Journal of Automata, Languages and Combinatorics **u** (v) w, x–y
© Otto-von-Guericke-Universität Magdeburg

ON BEHAVIOR EQUIVALENCE FOR PROBABILISTIC I/O AUTOMATA AND ITS RELATIONSHIP TO PROBABILISTIC BISIMULATION [1]

Eugene W. Stark
Dept. of Computer Science, State University of New York at Stony Brook
Stony Brook, NY 11794-4400 USA
e-mail: `stark@cs.sunysb.edu`

ABSTRACT

Previous work of the author has developed *probabilistic input/output automata* (PIOA) as a formalism for modeling systems that exhibit concurrent and probabilistic behavior. Central to that work was the notion of the "behavior map" associated with a state of a PIOA. The present paper presents a new, simpler definition for PIOA behavior maps, investigates the induced "same behavior map" equivalence relation, and compares it with the standard notion of probabilistic bisimulation equivalence. *Weighted finite automata* are used as a unifying formalism to facilitate the comparison. A general notion of congruence for weighted automata is defined, which relates *signed measures* on states, rather than just individual states. PIOA are defined as a class of weighted automata, as are the class of *probabilistic weighted automata* for which the standard definition of probabilistic bisimulation makes sense. A characterization is obtained of probabilistic bisimulation as the largest congruence that is in a sense generated by its restriction to a relation on states. This characterization is then used as the definition of *weighted bisimulation*, which generalizes probabilistic bisimulation to the full class of weighted automata. PIOA behavior equivalence is also shown to define a weighted automata congruence, which is strictly refined by weighted bisimulation equivalence. The relationship between these congruences and a notion of *composition* for weighted automata is also examined.

Keywords: probabilistic I/O automata, weighted automata, continuous-time Markov chains, probabilistic bisimulation, lumpability

1. Introduction

In previous work [22] we introduced *probabilistic I/O automata* (PIOA) as a formal model for systems that exhibit concurrent and probabilistic behavior. An important feature of the PIOA model is that it admits a *composition operation* by which a complex automaton can be constructed from simpler components. In [19, 20] we

[1]This research was supported in part by the National Science Foundation under Grant CCR-9988155 and the Army Research Office under Grants DAAD190110003 and DAAD190110019. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, the Army Research Office, or other sponsors.

presented algorithms for calculating certain kinds of performance parameters (such as mean time to failure) for such systems in a *compositional* fashion; that is, by treating the components of a composite system in succession rather than all at once. The compositional approach can help avoid the state-space explosion problem in such calculations by affording the opportunity to perform state-space reduction as each successive component is treated.

A central role in our previous work was played by so-called *behavior maps*, which are certain functions associated with the states of a PIOA. In [22], the assignment of behavior maps to states was shown to *respect composition* in the sense that the behavior map associated with a state of a composite PIOA is determined completely by the behavior maps associated with its component states. In addition, behavior maps were shown (for PIOA without internal actions) to be *fully abstract* with respect to a natural notion of probabilistic testing. That is, two states of a PIOA are indistinguishable by probabilistic testing precisely when they have the same associated behavior maps. Behavior maps also form the basis for the algorithms presented in [19, 20]. These algorithms work by starting with a so-called *observable*, which describes the performance parameter to be computed, and then successively applying the behavior map associated with each component of the system to obtain a new observable. Once all components have been treated, the final answer is extracted.

Although we felt that our notion of behavior map was justified in view of the results we were able to prove about it, the formal definitions we gave for this concept are messy and difficult to motivate. In addition, we did not have any understanding of how the induced "behavior equivalence" on PIOA states might relate to previously studied equivalences, such as probabilistic bisimulation [14], for probabilistic systems. The present paper attempts to rectify this situation: firstly by giving a much simpler definition for PIOA behavior maps than in our previous work, and secondly by studying the relationship of behavior equivalence for PIOA with probabilistic bisimulation equivalence.

Probabilistic bisimulation, as introduced by Larsen and Skou [14], and as studied further in [10, 11], does not apply directly to probabilistic I/O automata but rather to a somewhat different model, called *probabilistic transition systems*. In order to make a comparison of PIOA behavior equivalence and probabilistic bisimulation, we need a common framework within which both notions of equivalence make sense. In this paper, we use *weighted automata* for this purpose. For us, a weighted automaton consists of a finite set E of actions, a finite set Q of states, and for each $e \in E$ a linear operator T_e on Q-indexed vectors of real numbers. Each T_e can be represented by a matrix whose rows and colums are indexed by Q, and whose entries are "weights" assigned to state transitions. Both PIOA and "deterministic" probabilistic transition systems can be viewed as special cases of weighted automata, though the weights have a different interpretation in each case.

The class of weighted automata we consider in this paper is a rather restricted subclass of the most general kind of weighted automata that have been considered in the literature. A general definition of weighted automata (see *e.g.* [13]) permits the weights to be drawn from an arbitrary semiring, and indeed this generality is exploited in important application areas: the boolean semiring in the case of formal language theory, as well as more "exotic" semirings such as "max-plus" and its relatives in the case of optimization and scheduling applications. Here we are concerned with weights that represent either transition probabilities or transition weights, and for this reason we restrict our attention to weights drawn from the semiring of nonnegative real numbers with the usual addition and multiplication, or more conveniently, from the ring of all real numbers. For us, the use of weighted automata serves to emphasize the natural linear algebraic structure inherent in probabilistic and stochastic automata, and motivates an attempt to characterize important equivalences, such probabilistic bisimulation, as equivalences that respect this structure.

We begin our investigation by defining a notion of *congruence* for weighted automata. A congruence on any algebraic structure consists of an equivalence relation on the elements of the structure which respects the algebraic operations. In the case of weighted automata, the relevant algebraic operations are the transition maps, which naturally act not on individual elements of the state set Q, but rather on Q-indexed vectors of real numbers. These vectors can be thought of as "weightings" or "measures" on states. The transition maps are linear, which implies that the vector space structure also becomes relevant for the notion of congruence. Thus, we define a *congruence* for a weighted automaton A to be an equivalence relation on measures that has a certain *linearity* property and in addition is *invariant* under the transition maps T_e of A. Equivalence relations on individual states lift naturally to linear equivalences via the identification of an individual element q of Q with the "point measure" δ_q that assigns weight 1 to q and weight 0 to all other states.

After introducing the notion of weighted automata congruence, we next consider the class of *probabilistic weighted automata*, defined to be those weighted automata whose transition matrices have nonnegative entries and satisfy a certain stochastic condition. Probabilistic weighted automata amount to the the same thing as *deterministic* or *reactive* probabilistic transition systems [6, 14]. We consider how the notion of probabilistic bisimulation for such automata relates to the notion of congruence for weighted automata. Our main result here is a correspondence between probabilistic bisimulations and those congruences that are in a sense generated by an underlying equivalence relation on states. Although the original definition of probabilistic bisimulation involves probability distributions, and would therefore seem only to be applicable to probabilistic automata, our description of probabilistic bisimulations in terms of congruences indicates that the concept can be generalized sensibly to arbitrary weighted automata with nonnegative transition matrices. We use the term *weighted bisimulation* to refer to this generalization.

We next define probabilistic I/O automata as another class of weighted automata. PIOA have two types of actions: *input* actions and *output* actions. The weights associated with the input transitions of a PIOA are interpreted as probabilities and the transition matrix T_e associated with an input action e is required to satisfy a stochastic condition. In contrast, the output transitions of a PIOA describe a *continuous-time Markov chain*, in which the entry $T_e(q, q')$ is the parameter of an exponential probability distribution that describes the random amount of time that will be spent in state q before performing an action e and changing to state q'. The weights associated with the output transitions of a PIOA are interpreted as transition *rates*, not probabilities, and we therefore require that the transition matrix T_e associated with an output action e be nonnegative, but do not impose any stochastic condition.

After giving the definition of PIOA, we present a simple definition of the *behavior map* associated with a state q of a PIOA. We show that the "behavior equivalence" relation on a PIOA determines a congruence on the underlying weighted automaton. We compare the congruence that arises in this way with the congruence associated with the largest weighted bisimulation, and we find that weighted bisimulation equivalence strictly refines behavior equivalence. The intuitive reason for this is that whereas probabilistic bisimulation equivalence is in a sense generated by its restriction to point measures, behavior equivalence can include relationships between arbitrary measures that are not consequences of relationships between point measures. This is a key point of our work. If one limits consideration to congruences generated by underlying equivalences on individual states, then probabilistic bisimulation is the congruence that yields the maximum identifications between states. On the other hand, if one is willing to consider equivalences on measures which are not generated by underlying state equivalences, then one opens the possibility of congruences that relate point measures (*i.e.* states) with more general measures. Our PIOA behavior equivalence relation is an example of such a congruence, which is strictly coarser than probabilistic bisimulation, yet still separates states that can be distinguished on the basis of the probability distributions governing the production of outputs in those states.

Besides relating PIOA behavior equivalence and probabilistic bisimulation equivalence, we also consider the relationship between these equivalences and the notion of composition. We give a general definition for the composition $A_1 * A_2$ of weighted automata A_1 and A_2. This composition amounts to taking a kind of product of A_1 and A_2 where the actions in the intersection of the alphabets are required to synchronize and actions in the symmetric difference occur independently. This composition operation restricts sensibly to both probabilistic weighted automata and PIOA, in the sense that the classes of probabilistic weighted automata and PIOA are both closed under it. Moreover, for *compatible* PIOA A_1 and A_2, the composition $A_1 * A_2$ corresponds exactly to the PIOA composition used in our previous work. We show that composition respects both probabilistic bisimulation and PIOA behavior equivalence, and for PIOA behavior equivalence we give a simple new proof of this fact based on our simpler definition of PIOA behavior.

The remainder of the paper is organized as follows. Section 2 gives some preliminary definitions and terminology. Section 3 defines weighted automata and congruences on such automata. Section 4 defines the subclass of probabilistic weighted automata and characterizes probabilistic bisimulation among the congruences on such automata. Section 5 defines probabilistic I/O automata and behavior maps and characterizes PIOA behavior equivalence. Finally, Section 6 summarizes what has been achieved and indicates a direction for future research.

2. Preliminaries

A *signed measure* on a finite set Q is a real-valued function μ on Q. A signed measure on Q is called *nonnegative* if it takes on only nonnegative values. In the sequel, we use the unqualified term "measure" to mean signed measure, and we use the term "nonnegative measure" when we wish to restrict our attention to that case. We use R^Q to denote the set of all signed measures on Q. Note that R^Q is a finite-dimensional vector space with obvious pointwise definitions of sum and scalar product. For each state $q \in Q$, define the *point measure* δ_q by the conditions $\delta_q(q) = 1$ and $\delta_q(q') = 0$ if $q' \neq q$. When no confusion can result, we shall find it convenient to identify the elements of Q with the corresponding point measures in R^Q. Note that the set of point measures $\{\delta_q : q \in Q\}$ constitutes a basis for R^Q. We use the abbreviation $\mu(S)$ to denote the sum $\sum_{q \in S} \mu(q)$ for $S \subseteq Q$. The *support* of μ is defined to be the set $\mathrm{supp}(\mu) = \{q \in Q : \mu(q) \neq 0\}$.

Suppose $T : \mathrm{R}^Q \to \mathrm{R}^Q$ is a linear operator. We regard T as acting on the right, and hence use the notation μT for the result of applying T to μ. Each such operator T determines a function, which we also denote by T, from $Q \times Q$ to R, according to the definition

$$T(q, q') = (qT)(q').$$

We then have the familiar matrix multiplication formula

$$(\mu T)(q') = \sum_{q \in Q} \mu(q) T(q, q').$$

We say that T is *nonnegative* if it preserves nonnegative measures. This is equivalent to the condition that all $T(q, q')$ are nonnegative.

A *formal power series* over a set E is a function $\Phi : E^* \to \mathrm{R}$. where E^* denotes the set of finite words over E. Let $\mathrm{R}\langle\langle E\rangle\rangle$ denote the set of all formal power series over E. Note that $\mathrm{R}\langle\langle E\rangle\rangle$ is a vector space with the obvious pointwise sum and scalar product. The *derivative* of a formal power series Φ by an element $e \in E$ is the formal power series $e^{-1}\Phi$ that satisfies

$$(e^{-1}\Phi)(w) = \Phi(ew).$$

for all $w \in E^*$. Note that for each $e \in E$, the map taking a formal power series Φ to its derivative $e^{-1}\Phi$ is a linear operator on $\mathrm{R}\langle\langle E\rangle\rangle$.

3. Weighted Automata

A *weighted automaton* (WA) is a triple $A = (E, Q, \{T_e : e \in E\})$ where

- E is a finite set of *actions*.
- Q is a finite set of *states*.
- $T_e : \mathrm{R}^Q \to \mathrm{R}^Q$ is a linear map, called the *transition map* for action $e \in E$.

Note that the map taking $e \in E$ to the linear operator T_e on R^Q extends uniquely to a morphism of monoids:

$$T : E^* \to \mathrm{Lin}(\mathrm{R}^Q, \mathrm{R}^Q)$$

which assigns to each word $w = e_1 e_2 \ldots e_n$ in E^* the linear operator $T_\alpha = T_{e_1} T_{e_2} \ldots T_{e_n}$ on R^Q. (Here $\mathrm{Lin}(\mathrm{R}^Q, \mathrm{R}^Q)$ denotes the monoid of linear operators on R^Q.)

In this paper, we restrict our attention to weighted automata with finite sets of states. There are some significant complications that arise if one attempts to treat countable sets of states, due to the necessity of introducing a suitable notion of convergence for R^Q and keeping track of the conditions under which the various summations we use in fact converge. We do not know to what extent our results might generalize to infinite state sets.

The reader will also have noted that our definition of weighted automaton does not include any initial conditions such as a distinguished initial state. The separation of the dynamics of an automaton from the initial conditions allows us to focus our attention for the moment on notions (such as congruences) that are independent of the initial conditions. Of course, if we are interested in particular behavior of an automaton then we must specify particular initial conditions, but even so it is often useful to consider the behavior of a single automaton under a variety of initial conditions. Keeping the initial conditions separate from the automaton helps to facilitate this.

We now turn to defining the notion of congruence for weighted automata. Define a relation $\mathcal{R}$ on R^Q to be *linear* if the following conditions are satisfied:

- $\mu \; \mathcal{R} \; \mu'$ implies $c\mu \; \mathcal{R} \; c\mu'$.
- $\mu_1 \; \mathcal{R} \; \mu'_1$ and $\mu_2 \; \mathcal{R} \; \mu'_2$ imply $\mu_1 + \mu_2 \; \mathcal{R} \; \mu'_1 + \mu'_2$.

A *congruence* on a weighted automaton $A = (E, Q, \{T_e : e \in E\})$ is a linear equivalence relation $\mathcal{E}$ on R^Q which is *A-invariant* in the following sense:

- $\mu \; \mathcal{E} \; \mu'$ implies $\mu T_e \; \mathcal{E} \; \mu' T_e$ for all actions e.

As mentioned in the introduction, our usage of the term "congruence" here is consistent with the standard mathematical meaning of "an equivalence that respects relevant algebraic structure." The algebraic structure that is relevant for a weighted automaton consists of the transition maps T_e and the vector space structure on R^Q. However, those who study process algebra may be misled by an apparent conflict with the way the term "congruence" is used in that field. In fact, there is no conflict. Process algebra considers, besides the transition structure of an automaton, various additional algebraic operations such as those that combine automata A_1 and A_2 to form some kind of composite automaton $A_1 * A_2$. In process algebra the relevant algebraic structure consists, therefore, not only of the transitions between states, but also any additional algebraic operations that may be under consideration. One therefore expects a notion of congruence appropriate for process algebra to respect these additional algebraic operations on automata as well as the transition structure of individual automata. Indeed, the attempt to establish that a standard transition-respecting congruence such as bisimulation also respects the additional algebraic operations, forms a typical theme in process-algebraic investigations.

Define the *kernel* $K_{\mathcal{R}}$ of a binary relation $\mathcal{R}$ on R^Q to be the subspace of R^Q generated by the set $\{\mu - \mu' : \mu \; \mathcal{R} \; \mu'\}$. Then every element of $K_{\mathcal{R}}$ is a finite linear combination of measures of the form $\mu - \mu'$ where $\mu \; \mathcal{R} \; \mu'$. Define the *closure* $\mathcal{R}^\dagger$ of a binary relation $\mathcal{R}$ on R^Q by:

$$\mu \; \mathcal{R}^\dagger \; \mu' \text{ iff } \mu - \mu' \in K_{\mathcal{R}}.$$

A relation R on Q may be identified with the corresponding relation $\mathcal{R}$ on R^Q defined by

$$\mathcal{R} \;=\; \{(\delta_q, \delta_{q'}) : q \; R \; q'\}.$$

We use this correspondence to extend the notion of kernel and closure to relations on Q. Note, though, that the closure $R^\dagger$ of a relation R on Q is a relation on R^Q, not a relation on Q.

Lemma 3.1 *Let $\mathcal{R}$ be a binary relation on R^Q. Then $\mathcal{R}^\dagger$ is a linear equivalence relation, which is in fact the smallest such relation containing $\mathcal{R}$.*

Proof. It is straightforward to see from the definition that $\mathcal{R}^\dagger$ is a linear equivalence relation that contains $\mathcal{R}$. If $\mathcal{R}$ itself happens to be a linear equivalence, then $\{\mu - \mu' : \mu \; \mathcal{R} \; \mu'\}$ is already a subspace of R^Q, which must therefore equal $K_\mathcal{R}$. In this case we have that $\mathcal{R}^\dagger = \{(\mu, \mu') : \mu \; \mathcal{R} \; \mu'\} = \mathcal{R}$. Thus, if $\mathcal{R}$ is a linear equivalence then $\mathcal{R}^\dagger = \mathcal{R}$.

Now, suppose $\mathcal{E}$ is an an arbitrary linear equivalence that contains $\mathcal{R}$. Then clearly $K_\mathcal{R} \subseteq K_\mathcal{E}$, and consequently $\mathcal{R}^\dagger \subseteq \mathcal{E}^\dagger = \mathcal{E}$. Thus, $\mathcal{R}^\dagger$ is the smallest linear equivalence that contains $\mathcal{R}$. □

Lemma 3.2 *Let A be a weighted automaton with state set Q and action set E. A linear equivalence $\mathcal{E}$ on R^Q is a congruence for A if and only if $K_\mathcal{E}$ is closed under T_e for all $e \in E$.*

Proof. Suppose the linear equivalence $\mathcal{E}$ is a congruence on A, hence is A-invariant. Then $\mu \in K_\mathcal{E}$ iff $\mu \; \mathcal{E} \; 0$, so by the A-invariance of $\mathcal{E}$ it follows that $\mu \in K_\mathcal{E}$ implies $\mu T_e \; \mathcal{E} \; 0T_e = 0$; hence $K_\mathcal{E}$ is closed under T_e for all e.

Conversely, suppose $\mathcal{E}$ is a linear equivalence such that $K_\mathcal{E}$ is closed under T_e for all e. If $\mu \; \mathcal{E} \; \mu'$, then by definition $\mu - \mu' \in K_\mathcal{E}$. Since $K_\mathcal{E}$ is closed under T_e, it follows that $\mu T_e - \mu' T_e \in K_\mathcal{E}$, hence $\mu T_e \; \mathcal{E}^\dagger \; \mu' T_e$. But since $\mathcal{E}$ is linear and $\mathcal{E}^\dagger$ is the smallest linear equivalence containing $\mathcal{E}$ it follows that $\mathcal{E}^\dagger = \mathcal{E}$. Thus $\mu T_e \; \mathcal{E} \; \mu' T_e$, showing that $\mathcal{E}$ is A-invariant, hence a congruence on A. □

Example 1 Let $A = (E, Q, \{T_e : e \in E\})$, where

$$E = \{a, b, c, d\}, \qquad Q = \{q_0, q_1, q_1', q_2, q_2', q_3\} \cup \{r_0, r_1, r_2, r_2', r_3\},$$

and the transition matrices T_a, T_b, T_c, and T_d are defined so that the following are the only non-zero entries:

$$T_a(q_0, q_1) = 1/2 \qquad T_a(q_0, q_1') = 1/2 \qquad T_a(r_0, r_1) = 1$$

$$T_b(q_1, q_2) = 1 \qquad T_b(q_1', q_2') = 1 \qquad T_b(r_1, r_2) = 1/2 \qquad T_b(r_1, r_2') = 1/2$$

$$T_c(q_2, q_3) = 1 \qquad T_c(r_2, r_3) = 1$$

$$T_d(q_2', q_3) = 1 \qquad T_d(r_2', r_3) = 1.$$

A transition diagram for A is shown in Figure 1.

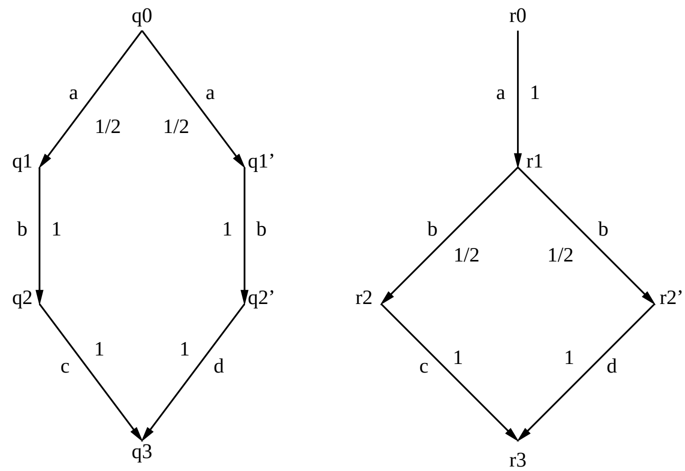

Figure 1: Transition Diagram for Example 1

Now, R^Q is an 11-dimensional space having as a basis the set:

$$\{\delta_{q_0}, \delta_{q_1}, \delta_{q_1'}, \delta_{q_2}, \delta_{q_2'}, \delta_{q_3}, \delta_{r_0}, \delta_{r_1}, \delta_{r_2}, \delta_{r_2'}, \delta_{r_3}\}$$

so that the elements of R^Q are all linear combinations of these basis elements. Let $\mathcal{K}$ be the subspace of R^Q generated by the following set:

$$\{\delta_{q_0} - \delta_{r_0},\ \ \frac{1}{2}\delta_{q_1} + \frac{1}{2}\delta_{q_1'} - \delta_{r_1},\ \ \delta_{q_2} - \delta_{r_2},\ \ \delta_{q_2'} - \delta_{r_2'},\ \ \delta_{q_3} - \delta_{r_3}\}.$$

It is easy to check that the above set is linearly independent, so that $\mathcal{K}$ is a 5-dimensional subspace of R^Q.

It is also easily checked by straightforward calculations that $\mathcal{K}$ is closed under T_e for all e. For example, we have:

$$(\delta_{q_0} - \delta_{r_0})T_a = \frac{1}{2}\delta_{q_1} + \frac{1}{2}\delta_{q_1'} - \delta_{r_1}$$

and

$$(\delta_{q_0} - \delta_{r_0})T_b = 0.$$

Let $\mathcal{E}$ be defined by $\mu\ \mathcal{E}\ \mu'$ iff $\mu - \mu' \in \mathcal{K}$; then $\mathcal{K} = K_{\mathcal{E}}$ and $\mathcal{E} = \mathcal{E}^{\dagger}$, so that $\mathcal{E}$ is a linear equivalence relation. Examples of elements of $\mathcal{E}$ are $(\delta_{q_0}, \delta_{r_0})$, $(\delta_{q_1} + \delta_{q_1'}, 2\delta_{r_1})$, and $(\delta_{q_2} - \delta_{r_2'}, \delta_{r_2} - \delta_{q_2'})$. Since $\mathcal{K}$ is closed under T_e for all e, it follows by Lemma 3.2 that $\mathcal{E}$ is a congruence on A.

We are interested in weighted automata congruences that respect certain basic attributes of measures. One such attribute is the total weight $\mu(Q)$ of the measure μ. A congruence $\mathcal{E}$ on A is called Σ-*respecting* if $\mu\ \mathcal{E}\ \mu'$ implies $\mu(Q) = \mu'(Q)$ for all $\mu, \mu' \in \mathrm{R}^Q$. Since it is easily verified that the identity relation is a Σ-respecting congruence, and the union of an arbitrary collection of Σ-respecting congruences is again a Σ-respecting congruence, it follows that there exists a largest Σ-respecting congruence for A. We call this relation the Σ-*congruence* relation for A.

The following result can be interpreted as saying that the Σ-congruent measures are precisely those that are indistinguishable by any "experiment" in which a series of actions is performed and then the total weight of the resulting measure is observed.

Lemma 3.3 *Let A be a weighted automaton with state set Q and action set E. Then measures μ and μ' in R^Q are Σ-congruent for A if and only if for all w in E^* we have $\mu\, T_w(Q) = \mu'\, T_w(Q)$.*

Proof. Suppose μ and μ' are Σ-congruent for A. We claim that for all w in E^* we have $\mu\, T_w(Q) = \mu'\, T_w(Q)$. The proof is by induction on the length of w. If w has length 0, then $w = \epsilon$, the empty word. In this case, we have $\mu\, T_\epsilon(Q) = \mu(Q) = \mu'(Q) = \mu'\, T_\epsilon(Q)$, by the fact that Σ-congruence is Σ-respecting. Now suppose we have established the result for all words w of length less than or equal to n, for some $n \geq 0$, and consider a word ew of length $n+1$. Since μ and μ' are Σ-congruent and Σ-congruence is A-invariant, it follows that μT_e and $\mu' T_e$ are also Σ-congruent. Then $\mu\, T_{ew}(Q) = \mu T_e\, T_w(Q) = \mu' T_e\, T_w(Q). = \mu'\, T_{ew}(Q)$, where we have used the definition of T_{ew} and the induction hypothesis.

Conversely, let $\mathcal{E}$ be the binary relation on R^Q that relates measures μ and μ' precisely when for all w in E^* we have $\mu\, T_w(Q) = \mu'\, T_w(Q)$. We claim that $\mathcal{E}$ is a Σ-respecting congruence, hence $\mathcal{E}$ is contained in Σ-congruence. The linearity of $\mathcal{E}$ is obvious from the form of its definition. To show that $\mathcal{E}$ is Σ-respecting, suppose $\mu\, \mathcal{E} \mu'$. Then $\mu(Q) = \mu T_\epsilon(Q) = \mu' T_\epsilon(Q) = \mu'(Q)$, as required. To show that $\mathcal{E}$ is invariant, suppose again that $\mu\, \mathcal{E} \mu'$. Let $e \in E$ be arbitrary. Then we have, for all $w \in E^*$:

$$\mu T_e T_w(Q) = \mu T_{ew}(Q) = \mu' T_{ew}(Q) = \mu' T_e T_w(Q).$$

Hence $\mu T_e\ \mathcal{E}\ \mu' T_e$, as required for invariance. □

Corollary 3.4 *Let A be a weighted automaton with state set Q and action set E. Suppose $\mathcal{E}$ is a Σ-respecting congruence on A. If μ and μ' in R^Q are related by $\mathcal{E}$, then for all $e \in E$ we have $\mu T_e(Q) = \mu' T_e(Q)$.*

Proof. If μ and μ' are related by the Σ-respecting congruence $\mathcal{E}$, then μ and μ' are Σ-congruent. Now apply Lemma 3.3 in the special case that $w = e$. □

Since $qT_e(Q)$ amounts to a kind of "total transition weight" for action $e \in E$ from state q, Corollary 3.4 shows that Σ-congruence is nontrivial in the sense that it never relates two states having distinct total transition weights for some action e. In particular, it never relates a state q in which e is enabled ($qT_e(Q) \neq 0$) with a state q' in which e is not enabled ($q'T_e(Q) = 0$). Thus, Σ-congruence is the largest congruence it is reasonable to consider, if one regards the enabling of each individual action e as an observable characteristic of a state.

Let $A_1 = (E_1, Q_1, \{T_{1,e} : e \in E_1\})$ and $A_2 = (E_2, Q_2, \{T_{2,e} : e \in E_2\})$ be weighted automata. We define the *composition* of A_1 and A_2 to be the weighted automaton

$$A_1 * A_2 = (E_1 \cup E_2, Q_1 \times Q_2, \{T_{1,e} \otimes T_{2,e} : e \in E\})$$

In the above, the symbol $\otimes$ denotes tensor (or Kronecker) product, so that

$$T_{1,e} \otimes T_{2,e}((q_1, q_2), (q_1', q_2')) \;=\; T_{1,e}(q_1, q_1') \cdot T_{2,e}(q_2, q_2').$$

Also, we have used the convention that $T_{1,e}$ denotes the identity transformation if $e \notin E_1$, and similarly for $T_{2,e}$. Thus, $A_1 * A_2$ models a system consisting of components A_1 and A_2, where actions $e \in E_1 \cap E_2$ are executed jointly by both components, and actions $e \in (E_1 \setminus E_2) \cup (E_2 \setminus E_1)$ are executed independently by one component or the other.

The above definition of composition is not new, as similar or identical notions of composition have previously been defined and studied in the literature on stochastic process algebras. Hillston [8] compares the merits of a number of such notions of composition that have appeared in the context of work on the stochastic process algebras TIPP [7] PEPA [9], MPA [4], and MPA/EMPA [1]. The observation that tensor notation provides a succinct way to define composition of automata is also not new, dating back at least to Plateau [16] for stochastic automata.

The following result together with Lemma 3.3 shows that composition of weighted automata respects Σ-congruence. Similar results are well-known in the stochastic automata literature.

Proposition 3.5 *Suppose A_1 and A_2 are weighted automata. Then for all measures μ_1 over Q_1 and measures μ_2 over Q_2, and all $w \in (E_1 \cup E_2)^*$ we have*

$$(\mu_1 \otimes \mu_2)\, T_w(Q_1 \times Q_2) = \mu_1 T_w(Q_1) \cdot \mu_2 T_w(Q_2).$$

Proof. By induction on w. If $w = \epsilon$, then

$$(\mu_1 \otimes \mu_2)\, T_w(Q_1 \times Q_2) = (\mu_1 \otimes \mu_2)(Q_1 \times Q_2) = \mu_1(Q_1) \cdot \mu_2(Q_2).$$

Suppose now that $w = ew'$ and that we have shown the result for w'. Then

$$\begin{aligned}
(\mu_1 \otimes \mu_2)\, T_w(Q_1 \times Q_2) &= (\mu_1 \otimes \mu_2)\, T_{ew'}(Q_1 \times Q_2) \\
&= (\mu_1 \otimes \mu_2)\, (T_{1,e} \otimes T_{2,e}) T_{w'}(Q_1 \times Q_2) \\
&= (\mu_1 T_{1,e} \otimes \mu_2 T_{2,e})\, T_{w'}(Q_1 \times Q_2) \\
&= \mu_1 T_{1,e} T_{w'}(Q_1) \cdot \mu_2 T_{2,e} T_{w'}(Q_2) \\
&= \mu_1 T_w(Q_1) \cdot \mu_2 T_w(Q_2).
\end{aligned}$$

□

4. Probabilistic Weighted Automata

In this section, we consider a class of weighted automata for which the weights can be interpreted as probabilities. The main result (comprising Theorems 4.3 and 4.4) is that for this class of automata, probabilistic bisimulations correspond to those weighted automata congruences that are in a sense generated by their restrictions to point measures. In addition, though every probabilistic bisimulation is automatically Σ-respecting, Σ-congruence on a probabilistic weighted automaton does not necessarily correspond to any probabilistic bisimulation. Thus, probabilistic bisimulation equivalence is a strict refinement of Σ-congruence.

Probabilistic bisimulation was introduced by Larsen and Skou [14], as an adaptation of Milner's bisimulation to probabilistic transition systems. For purely probabilistic transition systems, which determine Markov chains, probabilistic bisimulation turns out to be essentially the same concept as the older *lumpability* notion of Kemeny and Snell [12]. In that work, lumpability of a Markov chain with respect to an equivalence relation R on states is defined by the condition that the stochastic process that results when R-related states are identified is once again a Markov chain. In that case, the resulting "lumped" chain can serve for some purposes as a faithful abstraction of the original chain [3]. Kemeny and Snell identify the following *row sum condition* as necessary and sufficient for lumpability: whenever q and q' are related by R and C is an equivalence class of R, then the total of the transition probabilities from q to states in C equals the total of the transition probabilities from q' to states in C. Larsen and Skou's definition of probabilistic bisimulation is based on the same row sum condition, except that it is imposed on transitions of each action separately. Used as a definition, the row sum condition is rather concrete and thus not really satisfactory from an algebraic point of view. Here we build on ideas of Jonsson and Larsen [10] to obtain a more abstract characterization of probabilistic bisimulation in terms of weighted automata congruences.

Define a weighted automaton $A = (E, Q, \{T_e : e \in E\})$ to be *probabilistic* if for all $e \in E$ the operator T_e is nonnegative and in addition for all $q \in Q$, the "row sum" $qT_e(Q)$ is either 0 or 1. In case $qT_e(Q) = 1$, we say that e is *enabled* in state q, otherwise e is *not enabled* in state q. A measure μ on Q is called a *distribution* if it is nonnegative and in addition $\mu(Q) = 1$. Note that, though in general we have $\mu T_e(Q) \leq \mu(Q)$, equality need not hold unless e is enabled in all states. Thus, it is generally *not* true that μT_e is a distribution if μ is.

Jonsson and Larsen [10] define a lifting of an equivalence relation R on a set Q to a relation R^* on distributions over Q as follows: $\mu \; R^* \; \mu'$ if and only if there exists a distribution M on $Q \times Q$ with $\mathrm{supp}(M) \subseteq R$, such that the following conditions hold:

$$\sum_{q' \in Q} M(q, q') = \mu(q) \qquad\qquad \sum_{q \in Q} M(q, q') = \mu'(q').$$

We may think of M as a matrix, whose entries describe the way to redistribute probabilities from μ within equivalence classes so as to arrive at μ'.

In their definition of R^*, Jonsson and Larsen require that M be a distribution over $Q \times Q$, so that its entries sum to one. However, if μ and μ' are distributions, and M is any nonnegative measure that satisfies the above conditions with respect to μ and μ', then M is automatically a distribution. There is therefore no reason not to view R^* as a relation on measures, rather than distributions, satisfying the same conditions as in Jonsson and Larsen's definition except that we allow M to be a nonnegative measure on $Q \times Q$ rather than a distribution. We assume this generalization of Jonsson and Larsen's definition in the sequel.

Lemma 4.1 *Suppose R is a binary relation on Q. Then the following are equivalent statements about a measure μ on Q:*

1. $\mu \in K_R$.

2. μ *can be expressed in the form*

$$\sum_{q \in Q} \sum_{q' \in Q} M(q,q')(\delta_q - \delta_{q'})$$

where M *is a measure on* $Q \times Q$ *with* $\mathrm{supp}(M) \subseteq R$.

3. $\mu(C) = 0$ *for all equivalence classes* C *of the reflexive transitive closure* $\overline{R}$ *of* R.

In particular, $\delta_q - \delta_{q'} \in K_R$ *if and only if* (q,q') *is in* $\overline{R}$.

Proof. 1) implies 2)

By definition K_R is the subspace generated by all measures of the form $\delta_q - \delta_{q'}$ where $q\ R\ q'$; thus any measure in K_R is a linear combination of measures of this form. Given $\mu \in K_R$, we may express μ as a linear combination of measures of the form $\delta_q - \delta_{q'}$ where $q\ R\ q'$ and then we may define a measure M on $Q \times Q$ by taking $M(q,q')$ to be the coefficient of $\delta_q - \delta_{q'}$ in the chosen linear combination. Then M has the stated property by construction.

2) implies 3)

Suppose there exists a measure M on $Q \times Q$ such that $\mathrm{supp}(M) \subseteq R$ and such that

$$\mu = \sum_{q \in Q} \sum_{q' \in Q} M(q,q')(\delta_q - \delta_{q'}).$$

Let C be an arbitrary equivalence class of $\overline{R}$, and for $q \in Q$ let $[q]$ denote the equivalence class that contains q. Then

$$\begin{aligned}
\mu(C) &= \sum_{q \in Q} \sum_{q' \in Q} M(q,q')(\delta_q(C) - \delta_{q'}(C)) \\
&= \sum_{q \in Q} \sum_{q' \in [q]} M(q,q')(\delta_q(C) - \delta_{q'}(C)) \\
&= \sum_{q \in C} \sum_{q' \in [q]} M(q,q')(\delta_q(C) - \delta_{q'}(C)) \\
&\qquad + \sum_{q \notin C} \sum_{q' \in [q]} M(q,q')(\delta_q(C) - \delta_{q'}(C)) \\
&= \sum_{q \in C} \sum_{q' \in [q]} M(q,q')(1-1) + \sum_{q \notin C} \sum_{q' \in [q]} M(q,q')(0-0) \\
&= 0.
\end{aligned}$$

Thus, for any equivalence class C of $\overline{R}$ we have $\mu(C) = 0$.

3) implies 1)

Let $\#\mu$ denote the number of $q \in Q$ for which $\mu(q)$ is nonzero. Note that this quantity is finite for all μ, in view of the finiteness of Q. We show, by induction on $\#\mu$, that for all measures μ on Q, if $\mu(C) = 0$ for all equivalence classes C of $\overline{R}$, then $\mu \in K_R$. For the basis case, if $\#\mu = 0$ then $\mu = 0$ and $\mu \in K_R$ holds trivially. Suppose $\#\mu = n > 0$ and that we have established the result for all μ' with $\#\mu' < n$. Suppose $\mu(C) = 0$ for all equivalence classes C of $\overline{R}$. Since $\#\mu > 0$, there exists some

q for which $\mu(q) \neq 0$. Since $\mu([q]) = 0$, there must exist some $q' \in [q]$ such that $\mu(q')$ has sign opposite to that of $\mu(q)$. Suppose $|\mu(q)| \geq |\mu(q')|$; if the opposite relation holds the proof is symmetric. Define $\mu' = \mu + \mu(q')(\delta_q - \delta_{q'})$. Then $\mu'(r) = \mu(r)$ for all $r \notin \{q, q'\}$, also $\mu'(q') = 0$ and thus $\#\mu' < \#\mu$. Moreover, $\mu'(C) = \mu(C) = 0$ for all equivalence classes C of $\overline{R}$, so by induction we have $\mu' \in K_R$. But since $\mu = \mu' - \mu(q')(\delta_q - \delta_{q'})$ where $q' \in [q]$ it must also be the case that $\mu \in K_R$. □

The subspace K_R can in a sense be viewed as the theory generated by the relation R, in a logic whose sentences are equations of the form

$$\sum_{q \in S} a_q \delta_q \;\approx\; \sum_{q \in S'} b_q \delta_q,$$

where the coefficients a_q and b_q are positive and where the identity sign is interpreted as $R^\dagger$. More precisely, a measure $\mu \in K_R$ has a unique representation as a linear combination $\sum_{q \in Q} a_q \delta_q$. Let P be the set of all $q \in Q$ for which the coefficient a_q is positive, and let N be the set of all $q \in Q$ for which the coefficient a_q is negative. Let $b_q = -a_q$ for $q \in N$. Then the measures $\sum_{q \in P} a_q \delta_q$ and $\sum_{q \in N} b_q \delta_q$ differ by $\mu \in K_R$, hence they are related by $R^\dagger$. From this point of view, Lemma 4.1 can be seen as a conservative extension result which implies that the only equations between point measures that are logical consequences of the "axioms" R are equations of the form $\delta_q \approx \delta_{q'}$ where q and q' are related by the reflexive transitive closure of R.

Lemma 4.2 *Suppose R is an equivalence relation on Q. Then the following are equivalent statements about* nonnegative *measures μ and μ' on Q:*

1. $\mu \; R^* \; \mu'$.
2. $\mu \; R^\dagger \; \mu'$.

Proof. 1) implies 2)

Suppose $\mu \; R^* \; \mu'$. Then there exists a nonnegative measure M on $Q \times Q$ with $\mathrm{supp}(M) \subseteq R$ such that the following both hold.

$$\mu(q) = \sum_{q' \in Q} M(q, q') \qquad\qquad \mu'(q') = \sum_{q \in Q} M(q, q').$$

Then since $\mu = \sum_{q \in Q} \mu(q)\delta_q$ and $\mu' = \sum_{q' \in Q} \mu(q')\delta_{q'}$, it follows that

$$\mu = \sum_{q \in Q} \sum_{q' \in Q} M(q, q')\delta_q \qquad\qquad \mu' = \sum_{q' \in Q} \sum_{q \in Q} M(q, q')\delta_{q'},$$

and hence

$$\mu - \mu' = \sum_{q \in Q} \sum_{q' \in Q} M(q, q')(\delta_q - \delta_{q'}).$$

By Lemma 4.1, $\mu - \mu' \in K_R$, hence $\mu \; R^\dagger \; \mu'$.

2) implies 1) (*cf.* [11])

Suppose $\mu \, R^\dagger \, \mu'$; then $\mu - \mu' \in K_R$. By Lemma 4.1 for any equivalence class C of R we have $(\mu - \mu')(C) = 0$, hence $\mu(C) = \mu'(C)$. Define $M(q, q') = \mu(q)\mu'(q')/\mu'([q])$

if $q \; R \; q'$ and $\mu'([q]) > 0$, and define $M(q,q') = 0$ otherwise. Then $M(q,q') \geq 0$ for all $q, q' \in Q$, so that M is nonnegative. Since $\mathrm{supp}(M) \subseteq R$, it follows that

$$\sum_{q \in Q} \sum_{q' \in Q} M(q,q') = \sum_{q \in Q} \sum_{q' \in [q]} M(q,q').$$

Now, let q be an arbitrary element of Q. If $\mu'([q]) = 0$ then also $\mu'(q) = 0$ (because μ' is assumed nonnegative) and $M(q,q') = 0$ for all $q' \in Q$. In addition $\mu'([q]) = \mu([q])$ so that $\mu(q) = 0$. Thus in this case we have $0 = \sum_{q' \in Q} M(q,q') = \mu(q)$. On the other hand, if $\mu'([q]) > 0$, then

$$\begin{aligned} \sum_{q' \in Q} M(q,q') &= \sum_{q' \in [q]} \mu(q)\mu(q')/\mu'([q]) \\ &= (\mu(q)/\mu'([q])) \sum_{q' \in [q]} \mu'(q') \\ &= (\mu(q)/\mu'([q]))\mu'([q]) \\ &= \mu(q). \end{aligned}$$

Since q was arbitrary, it follows that $\sum_{q' \in Q} M(q,q') = \mu(q)$ for all $q \in Q$. Similar reasoning shows that also $\sum_{q \in Q} M(q,q') = \mu'(q')$ for all $q' \in Q$. Thus, $\mu \; R^* \; \mu'$. □

Note that the construction of M in the second part of the proof of Lemma 4.2 does not work for general signed measures, since in that case we could have $\mu'([q]) = 0$ without having $\mu'(q) = 0$.

Jonsson, Larsen, and Yi [11] define the notion of probabilistic bisimulation for the general class of *probabilistic transition systems.* Their definition, adapted to the special case of probabilistic weighted automata which is our present interest, defines a *probabilistic bisimulation* on a probabilistic weighted automaton $A = (E, Q, \{T_e : e \in E\})$ to be an equivalence relation R on Q such that whenever $q \; R \; q'$, then for all $e \in E$ we have $qT_e \; R^* \; q'T_e$. In view of Lemma 4.2 and the fact that the transition maps T_e are nonnegative, this is equivalent to the condition that whenever $q \; R \; q'$, then for all $e \in E$ we have $qT_e \;\; R^\dagger \;\; q'T_e$.

We are now able to relate the notions of probabilistic bisimulation and weighted automata congruence. We first show that if R is a probabilistic bisimulation on A, then R^* is "almost" a congruence on A, failing to be a congruence only because it is defined only for nonnegative measures.

Theorem 4.3 *Let A be a probabilistic weighted automaton with state set Q, and suppose R is a probabilistic bisimulation on Q. Then $R^\dagger$ is a congruence on A that coincides with R^* on nonnegative measures.*

Proof. That $R^\dagger$ coincides with R^* on nonnegative measures is the content of Lemma 4.2. It remains to be shown that $R^\dagger$ is a congruence on A. By Lemma 3.2, it suffices to show that $\mu \in K_R$ implies $\mu T_e \in K_R$ for all e. Suppose $\mu \in K_R$. Then by Lemma 4.1, there exists a measure M on $Q \times Q$ with $\mathrm{supp}(M) \subseteq R$ such that

$$\mu = \sum_{q \in Q} \sum_{q' \in Q} M(q,q')(\delta_q - \delta_{q'}).$$

But then

$$\mu T_e = \sum_{q\in Q}\sum_{q'\in Q} M(q,q')(qT_e - q'T_e).$$

Since R is a probabilistic bisimulation, $q\ R\ q'$ implies $qT_e\ R^*\ q'T_e$, hence $qT_e\ R^\dagger\ q'T_e$ by Lemma 4.2, and thus $qT_e - q'T_e \in K_R$. From this it follows that $\mu T_e \in K_R$, because K_R is a subspace of R^Q and μT_e is a linear combination of elements of K_R. □

The next result shows that the probabilistic bisimulations correspond to those weighted automata congruences that are in a sense generated by their restrictions to point measures.

Theorem 4.4 *Let A be a probabilistic weighted automaton with state set Q, and let $\mathcal{E}$ be a congruence on A. Let R be the binary relation on Q defined by $q\ R\ q'$ if and only if $\delta_q\ \mathcal{E}\ \delta_{q'}$. If $R^\dagger = \mathcal{E}$, then R is a probabilistic bisimulation.*

Proof. Suppose $R^\dagger = \mathcal{E}$. Suppose further that $q\ R\ q'$, then it follows by definition of R that $\delta_q\ \mathcal{E}\ \delta_{q'}$. Since $\mathcal{E}$ is a congruence, we have $qT_e\ \mathcal{E}\ q'T_e$ for all e. Since A is probabilistic, qT_e and $q'T_e$ are nonnegative measures for all $e \in E$. Since $\mathcal{E} = R^\dagger$, and $R^\dagger$ coincides with R^* on nonnegative measures by Lemma 4.2, it follows that $qT_e\ R^*\ q'T_e$ for all e, showing that R is a probabilistic bisimulation. □

If A is a probabilistic weighted automaton, then it can be shown that there always exists a largest probabilistic bisimulation relation on A. This relation is called *probabilistic bisimulation equivalence.*

Corollary 4.5 *Let A be a probabilistic weighted automaton with state set Q, and let $\simeq$ denote the probabilistic bisimulation equivalence relation for A. Then $\simeq^\dagger$ is the largest congruence $\mathcal{E}$ on A with the property that $\mathcal{E} = R^\dagger$ for some equivalence relation R on Q.*

Proof. By Theorem 4.3, $\simeq^\dagger$ is a congruence on A and it obviously has the stated property. To complete the proof we must show that $\simeq^\dagger$ is the largest such congruence. So, suppose $\mathcal{E}$ is an arbitrary congruence with the property that $\mathcal{E} = R^\dagger$ for some equivalence relation R on Q. Then $\delta_q\ \mathcal{E}\ \delta_{q'}$ iff $\delta_q - \delta_{q'} \in K_{R^\dagger} = K_R$. By Lemma 4.1, $\delta_q - \delta_{q'} \in K_R$ if and only if $q\ R\ q'$. But then it follows from Theorem 4.4 that R is a probabilistic bisimulation. □

The next result shows that the congruences determined by probabilistic bisimulation relations are Σ-respecting, hence probabilistic bisimulation equivalence is a refinement of Σ-congruence.

Theorem 4.6 *Let A be a probabilistic weighted automaton with state set Q, and let R be a probabilistic bisimulation on A. Then $R^\dagger$ is Σ-respecting.*

Proof. It suffices to show that $\mu(Q) = 0$ whenever $\mu \in K_R$. Suppose $\mu \in K_R$. Then by Lemma 4.1, there exists a measure M on $Q \times Q$ with $\mathrm{supp}(M) \subseteq R$ such that

$$\mu = \sum_{q \in Q} \sum_{q' \in Q} M(q, q')(\delta_q - \delta_{q'}).$$

But then

$$\begin{aligned}
\mu(Q) &= \sum_{r \in Q} \sum_{q \in Q} \sum_{q' \in Q} M(q, q')(\delta_q(r) - \delta_{q'}(r)) \\
&= \sum_{q \in Q} \sum_{q' \in Q} M(q, q') \sum_{r \in Q} (\delta_q(r) - \delta_{q'}(r)) \\
&= \sum_{q \in Q} \sum_{q' \in Q} M(q, q')(1 - 1) \\
&= 0.
\end{aligned}$$

□

The following example shows that Σ-congruence for probabilistic weighted automata does not necessarily correspond to any probabilistic bisimulation, and thus probabilistic bisimulation equivalence is a strict refinement of Σ-congruence.

Example 2 Let $A = (E, Q, \{T_e : e \in E\})$ be defined as in Example 1. It is easy to check that A is a probabilistic weighted automaton. Let $\mathcal{K}$ and $\mathcal{E}$ also be defined as in Example 1, then $\mathcal{E}$ is a congruence on A with $K_{\mathcal{E}} = \mathcal{K}$. It is also easy to check that $\mu(Q) = 0$ for all $\mu \in \mathcal{K}$; for example, if $\mu = (1/2)\delta_{q_1} + (1/2)\delta_{q_1'} - \delta_{r_1}$, then $\mu(Q) = (1/2) + (1/2) - 1 = 0$. Thus, $\mathcal{E}$ is a Σ-respecting congruence on A.

Since $\delta_{q_0}\ \mathcal{E}\ \delta_{r_0}$, it follows that δ_{q_0} and δ_{r_0} are Σ-congruent. However any probabilistic bisimulation R containing (q_0, r_0) would have to contain (q_1, r_1) and (q_1', r_1), hence also (q_1, q_1'), so that $\delta_{q_1} - \delta_{q_1'} \in K_R$. By the invariance of R we would then also have to have $\delta_{q_3} = (\delta_{q_1} - \delta_{q_1'})T_b T_c \in K_R$. This is impossible because $\delta_{q_3}(Q) = 1 \neq 0$, hence $R^\dagger$ cannot be Σ-respecting.

Example 2 illustrates the key reason why Σ-congruence can be a coarser relation than probabilistic bisimulation equivalence. Specifically, Σ-congruence does not require that states (point measures) always be related to other states, but also permits states to be related to arbitrary measures. In the example, the relation $\mathcal{E}$ relates the state r_1 to the measure $\frac{1}{2}\delta_{q_1} + \frac{1}{2}\delta_{q_1'}$. This relationship is necessary for invariance if q_0 is related to r_0.

Lemma 3.3 gives us an even clearer picture of the nature of Σ-congruence. For a probabilistic weighted automaton A with action set E and state set Q, if μ is a distribution, then for each word w the quantity $\mu\, T_w(Q)$ corresponds to the probability that action sequence w is performed if A is started in initial distribution μ. Then by Lemma 3.3, distributions μ and μ' are Σ-congruent if and only if they are indistinguishable by experiments that estimate the probability of performing a given action sequence.

A weighted automaton can be factored by Σ-congruence just as it can by probabilistic bisimulation equivalence. The states of the quotient automaton obtained in this way will in general not be equivalence classes of states of the original automaton, but will instead be subspaces of R^Q. Since Σ-congruence can be a coarser relation than probabilistic bisimulation equivalence, the quotient automaton that results from factoring by Σ-congruence can be smaller than that obtained from factoring by probabilistic bisimulation equivalence. We shall see in Section 5 that the same situation occurs for PIOA behavior equivalence. This way of constructing a reduced automaton, which is naturally suggested by the linear algebraic structure associated with weighted automata, seems not to have been studied in the context of stochastic process algebras.

We now consider composition of probabilistic weighted automata. We first note that the class of probabilistic weighted automata is closed under composition. This straightforward observation has already been made in the literature in various settings (for example, see *e.g.* [4]) but we restate it here for the sake of continuity of the presentation.

Lemma 4.7 *Suppose A_1 and A_2 are probabilistic weighted automata. Then their composition $A_1 * A_2$ is also probabilistic.*

Proof. Since

$$(q_1, q_2)(T_{1,e} \otimes T_{2,e})(Q_1 \times Q_2) = q_1 T_{1,e}(Q_1) \cdot q_2 T_{2,e}(Q_2),$$

and $q_1 T_{1,e}(Q_1)$ and $q_2 T_{2,e}(Q_2)$ are either 0 or 1, it follows that the transition matrices $T_e = T_{1,e} \otimes T_{2,e}$ have the same 0/1 property. □

Just as for Σ-congruence, composition of probabilistic weighted automata respects probabilistic bisimulation. This fact about probabilistic bisimulation is already known (see, *e.g.* [2]), though our presentation here in terms of weighted automata congruences is new.

Proposition 4.8 *Suppose A_1 and A_2 are probabilistic weighted automata, and that R_1 is a probabilistic bisimulation on A_1 and R_2 is a probabilistic bisimulation on A_2. Let the relation R on $Q_1 \times Q_2$ be defined by*

$$(q_1, q_2)\ R\ (q_1', q_2') \text{ iff } q_1\ R_1\ q_1' \text{ and } q_2\ R_2\ q_2'.$$

Then R is a probabilistic bisimulation on $A_1 * A_2$.

Proof. The relation $R^\dagger$ is clearly invariant under $T_{1,e} \otimes T_{2,e}$ and hence is a congruence on $A_1 * A_2$. It then follows by Theorem 4.4 that R is a probabilistic bisimulation. □

5. Probabilistic I/O Automata

A *probabilistic I/O automaton* (PIOA) is a weighted automaton

$$A = (E^{\mathrm{in}} \uplus E^{\mathrm{out}}, Q, \{T_e : e \in E\})$$

such that

- If $e \in E^{\mathrm{in}}$, then T_e is nonnegative and $\sum_{r \in Q} T_e(q,r) = 1$ for all $q \in Q$. In this case, the entries of T_e are interpreted as *transition probabilities*.
- If $e \in E^{\mathrm{out}}$, then T_e is nonnegative. In this case, the entries of T_e for $e \in E^{\mathrm{out}}$ are interpreted as *transition rates*.

The set E^{in} is the set of *input actions* and E^{out} is the set of *output actions* for A. Note that the requirement that $\sum_{r \in Q} T_e(q,r) = 1$ for all $e \in E^{\mathrm{in}}$ and all $q \in Q$ means that a PIOA is *input enabled*: all input actions are enabled in each state.

We regard the output transitions enabled in a state q as independent, competing activities. When state q is entered, each output transition "chooses" a random time according to its associated exponential probability distribution. The output transition that will be taken (assuming that no input occurs first) is the one that chooses the smallest time. Thus, the sojourn time in state q before the next output transition is executed is a random variable that is the minimum of the random variables associated with each of the output transitions enabled in state q. This variable also has an exponential distribution, whose parameter is the *total output rate* from state q:

$$\mathrm{rt}(q) = \sum_{e \in E^{\mathrm{out}}} \sum_{r \in Q} T_e(q,r).$$

The expected sojourn time in state q is the mean of this distribution, which is given by $1/\mathrm{rt}(q)$.

Probabilistic I/O automata [21, 22] have features in common with *stochastic automata* (SA), for which there is an established body of research. One definition of stochastic automata is given by Buchholz [5], who cites earlier work by Plateau and her co-workers [16, 17, 18]. In Buchholz' definition, a stochastic automaton includes a finite set of *states*, a set of *action labels* including a distinguished *internal action* label τ, and a function that maps state-action-state triples to nonnegative real *transition weights*. In addition, Buchholz includes as part of a stochastic automaton an initial probability distribution on states and a mapping that assigns to each state a nonnegative *reward vector*. In the absence of synchronization, such an automaton can be regarded as determining a continuous-time Markov chain (CTMC) whose generator matrix is derived in an evident fashion from the transition weight function. The entries of the reward vector associated with a state are interpreted as rates of linear reward accumulation during a sojourn in that state.

Perhaps the best way to compare stochastic automata and probabilistic I/O automata is to think of a PIOA as a stochastic automaton with the initial conditions and reward structure removed, but having an added input structure that describes the automaton's response to stimuli applied by its environment. The transition weight matrices can be used to associate a CTMC with a PIOA just as for a stochastic automaton, except for the fact that with a PIOA it is only the matrices for output actions, rather than all actions, that contribute to the CTMC. The matrices associated with the input actions of a PIOA provide a way for the environment of a PIOA to cause it to make abrupt state transitions that interrupt the autonomous evolution described by the associated CTMC. As a consequence of the definition of PIOA composition given later in this section, synchronization between a PIOA and its environment is asymmetric in the sense that a PIOA cannot constrain the application of input stimuli to it by its environment, and conversely, the environment of a PIOA cannot constrain its trajectory and associated output actions other than through the application of input stimuli.

We now give the definition of the behavior map $\mathcal{B}_q^A$ associated with a state q of a PIOA A. As we shall see, $\mathcal{B}_q^A$ can be thought of as a description of how to calculate the rewards associated with certain sets of trajectories of A starting from state q. To state the definition a few preliminary notions are required. If E is a set, then a *rated trace* over E is an element of the set $([0,\infty) \times E)^*$; that is, a finite word over the alphabet of pairs (d, e), with $d \in [0,\infty)$ and $e \in E$. We use ϵ to denote the empty word. An *observable* over E is a formal power series $\Phi \in \mathrm{R}\langle\langle [0,\infty) \times E \rangle\rangle$; that is, a mapping from rated traces to real numbers.

Suppose we have fixed in advance a countably infinite "universal" set of actions U, and let $A = (E, Q, \{T_e : e \in E\})$ be a PIOA such that $E \subseteq U$. Let $\mathcal{B}^A$ be the mapping that assigns to each state $q \in Q$ the *transformation of observables*:

$$\mathcal{B}_q^A : \mathrm{R}\langle\langle [0,\infty) \times U \rangle\rangle \to \mathrm{R}\langle\langle [0,\infty) \times U \rangle\rangle$$

according to the following inductive definition:

$$\mathcal{B}_q^A[\Phi](\epsilon) = \Phi(\epsilon)$$

$$\mathcal{B}_q^A[\Phi]((d,e)\alpha) = \sum_{q' \in Q} T_e(q,q')\ \mathcal{B}_{q'}^A[(d + \mathrm{rt}(q), e)^{-1}\Phi](\alpha).$$

In the above, we have used the convention that T_e is the identity transformation when $e \in U \setminus E$.

To gain some intuitive understanding of $\mathcal{B}_q^A[\Phi](\alpha)$, one should think of α as giving certain partial information about a particular set of execution trajectories that might be traversed by A in combination with its environment. In particular, if $\alpha = (d_1, e_1)(d_2, e_2) \dots (d_n, e_n)$, then $e_1 e_2 \dots e_n$ is the sequence of actions performed in such a trajectory (including both input and output actions) and $d_1 d_2 \dots d_n$ is the sequence of output rates associated with the successive states visited by the environment in such a trajectory. The observable Φ should be thought of as a way of associating some numeric measure, or reward, with trajectories. By "unwinding" the definition of $\mathcal{B}_q^A[\Phi](\alpha)$, one can see that it amounts to a weighted summation, over all trajectories of A starting from state q and matching α, of a certain reward $\Phi(\alpha')$ associated with this trajectory, where α' is obtained from α by combining the output rates of the states visited by A with the rates of the corresponding environment states. The weight associated with the term $\Phi(\alpha')$ is the product of the weights assigned by A to each of the transitions along the trajectory.

The concept of $\mathcal{B}_q^A$ was introduced and studied in our previous work, though the inductive definition given above is much simpler than the definitions we previously used. The precise notion defined above is in spirit the same as that studied in [20], but differs formally from it in two respects: (1) here we do not treat internal actions, and (2) the rated traces we use here do not have a "final rate" after the last action. Although it is possible to handle internal actions using a definition of behavior similar to that given above, the definition turns out to be a recursive definition requiring least-fixed-point techniques, rather than a straightforward induction, and so we leave this for future work. The omission of the final rate from rated traces has the effect of making the total output rate of a state not directly observable, thereby coarsening the induced “behavior equivalence” relation somewhat. This permits a comparison to be made between behavior and bisimulation equivalences.

The reason for our interest in $\mathcal{B}_q^A$ is because, as shown in [22], it exactly captures those distinctions between states that can be made on the basis of a certain kind of probabilistic testing, and as further shown in [20, 19], interesting performance parameters for A when started from state q can be extracted from $\mathcal{B}_q^A$.

To illustrate what can be done with $\mathcal{B}_q^A$, consider the observable Π_a defined as follows:

$$\Pi_a((d_1, e_1)(d_2, e_2)\dots(d_n, e_n)) = \begin{cases} \prod_{k=1}^n \frac{1}{d_k}, & \text{if } e_n = a \text{ and } e_k \neq a \text{ for } k < n, \\ 0, & \text{otherwise.} \end{cases}$$

Then if A is “closed” in the sense that $E^{\text{in}} = \emptyset$, it can be shown that

$$\sum_{\alpha \in (\{0\} \times E)^*} \mathcal{B}_q^A[\Pi_a](\alpha)$$

is the probability that A will eventually perform action a if it is started from state q. Moreover, suppose A eventually performs action a with probability 1 if it is started from state q. Define observable Λ_a as follows:

$$\Lambda_a((d_1, e_1)(d_2, e_2)\dots(d_n, e_n)) = \begin{cases} \left(\prod_{k=1}^n \frac{1}{d_k}\right)\left(\sum_{k=1}^n \frac{1}{d_k}\right), & \text{if } e_n = a \text{ and } e_k \neq a \text{ for } k < n, \\ 0, & \text{otherwise.} \end{cases}$$

Then

$$\sum_{\alpha \in (\{0\} \times E)^*} \mathcal{B}_q^A[\Lambda_a](\alpha)$$

is the expected time for A to perform action a when started from state q.

To understand in detail how behavior maps work, it is instructive to use the definitions given above to actually carry out an expected time calculation for a simple example.

Example 3 Consider the PIOA A having state set $Q = \{q_0, q_1\}$, action set $E_A = E_A^{\text{in}} \uplus E_A^{\text{out}}$ with $E_A^{\text{in}} = \emptyset$ and $E_A^{\text{out}} = \{b\}$, and with transition map T_b^A such that

$$T_b^A(q_0, q_1) = 2$$

and such that $T_b^A(q, q') = 0$ in all other cases. A transition diagram for A is shown in Figure 2(a).

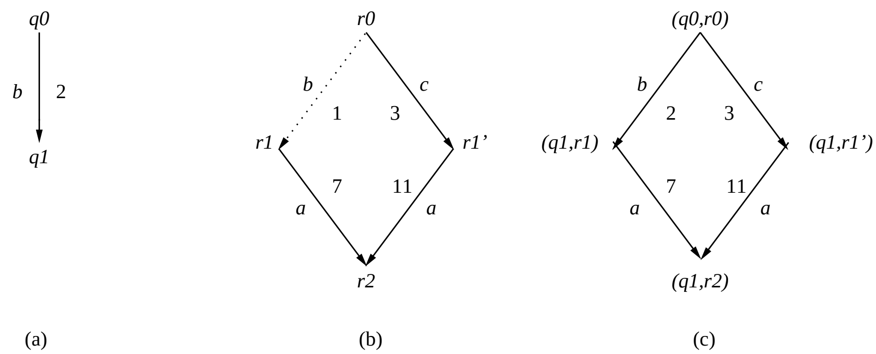

Figure 2: Transition Diagrams for Example 3

Using the definition of behavior map, we may calculate as follows:

$$\begin{aligned}
\mathcal{B}_{q_0}^A[\Lambda_a]((d_0,b)(d_1,a)) &= 2 \cdot \mathcal{B}_{q_1}^A[(d_0+2,b)^{-1}\Lambda_a]((d_1,a)) \\
&= 2 \cdot 1 \cdot \mathcal{B}_{q_1}^A[(d_1+0,a)^{-1}(d_0+2,b)^{-1}\Lambda_a](\epsilon) \\
&= 2 \cdot 1 \cdot \Lambda_a((d_0+2,b)(d_1,a)) \\
&= \left(\frac{2}{d_0+2} \cdot \frac{1}{d_1}\right)\left(\frac{1}{d_0+2} + \frac{1}{d_1}\right).
\end{aligned}$$

Similarly, we have

$$\mathcal{B}_{q_0}^A[\Lambda_a]((d_0,c)(d_1,a)) = \left(\frac{1}{d_0+2} \cdot \frac{1}{d_1}\right)\left(\frac{1}{d_0+2} + \frac{1}{d_1}\right).$$

Now, consider the PIOA B having state set $Q = \{r_0, r_1, r_1', r_2\}$, action set $E_B = E_B^{\text{in}} \uplus E_B^{\text{out}}$ with $E_B^{\text{in}} = \{b\}$ and $E_B^{\text{out}} = \{a, c\}$, with transition maps T_a^B, T_b^B, and T_c^B having

$$T_b^B(r_0, r_1) = 1, \;\; T_c^B(r_0, r_1') = 3, \;\; T_a^B(r_1, r_2) = 7, \;\; T_a^B(r_1', r_2) = 11$$

and with $T_a^B(q,q') = 0$, $T_b^B(q,q') = 0$, and $T_c^B(q,q') = 0$ in all other cases. A transition diagram for B is shown in Figure 2(b).

The weighted automata composition $A * B$ of A and B is a PIOA having action set $E = E^{\text{in}} \uplus E^{\text{out}}$ with $E^{\text{in}} = \emptyset$ and $E^{\text{out}} = \{a, b, c\}$, with transition diagram shown in Figure 2 (c). Suppose we wish to calculate the quantity

$$\sum_{\alpha \in (\{0\} \times E)^*} \mathcal{B}_{(q_0,r_0)}^{A*B}[\Lambda_a](\alpha),$$

which will be the expected time for the composition $A * B$ to perform action a when started from state (q_0, r_0). Though we could calculate this quantity directly using the definition of the behavior map for $A * B$, in order to illustrate the way in which behavior maps handle input/output interactions, we calculate instead the quantity

$$\sum_{\alpha \in (\{0\} \times E)^*} \mathcal{B}_{r_0}^B[\mathcal{B}_{q_0}^A[\Lambda_a]](\alpha).$$

According to Theorem 5.9 proved later in this section, this will produce the same result.

We proceed as follows:

$$
\begin{aligned}
&\sum_{\alpha\in(\{0\}\times E)^*} \mathcal{B}^B_{r_0}[\mathcal{B}^A_{q_0}[\Lambda_a]](\alpha) \\
&\quad= \sum_{\alpha'\in(\{0\}\times E)^*} 1\cdot\mathcal{B}^B_{r_1}[(0+3,b)^{-1}\mathcal{B}^A_{q_0}[\Lambda_a]](\alpha') \\
&\qquad\quad+ \sum_{\alpha'\in(\{0\}\times E)^*} 3\cdot\mathcal{B}^B_{r_1'}[(0+3,c)^{-1}\mathcal{B}^A_{q_0}[\Lambda_a]](\alpha') \\
&\quad= \sum_{\alpha''\in(\{0\}\times E)^*} 1\cdot 7\cdot\mathcal{B}^B_{r_2}[(0+7,a)^{-1}(3,b)^{-1}\mathcal{B}^A_{q_0}[\Lambda_a]](\alpha'') \\
&\qquad\quad+ \sum_{\alpha''\in(\{0\}\times E)^*} 3\cdot 11\cdot\mathcal{B}^B_{r_2}[(0+11,a)^{-1}(3,c)^{-1}\mathcal{B}^A_{q_0}[\Lambda_a]](\alpha'') \\
&\quad= 1\cdot 7\cdot((7,a)^{-1}(3,b)^{-1}\mathcal{B}^A_{q_0}[\Lambda_a])(\epsilon) \\
&\qquad\quad+\ 3\cdot 11\cdot((11,a)^{-1}(3,c)^{-1}\mathcal{B}^A_{q_0}[\Lambda_a])(\epsilon) \\
&\quad= 1\cdot 7\cdot\mathcal{B}^A_{q_0}[\Lambda_a]((3,b)(7,a)) + 3\cdot 11\cdot\mathcal{B}^A_{q_0}[\Lambda_a]((3,c)(11,a)) \\
&\quad= 1\cdot 7\cdot\left(\frac{2}{3+2}\cdot\frac{1}{7}\right)\left(\frac{1}{3+2}+\frac{1}{7}\right) \\
&\qquad\quad+\ 3\cdot 11\cdot\left(\frac{1}{3+2}\cdot\frac{1}{11}\right)\left(\frac{1}{3+2}+\frac{1}{11}\right) \\
&\quad= \left(\frac{2}{5}\cdot\frac{7}{7}\right)\left(\frac{1}{5}+\frac{1}{7}\right)+\left(\frac{3}{5}\cdot\frac{11}{11}\right)\left(\frac{1}{5}+\frac{1}{11}\right) \\
&\quad= \left(\frac{2}{5}\right)\left(\frac{1}{5}+\frac{1}{7}\right)+\left(\frac{3}{5}\right)\left(\frac{1}{5}+\frac{1}{11}\right) \\
&\quad= \frac{1}{5}+\left(\frac{2}{5}\cdot\frac{1}{7}+\frac{3}{5}\cdot\frac{1}{11}\right)
\end{aligned}
$$

In eliminating the summations above we have used the fact that $T^B_a(r_2,q) = T^B_b(r_2,q) = T^B_c(r_2,q) = 0$ for all $q\in Q$, hence $\mathcal{B}^B_{r_2}[\Phi](\alpha'')$ is nonzero only when $\alpha''=\epsilon$, and in that case $\mathcal{B}^B_{r_2}[\Phi](\alpha'')=\Phi(\epsilon)$. In the above calculation, observe in particular how the total output rates of the various states eventually contribute to the computation of sojourn times, and how the transition rates that "stack up" each time the behavior map is unwound eventually find their corresponding denominators and become probabilities. Also notice how these denominators are built by an incremental process in which the rates of the states traversed by each component are summed into the appropriate denominator, no matter whether the action performed from a state is an input or an output action. This is because the occurrence of an input transition from a state involves foregoing the opportunity to perform an output transition from that state, and this must be taken into account in calculating the transition probability.

We can generalize from Example 3 to see that observables serve essentially the same purpose for probabilistic I/O automata as reward functions do for stochastic automata. However, although the class of rewards that can be expressed in terms of observables overlaps the class that can be expressed via state rewards, neither of the two classes is entirely contained in the other. Observables correspond to the concept of a reward that is tallied by an external observer who has access to the sequence of actions performed and the sojourn time of each of the states that is traversed. Such rewards can be history-dependent, but cannot make arbitrary distinctions between individual states. On the other hand, reward measures for stochastic automata have access to information about the precise states traversed, but they cannot depend on the history of actions performed.

Lemma 5.1 *Suppose A is a PIOA. Then*

$$(d,e)^{-1}\mathcal{B}_q^A[\Phi] = \sum_{q' \in Q} T_e(q,q')\mathcal{B}_{q'}^A[(d+\mathrm{rt}(q),e)^{-1}\Phi].$$

Proof. Straightforward from the definitions. □

In order to consider what happens when a PIOA is executed from a particular starting distribution, rather than a specific starting state, it is convenient to extend the notation $\mathcal{B}_q^A$ to $\mathcal{B}_\mu^A$, where $\mu \in \mathrm{R}^Q$. This is done by linearity as follows:

$$\mathcal{B}_\mu^A[\Phi](\alpha) = \sum_{q \in Q} \mu(q)\mathcal{B}_q^A[\Phi](\alpha).$$

Clearly we then have

$$\mathcal{B}_{a\mu+b\nu}^A = a\,\mathcal{B}_\mu^A + b\,\mathcal{B}_\nu^B.$$

Lemma 5.2 *For each measure $\mu \in \mathrm{R}^Q$, the mapping $\mathcal{B}_\mu^A$ is a linear operator on $\mathrm{R}\langle\langle[0,\infty) \times E\rangle\rangle$.*

Proof. We first show that given $q \in Q$, for all rated traces α, for all observables $\Phi, \Psi \in \mathrm{R}\langle\langle[0,\infty) \times E\rangle\rangle$, and all $a, b \in \mathrm{R}$ we have

$$\mathcal{B}_q^A[a\Phi + b\Psi](\alpha) = a\,\mathcal{B}_q^A[\Phi](\alpha) + b\,\mathcal{B}_q^A[\Psi](\alpha).$$

The proof is by induction on α. In case $\alpha = \epsilon$, then

$$\begin{aligned}\mathcal{B}_q^A[a\Phi + b\Psi](\alpha) &= (a\Phi + b\Psi)(\epsilon)\\ &= a\,\Phi(\epsilon) + b\,\Psi(\epsilon)\\ &= a\,\mathcal{B}_q^A[\Phi](\alpha) + b\,\mathcal{B}_q^A[\Psi](\alpha).\end{aligned}$$

Now suppose $\alpha = (d, e)\alpha'$, and that we have established the result for α'. Then

$$\begin{aligned}
\mathcal{B}^A_q[a\Phi + b\Psi](\alpha) &= \mathcal{B}^A_q[a\Phi + b\Psi]((d,e)\alpha') \\
&= \sum_{q' \in Q} T_e(q,q')\mathcal{B}^A_{q'}[(d + \mathrm{rt}(q), e)^{-1}(a\Phi + b\Psi)](\alpha') \\
&= \sum_{q' \in Q} T_e(q,q') \cdot \\
&\qquad\qquad \mathcal{B}^A_{q'}[a(d + \mathrm{rt}(q), e)^{-1}\Phi + b(d + \mathrm{rt}(q), e)^{-1}\Psi](\alpha') \\
&= a \sum_{q' \in Q} T_e(q,q')\mathcal{B}^A_{q'}[(d + \mathrm{rt}(q), e)^{-1}\Phi](\alpha') \\
&\qquad\qquad + b \sum_{q' \in Q} T_e(q,q')\mathcal{B}^A_{q'}[(d + \mathrm{rt}(q), e)^{-1}\Psi](\alpha') \\
&= a \sum_{q' \in Q} T_e(q,q')\mathcal{B}^A_{q'}[(d + \mathrm{rt}(q), e)^{-1}\Phi](\alpha') \\
&\qquad\qquad + b \sum_{q' \in Q} T_e(q,q')\mathcal{B}^A_{q'}[(d + \mathrm{rt}(q), e)^{-1}\Psi](\alpha') \\
&= a\, \mathcal{B}^A_q[\Phi](\alpha) + b\, \mathcal{B}^A_q[\Psi](\alpha),
\end{aligned}$$

where we have used the linearity of derivative in the second step and the induction hypothesis in the third step.

The result for $\mathcal{B}^A_\mu$ for μ an arbitrary measure now follows immediately from the special case established above, as a result of the linear form of the definition of $\mathcal{B}^A_\mu$. □

Define measures μ and μ' in R^Q to be *behavior equivalent for A* if $\mathcal{B}^A_\mu = \mathcal{B}^A_{\mu'}$.

Theorem 5.3 *Let A be a PIOA. Then behavior equivalence for A is a Σ-respecting congruence on A.*

Proof. Linearity is obvious from the form of the definition. To show Σ-respecting, suppose μ and μ' are A-behavior-equivalent, so that

$$\sum_{q \in Q} \mu(q)\mathcal{B}^A_q = \sum_{q \in Q} \mu'(q)\mathcal{B}^A_q.$$

Letting $\mathbf{1} \in \mathrm{R}\langle\langle[0,\infty) \times E\rangle\rangle$ denote the identically 1 observable, we have

$$\mu(Q) = \sum_{q \in Q} \mu(q)\mathcal{B}^A_q[\mathbf{1}](\epsilon) = \sum_{q \in Q} \mu'(q)\mathcal{B}^A_q[\mathbf{1}](\epsilon) = \mu'(Q),$$

as required.

To show invariance, suppose again that μ and μ' are A-behavior-equivalent. We must show that

$$\sum_{q \in Q} (\mu T_e)(q) \cdot \mathcal{B}^A_q[\Phi](\alpha) = \sum_{q \in Q} (\mu' T_e)(q) \cdot \mathcal{B}^A_q[\Phi](\alpha)$$

for all $e \in E$, all observables Φ and all rated traces α. Note that

$$(\mu T_e)(q) = \sum_{q' \in Q} \mu(q') T_e(q', q),$$

and thus

$$\begin{aligned}\sum_{q \in Q} (\mu T_e)(q) \cdot \mathcal{B}^A_q[\Phi](\alpha) &= \sum_{q \in Q} \sum_{q' \in Q} \mu(q') T_e(q', q) \mathcal{B}^A_q[\Phi](\alpha) \\ &= \sum_{q' \in Q} \mu(q') \sum_{q \in Q} T_e(q', q) \mathcal{B}^A_q[\Phi](\alpha).\end{aligned}$$

Define the observable Φ' by $\Phi'((0,e)\alpha) = \Phi(\alpha)$ and $\Phi'(\alpha) = 1$ for all rated traces α not beginning with $(0,e)$, so that we have $\Phi = (0,e)^{-1}\Phi'$. Then

$$\sum_{q' \in Q} \mu(q') \sum_{q \in Q} T_e(q', q) \mathcal{B}^A_q[\Phi](\alpha) = \sum_{q' \in Q} \mu(q') \mathcal{B}^A_{q'}[\Phi']((0,e)\alpha).$$

We have thus shown

$$\sum_{q \in Q} (\mu T_e)(q) \cdot \mathcal{B}^A_q[\Phi](\alpha) = \sum_{q' \in Q} \mu(q') \mathcal{B}^A_{q'}[\Phi']((0,e)\alpha).$$

Using the same reasoning for μ' we obtain

$$\sum_{q \in Q} (\mu' T_e)(q) \cdot \mathcal{B}^A_q[\Phi](\alpha) = \sum_{q' \in Q} \mu'(q') \mathcal{B}^A_{q'}[\Phi']((0,e)\alpha).$$

The assumption that μ and μ' are A-behavior-equivalent now implies that

$$\sum_{q \in Q} (\mu T_e)(q) \cdot \mathcal{B}^A_q[\Phi](\alpha) = \sum_{q \in Q} (\mu' T_e)(q) \cdot \mathcal{B}^A_q[\Phi](\alpha)$$

as required. □

Close examination of the argument used in the proof of Theorem 5.3 above reveals that no specific properties of the derivative operation on observables were used, other than that given any (d,e) and Φ, there exists Φ' such that $(d,e)^{-1}\Phi' = \Phi$. Note that the identically 1 observable $\mathbf{1}$ has the property that $(d,e)^{-1}\mathbf{1} = \mathbf{1}$ for all (d,e). Thus, the arguments above remain valid if Φ and Φ' are replaced by $\mathbf{1}$ throughout, and then the above argument shows that the equivalence relating μ and μ' exactly when $\mathcal{B}^A_\mu[\mathbf{1}] = \mathcal{B}^A_{\mu'}[\mathbf{1}]$, is also a Σ-respecting congruence. In fact this relation is precisely the relation of Σ-congruence.

Although the original motivation for probabilistic bisimulation only makes sense for probabilistic weighted automata, the formal characterization obtained from Theorems 4.3 and 4.4 makes sense for arbitrary weighted automata with nonnegative transition maps. In view of this fact, we define a *weighted bisimulation* on a weighted automaton A with state set Q, action set E, and nonnegative transition maps T_e to be an equivalence relation R on the states of A such that $q \; R \; q'$ implies $qT_e \; R^\dagger \; q'T_e$ for all $q, q' \in Q$ and $e \in E$. In case the weights are interpreted as transition rates, then weighted bisimulation equivalence coincides with the the kind of bisimulation introduced by Hillston for the language PEPA [9], as well as one of the two studied by Buchholz in the context of Markovian Process Algebra [4]. The following result shows that when applied to probabilistic I/O automata, weighted bisimulation equivalence is a refinement of PIOA behavior equivalence.

Theorem 5.4 *Let A be a PIOA with state set Q and action set E, and let R be a weighted bisimulation on A. Then $q\ R\ q'$ implies $\mathcal{B}^A_q = \mathcal{B}^A_{q'}$ for all $q, q' \in Q$.*

Proof. We show by induction on α that for all $\Phi \in \mathrm{R}\langle\langle[0,\infty) \times E\rangle\rangle$ and all rated traces α, if $q\ R\ q'$, then $\mathcal{B}^A_q[\Phi](\alpha) = \mathcal{B}^A_{q'}[\Phi](\alpha)$.

For the basis case, if $\alpha = \epsilon$, then $\mathcal{B}^A_q[\Phi](\alpha) = \Phi(\epsilon) = \mathcal{B}^A_{q'}[\Phi](\alpha)$.

For the induction step, suppose $\alpha = (d, e)\alpha'$ and that we have already established the result for α'. Then

$$\mathcal{B}^A_q[\Phi](\alpha) = \sum_{r \in Q} T_e(q, r)\mathcal{B}^A_r[(d + \mathrm{rt}(q), e)^{-1}\Phi](\alpha').$$

Since $q\ R\ q'$ and since R is a bisimulation, by Corollary 3.4 we have $\mathrm{rt}(q) = \mathrm{rt}(q')$. Let $\Phi' = (d + \mathrm{rt}(q), e)^{-1}\Phi = (d + \mathrm{rt}(q'), e)^{-1}\Phi$, so that

$$\mathcal{B}^A_q[\Phi](\alpha) = \sum_{q'' \in Q} T_e(q, q'')\mathcal{B}^A_{q''}[\Phi'](\alpha').$$

Similarly, we have

$$\mathcal{B}^A_{q'}[\Phi](\alpha) = \sum_{q'' \in Q} T_e(q', q'')\mathcal{B}^A_{q''}[\Phi'](\alpha').$$

Thus

$$\mathcal{B}^A_q[\Phi](\alpha) - \mathcal{B}^A_{q'}[\Phi](\alpha) = \sum_{q'' \in Q} (T_e(q, q'') - T_e(q', q''))\mathcal{B}^A_{q''}[\Phi'](\alpha').$$

Note that

$$T_e(q, q'') - T_e(q', q'') = (qT_e - q'T_e)(q'').$$

Since $q\ R\ q'$, by the assumption that R is a bisimulation we have $qT_e \ \ R^\dagger \ \ q'T_e$. By Lemma 4.1, there exists a measure M on $Q \times Q$ with $\mathrm{supp}(M) \subseteq R$ such that

$$qT_e - q'T_e = \sum_{r \in Q} \sum_{r' \in Q} M(r, r')(\delta_r - \delta_{r'}).$$

Then

$$\begin{aligned}(qT_e - q'T_e)(q'') &= \sum_{r \in Q} \sum_{r' \in Q} M(r, r')\delta_r(q'') - \sum_{r \in Q} \sum_{r' \in Q} M(r, r')\delta_{r'}(q'') \\ &= \sum_{r' \in Q} M(q'', r') - \sum_{r \in Q} M(r, q'').\end{aligned}$$

We therefore have

$$\begin{aligned}
\mathcal{B}_q^A[\Phi](\alpha) - \mathcal{B}_{q'}^A[\Phi](\alpha) &= \sum_{q''\in Q}(\sum_{r'\in Q} M(q'',r') - \sum_{r\in Q} M(r,q''))\mathcal{B}_{q''}^A[\Phi'](\alpha') \\
&= \sum_{q''\in Q}\sum_{r''\in Q} M(q'',r'')\mathcal{B}_{q''}^A[\Phi'](\alpha') \\
&\qquad - \sum_{q''\in Q}\sum_{r''\in Q} M(r'',q''))\mathcal{B}_{q''}^A[\Phi'](\alpha') \\
&= \sum_{q''\in Q}\sum_{r''\in Q} M(q'',r'')\mathcal{B}_{q''}^A[\Phi'](\alpha') \\
&\qquad - \sum_{r''\in Q}\sum_{q''\in Q} M(q'',r''))\mathcal{B}_{r''}^A[\Phi'](\alpha') \\
&= \sum_{q''\in Q}\sum_{r''\in Q} M(q'',r'')(\mathcal{B}_{q''}^A[\Phi'](\alpha') - \mathcal{B}_{r''}^A[\Phi'](\alpha')).
\end{aligned}$$

It follows from the induction hypothesis that $\mathcal{B}_{q''}^A[\Phi'](\alpha') = \mathcal{B}_{r''}^A[\Phi'](\alpha')$ if $q''\ R\ r''$, and since $\mathrm{supp}(M) \subseteq R$, we have $\mathcal{B}_q^A[\Phi](\alpha) - \mathcal{B}_{q'}^A[\Phi](\alpha) = 0$, hence $\mathcal{B}_q^A[\Phi](\alpha) = \mathcal{B}_{q'}^A[\Phi](\alpha)$, completing the induction step and the proof. □

The following example illustrates that the refinement relationship given by the previous theorem is in fact strict, in the sense that there exists a PIOA A with states q and q', such that $\mathcal{B}_q^A = \mathcal{B}_{q'}^A$ but such that we do not have $q\ \mathcal{E}\ q'$ for any weighted bisimulation $\mathcal{E}$.

Example 4 Consider the PIOA A having state set

$$Q = \{q_0, q_1, q_1', q_2\} \cup \{r_0, r_1, r_2\},$$

action set $E = E^{\mathrm{in}} \uplus E^{\mathrm{out}}$ with $E^{\mathrm{in}} = \emptyset$ and $E^{\mathrm{out}} = \{a, b, c\}$, and having transition matrices T_a, T_b, and T_c defined so that their only non-zero entries are as follows:

$$\begin{array}{lll}
T_a(q_0,q_1) = 1/2 & T_a(q_0,q_1') = 1/2 & T_a(r_0,r_1) = 1 \\
T_b(q_1,q_2) = 1 & T_b(r_1,r_2) = 1/2 & \\
T_c(q_1',q_2) = 1 & T_c(r_1,r_2) = 1/2. &
\end{array}$$

A transition diagram for A is shown in Figure 3.

Clearly $\mathcal{B}_{q_2}^A = \mathcal{B}_{r_2}^A$, since neither state has any outgoing transitions. Also, $\mathrm{rt}(q_1) = \mathrm{rt}(q_1') = \mathrm{rt}(r_1) = 1$, from which it is easy to verify that

$$\mathcal{B}_{r_1}^A[\Phi](\alpha) = \frac{1}{2}\mathcal{B}_{q_1}^A[\Phi](\alpha) + \frac{1}{2}\mathcal{B}_{q_1'}^A[\Phi](\alpha)$$

for all Φ and α. It follows immediately from this that $\mathcal{B}_{q_0}^A[\Phi](\alpha) = \mathcal{B}_{r_0}^A[\Phi](\alpha)$ for all Φ and α. However any weighted bisimulation R containing (q_0, r_0) would have to contain (q_1, r_1) and (q_1', r_1), hence also (q_1, q_1'), which is impossible because different sets of actions are enabled in q_1 and q_1'.

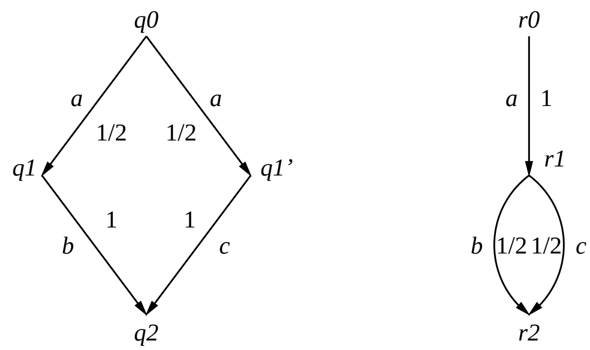

Figure 3: Transition Diagram for Example 4

In Example 4 as in Example 2, we see how the possibility of relating states to measures can result in a coarser congruence than if we require that states always be related to other states. That is, states q_0 and r_0 are behavior equivalent as a result of our freedom to relate state r_1 to the measure $\frac{1}{2}\delta_{q_1} + \frac{1}{2}\delta_{q_1'}$. So PIOA behavior equivalence, like Σ-congruence, is in general a coarser equivalence than probabilistic bisimulation equivalence and factoring by PIOA behavior equivalence can lead to smaller quotient automata than factoring by probabilistic bisimulation equivalence.

On the other hand, the following example shows that PIOA behavior equivalence is a strict refinement of Σ-congruence, and that Σ-congruence is "wrong" for probabilistic I/O automata (or, indeed, for stochastic automata) in the sense that in general it relates states that are distinguishable by expected time experiments.

Example 5 Consider the PIOA A having state set

$$Q = \{q_0, q_1, q_1', q_2\} \cup \{r_0, r_1, r_2\}.$$

action set $E = E^{\text{in}} \uplus E^{\text{out}}$ with $E^{\text{in}} = \emptyset$ and $E^{\text{out}} = \{a, b\}$, and having transition matrices T_a and T_b be defined so that their only nonzero entries are as follows:

$$T_a(q_0, q_1) = 2 \qquad T_a(q_0, q_1') = 1 \qquad T_a(r_0, r_1) = 3$$

$$T_b(q_1, q_2) = 1 \qquad T_b(q_1', q_2) = 2 \qquad T_b(r_1, r_2) = 4/3$$

A transition diagram for A is shown in Figure 4.

It is straightforward to check that the states q_0 and r_0 are Σ-congruent, but they are not behavior equivalent due to the fact that $\text{rt}(q_1)$ and $\text{rt}(q_1')$ are distinct from each other as well as from $\text{rt}(r_1)$. If we know with certainty that the automaton is in state r_1, then we would expect to observe a sojourn time distribution that is exponentially distributed with rate $\text{rt}(r_1) = 4/3$. On the other hand, if we know only that the automaton is in the mixture of states $\frac{2}{3}\delta_{q_1} + \frac{1}{3}\delta_{q_1'}$, then the sojourn time distribution will be a nontrivial mixture of two exponentials, one with rate $\text{rt}(q_1) = 1$ and one with rate $\text{rt}(q_1') = 2$. This difference is something that is directly discernable by an observer who can perform expected time measurements. The expected time to

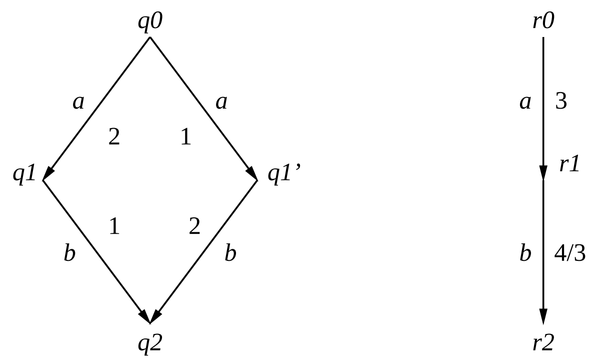

Figure 4: Transition Diagram for Example 5

execute ab starting from state q_0 is $(1/3) + (2/3)(1/1) + (1/3)(1/2) = 7/6$ but the expected time to execute ab starting from state r_0 is $(1/3) + (3/4) = 7/12$.

To further clarify the nature of behavior equivalence and its relationship to weighted bisimulation, we now obtain a characterization of behavior equivalence as the largest Σ-respecting congruence that in a sense separates states having distinct total output rates.

Suppose A is a probabilistic I/O automaton with state set Q. A subset S of R^Q is called *rate-homogeneous* if there exists $d \in [0, \infty)$ such that for all $\mu \in S$ and all $q \in Q$, if $\mu(q) \neq 0$ then $\mathrm{rt}(q) = d$.

Theorem 5.5 *Suppose A is a probabilistic I/O automaton with state set Q and action set E. Let $\equiv$ denote the relation of A-behavior equivalence on R^Q, then $K_\equiv$ is the direct sum of rate-homogeneous subspaces of R^Q.*

Proof. Since Q is a finite set, the set $\{\mathrm{rt}(q) : q \in Q\}$ is also a finite set, which we may enumerate as $\{d_1, d_2, \ldots, d_n\}$. Given any $\mu \in \mathrm{R}^Q$, define

$$\mu|_{d_i} = \sum_{\{q \in Q : \mathrm{rt}(q) = d_i\}} \mu(q)\delta_q.$$

Define

$$K|_{d_i} = \{\mu|_{d_i} : \mu \in K_\equiv\}.$$

Clearly $K|_{d_i}$ is a rate-homogeneous subspace of R^Q for $1 \leq i \leq n$, and $\mu = \sum_{i=1}^n \mu|_{d_i}$ for all $\mu \in K_\equiv$. Moreover, if $\sum_{i=1}^n \mu|_{d_i} = 0$, then $\mu|_{d_i} = 0$ for $1 \leq i \leq n$. We claim that $K|_{d_i} \subseteq K_\equiv$ for $1 \leq i \leq n$ and hence we have the direct sum decomposition $K_\equiv = \oplus K|_{d_i}$. We prove this claim by showing that $\mathcal{B}^A_{\mu|_{d_i}} = 0$ for all $\mu \in K_\equiv$ and all $1 \leq i \leq n$.

For $\Phi \in \mathrm{R}\langle\langle [0, \infty) \times E \rangle\rangle$ define $\Phi|_{d_i} \in \mathrm{R}\langle\langle [0, \infty) \times E \rangle\rangle$ for $1 \leq i \leq n$ by

$$\Phi|_{d_i}(\alpha) = \begin{cases} \Phi(\alpha), & \text{if } \alpha = ((d_i, e)\alpha'), \text{ for some } e \in E \text{ and } \alpha' \in ([0, \infty) \times E)^* \\ 0, & \text{otherwise.} \end{cases}$$

We claim that $\Phi|_{d_i}$ has the following properties:

1. $\mathcal{B}^A_\mu[\mathbf{1}|_{d_i}]((0,e)) = \mu|_{d_i}(Q)$, for all $e \in U \setminus E$, where $\mathbf{1}$ denotes the identically 1 observable.
2. $\mathcal{B}^A_\mu[\Phi|_{d_i+d}]((d,e)\alpha) = \mathcal{B}^A_{\mu|_{d_i}}[\Phi]((d,e)\alpha)$, for all $d \in [0,\infty)$ and all $e \in E$.

To prove property (1), we calculate as follows:

$$\begin{aligned}\mathcal{B}^A_\mu[\mathbf{1}|_{d_i}]((0,e)) &= \sum_{q\in Q}\mu(q)\sum_{q'\in Q}T_e(q,q')\cdot\mathbf{1}|_{d_i}((\mathrm{rt}(q),e)) \\ &= \mu(\{q\in Q : \mathrm{rt}(q) = d_i\}) \\ &= \mu|_{d_i}(Q),\end{aligned}$$

where we have used the definition of $\mathbf{1}|_{d_i}$ and the fact that $e \in U\setminus E$ implies $T_e(q,q') = 1$ if $q = q'$ and 0 otherwise. To prove property (2), we calculate as follows:

$$\begin{aligned}&\mathcal{B}^A_\mu[\Phi|_{d_i+d}]((d,e)\alpha) \\ &\quad= \textstyle\sum_{q\in Q}\mu(q)\ \sum_{q'\in Q}T_e(q,q')\ \mathcal{B}^A_{q'}[(d+\mathrm{rt}(q),e)^{-1}\Phi|_{d+d_i}](\alpha) \\ &\quad= \textstyle\sum_{\{q\in Q: d+\mathrm{rt}(q)=d+d_i\}}\mu(q)\ \sum_{q'\in Q}T_e(q,q')\ \mathcal{B}^A_{q'}[(d+\mathrm{rt}(q),e)^{-1}\Phi](\alpha) \\ &\quad= \textstyle\sum_{\{q\in Q:\mathrm{rt}(q)=d_i\}}\mu(q)\ \sum_{q'\in Q}T_e(q,q')\ \mathcal{B}^A_{q'}[(d+\mathrm{rt}(q),e)^{-1}\Phi](\alpha) \\ &\quad= \mathcal{B}^A_{\mu|_{d_i}}[\Phi]((d,e)\alpha).\end{aligned}$$

We now return to proving that if $\mu \in K_\equiv$, then $\mathcal{B}^A_{\mu|_{d_i}} = 0$ for $1 \le i \le n$. Suppose $\mu \in K_\equiv$. We first show that $\mathcal{B}^A_{\mu|_{d_i}}[\Phi](\epsilon) = 0$ for all $\Phi \in \mathrm{R}\langle\langle[0,\infty)\times E\rangle\rangle$ and all i with $1 \le i \le n$. By definition, $\mathcal{B}^A_{\mu|_{d_i}}[\Phi](\epsilon) = \mu|_{d_i}(Q)\cdot\Phi(\epsilon)$ for $1 \le i \le n$. Let e be an arbitrarily chosen element of $U \setminus E$, which is nonempty because U is countably infinite and E is finite. By property (1) shown above,

$$\mathcal{B}^A_\mu[\mathbf{1}|_{d_i}]((0,e)) = \mu|_{d_i}(Q)$$

and thus

$$\mathcal{B}^A_{\mu|_{d_i}}[\Phi](\epsilon) = \mathcal{B}^A_\mu[\mathbf{1}|_{d_i}]((0,e))\cdot\Phi(\epsilon)$$

for $1 \le i \le n$. Since $\mu \in K_\equiv$ by hypothesis, we have $\mathcal{B}^A_\mu = 0$, and hence $\mathcal{B}^A_{\mu|_{d_i}}[\Phi](\epsilon) = 0\cdot\Phi(\epsilon) = 0$ for $1 \le i \le n$.

To complete the proof, we show that $\mathcal{B}^A_{\mu|_{d_i}}[\Phi](\alpha) = 0$ for all i with $1 \le i \le n$, all $\Phi \in \mathrm{R}\langle\langle[0,\infty)\times E\rangle\rangle$, and all α of the form $(d,e)\alpha'$ for some $d \in [0,\infty)$, $e \in E$, and $\alpha' \in ([0,\infty)\times E)^*$. To see this, simply observe that by property (2) we have

$$\mathcal{B}^A_{\mu|_{d_i}}[\Phi]((d,e)\alpha') = \mathcal{B}^A_\mu[\Phi|_{d_i+d}]((d,e)\alpha'),$$

which equals 0 by the assumption that $\mu \in K_\equiv$. □

Theorem 5.6 *Suppose A is a probabilistic I/O automaton with state set Q and action set E. Suppose $\mathcal{E}$ is a Σ-respecting congruence on A with the property that $K_{\mathcal{E}}$ is the* *direct sum of rate-homogeneous subspaces of* R^Q. *Then* $\mathcal{E}$ *is contained in A-behavior equivalence.*

Proof. Suppose $\mathcal{E}$ is a Σ-respecting congruence on A such that $K_\mathcal{E}$ is the direct sum $\oplus_{i=1}^n K_i$, where each K_i is rate-homogeneous. To show that $\mathcal{E}$ is contained in A-behavior equivalence, it suffices to prove that $\mu \in K_\mathcal{E}$ implies $\mathcal{B}_\mu^A = 0$ for all $\mu \in \mathrm{R}^Q$. To do this we show by induction on α that for all $\alpha \in ([0,\infty) \times U)^*$, if $\mu \in K_\mathcal{E}$ then $\mathcal{B}_\mu^A[\Phi](\alpha) = 0$ for all $\Phi \in \mathrm{R}\langle\langle [0,\infty) \times U \rangle\rangle$.

For the basis case, suppose $\alpha = \epsilon$. Then $\mathcal{B}_\mu^A[\Phi](\alpha) = \mu(Q) \cdot \Phi(\epsilon)$. If $\mu \in K_\mathcal{E}$, then since μ is Σ-respecting we have $\mu(Q) = 0$, hence $\mathcal{B}_\mu^A[\Phi](\alpha) = 0$.

For the induction step, suppose $\alpha = (d,e)\alpha'$, and that we have already established the result for α'. Suppose $\mu \in K_\mathcal{E}$, then $\mu = \sum_{i=1}^n \mu_i$, where $\mu_i \in K_i$ for $1 \leq i \leq n$. Then

$$\begin{aligned}
\mathcal{B}_\mu^A[\Phi](\alpha) &= \sum_{q \in Q} \mu(q) \cdot \sum_{q' \in Q} T_e(q,q')\ \mathcal{B}_{q'}^A[(d + \mathrm{rt}(q), e)^{-1}\Phi](\alpha') \\
&= \sum_{i=1}^n \sum_{q \in Q} \mu_i(q) \cdot \sum_{q' \in Q} T_e(q,q')\ \mathcal{B}_{q'}^A[(d + \mathrm{rt}(q), e)^{-1}\Phi](\alpha').
\end{aligned}$$

Since K_i is rate-homogeneous, for each i with $1 \leq i \leq n$ there exists $d_i \in \mathrm{R}$ such that $\mathrm{rt}(q) = d_i$ for all $q \in \mathrm{supp}(\mu_i)$. Let $\Phi_i = (d + d_i, e)^{-1}\Phi$; we then have

$$\begin{aligned}
\mathcal{B}_\mu^A[\Phi](\alpha) &= \sum_{i=1}^n \sum_{q \in Q} \mu_i(q) \cdot \sum_{q' \in Q} T_e(q,q')\mathcal{B}_{q'}^A[\Phi_i](\alpha') \\
&= \sum_{i=1}^n \sum_{q' \in Q} (\mu_i T_e)(q')\mathcal{B}_{q'}^A[\Phi_i](\alpha') \\
&= \sum_{i=1}^n \mathcal{B}_{\mu_i T_e}^A[\Phi_i](\alpha').
\end{aligned}$$

Since $\mu_i \in K_i \subseteq K_\mathcal{E}$, it follows by invariance that $\mu_i T_e \in K_\mathcal{E}$. We may therefore apply the induction hypothesis to conclude that $\mathcal{B}_{\mu_i T_e}^A[\Phi_i](\alpha') = 0$ for $1 \leq i \leq n$, and thus $\mathcal{B}_\mu^A[\Phi](\alpha) = 0$, to complete the induction step and the proof. □

Corollary 5.7 *Suppose A is a probabilistic I/O automaton with state set Q. Then A-behavior-equivalence is the largest* Σ*-respecting congruence* $\mathcal{E}$ *on A with the property that* $K_\mathcal{E}$ *is the direct sum of rate-homogeneous subspaces of* R^Q.

Proof. Immediate from Theorems 5.5 and 5.6. □

We now consider the relationship between behavior maps and PIOA composition. The main result (Theorem 5.9) was established in our previous work [22, 20], however here we give a simpler proof that exploits the weighted automata formulation and our new definition of PIOA behavior map.

We first state for completeness of exposition the obvious fact that the formal conditions for being a probabilistic I/O automaton are preserved by weighted automata composition.

Lemma 5.8 *Suppose A_1 and A_2 are probabilistic I/O automata. Then their composition $A_1 * A_2$ is also a probabilistic I/O automaton, if we take $(E_1 \cup E_2)^{\mathrm{out}} = E_1^{\mathrm{out}} \cup E_2^{\mathrm{out}}$ and $(E_1 \cup E_2)^{\mathrm{in}} = (E_1 \cup E_2) \setminus (E_1^{\mathrm{out}} \cup E_2^{\mathrm{out}})$.*

Proof. Immediate from the fact that the tensor product of stochastic matrices is stochastic. □

Although the previous result shows that the composition of arbitrary probabilistic I/O automata is formally again a probabilistic I/O automaton, we must regard as suspect the idea that the tensor product of transition rate matrices can again be regarded as a transition rate matrix. For one thing, the result is dimensionally incorrect, since its entries have units of rate-squared, rather than rate. One way around this problem, adopted in [5], involves the assumption of a "basic rate" for each action. The entries of the transition rate matrices for action e are first normalized by dividing by the basic rate of action e to obtain matrices of dimensionless quantities, then the tensor product is taken, and finally the dimensionality is restored by multiplying the result by the basic rate. For probabilistic I/O automata, we take a more direct approach: we only consider as meaningful the composition of probabilistic I/O automata A_1 and A_2 that are *compatible* in the sense that $E_1^{\mathrm{out}} \cap E_2 \subseteq E_2^{\mathrm{in}}$ and $E_2^{\mathrm{out}} \cap E_1 \subseteq E_1^{\mathrm{in}}$. In forming the composition of compatible PIOA, therefore, we only ever form the tensor product of stochastic matrices, whose entries are dimensionless probabilities, or the tensor product of a stochastic matrix and a transition rate matrix. This kind of passive/active synchronization is the traditional way of defining composition for I/O automata [15], and was used by us for probabilistic I/O automata in [21]. It has also been considered elsewhere in the stochastic process algebra literature (see, *e.g.* [8]).

Theorem 5.9 *Suppose A_1 and A_2 are probabilistic I/O automata with state sets Q_1 and Q_2, respectively. Then for all $\mu_1 \in \mathrm{R}^{Q_1}$ and $\mu_2 \in \mathrm{R}^{Q_2}$ we have $\mathcal{B}^{A_1 * A_2}_{\mu_1 \otimes \mu_2} = \mathcal{B}^{A_1}_{\mu_1} \circ \mathcal{B}^{A_2}_{\mu_2}$.*

Proof. We first prove by induction on α that for all rated traces α over $E_1 \cup E_2$, all states $q_1 \in Q_1$ and all states $q_2 \in Q_2$, and all observables $\Phi \in \mathrm{R}\langle\langle [0,\infty) \times (E_1 \cup E_2) \rangle\rangle$ we have

$$\mathcal{B}^{A_1 * A_2}_{(q_1,q_2)}[\Phi](\alpha) = \mathcal{B}^{A_1}_{q_1}[\mathcal{B}^{A_2}_{q_2}[\Phi]](\alpha).$$

In case $\alpha = \epsilon$ we have

$$\begin{aligned}
\mathcal{B}^{A_1}_{q_1}[\mathcal{B}^{A_2}_{q_2}[\Phi]](\alpha) &= \mathcal{B}^{A_1}_{q_1}[\mathcal{B}^{A_2}_{q_2}[\Phi]](\epsilon) \\
&= \mathcal{B}^{A_2}_{q_2}[\Phi](\epsilon) \\
&= \Phi(\epsilon) \\
&= \mathcal{B}^{A_1 * A_2}_{(q_1,q_2)}[\Phi](\alpha).
\end{aligned}$$

Now suppose $\alpha = (d,e)\alpha'$ and we have already established the result for α'. Then

$$\begin{aligned}
\mathcal{B}^{A_1}_{q_1}[\mathcal{B}^{A_2}_{q_2}[\Phi]](\alpha) &= \sum_{q_1' \in Q_1} T_{1,e}(q_1,q_1') \cdot \mathcal{B}^{A_1}_{q_1'}[(d+\mathrm{rt}(q_1),e)^{-1}\mathcal{B}^{A_2}_{q_2}[\Phi]](\alpha') \\
&= \sum_{q_1' \in Q_1} T_{1,e}(q_1,q_1') \cdot \mathcal{B}^{A_1}_{q_1'}[\sum_{q_2' \in Q_2} T_{2,e}(q_2,q_2') \cdot \\
&\qquad \mathcal{B}^{A_2}_{q_2'}[(d+\mathrm{rt}(q_1)+\mathrm{rt}(q_2),e)^{-1}\Phi]](\alpha') \\
&= \sum_{q_1' \in Q_1} T_{1,e}(q_1,q_1') \cdot \sum_{q_2' \in Q_2} T_{2,e}(q_2,q_2') \cdot \\
&\qquad \mathcal{B}^{A_1}_{q_1'}[\mathcal{B}^{A_2}_{q_2'}[(d+\mathrm{rt}(q_1)+\mathrm{rt}(q_2),e)^{-1}\Phi]](\alpha') \\
&= \sum_{(q_1',q_2') \in Q_1 \times Q_2} (T_{1,e} \otimes T_{2,e})((q_1,q_2),(q_1',q_2')) \cdot \\
&\qquad \mathcal{B}^{A_1 * A_2}_{(q_1',q_2')}[(d+\mathrm{rt}(q_1)+\mathrm{rt}(q_2),e)^{-1}\Phi](\alpha') \\
&= \mathcal{B}^{A_1 * A_2}_{(q_1,q_2)}[\Phi](\alpha).
\end{aligned}$$

where we have used Lemma 5.1, Lemma 5.2, the induction hypothesis and simple properties of tensor product. This completes the induction step.

We now use the special case just established to prove the stated result. We have

$$\begin{aligned}
\mathcal{B}^{A_1}_{\mu_1}[\mathcal{B}^{A_2}_{\mu_2}[\Phi]](\alpha) &= \sum_{q_1 \in Q_1} \mu_1(q_1)\mathcal{B}^{A_1}_{q_1}[\sum_{q_2 \in Q_2} \mu_2(q_2)\mathcal{B}^{A_2}_{q_2}[\Phi]](\alpha) \\
&= \sum_{q_1 \in Q_1} \mu_1(q_1) \sum_{q_2 \in Q_2} \mu_2(q_2)\mathcal{B}^{A_1}_{q_1}[\mathcal{B}^{A_2}_{q_2}[\Phi]](\alpha) \\
&= \sum_{(q_1,q_2) \in Q_1 \times Q_2} (\mu_1 \otimes \mu_2)(q_1,q_2) \cdot \mathcal{B}^{A_1 * A_2}_{(q_1,q_2)}[\Phi](\alpha) \\
&= \mathcal{B}^{A_1 * A_2}_{\mu_1 \otimes \mu_2}[\Phi](\alpha),
\end{aligned}$$

completing the proof. □

6. Conclusion

We have made a comparison of PIOA behavior equivalence and probabilistic bisimulation equivalence by formulating both as a kind of congruence on weighted finite automata. The congruences we use relate signed measures on states, rather than just individual states. We found that probabilistic bisimulation equivalence is a strict refinement of PIOA behavior equivalence, basically due to the fact that PIOA behavior equivalence can contain relationships between measures that are not consequences of an underlying set of equivalences between individual states. We obtained a characterization of both bisimulation equivalence and PIOA behavior equivalence as the largest congruences generated by basic equivalences of a certain form. In the case of bisimulation equivalence, the basic equivalences simply relate individual states. In the case of PIOA behavior equivalence, the basic equivalences relate rate-homogeneous measures. The characterization of PIOA behavior equivalence helps to point the way to a sound and complete axiomatization of behavior equivalence in a process-algebraic logic for PIOA, which is a subject of our current research.

As practical applications of the PIOA model to system specifications often involve the use of state variables with infinitely many possible values, it would also be useful to generalize the results of this paper to weighted automata with countably infinite state sets. Such a generalization was not attempted in the present paper, because it was found that the close attention that needs to be paid to convergence and continuity considerations was preventing a clear exposition of the basic results. Since many of the proofs given in this paper appear to depend in a significant way on the finiteness of the set of states, it is not clear to what extent the results generalize to infinite state sets.

Acknowledgements

The author has benefited during the course of this work from discussions with colleagues Rance Cleaveland and Scott Smolka. Their contribution is gratefully acknowledged.

References

[1] M. Bernardo, L. Donatiello, and R. Gorrieri. A formal approach to the integration of performance aspects in the modeling and analysis of concurrent systems. *Information and Computation*, 144(2):83–154, August 1998.

[2] P. Buchholz. Equivalence relations for stochastic automata networks. In W. J. Stewart, editor, *Proceedings of the 2nd International Workshop on the Numerical Solution of Markov Chains*. Kluwer, 1995.

[3] P. Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, 31:59–75, 1994.

[4] P. Buchholz. Markovian process algebra. In U. Herzog and M. Rettelbach, editors, *Proceedings of the 2nd Workshop on Process Algebra and Performance Modeling*, pages 11–30, University of Erlangen, July 1994.

[5] P. Buchholz. Exact performance equivalence: An equivalence relation for stochastic automata. *Theoretical Computer Science*, 215:263–287, 1999.

[6] R. J. van Glabbeek, S. A. Smolka, and B. Steffen. Reactive, generative, and stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, August 1995.

[7] N. Götz, U. Herzog, and M. Rettelbach. Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebra. In L. Donatiello and R. Nelson, editors, *Performance '93*, volume 729 of *Lecture Notes in Computer Science*, pages 121–146, Berlin, 1993. Springer-Verlag.

[8] J. Hillston. The nature of synchronization. In U. Herzog and M. Rettelbach, editors, *Proceedings of the 2nd Workshop on Process Algebra and Performance Modeling*, pages 51–70, University of Erlangen, July 1994.

[9] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[10] 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.

[11] B. Jonsson, K. G. Larsen, and W. Yi. Probabilistic extensions of process algebras. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*. Elsevier, 2001.

[12] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Springer-Verlag, New York, 1976.

[13] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1986.

[14] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, September 1991.

[15] 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*, pages 137–151, 1987.

[16] B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. *Performance Evaluation Review*, 13:147–154, 1985.

[17] B. Plateau and K. Atif. Stochastic automata networks for modeling parallel systems. *IEEE Transactions on Software Engineering*, 17:1093–1108, 1991.

[18] B. Plateau and J. M. Fourneau. A methodology for solving Markov models of parallel systems. *Journal of Parallel and Distributed Computing*, 12:370–387, 1991.

[19] E. W. Stark and G. Pemmasani. Implementation of a compositional performance analysis algorithm for probabilistic I/O automata. In *Proceedings of 1999 Workshop on Process Algebra and Performance Modeling (PAPM99)*. Prensas Universitarias de Zaragoza, September 1999.

[20] E. W. Stark and S. Smolka. Compositional analysis of expected delays in networks of probabilistic I/O automata. In *Proc. 13th Annual Symposium on Logic in Computer Science*, pages 466–477, Indianapolis, IN, June 1998. IEEE Computer Society Press.

[21] S.-H. Wu, S. A. Smolka, and E. W. Stark. Compositionality and full abstraction for probabilistic I/O automata. In *Proceedings of CONCUR '94 — Fifth International Conference on Concurrency Theory*, Uppsala, Sweden, August 1994.

[22] S.-H. Wu, S. A. Smolka, and E. W. Stark. Composition and behaviors of probabilistic I/O automata. *Theoretical Computer Science*, 176(1-2):1–38, 1997.