots, d_m$. It will be convenient to denote such a delay sequence as the pair $(\mathbf{d}, \mathbf{n})$, where $\mathbf{d} \in \mathcal{R}^{m+1}$ is the sequence of delay parameters $d_0 d_1 \cdots d_m$, and $\mathbf{n} \in \mathcal{N}^m$ is the sequence of nonnegative integers $n_0 n_1 \cdots n_{m-1}$, giving the number of internal actions between each successive pair of external actions.

Let $A = (Q, q^{\mathrm{I}}, E, \Delta, \mu, \delta)$ be a delay-restricted probabilistic I/O automaton. Given an external trace $\alpha = e_0 e_1 \ldots e_{m-1}$, for each delay sequence $(\mathbf{d}, \mathbf{n})$, define the quantity $p^A_\alpha(\mathbf{d}, \mathbf{n})$ by:

$$p^A_\alpha(\mathbf{d},\mathbf{n}) = \sum_\sigma \prod_{k=0}^{m-1}\left(\prod_{i=0}^{n_k-1}\mu_A(\sigma(k,i)\xrightarrow{\tau_{k,i}}\sigma(k,i+1))\right)\cdot\mu_A(\sigma(k,n_k),e_k,\sigma(k+1,0)),$$

where the summation is taken over all executions σ in Ext^A_α having delay sequence $(\mathbf{d}, \mathbf{n})$. Since the set Ext^A_α is finite, the summation converges, and for a fixed α, the set of all $(\mathbf{d}, \mathbf{n})$ for which $p^A_\alpha(\mathbf{d}, \mathbf{n})$ is nonzero, is finite.

Now, in a fashion analogous to Section 4, if

$$g : \mathcal{R}^{m+1} \times \mathcal{N}^m \to \mathcal{R}$$

is a real-valued function, define

$$\mathcal{E}^A_\alpha[g(\mathbf{D},\mathbf{N})] = \sum_{(\mathbf{d},\mathbf{n})} g(\mathbf{d},\mathbf{n})p^A_\alpha(\mathbf{d},\mathbf{n}),$$

where the summation ranges over all pairs $(\mathbf{d}, \mathbf{n}) \in \mathcal{R}^{m+1} \times \mathcal{N}^m$ for which $p^A_\alpha(\mathbf{d}, \mathbf{n})$ is nonzero. We may view $\mathcal{E}^A_\alpha$ as a functional

$$\mathcal{E}^A_\alpha : (\mathcal{R}^{m+1} \times \mathcal{N}^m \to \mathcal{R}) \to \mathcal{R}.$$

Our abstract representation for delay-restricted probabilistic I/O automata assigns, to each automaton A, the mapping $\mathcal{E}^A$ that takes each external trace $\alpha \in (U^{\text{ext}})^*$ of length m to the functional $\mathcal{E}^A_\alpha$ on $\mathcal{R}^{m+1} \times \mathcal{N}^m \to \mathcal{R}$. We also call the mapping $\mathcal{E}^A$ the *probabilistic behavior map* associated with A. The compositionality of the representation of delay-restricted automata by probabilistic behavior maps is established in the following theorem:

Theorem 3 *Suppose A and B are compatible delay-restricted probabilistic I/O automata and $\alpha = e_0 e_1 \dots e_{m-1}$ is an external trace. Then*

$$\mathcal{E}_\alpha^{A|B}[g(\mathbf{D}^{A|B}, \mathbf{N}^{A|B})] = \mathcal{E}_\alpha^{B}[\mathcal{E}_\alpha^{A}[g(\mathbf{D}^{A}+\mathbf{D}^{B}, \mathbf{N}^{A}+\mathbf{N}^{B}) \cdot h((\mathbf{D}^{A}, \mathbf{N}^{A}), (\mathbf{D}^{B}, \mathbf{N}^{B}))]\,],$$

where

$$\begin{aligned} h((\mathbf{D}^{A}, \mathbf{N}^{A}), (\mathbf{D}^{B}, \mathbf{N}^{B})) \;&=\; \left(\prod_{k=0}^{m-1} \binom{N_k^A + N_k^B}{N_k^A} \cdot \left(\frac{D_k^A}{D_k^A + D_k^B}\right)^{N_k^A} \cdot \left(\frac{D_k^B}{D_k^A + D_k^B}\right)^{N_k^B}\right) \\ &\quad \left(\prod_{k \in K_A^{\text{out}}} \frac{D_k^A}{D_k^A + D_k^B}\right)\left(\prod_{k \in K_B^{\text{out}}} \frac{D_k^B}{D_k^A + D_k^B}\right) \end{aligned}$$

and

$$K_A^{\text{out}} = \{k : 0 \le k < m,\ e_k \in E_A^{\text{out}}\} \qquad K_B^{\text{out}} = \{k : 0 \le k < m,\ e_k \in E_B^{\text{out}}\}$$

Proof – By definition,

$$\mathcal{E}_\alpha^{A|B}[g(\mathbf{D}^{A|B}, \mathbf{N}^{A|B})] = \sum_{(\mathbf{d},\mathbf{n})} g(\mathbf{d},\mathbf{n}) p_\alpha^{A|B}(\mathbf{d},\mathbf{n}),$$

in which the quantity $p_\alpha^{A|B}(\mathbf{d},\mathbf{n})$ is given by the following:

$$p_\alpha^{A|B}(\mathbf{d},\mathbf{n}) = \sum_\sigma \prod_{k=0}^{m-1} \left(\prod_{i=0}^{n_k-1} \mu_{A|B}(\sigma(k,i) \xrightarrow{\tau_{k,i}} \sigma(k,i+1))\right) \cdot \mu_{A|B}(\sigma(k,n_k), e_k, \sigma(k+1,0)),$$

where the summation is taken over all executions σ in $\mathrm{Ext}_\alpha^{A|B}$ having delay sequence $(\mathbf{d},\mathbf{n})$.

By Proposition 2.1, the executions σ in $\mathrm{Ext}_\alpha^{A|B}$ having delay sequence $(\mathbf{d},\mathbf{n})$ are in bijective correspondence with pairs (σ_A, σ_B), where σ_A is an execution of A having external trace α and delay sequence $(\mathbf{d}^A, \mathbf{n})$, σ_B is an execution of B having external trace α and delay sequence $(\mathbf{d}^B, \mathbf{n})$, such that $\mathbf{d} = \mathbf{d}^A + \mathbf{d}^B$. However, σ_A is not necessarily in Ext_α^A since some of the internal actions in σ_A are from B. More precisely, suppose

$$\sigma = \sigma(0,0) \xRightarrow{\tau^{n_0}} \sigma(0,n_0) \xrightarrow{e_0} \sigma(1,0) \xRightarrow{\tau^{n_1}} \cdots \xRightarrow{\tau^{n_{m-1}}} \sigma(m-1, n_{m-1}) \xrightarrow{e_{m-1}} \sigma(m,0)$$

is an execution in $\mathrm{Ext}_\alpha^{A|B}$ with delay sequence $(\mathbf{d},\mathbf{n})$. By Proposition 2.1, projecting σ onto A gives execution

$$\begin{aligned} \sigma_A \;&=\; \sigma_A(0,0) \xRightarrow{\tau^{n_0}} \sigma_A(0,n_0) \xrightarrow{e_0} \sigma_A(1,0) \xRightarrow{\tau^{n_1}} \cdots \\ &\quad \cdots \xRightarrow{\tau^{n_{m-1}}} \sigma_A(m-1, n_{m-1}) \xrightarrow{e_{m-1}} \sigma_A(m,0) \end{aligned}$$

with delay sequences $(\mathbf{d}^A, \mathbf{n})$. But since all the n_k internal steps in σ are from either A or B, some of the internal steps in σ_A are from B. By convention $q \xrightarrow{e} q$ and $\mu_A(q,e,q) = 1$ if $e \notin E_A$, we can remove those internal transitions generated by B from σ_A and have

$$\begin{aligned} \sigma_A' \;&=\; \sigma_A(0,0) \xRightarrow{\tau^{n_0^A}} \sigma_A(0,n_0^A) \xrightarrow{e_0} \sigma_A(1,0) \xRightarrow{\tau^{n_1^A}} \cdots \\ &\quad \cdots \xRightarrow{\tau^{n_{m-1}^A}} \sigma_A(m-1, n_{m-1}^A) \xrightarrow{e_{m-1}} \sigma_A(m,0) \end{aligned}$$

Now, σ'_A is in Ext^A_α since it contains internal actions in A only. Similarly, we can remove the internal steps generated by A from σ_B and have an execution σ'_B that is in Ext^B_α. Thus, for a given external trace $\alpha = e_0 e_1 \cdots e_{m-1}$, choosing an execution $\sigma \in \mathrm{Ext}^{A|B}_\alpha$ with delay sequence $(\mathbf{d}, \mathbf{n})$ amounts to choosing a pair (σ_A, σ_B), where $\sigma_A \in \mathrm{Ext}^A_\alpha$ has delay sequence $(\mathbf{d}^A, \mathbf{n}^A)$, $\sigma_B \in \mathrm{Ext}^B_\alpha$ has delay sequence $(\mathbf{d}^B, \mathbf{n}^B)$, such that $\mathbf{d}^A + \mathbf{d}^B = \mathbf{d}$ and $\mathbf{n}^A + \mathbf{n}^B = \mathbf{n}$, and for $0 \le k < m$, choosing a particular interleaving of the n^A_k internal actions of A and the n^B_k internal actions of B. For a fixed choice of (σ_A, σ_B), the total number of such interleaving is given by:

$$\prod_{k=0}^{m-1} \binom{n^A_k + n^B_k}{n^A_k}.$$

Then by the definition of composition for probabilistic I/O automata, we have

$$\begin{aligned}
& p^{A|B}_\alpha(\mathbf{d}, \mathbf{n}) \\
&= \sum_\sigma \prod_{k=0}^{m-1} \left(\prod_{i=0}^{n_k - 1} \mu_{A|B}(\sigma(k,i) \xrightarrow{\tau_{k,i}} \sigma(k, i+1)) \right) \cdot \mu_{A|B}(\sigma(k, n_k), e_k, \sigma(k+1, 0)) \\
&= \sum_{\sigma_A} \sum_{\sigma_B} \left(\prod_{k=0}^{m-1} \binom{n^A_k + n^B_k}{n^A_k} \cdot \left(\frac{d^A_k}{d^A_k + d^B_k}\right)^{n^A_k} \cdot \left(\frac{d^B_k}{d^A_k + d^B_k}\right)^{n^B_k} \cdot \left(\frac{\lambda_k}{d^A_k + d^B_k}\right) \right) \\
&\quad \cdot \left(\prod_{k=0}^{m-1} \left(\prod_{i=0}^{n^A_k - 1} \mu_A(\sigma_A(k,i) \xrightarrow{\tau_{k,i}} \sigma_A(k, i+1)) \right) \cdot \mu_A(\sigma_A(k, n^A_k), e_k, \sigma_A(k+1, 0)) \right) \\
&\quad \cdot \left(\prod_{k=0}^{m-1} \left(\prod_{i=0}^{n^B_k - 1} \mu_B(\sigma_B(k,i) \xrightarrow{\tau_{k,i}} \sigma_B(k, i+1)) \right) \cdot \mu_B(\sigma_B(k, n^B_k), e_k, \sigma_B(k+1, 0)) \right)
\end{aligned}$$

where

$$\lambda_k = \begin{cases} d^A_k & \text{if } e_k \in E^{\mathrm{out}}_A \\ d^B_k & \text{if } e_k \in E^{\mathrm{out}}_B \\ d^A_k + d^B_k & \text{otherwise,} \end{cases}$$

σ, σ_A, and σ_B range over $\mathrm{Ext}^{A|B}_\alpha$, Ext^A_α, and Ext^B_α, respectively, $(\mathbf{d}^A, \mathbf{n}^A)$ denotes the delay sequence of σ^A, and $(\mathbf{d}^B, \mathbf{n}^B)$ denotes the delay sequence of σ^B.

Substituting the above into the definition of $\mathcal{E}^{A|B}_\alpha$ and rearranging terms gives

$$\begin{aligned}
& \mathcal{E}^{A|B}_\alpha[g(\mathbf{D}^{A|B}, \mathbf{N}^{A|B})] \\
&= \sum_{(\mathbf{d},\mathbf{n})} g(\mathbf{d}, \mathbf{n}) p^{A|B}_\alpha(\mathbf{d}, \mathbf{n}) \\
&= \sum_{(\mathbf{d}^B, \mathbf{n}^B)} \sum_{(\mathbf{d}^A, \mathbf{n}^A)} g(\mathbf{d}^A + \mathbf{d}^B, \mathbf{n}^A + \mathbf{n}^B) \left(\prod_{k \in K^{\mathrm{out}}_A}^{m-1} \frac{d^A_k}{d^A_k + d^B_k} \right) \left(\prod_{k \in K^{\mathrm{out}}_B}^{m-1} \frac{d^B_k}{d^A_k + d^B_k} \right) \\
&\quad \cdot \left(\prod_{k=0}^{m-1} \binom{n^A_k + n^B_k}{n^A_k} \cdot \left(\frac{d^A_k}{d^A_k + d^B_k}\right)^{n^A_k} \cdot \left(\frac{d^B_k}{d^A_k + d^B_k}\right)^{n^B_k} \right)
\end{aligned}$$

$$\cdot \left(\sum_{\sigma_A} \prod_{k=0}^{m-1} \left(\prod_{i=0}^{n_k^A - 1} \mu_A(\sigma_A(k,i) \xrightarrow{\tau_{k,i}} \sigma_A(k,i+1)) \right) \cdot \mu_A(\sigma_A(k, n_k^A), e_k, \sigma_A(k+1, 0)) \right)$$
$$\cdot \left(\sum_{\sigma_B} \prod_{k=0}^{m-1} \left(\prod_{i=0}^{n_k^B - 1} \mu_B(\sigma_B(k,i) \xrightarrow{\tau_{k,i}} \sigma_B(k,i+1)) \right) \cdot \mu_B(\sigma_B(k, n_k^B), e_k, \sigma_B(k+1, 0)) \right)$$

or more simply,

$$\sum_{(\mathbf{d}^B, \mathbf{n}^B)} \sum_{(\mathbf{d}^A, \mathbf{n}^A)} g(\mathbf{d}^A + \mathbf{d}^B, \mathbf{n}^A + \mathbf{n}^B) \cdot h((\mathbf{d}^A, \mathbf{n}^A), (\mathbf{d}^B, \mathbf{n}^B)) \cdot p_\alpha^A(\mathbf{d}^A, \mathbf{n}^A) \cdot p_\alpha^B(\mathbf{d}^B, \mathbf{n}^B).$$

But this is easily recognized as

$$\mathcal{E}_\alpha^B[\mathcal{E}_\alpha^A[g(\mathbf{D}^A + \mathbf{D}^B, \mathbf{N}^A + \mathbf{N}^B) \cdot h((\mathbf{D}^A, \mathbf{N}^A), (\mathbf{D}^B, \mathbf{N}^B))]\,],$$

as was to be shown. ∎

Observe that Theorem 3 directly generalizes Theorem 1. If we apply Theorem 3 to probabilistic I/O automata without internal actions, then

$$\begin{aligned} h((\mathbf{D}^A, \mathbf{N}^A), (\mathbf{D}^B, \mathbf{N}^B)) &= \left(\prod_{k=0}^{m-1} \binom{N_k^A + N_k^B}{N_k^A} \cdot (\frac{D_k^A}{D_k^A + D_k^B})^{N_k^A} \cdot (\frac{D_k^B}{D_k^A + D_k^B})^{N_k^B} \right) \\ &\qquad \left(\prod_{k \in K_A^{\mathrm{out}}} \frac{D_k^A}{D_k^A + D_k^B} \right) \left(\prod_{k \in K_B^{\mathrm{out}}} \frac{D_k^B}{D_k^A + D_k^B} \right) \\ &= \left(\prod_{k \in K_A^{\mathrm{out}}} \frac{D_k^A}{D_k^A + D_k^B} \right) \left(\prod_{k \in K_B^{\mathrm{out}}} \frac{D_k^B}{D_k^A + D_k^B} \right), \end{aligned}$$

where the last equality holds because there are no internal actions between external actions and, as a result, the first product (of arity m) equals 1. By eliding the variables $\mathbf{N}^A$ and $\mathbf{N}^B$, which have constant value zero in case there are no internal actions, we arrive at the same equation for $\mathcal{E}_\alpha^{A|B}$ as in Section 4.

6.2 Testing Equivalence and Full Abstraction with Internal Actions

In this section we show that probabilistic behavior maps for delay-restricted probabilistic I/O automata are fully abstract with respect to the notion of probabilistic testing equivalence given in Section 5. We restate the lemmas and theorem of Section 5, with appropriate modifications, and give new proofs as required.

Recall that a *test* is simply a probabilistic I/O automaton T that has a distinguished output action ω.

Lemma 6.1 *Suppose T is a closed test. Then the set of all successful native executions of T is measurable. The probability of this set is given by the formula*

$$\sum_{\alpha \in \bar{\Omega}} \mathcal{E}^T_{\alpha\omega}[1],$$

where $\bar{\Omega}$ is the set of all external traces that do not contain ω.

Proof – The success probability of T is the probability of the set of all native executions of T that contain an occurrence of ω. This set can be represented as a countable union of sets of the form $[\sigma]$, where σ ranges over all finite native executions

$$\begin{aligned}\sigma \;=\; & \sigma(0,0) \overset{\tau^{n_0}}{\Longrightarrow} \sigma(0,n_0) \xrightarrow{e_0} \sigma(1,0) \overset{\tau^{n_1}}{\Longrightarrow} \cdots \\ & \overset{\tau^{n_{m-1}}}{\Longrightarrow} \sigma(m-1,n_{m-1}) \xrightarrow{e_{m-1}} \sigma(m,0) \overset{\tau^{n_m}}{\Longrightarrow} \sigma(m,n_m) \xrightarrow{\omega} \sigma(m+1,0),\end{aligned}$$

in which n_k (≥ 0) number of internal steps occur before each external action e_k, $0 \leq k < m$, and ω appears for the first time as the last action. Let $e_m = \omega$, then the probability of each set $[\sigma]$ is given by

$$\Pr([\sigma]) = \prod_{k=0}^{m} \left(\prod_{i=0}^{n_k-1} \mu_T(\sigma(k,i) \xrightarrow{\tau_{k,i}} \sigma(k,i+1)) \right) \cdot \mu_T(\sigma(k,n_k), e_k, \sigma(k+1,0)).$$

Thus, the success probability of T can be expressed as

$$\sum_{\alpha \in \bar{\Omega}} \sum_{\sigma \in \mathrm{Ext}^T_{\alpha\omega}} \prod_{k=0}^{m} \left(\prod_{i=0}^{n_k-1} \mu_T(\sigma(k,i) \xrightarrow{\tau_{k,i}} \sigma(k,i+1)) \right) \cdot \mu_T(\sigma(k,n_k), e_k, \sigma(k+1,0)).$$

The above equation is just

$$\sum_{\alpha \in \bar{\Omega}} \mathcal{E}^T_{\alpha\omega}[1],$$

as was to be shown. ∎

Let a set of actions $E = E_0 \cup E_1$ be fixed, with E_0 and E_1 disjoint. For each external trace $\alpha = e_0 e_1 \cdots e_{m-1}$ and each sequence $\mathbf{x} = x_0 x_1 \cdots x_{m-1}$ of positive real numbers, test $T_{\alpha,\mathbf{x}}$ is defined as in Section 5.

Lemma 6.2 *Suppose A is a delay-restricted probabilistic I/O automaton. Then for each external trace $\alpha = e_0 e_1 \ldots e_{m-1} \in (E^{\mathrm{ext}}_A)^*$, and for each sequence $\mathbf{x} = x_0, x_1, \ldots, x_m$ of positive real numbers, the test $T_{\alpha,\mathbf{x}}$ (with $E_0 = E^{\mathrm{in}}_A$ and $E_1 = E^{\mathrm{out}}_A$) is a proper test for A. Moreover, the success probability of $A|T_{\alpha,\mathbf{x}}$ is given by:*

$$2^{-c} \cdot \mathcal{E}^A_{\alpha\omega}[\prod_{k=0}^{m} \frac{y_k \cdot (D^A_k)^{N^A_k}}{(x_k + D^A_k)^{N^A_k+1}}],$$

where for all $0 \le k < m$ *we have*

$$y_k = \begin{cases} D_k^A, & \text{if } e_k \in E_A^{\text{out}} \\ x_k & \text{otherwise,} \end{cases}$$

$y_m = x_m$*, and* c *is the number of* $k \in \{0, 1, \dots, m-1\}$ *for which* $e_k \in E_A^{\text{in}}$*.*

Proof – Fix α and $\mathbf{x}$, and let T abbreviate $T_{\alpha,\mathbf{x}}$. The proof that T is a proper test for A is the same as in Lemma 5.2.

By Lemma 6.1, the success probability of $A|T$ is given by:

$$\sum_{\alpha' \in \bar{\Omega}} \mathcal{E}_{\alpha'\omega}^{A|T}[1],$$

where $\bar{\Omega}$ is the set of all traces that do not contain ω. By Theorem 3, this may be rewritten as follows:

$$\sum_{\alpha' \in \bar{\Omega}} \mathcal{E}_{\alpha'\omega}^{A}[\mathcal{E}_{\alpha'\omega}^{T}[h((\mathbf{D}^A, \mathbf{N}^A), (\mathbf{D}^T, \mathbf{N}^T))]\,],$$

where

$$\begin{aligned} h((\mathbf{D}^A, \mathbf{N}^A), (\mathbf{D}^T, \mathbf{N}^T)) \;\;=\;\; & \left(\prod_{k=0}^{m} \binom{N_k^A + N_k^T}{N_k^A} (\frac{D_k^A}{D_k^A + D_k^T})^{N_k^A} (\frac{D_k^T}{D_k^A + D_k^T})^{N_k^T}\right) \\ & \left(\prod_{k \in K_A^{\text{out}}} \frac{D_k^A}{D_k^A + D_k^T}\right) \left(\prod_{k \in K_T^{\text{out}}} \frac{D_k^T}{D_k^A + D_k^T}\right) \end{aligned}$$

and index sets K_A^{out} and K_T^{out} contain the indices of actions in $\alpha'\omega$ that are output actions of A and T respectively.

Now, observe that T has just one execution that produces a trace of the form $\alpha'\omega$, namely

$$\sigma = 0 \xrightarrow{e_0} 1 \xrightarrow{e_1} 2 \xrightarrow{e_2} \dots \xrightarrow{e_{m-1}} m \xrightarrow{\omega} m+1,$$

which produces trace $\alpha\omega$. From this it is easy to see that $\mathcal{E}_{\alpha'\omega}^{T}[h((\mathbf{D}^A, \mathbf{N}^A), (\mathbf{D}^T, \mathbf{N}^T))] = 0$ unless $\alpha' = \alpha$, and since the delay sequence of σ is $x_0 x_1 \cdots x_m$, we have

$$\mathcal{E}_{\alpha\omega}^{T}[h((\mathbf{D}^A, \mathbf{N}^A), (\mathbf{D}^T, \mathbf{N}^T))] = 2^{-c} \left(\prod_{k=0}^{m} (\frac{D_k^A}{D_k^A + x_k})^{N_k^A}\right) \left(\prod_{k \in K_A^{\text{out}}} \frac{D_k^A}{D_k^A + x_k}\right) \left(\prod_{k \in K_T^{\text{out}}} \frac{x_k}{D_k^A + x_k}\right)$$

where c is the number of $k \in \{0, 1, \dots, m-1\}$ such that $e_k \in E_A^{\text{in}}$. Using this information and the fact that each e_k is an output action of either A or T, the success probability of $A|T$ may be rewritten as:

$$2^{-c} \cdot \mathcal{E}_{\alpha\omega}^{A}[\prod_{k=0}^{m} \frac{y_k \cdot (D_k^A)^{N_k^A}}{(x_k + D_k^A)^{N_k^A + 1}}],$$

where for all $0 \leq k < m$ we have

$$y_k = \begin{cases} D_k^A, & \text{if } e_k \in E_A^{\text{out}} \\ x_k & \text{otherwise,} \end{cases}$$

and $y_m = x_m$. ∎

Theorem 4 *Suppose A and B are delay-restricted probabilistic I/O automata with the same set of actions. Then A and B are testing equivalent if and only if the associated probabilistic behavior maps $\mathcal{E}^A$ and $\mathcal{E}^B$ are equal.*

Proof Sketch – The proof is similar to that of Theorem 2, using Theorem 3 and Lemma 6.2 instead of Theorem 1 and Lemma 5.2. Also, Lemma 5.3 is applied to rational functions with nonlinear denominators; in the proof of Theorem 2, Lemma 5.3 was applied to rational functions with linear denominators. ∎

7 Summary and Conclusion

We have presented a framework in which probability can be added to I/O automata. To capture the asymmetric treatment of input and output indigenous to I/O automata, a separate distribution is associated with each input action, in the reactive style, and a single distribution is associated with all locally controlled actions, in the generative style. No relative probabilities are defined among different input actions nor between input and locally controlled actions. Moreover, the pleasant notion of I/O automaton asynchronous composition is retained, in part, through the introduction of state delay parameters. Delay parameters admit a natural probabilistic description of the outcome of the competition between automata vying for control of the next action.

As is the practice with ordinary I/O automata, we introduced a more abstract representation, *probabilistic behavior maps*, for the external behaviors of certain classes of probabilistic I/O automata. This representation maps finite action sequences to a set of expectation functionals which give information not only about the probabilities of action sequences but also delay sequences. This latter information is essential for achieving compositionality. We also showed that probabilistic behavior maps are fully abstract with respect to a natural notion of probabilistic testing.

As future work, we would like to extend the entire setup to handle time as well as probability. The presence of the state delay function in our model provides a convenient mechanism on which to base this work. Another interesting research direction concerns simulation relations for probabilistic I/O automata, in the style of [LT87, LV91, SL94].

To conclude, we would like to comment on the issue of whether the model we have defined is "realistic" in the sense that it could be used in the design and analysis of actual systems. One might question, for example, our assumption that component automata experience *independent* delays. However, this assumption reflects our view of a concurrent system as a collection of autonomous component automata, each of which executes its locally controlled actions without interference from another component.

Another potential source of objection to the model might be the assumption, underlying the definition of composition, that the delay time in a state is exponentially distributed. However, we would argue as follows: there are at least two distinct reasons for defining a model of probabilistic systems like the one we have studied here. First, one might construct the model such that it accurately models a pre-existing class of systems. Certainly in this case one should be very concerned about whether the assumption of exponential delays is reasonable. However, a second reason for constructing a model would be as a theoretical tool with which to design and build real systems. In this case, one is not concerned with whether the model describes a pre-existing class of systems, but rather with whether systems can be engineered whose behavior closely matches that predicted by the model. Clearly, it would be possible, by introducing exponentially distributed random delays between each step of a program, to simulate arbitrarily closely the type of probabilistic behavior modeled by our probabilistic I/O automata. The price paid for a close match between the actual behavior of a system implemented in this way and the theoretical behavior predicted by the model would be a slowdown in performance introduced by the delays. In return for this loss of performance, however, one would receive the ability to make quantitative statements about the probability of various kinds of system behavior. Having this ability is important to many applications.

Acknowledgment: We thank the anonymous referees for their helpful comments.

References

[Chr90] I. Christoff. Testing equivalences and fully abstract models for probabilistic processes. In J. C. M. Baeten and J. W. Klop, editors, *Proceedings of CONCUR '90 – First International Conference on Concurrency Theory*, Lecture Notes in Computer Science, Volume 458, pages 126–140. Springer-Verlag, 1990.

[CSZ92] R. Cleaveland, S. A. Smolka, and A. E. Zwarico. Testing preorders for probabilistic processes. In *Proceedings of the 19th ICALP*, July 1992.

[dNH83] R. de Nicola and M. C. B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1983.

[GHR92] N. Götz, U. Herzog, and M. Rettlebach. TIPP — a language for timed processes and performance evaluation. Technical Report 4/92, University of Erlangen-Nürnberg, Germany, November 1992.

[Han91] H. A. Hansson. *Time and Probability in Formal Design of Distributed Systems.* PhD thesis, Department of Computer Systems, Uppsala University, 1991.

[Hen88] M. C. B. Hennessy. *Algebraic Theory of Processes.* MIT Press, Boston, Mass., 1988.

[Hil93] J. Hillston. PEPA: Performance enhanced process algebra. Technical Report CSR-24-93, Department of Computer Science, University of Edinburgh, Edinburgh, Great Britain, March 1993.

[HJ90] H. A. Hansson and B. Jonsson. A calculus for communicating systems with time and probabilities. In *Proceedings of the 11th IEEE Symposium on Real-Time Systems*, 1990.

[JL91] B. Jonsson and K. G. Larsen. Specification and refinement of probabilistic processes. In *Proceedings of the 6th IEEE Symposium on Logic in Computer Science*, Amsterdam, July 1991.

[Lan90] S. Lang. *Undergraduate Algebra.* Springer-Verlag, New York, 1990.

[LS92] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, September 1992. Preliminary versions of this paper appeared as University of Aalborg technical reports R 88-18 and R 88-29, and in *Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages*, Austin, Texas, 1989.

[LT87] N. A. Lynch and M. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing*, 1987.

[LV91] N. A. Lynch and F. W. Vaandrager. Forward and backward simulations for timing-based systems. In *Proceedings of the REX Workshop "Real-Time: Theory in Practice"*, Lecture Notes in Computer Science, Volume 600, pages 397–446. Springer-Verlag, 1991.

[Mil89] R. Milner. *Communication and Concurrency.* International Series in Computer Science. Prentice Hall, 1989.

[Mol82] M. K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Trans. Comput.*, C-31(9), September 1982.

[Rab63] M. O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.

[Sei92] K. Seidel. *Probabilistic CSP.* PhD thesis, Technical Monograph PRG-102, Programming Research Group, Oxford University Computing laboratory, 1992.

[SL94] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In B. Jonsson and J. Parrow, editors, *Proceedings of CONCUR '94 – Fifth International Conference on Concurrency Theory*, Lecture Notes in Computer Science, Volume 836, pages 481–496. Springer-Verlag, 1994.

[Tri82] K. S. Trivedi. *Probability & Statistics with Reliability, Queuing, and Computer Science Applications.* Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[Tut87] M. Tuttle. Hierarchical correctness proofs for distributed algorithms. Master's thesis, MIT, April 1987.

[vGSST90] R. J. van Glabbeek, S. A. Smolka, B. Steffen, and C. M. N. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Proceedings of the 5th IEEE Symposium on Logic in Computer Science*, pages 130–141, Philadelphia, PA, 1990. Extended version to appear in *Information and Computation.*

[Wei74] A. J. Weir. *General Integration and Measure.* Cambridge University Press, 1974.

[YCDS94] S. Yuen, R. Cleaveland, Z. Dayar, and S. A. Smolka. Fully abstract characterizations of testing preorders for probabilistic processes. In B. Jonsson and J. Parrow, editors, *Proceedings of CONCUR '94 – Fifth International Conference on Concurrency Theory*, Lecture Notes in Computer Science, Volume 836, pages 497–512. Springer-Verlag, 1994.

Appendix

A Proof of Lemma 5.3

Lemma A.1 is a uniqueness theorem for partial fraction expansions of rational functions. It is used in the proof of Lemma 5.3.

Lemma A.1 *Suppose f and f' are two rational functions of variable x defined as follows:*

$$f = \sum_{i \in I} \frac{a_i}{(x+b_i)^{r_i}} \qquad\qquad f' = \sum_{i \in I'} \frac{c_i}{(x+d_i)^{s_i}},$$

where I and I' are finite sets, for each $i \in I$, the exponent r_i is a positive integer and $a_i \in (0,\infty)$, for each $i \in I'$, the exponent s_i is a positive integer and $c_i \in (0,\infty)$, for each distinct $i,j \in I$ the pairs (b_i, r_i) and (b_j, r_j) are distinct, and for each distinct $i,j \in I'$ the pairs (d_i, s_i) and (d_j, s_j) are distinct. If $f = f'$, then there exists a bijection $(\text{ - })' : I \to I'$ such that $a_i = c_{i'}$, $b_i = d_{i'}$ and $r_i = s_{i'}$ for all $i \in I$.

Proof – The equivalence relation $\sim$ on I, defined by $i \sim j$ iff $b_i = b_j$, induces a partition of I. Let $I_1, I_2, \cdots, I_m$ denote the equivalence classes. Let b_l denote the common value of b_i for $i \in I_l$ and let $r_l = \sum_{i \in I_l} r_i$ denote the sum of all r_i for $i \in I_l$. We may then write

$$f = \sum_{l=1}^{m} \frac{g_l}{(x+b_l)^{r_l}} \qquad \text{where} \qquad g_l = \sum_{i \in I_l} a_i (x+b_l)^{(r_l - r_i)}.$$

Similarly, we may write

$$f' = \sum_{l=1}^{m'} \frac{g'_l}{(x+d_l)^{s_l}} \qquad \text{where} \qquad g'_l = \sum_{i \in I'_l} c_i (x+d_l)^{(s_l - s_i)}.$$

Since by construction, the b_l are all distinct for $1 \leq l \leq m$ and g_l is a polynomial of degree less than r_l, similar for d_l and g'_l for $1 \leq l \leq m'$, the above expressions amount to *partial fraction expansions* of f and f'. If $f = f'$, then by the uniqueness of partial fraction expansions [Lan90], we conclude that $m = m'$, and by choosing appropriately the order of the terms in the summations we may assume that $b_l = d_l$ and the polynomials g_l and g'_l are equal for $1 \leq l \leq m$. Now,

$$g_l = \sum_{i \in I_l} a_i (x+b_l)^{(r_l - r_i)} \qquad \text{and} \qquad g'_l = \sum_{i \in I'_l} c_i (x+d_l)^{(s_l - s_i)}$$

where I_l and I'_l are finite sets, $a_i \in (0,\infty)$ for $i \in I_l$, $c_i \in (0,\infty)$ for $i \in I'_l$, for each distinct $i,j \in I_l$ we have $r_i \neq r_j$, and for each distinct $i,j \in I'_l$ we have $s_i \neq s_j$. The polynomials g_l and g'_l may thus be regarded as polynomials in the quantity $y = x + b_l$ $(= x + d_l)$, with positive coefficients a_i and c_i, and with each term having a distinct exponent applied to y. Since g_l and g'_l are equal, we may conclude the equality of the coefficients of corresponding terms; thus, there exists a bijection $(\text{ - })' : I_l \to I'_l$ such that $a_i = c_{i'}$ and $r_i = s_{i'}$ for all $i \in I_l$.

Taking the union of the bijections $(\text{ - })' : I_l \to I'_l$ for $1 \leq l \leq m$, yields the required bijection $(\text{ - })' : I \to I'$, completing the proof. ∎

Lemma 5.3 *Suppose f and f' are two rational functions of variables $x_0, x_1, \ldots, x_{n-1}$ ($n \geq 0$), defined as follows:*

$$f = \sum_{i \in I} \frac{a_i}{\prod_{k=0}^{n-1} (x_k + b_{i,k})^{r_{i,k}}} \qquad\qquad f' = \sum_{i \in I'} \frac{c_i}{\prod_{k=0}^{n-1} (x_k + d_{i,k})^{s_{i,k}}},$$

where I and I' are finite sets, for each $i \in I$ and $0 \leq k < n$, the exponent $r_{i,k}$ is a positive integer and $a_i \in (0,\infty)$, for each $i \in I'$ and $0 \leq k < n$, the exponent $s_{i,k}$ is a positive integer and $c_i \in (0,\infty)$, for each distinct $i,j \in I$ the sets $\{(k, b_{i,k}, r_{i,k}) : 0 \leq k < n\}$ and $\{(k, b_{j,k}, r_{j,k}) : 0 \leq k < n\}$ are distinct, and for each distinct $i,j \in I'$ the sets $\{(k, d_{i,k}, s_{i,k}) : 0 \leq k < n\}$ and $\{(k, d_{j,k}, s_{j,k}) : 0 \leq k < n\}$ are distinct. If $f = f'$, then there exists a bijection $(\text{ - })' : I \to I'$ such that $a_i = c_{i'}$, $b_{i,k} = d_{i',k}$ and $r_{i,k} = s_{i',k}$ for all $i \in I$ and $0 \leq k < n$.

Proof – The proof proceeds by induction on n. In case $n = 0$, then the sets $\{(k, b_{i,k}, r_{i,k}) : 0 \leq k < n\}$ are empty for all $i \in I$. In view of the assumption that the distinctness of $i,j \in I$ implies the distinctness of the sets $\{(k, b_{i,k}, r_{i,k}) : 0 \leq k < n\}$ and $\{(k, b_{j,k}, r_{j,k}) : 0 \leq k < n\}$, it follows that I can contain at most one element. Similar reasoning shows that I' also contains at most one element. If $I = \emptyset$, then $f = 0$, so if $f' = f$, then we must have $I' = \emptyset$ as well. In this case the trivial bijection from I to I' has the required properties. Suppose I has exactly one element $*$. Then $f = a_*$, so if $f' = f$ we know that I' also has exactly one element $*'$ and $f' = c_{*'}$. In this case, we may take as our bijection the map $(\text{ - })' : I \to I'$ that takes $*$ to $*'$.

Now suppose the result has been shown for n, and consider the situation for $n+1$. We may write

$$f = \sum_{i \in I} \frac{g_i}{(x_n + b_{i,n})^{r_{i,n}}},$$

where

$$g_i = \frac{a_i}{\prod_{k=0}^{n-1} (x_k + b_{i,k})^{r_{i,k}}}.$$

The equivalence relation $\sim$ on I, defined by $i \sim j$ iff $(b_{i,n}, r_{i,n}) = (b_{j,n}, r_{j,n})$, induces a partition of I. Let $I_1, I_2, \ldots, I_m$ denote the equivalence classes, and let b_l, r_l denote the the common values of $b_{i,n}$, $r_{i,n}$ for $i \in I_l$. We may then write

$$f = \sum_{l=1}^{m} \frac{\sum_{i \in I_l} g_i}{(x_n + b_l)^{r_l}}.$$

Similarly, we may write

$$f' = \sum_{l=1}^{m'} \frac{\sum_{i \in I'_l} g'_i}{(x_n + d_l)^{s_l}}.$$

Since by construction, the pairs (b_l, r_l) are all distinct for $1 \le l \le m$, and the pairs (d_l, s_l) are all distinct for $1 \le l \le m'$, the above expressions amount to the rational functions in Lemma A.1, viewed as rational functions of x_n with $x_1, \ldots, x_{n-1}$ held constant. If $f = f'$, then by Lemma A.1, we conclude that $m = m'$, and by choosing appropriately the order of the terms in the summations we may assume that $b_l = d_l$, $r_l = s_l$ for $1 \le l \le m$, and the rational functions $g_l = \sum_{i \in I_l} g_i$ and $g'_l = \sum_{i \in I'_l} g'_i$ are equal for $1 \le l \le m$.

Now,

$$g_l = \sum_{i \in I_l} \frac{a_i}{\prod_{k=0}^{n-1} (x_k + b_{i,k})^{r_{i,k}}} \qquad \text{and} \qquad g'_l = \sum_{i \in I'_l} \frac{c_i}{\prod_{k=0}^{n-1} (x_k + d_{i,k})^{s_{i,k}}},$$

where I_l and I'_l are finite sets, $a_i \in (0, \infty)$ for $i \in I_l$, and $c_i \in (0, \infty)$ for $i \in I'_l$. Moreover, by the assumption that the sets $\{(k, b_{i,k}, r_{i,k}) : 0 \le k < n+1\}$ and $\{(k, b_{j,k}, r_{j,k}) : 0 \le k < n+1\}$ are distinct for $i, j \in I$ and the fact that $(b_{i,n}, r_{i,n}) = (b_{j,n}, r_{j,n})$ for $i, j \in I_l$, we may conclude that the sets $\{(k, b_{i,k}, r_{i,k}) : 0 \le k < n\}$ and $\{(k, b_{j,k}, r_{j,k}) : 0 \le k < n\}$ are distinct for $i, j \in I_l$. Similar reasoning shows that the sets $\{(k, d_{i,k}, s_{i,k}) : 0 \le k < n\}$ and $\{(k, d_{j,k}, s_{j,k}) : 0 \le k < n\}$ are distinct for $i, j \in I'_l$. Thus, for each l, we may apply the induction hypothesis to g_l and g'_l, to conclude the existence of a bijection $(\text{ - })' : I_l \to I'_l$ such that $a_i = c_{i'}$ and $b_{i,k} = d_{i',k}$ for all $i \in I$ and all $0 \le k < n$.

Taking the union of the bijections $(\text{ - })' : I_l \to I'_l$, for $1 \le l \le m$, yields the required bijection $(\text{ - })' : I \to I'$, completing the induction step and the proof. ∎