

Branching Bisimulation Congruence for Probabilistic Systems

Nikola Trčka^{a,1} and Sonja Georgievska^{a,2}

^a *Department of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands*

Abstract

The notion of branching bisimulation for the alternating model of probabilistic systems is not a congruence with respect to parallel composition. In this paper we first define another branching bisimulation in the more general model allowing consecutive probabilistic transitions, and we prove that it is compatible with parallel composition. We then show that our bisimulation is actually the coarsest congruence relation included in the existing branching bisimulation when restricted to the alternating model.

Keywords: probabilistic systems, branching bisimulation, alternating model, congruence

1 Introduction

Labeled transition systems are a well established formalism for modeling of the qualitative aspects of systems, focusing on the behavioral part. A transition system is a directed graph where nodes represent states of the system, and labels on edges represent the actions that the system performs when moving from one state to another. Reasoning on transition systems is usually done by relating states according to some behavioral equivalence. If two states agree on every step they can take, then they are equivalent modulo *strong bisimulation* equivalence [15,14]. If a system can perform internal, unobservable steps, then strong bisimulation becomes less appropriate because it equates too few states. To solve this problem weaker equivalences are introduced that abstract from internal steps but require that the observable steps are exactly simulated. The two most commonly used equivalences of this type are *weak* bisimulation [14] and *branching* bisimulation [8,6] equivalence.

¹ Email: n.trcka@tue.nl

² Email: s.georgievska@tue.nl

The difference between the two is that the latter is known to preserve the branching structure of a transition system better [8].

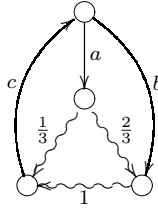


Fig. 1. A generalized alternating model of probabilistic systems

To model random behavior in systems several probabilistic extensions of transition systems have been proposed, different in the way they combine probability with non-determinism [5]. One of the models that have attracted attention is the *alternating* model introduced in [21]. This model makes a distinction between non-deterministic states, in which non-deterministic choice is resolved, and probabilistic states, in which probabilistic (random) choice is resolved according to a given distribution. In the non-strict version of the model [16] it is only required that probabilistic states are succeeded by nondeterministic ones; in the strictly alternating model [10] an action transition must also end in a probabilistic state. In this paper we consider a more general alternating model in which consecutive probabilistic steps are also allowed. An example of this system is depicted in Figure 1.

The notion of branching bisimulation in the probabilistic framework was first introduced in [18] for the *non-alternating* model (a transition system in which every action transition leads to a probability distribution of states instead to a single state [17]). Branching bisimulation was also defined for the *fully-probabilistic model* in [4]. This model, however, disallows any non-determinism by attaching probabilities to action labels so that the probability of all outgoing transitions from a state sums up to one. For the strictly alternating model a definition of branching bisimulation was given in [2]. By using colored traces, as in the original definition of non-deterministic branching bisimulation [8], they proved that their notion relates exactly those states that are intuitively branching bisimilar. Although originally defined for the strictly alternating model only, the same definition applies in the (non-strictly) alternating setting, too. They also point out that their branching bisimulation is incomparable to the one of [18].

Transition systems and their extensions are mostly used as a semantical model of some process algebra [3,11,13]. For a behavioral equivalence to be useful in a process algebraic setting, it is essential that it is compatible with all the operators under consideration, i.e. that it is a congruence relation. In the non-deterministic setting branching bisimulation is known to have the congruence property (with a slight adaptation needed for the compatibility with alternative composition) [8]. In the probabilistic setting, however, this issue has only been considered in [18] for the non-alternating model. In fact, branching bisimulation of [2] for the strictly alternating model fails to be a congruence for parallel composition [1]. Consider the following example.

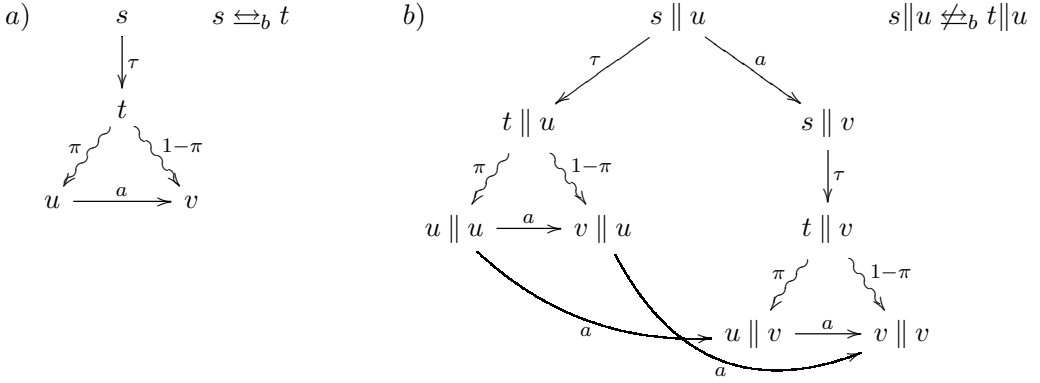


Fig. 2. Branching bisimulation of [2] is not a congruence for parallel composition

Example 1.1 States s and t in the alternating model from Figure 2a are branching bisimilar; each one can perform a with probability π and deadlock, or just deadlock with probability $1-\pi$ (τ is an internal step). However, states $s \parallel u$ and $t \parallel u$ are not branching bisimilar, as is easily seen from Figure 2b. State $s \parallel u$ can perform an action a and proceed to state $s \parallel v$, from which action a can be observed exactly with probability π . State $t \parallel u$, on the other hand, by executing action a (possibly preceded by τ or probabilistic steps) cannot at all reach a state that behaves as $s \parallel v$, i.e. that performs a with probability π . (Note that the usual addition of the root condition which solves the problem for alternative composition does not solve the problem here.)

The standard parallel composition for the alternating model, as defined in [10], is based on the resource oriented view of non-determinism: a non-deterministic choice is deduced only when the user knows which resources are offered. This means that parallel components must resolve their probabilistic choices first, and then start interleaving. In Figure 2b state $t \parallel u$ has to make the random choice in t before doing the action a from u , while state $s \parallel u$ can choose to perform a immediately. This leads to a problem since, as previously explained, the branching structure of the two states is not the same. Note that, as the definition of parallel composition is the same for any version of the alternating model, the problem remains when switching from one setting to another.

There are two ways of solving the congruence problem for a given equivalence satisfying the (rest of the) desired properties. The first way is to adapt the problematic operator, i.e. change its semantics. This approach can sometimes be hard to motivate since it also affects other behavioral equivalences. The other, more popular, approach is to strengthen the equivalence in consideration, preferably in such a way that the obtained notion is the coarsest congruence contained in the original equivalence. For branching bisimulation this idea has been employed many times, for instance in the extensions of process theory with priorities [7] and with timing [20].

In this paper we define the notion of branching bisimulation for the (generalized) alternating model, and prove this notion compatible with parallel composition. In

our bisimulation probabilistic transitions can be taken as silent when action behavior is simulated, but they themselves must be simulated in the strong sense (abstracting only from intermediate probabilistic transitions which are allowed in our model). This restriction implies that states s and t from Figure 2a are not branching bisimilar in our definition. Our motivation comes from the fact that probabilistic steps are always given priority in parallel contexts, and therefore a probabilistic state that leads to different classes cannot be related to a (non-prioritized) non-deterministic state. In this manner we follow a similar line of reasoning as in [7] where non-probabilistic branching bisimulation is adapted to become compatible with action priorities.

In contrast to [2] our definition does not need the notion of scheduler. However, by giving an alternative definition involving schedulers we prove that our bisimulation, when applied to the alternating model, is actually the coarsest equivalence relation included in the branching bisimulation of [2] that is compatible with parallel composition. We thus solve the congruence problem of [2] by using the second approach described above.

When used for simplification of systems, our bisimulation appears to be too strong at first since, in general, it eliminates less τ 's than the one from [2]. However, the examples we provide illustrate that the equivalence is still powerful enough for the elimination of internal non-determinism. In other words, the non-determinism that remains after the branching bisimulation reduction procedure is actual, and does not artificially persist because of our decision to preserve those τ 's that precede probabilistic choices. Note that the remaining τ 's can later be ignored when the congruence property is no longer needed (e.g. for performance analysis of a closed system).

The structure of the paper is as follows. In Section 2 we introduce probabilistic transition systems as the generalized alternating model, and we define the notion of branching bisimulation. In Section 3 we define parallel composition without synchronization for our model, and prove that our branching bisimulation is a congruence relation with respect to this operator. In Section 4 we show that our bisimulation is the smallest congruence in the branching bisimulation from [2]. Finally, the last section gives some conclusions and some directions for future work.

2 Model and Branching Bisimulation

In this section we define our model, a probabilistic transition system, as the generalized alternating system. We then define the notion of branching bisimulation for this model. Throughout the rest of the paper we presuppose a set A of actions that includes the special action τ .

Definition 2.1 [Probabilistic transition system] A *probabilistic transition system* is a tuple $(S, \rightarrow, \rightsquigarrow)$, where S is a finite set of states, $\rightarrow \subseteq S \times A \times S$ is an *action transition relation*, and $\rightsquigarrow \subseteq S \times (0, 1] \times S$ is a *probabilistic transition relation* such that, for all $s \in S$, $\sum_{(s, \pi, t) \in \rightsquigarrow} \pi \in \{0, 1\}$. Moreover, for a given $s \in S$, if there exist $t \in S$ and $\pi \in (0, 1]$ such that $(s, \pi, t) \in \rightsquigarrow$, then $(s, a, t') \notin \rightarrow$ for all $t' \in S$, $a \in A$.

We write $s \xrightarrow{a} t$ rather than $(s, a, t) \in \rightarrow$, and $s \xrightarrow{\pi} t$ rather than $(s, \pi, t) \in \rightsquigarrow$. We also write $s \rightsquigarrow t$ when there exists a $\pi \in (0, 1]$ such that $s \xrightarrow{\pi} t$. Note that the relations \rightarrow and \rightsquigarrow induce a partitioning of the set S into two sets S_n and S_p , denoting *nondeterministic* and *probabilistic* states, respectively. Formally, $S_p = \{s \in S \mid \exists \pi \in (0, 1], \exists t \in S, s \xrightarrow{\pi} t\}$ and $S_n = S \setminus S_p$. Note that a state without any outgoing transitions, i.e. a deadlock state, belongs to S_n .

We now introduce some preliminaries necessary for defining our branching bisimulation.

To be able to use some results from Markov chains theory, we first represent the probabilistic transition relation as a matrix. Given a probabilistic transition system $(S, \rightarrow, \rightsquigarrow)$, we define the matrix $P : S \times S \rightarrow [0, 1]$ by:

$$P(s, t) = \begin{cases} \pi, & \text{if } s \xrightarrow{\pi} t \\ 1, & \text{if } s = t \text{ and } s \in S_n \\ 0, & \text{otherwise.} \end{cases}$$

Since consecutive probabilistic transitions exist in $(S, \rightarrow, \rightsquigarrow)$, we next compute the overall probabilities. P is clearly a probabilistic matrix, which implies that there exists its *Cesaro limit* [12], defined by $\Pi = \lim_{n \rightarrow \infty} (I + P + P^2 + \dots + P^n)/(n+1)$. For $t \in S_n$, $\Pi(s, t)$ denotes the probability with which s reaches t via zero or more probabilistic transitions; for $t \in S_p$, $\Pi(s, t)$ holds the probability that the system is found in t in the long run of probabilistic transitions. We write $s \xrightarrow{\pi}_{\text{ces}} t$ when $\Pi(s, t) = \pi > 0$, and $s \rightsquigarrow_{\text{ces}} t$ if there is such π . For a set $D \subseteq S$, we define $\Pi(s, D) = \sum_{t \in D} \Pi(s, t)$.

Remark 2.2 If $(S, \rightarrow, \rightsquigarrow)$ is a probabilistic transition system belonging to the alternating model, then matrices Π and P coincide.

In the sequel the following notation will be used: $s \dashrightarrow t$ denotes $s \xrightarrow{\tau} t$ or $s \rightsquigarrow_{\text{ces}} t$; $s \xrightarrow{(a)} t$ denotes $s \xrightarrow{a} t$, or $s = t$ and $a = \tau$. Given an equivalence relation $R \subseteq S \times S$ and a state $s \in S$, S/R denotes the partitioning of S induced by R , and $[s]_R$ denotes the equivalence class of s with respect to R .

Definition 2.3 [Branching bisimulation] Let $(S, \rightarrow, \rightsquigarrow)$ be a probabilistic transition system. An equivalence relation $R \subseteq S \times S$ is a *branching bisimulation* iff, for every $(s, t) \in R$, the following holds:

- (i) if $s \in S_n$ and $s \xrightarrow{a} s'$ for some $a \in A$ and $s' \in S$, then there exist $t_0, \dots, t_n, t' \in S$ such that $t_0 = t$, $t_0 \dashrightarrow t_1 \dashrightarrow \dots \dashrightarrow t_n \xrightarrow{(a)} t'$, $(s, t_i) \in R$ for all $0 \leq i \leq n$, and $(s', t') \in R$; and
- (ii) if $s \in S_p$, then
 - (a) $\Pi(s, D) = \Pi(t, D)$ for all $D \in S/R$; and
 - (b) $\Pi(s, s) > 0$ implies $t \in S_p$.

We write $s \sim_b t$ if $(s, t) \in R$ for some branching bisimulation relation R .

The first condition of branching bisimulation says that, as usual, when action

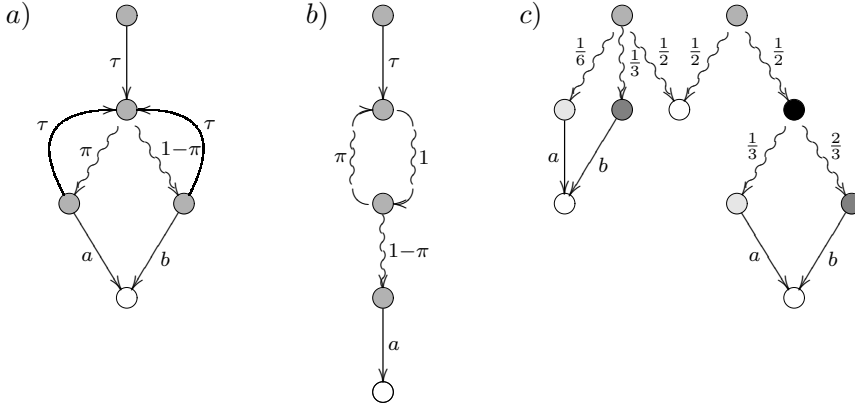


Fig. 3. Probabilistic transition systems with branching bisimulations

execution is simulated, it can possibly be preceded by a sequence of internal transitions connecting states within the same equivalence class. For an internal step we take either a step labeled with the τ action, or a probabilistic Cesaro transition. The second condition ensures that probabilistic steps are (more) strongly simulated. Generally we do not allow any τ 's in this case, and the only abstraction is the one induced by using the matrix Π instead of P . Due to this condition, in our setting states s and t of Figure 2a cannot be related. The extra requirement in the second condition, namely that $\Pi(s, s) > 0$ implies $t \in S_p$, prevents a deadlock state to be related to a state that is in a class of consecutive probabilistic states, from which no other class is reachable. This requirement is needed to establish the congruence property since such a class blocks any action in a parallel context.

It is not hard to show that the conditions of Definition 2.3 are compositional, and that the transitive closure of the union of all branching bisimulations is itself a branching bisimulation. From the last it follows easily that \sim_b is the maximal branching bisimulation.

Remark 2.4 Note that the type of abstraction induced by Π is more of a weak bisimulation type than of a branching bisimulation type since it allows intermediate probabilistic steps to be in different equivalence classes. It is, however, not hard to show that these intermediate steps are all branching bisimilar whenever the last state is non-deterministic and branching bisimilar to the first one. From this it follows that our branching bisimulation satisfies the stuttering property or, in other words, that using the relation \sim instead of \sim_{ces} in the first condition would lead to the same notion of branching bisimilarity.

The following illustrates our notion of branching bisimulation.

Example 2.5 Figure 3 depicts three probabilistic transition systems with branching bisimulations on them; bisimilar states are given the same color. Figures 3a and 3b illustrate the application of the first condition. They also show that not every state that guards a probabilistic choice with a τ must be in a different bisimulation class. The second and the third state in Figure 3b are bisimilar because Π ensures that these states must both leave the loop eventually with probability

one (the loop is not closed). Figure 3c shows another case, where Π abstracts from intermediate probabilistic transitions, and illustrates the point made in the above remark.

3 Parallel Composition

In this section we define the merge operator parallel composition without synchronization for probabilistic transition systems, and prove that \sim_b is compositional with respect to this operator. (For now we omit synchronization, since it does not add complexity to the problem).

Definition 3.1 [Merge] The operation *merge* transforms a probabilistic transition system $(S, \rightarrow, \rightsquigarrow)$ into the probabilistic transition system $(S \times S, \rightarrow, \rightsquigarrow)$ ³ defined as follows (we standardly write $s \parallel t$ instead of (s, t)):

- (i) $s \parallel t \xrightarrow{a} u$ iff $s, t \in S_n$ and
 - there exists $s' \in S$ such that $s \xrightarrow{a} s'$ and $u = s' \parallel t$, or
 - there exists $t' \in S$ such that $t \xrightarrow{a} t'$ and $u = s \parallel t'$; and
- (ii) for all $\pi \in (0, 1]$, $s \parallel t \rightsquigarrow^\pi u$ iff
 - $t \in S_n$, there exists $s' \in S$ such that $s \rightsquigarrow^\pi s'$, and $u = s' \parallel t$, or
 - $s \in S_n$, there exists $t' \in S$ such that $t \rightsquigarrow^\pi t'$, and $u = s \parallel t'$, or
 - there exist $\pi_1, \pi_2 \in (0, 1]$ and $s', t' \in S$, such that $s \rightsquigarrow^{\pi_1} s'$, $t \rightsquigarrow^{\pi_2} t'$, $\pi = \pi_1 \pi_2$, and $u = s' \parallel t'$.

The rest of this section is devoted to proving that branching bisimulation is a congruence for the merge operator. We need some lemmas for that. The first lemma establishes the connection between P and Π on the level of elements. Its proof can be found in any book on standard Markov chain theory (see e.g. [12]).

Lemma 3.2 *Let $s, t \in S$. Then*

- (i) *if $\Pi(s, t) > 0$, then $\Pi(t, t) > 0$;*
- (ii) *if $P(s, s) = 1$, then $\Pi(s, s) = 1$.*

The next lemma shows that a probabilistic state related to a non-deterministic state can reach only its own class via a Cesaro transition; moreover, the transition must be to a non-deterministic state.

Lemma 3.3 *Let $R \subseteq S \times S$ be a branching bisimulation and let $s \in S_p$. If $(s, t) \in R$ for some $t \in S_n$, then $\Pi(s, [s]_R) = 1$; moreover, for all $r \in S$ such that $\Pi(s, r) > 0$, it holds that $r \in S_n$.*

Proof. From $t \in S_n$ we have $P(t, t) = 1$, and so, by Lemma 3.2(ii), we have $\Pi(t, t) = 1$. Since $(s, t) \in R$, we have $\Pi(s, [s]_R) = 1$. This means that any state reachable from s via a Cesaro transition belongs to $[s]_R$. For the second part of the lemma, let us assume that $r \in S$ is such that $\Pi(s, r) > 0$. Then $\Pi(r, r) > 0$ by

³ We overload symbols \rightarrow and \rightsquigarrow for simplicity

Lemma 3.2(i). Suppose that $r \in S_p$. By Definition 2.3 it follows that $t \in S_p$, which contradicts the assumption that $t \in S_n$. Therefore, $r \in S_n$. \square

Finally, the last lemma shows how the Cesaro limit of the merge can be calculated by the Cesaro limits of the individual components.

Lemma 3.4 *For all $s, t, s', t' \in S$, $\Pi(s \parallel t, s' \parallel t') = \Pi(s, s') \cdot \Pi(t, t')$.*

Proof. Note that the probabilistic transition matrix of the merge is just the Kronecker product [9] of the probabilistic transition matrices of the parallel components. The proof then easily follows from the fact that matrix powers distribute over Kronecker products. \square

We can now prove the main theorem of this section.

Theorem 3.5 (Congruence theorem) *Branching bisimulation is a congruence with respect to the merge operator.*

Proof. Let $R = \{(s \parallel u, t \parallel v) \mid s, t, u, v \in S, s \sim_b t, u \sim_b v\}$. We show that R is a branching bisimulation relation. It is clearly an equivalence relation.

Let $s, t, u, v \in S$ be such that $(s \parallel u, t \parallel v) \in R$.

(i) Suppose that $s \parallel u \xrightarrow{a} r$ for some $r \in S \times S$ and $a \in A$. Without loss of generality we assume that $s \xrightarrow{a} s'$ for some $s' \in S$, therefore $u \in S_n$ and $r = s' \parallel u$. From $s \sim_b t$ it follows that there exist $t_0, \dots, t_n, t' \in S$ such that $t_0 = t, t \sim_b t_i$ for $0 \leq i \leq n$, $s' \sim_b t'$, and $t_0 \dashrightarrow t_1 \dashrightarrow \dots \dashrightarrow t_n \xrightarrow{(a)} t'$. By induction on n we now show that there exist $\bar{t}_0 = t, \bar{t}_1, \dots, \bar{t}_k, \bar{v}_0 = v, \bar{v}_1, \dots, \bar{v}_k$, and $\bar{r} \in S \times S$, such that $(t \parallel v, \bar{t}_i \parallel \bar{v}_i) \in R$ for $0 \leq i \leq k$, $(r, \bar{r}) \in R$, and $\bar{t}_0 \parallel \bar{v}_0 \dashrightarrow \bar{t}_1 \parallel \bar{v}_1 \dashrightarrow \dots \dashrightarrow \bar{t}_k \parallel \bar{v}_k \xrightarrow{(a)} \bar{r}$. We distinguish two cases: when $v \in S_n$ and when $v \in S_p$. Suppose first that $v \in S_n$.

- If $n = 0$, then there is a $t' \in S$ such that $t \xrightarrow{(a)} t'$ and $t' \sim_b s'$. From this $t \parallel v \xrightarrow{(a)} t' \parallel v$.
- Suppose that $n > 0$. If $t_0 \xrightarrow{\tau} t_1$, then $t_0 \parallel v \xrightarrow{\tau} t_1 \parallel v$. If $t_0 \rightsquigarrow_{\text{ces}} t_1$, then $t_0 \parallel v \rightsquigarrow_{\text{ces}} t_1 \parallel v$, by Lemma 3.4 and because $\Pi(v, v) = 1$. The rest follows by the inductive assumption.

Suppose now that $v \in S_p$. From $v \sim_b u$ and Lemma 3.3, it follows that there exists a $\bar{v} \in S_n$ such that $\Pi(v, \bar{v}) > 0$ and $v \sim_b \bar{v}$.

- If $n = 0$, then there is a $t' \in S$ such that $t \xrightarrow{(a)} t'$ and $t' \sim_b s'$. If $t = t'$, then $t \parallel v \xrightarrow{(a)} t' \parallel v$. If $t \neq t'$, then by Lemma 3.4 we have $t \in S_n$, $\Pi(t, t) = 1$, and $t \parallel v \rightsquigarrow_{\text{ces}} t \parallel \bar{v} \xrightarrow{(a)} t' \parallel \bar{v}$.
- Suppose that $n > 0$. If $t_0 \xrightarrow{\tau} t_1$, then $t_0 \parallel v \rightsquigarrow_{\text{ces}} t_0 \parallel \bar{v} \xrightarrow{\tau} t_1 \parallel \bar{v}$. If $t_0 \rightsquigarrow_{\text{ces}} t_1$, then $t_0 \parallel v \rightsquigarrow_{\text{ces}} t_1 \parallel \bar{v}$ by Lemma 3.4. In both cases the rest follows by the inductive assumption.

(ii) Suppose that $s \parallel u$ is a probabilistic state.

- Let $p, q \in S$ and $D = [p \parallel q]_R$. Then, using Lemma 3.4, we have $\Pi(s \parallel u, D) = \sum_{\bar{p} \parallel \bar{q} \in D} \Pi(s \parallel u, \bar{p} \parallel \bar{q}) = \sum_{\bar{p} \parallel \bar{q} \in D} \Pi(s, \bar{p}) \Pi(u, \bar{q}) = \sum_{\bar{p} \sim_b p, \bar{q} \sim_b q} \Pi(s, \bar{p}) \Pi(u, \bar{q}) = (\sum_{\bar{p} \sim_b p} \Pi(s, \bar{p})) (\sum_{\bar{q} \sim_b q} \Pi(u, \bar{q})) = (\sum_{\bar{p} \sim_b p} \Pi(t, \bar{p})) (\sum_{\bar{q} \sim_b q} \Pi(v, \bar{q})) = \sum_{\bar{p} \sim_b p, \bar{q} \sim_b q} \Pi(t, \bar{p}) \Pi(v, \bar{q}) = \sum_{\bar{p} \parallel \bar{q} \in D} \Pi(t \parallel v, \bar{p} \parallel \bar{q}) = \Pi(t \parallel v, D)$.
- Without loss of generality we assume that $s \in S_p$. Suppose $\Pi(s \parallel u, s \parallel u) > 0$ and $t \parallel v$ is nondeterministic. Then $t \in S_n$ and $v \in S_n$. If $\Pi(s, s) > 0$, then $t \in S_p$. If $\Pi(s, s) = 0$, then because of Lemma 3.4, $\Pi(s \parallel u, s \parallel u) = \Pi(s, s) \Pi(u, u) = 0$. In both cases we obtain a contradiction. Therefore, $t \parallel v \in S_p$. \square

4 The Coarsest Congruence

In this section we show that our branching bisimulation is the coarsest congruence included in the branching bisimulation of [2]. We proceed with definitions involved in formalizing schedulers, necessary for the comparison of the two notions of branching bisimulation.

Definition 4.1 [Paths] A *finite path* (from a state s) is a sequence $s_0 \ l_1 \ s_1 \ \dots \ l_k \ s_k$ such that $s = s_0$, $s_i \in S$, $l_i \in (0, 1] \cup A$, and $s_{i-1} \xrightarrow{l_i} s_i$ or $s_{i-1} \rightsquigarrow_{\text{ces}}^{l_i} s_i$ for all $1 \leq i \leq k$. An *infinite path* from a state $s \in S$ is an infinite sequence $s_0 \ l_1 \ s_1 \ \dots$ such that $s = s_0$, $s_i \in S$, $l_i \in (0, 1] \cup A$, and $s_{i-1} \xrightarrow{l_i} s_i$ or $s_{i-1} \rightsquigarrow_{\text{ces}}^{l_i} s_i$ for all $1 \leq i$. A *path* is a finite or infinite path. The set of all finite paths is denoted as Paths_f .

Given a finite path $c = s_0 \ l_1 \ s_1 \ \dots \ l_k \ s_k$, we define $\text{last}(c) = s_k$. The probability of c is the product of all probability labels found on it, if any, or 1 otherwise, i.e.

$$\text{Prob}(c) = \begin{cases} \prod_{l_i \in (0, 1]} l_i, & \text{if } l_j \in (0, 1] \text{ for some } 1 \leq j \leq k \\ 1, & \text{otherwise.} \end{cases}$$

We now introduce the notion of scheduler; a scheduler is used to resolve non-determinism.

Definition 4.2 [Scheduler] A *scheduler* is a function $\sigma: \text{Paths}_f \mapsto (\rightarrow \cup \{\perp\})$, such that, if $\sigma(c) = s \xrightarrow{a} s'$ for some $s, s' \in S$ and $a \in A$, then $\text{last}(c) = s$.

If a path ends with a non-deterministic state, a scheduler schedules the next action transition or it schedules nothing (i.e. \perp). If a path ends with a probabilistic state, a scheduler schedules nothing (\perp) since in this case the next state is determined by the probability distribution.

Definition 4.3 Let σ be a scheduler. A *scheduled path* by σ is a finite path $s_0 \ l_1 \ s_1 \ \dots \ s_k$ where, for each $0 \leq i < k$, $s_i \in S_n$ implies $\sigma(s_0 \ l_1 \ s_1 \ \dots \ s_i) = s_i \xrightarrow{l_{i+1}} s_{i+1}$, or an infinite path $s_0 \ l_1 \ s_1 \ \dots$, where for each $i \geq 0$, $s_i \in S_n$ implies $\sigma(s_0 \ l_1 \ s_1 \ \dots \ s_i) = s_i \xrightarrow{l_{i+1}} s_{i+1}$. A *maximal scheduled path* is either an infinite scheduled path, or a finite scheduled path c for which $\text{last}(c) \in S_n$ and $\sigma(c) = \perp$. The set of all maximal scheduled paths by σ is denoted by $\text{Paths}_m(\sigma)$.

Any scheduler σ induces a probability space on $\text{Paths}_m(\sigma)$ in the usual way [2].

Next, given an equivalence relation on the states, we formally define paths that silently traverse through equivalent states, perform an action and then reach another class.

Definition 4.4 Let $R \subseteq S \times S$ be an equivalence relation and let $D \in S/R$. Let $c = s_0 l_1 s_1 \dots l_k s_k$ be a finite path such that $s_k \in D$, and for all $1 \leq i \leq k-1$, $s_i \in [s_0]_R$ and $l_i \in A$ implies $l_i = \tau$. For a given $a \in A \setminus \{\tau\}$, we say that c is of type $s_0 \xrightarrow{a} D$ if $l_k = a$, and we say that c is of type $s_0 \xrightarrow{\tau} D$ if $l_k = \tau$, $l_k \in (0, 1]$, or $k = 0$. We write $s_0 \Rightarrow D$ rather than $s_0 \xrightarrow{\tau} D$.

For a scheduler σ , by $\text{Paths}_m(\sigma)_{/t \xrightarrow{a} D}$ we denote the set of all paths in $\text{Paths}_m(\sigma)$ that are of type $t \xrightarrow{a} D$.

The following lemma is needed in the comparison of the two notions of branching bisimulation.

Lemma 4.5 Let $R \subseteq S \times S$ be a branching bisimulation relation and let $(s, t) \in R$. If $s \xrightarrow{a} s'$ for some $a \in A$ and $s' \in S$, then there exists a scheduler σ such that $\text{Prob}(\text{Paths}_m(\sigma)_{/t \xrightarrow{a} [s']_R}) = 1$.

Proof. Since $s \xrightarrow{a} s'$ and $(s, t) \in R$, there exists at least one path of type $t \xrightarrow{a} [s']_R$. The proof goes by induction on the minimal number of non-deterministic states that appear in a path of type $t \xrightarrow{a} [s']_R$, not counting the last node. To formalize this, for a given path c , we define $N(c) = \{i \mid i \in S_n, i \in c, i \neq \text{last}(c)\}$ and $\min_n(t) = \min\{|N(c)|, \text{where } c \text{ is of type } t \xrightarrow{a} [s']_R\}$. The proof is by induction on $\min_n(t)$.

(i) Let $t \in S$ be such that $(s, t) \in R$ and $\min_n(t) = 0$. Then $t \in S_p$, $\Pi(t, [s']_R) > 0$, and $a = \tau$. Since $(s, t) \in R$ and $\Pi(s, [s]_R) = 1$, it follows that $\Pi(t, [t]_R) = 1$. From $\Pi(t, [s']_R) > 0$, we obtain $[s']_R = [t]_R$. The scheduler σ , defined by $\sigma(c) = \perp$ for every path c , satisfies $\text{Prob}(\text{Paths}_m(\sigma)_{/t \xrightarrow{\tau} [s']_R}) = 1$.

(ii) Suppose that, for every $t \in S$ such that $(s, t) \in R$ and $\min_n(t) = m \geq 0$, there exists a scheduler σ such that $\text{Prob}(\text{Paths}_m(\sigma)_{/t \xrightarrow{a} [s']_R}) = 1$. Let $t \in S$ be such that $(s, t) \in R$ and $\min_n(t) = m + 1$.

Suppose first that $t \in S_n$. Then either $t \xrightarrow{a} t'$, for some $t' \in S$ such that $(s', t') \in R$, or there exists a $t'' \in S$ such that $(t'', t) \in R$, $t \xrightarrow{\tau} t''$, and $\min_n(t'') = m$. In the first case, the required scheduler is defined by

$$\sigma(c) = \begin{cases} t \xrightarrow{a} t', & \text{if } \text{last}(c) = t \\ \perp, & \text{otherwise.} \end{cases}$$

In the second case, by the inductive assumption there exists a scheduler ρ , such

that $Prob(\text{Paths}_m(\rho)_{/t'' \xrightarrow{a} [s']_R}) = 1$. Define scheduler σ by

$$\sigma(c) = \begin{cases} t \xrightarrow{\tau} t'', & \text{if } \text{last}(c) = t \\ \rho(c), & \text{otherwise.} \end{cases}$$

The scheduler σ satisfies $Prob(\text{Paths}_m(\sigma)_{/t \xrightarrow{a} [s']_R}) = 1$.

Suppose now that $t \in S_p$. Then $\Pi(t, [t]_R) = 1$. Let $T = \{u \mid t \rightsquigarrow_{\text{ces}} u\}$ be the set of all states that can be reached from t by Cesaro transitions. By Lemma 3.3 it follows that $u \in S_n$ for all $u \in T$. Thus, for every $u \in T$, either there exists a $u'' \in S$ such that $u \xrightarrow{\tau} u''$, $(u, u'') \in R$, and $\min_n(u'') = m$, or there exists a $u' \in S$ such that $u \xrightarrow{a} u'$ and $(u', s') \in R$. In the first case, by the inductive assumption there exists a scheduler ρ_u that satisfies $Prob(\text{Paths}_m(\rho_u)_{/u'' \xrightarrow{a} [s']_R}) = 1$. Define schedulers ρ and σ by

$$\rho(c) = \begin{cases} u \xrightarrow{\tau} u'', & \text{if } \text{last}(c) = u, (u, u'') \in R, \text{ and } \min_n(u'') = m \\ \rho_u(c), & \text{otherwise.} \end{cases}$$

and

$$\sigma(c) = \begin{cases} u \xrightarrow{a} u', & \text{if } \text{last}(c) = u \text{ and } (u', s') \in R \\ \rho(c), & \text{otherwise.} \end{cases}$$

Using that $\Pi(t, t) = 0$, we can calculate $Prob(\text{Paths}_m(\sigma)_{/t \xrightarrow{a} [s']_R}) = \sum_{u \in T} \Pi(t, u) \cdot Prob(\text{Paths}_m(\sigma)_{/u \xrightarrow{a} [s']_R}) = \sum_{u \in T} \Pi(t, u) = 1$. \square

Note that our branching bisimulation can now be defined in terms of schedulers by replacing condition (i) of Definition 2.3 with the following condition: If $s \in S_n$ and $s \xrightarrow{a} s'$ for some $a \in A$ and $s' \in S$, then there exists a scheduler σ such that $Prob(\text{Paths}_m(\sigma)_{/t \xrightarrow{a} [s']_R}) = 1$. One direction of the proof follows immediately from Lemma 4.5, while the other direction is trivial.

For the rest of this section we restrict ourselves to the alternating model in which the probabilistic transition relation \rightsquigarrow is a subset of $S \times (0, 1] \times S_n$. Recall that in this case the matrices Π and P coincide.

For an equivalence relation $R \subseteq S \times S$ we define the probabilistic function $\mu_R: S \times S/R \mapsto [0, 1]$ by:

$$\mu_R(s, D) = \begin{cases} \frac{\Pi(s, D)}{1 - \Pi(s, [s]_R)}, & \text{if } s \in S_p, D \neq [s]_R, \text{ and } \Pi(s, [s]_R) \neq 1 \\ 0, & \text{otherwise.} \end{cases}$$

$\mu_R(s, D)$ represents the conditional probability that in one Cesaro transition s can reach class D , assuming that it leaves its own class.

We now give the definition of branching bisimulation as defined in to [2].

Definition 4.6 [Branching bisimulation according to [2]] An equivalence relation $R \subseteq S \times S$ is a *branching bisimulation* iff, for every $(s, t) \in R$ the following holds:

- (i) if $s \xrightarrow{a} s'$ for some $a \in A$ and $s' \in S$, then there exists a scheduler σ , such that $Prob(\text{Paths}_m(\sigma)_{/t \xrightarrow{a} [s']_R}) = 1$; and
- (ii) if $s \in S_p$, then there is a scheduler σ such that $\mu_R(s, D) = Prob(\text{Paths}_m(\sigma)_{/t \Rightarrow D})$ for all $D \in S/R$, $D \neq [s]_R$.

We write $s \rightleftharpoons_b t$ if there exists a branching bisimulation relation $R \subseteq S \times S$ as of [2] such that $(s, t) \in R$. The relation \rightleftharpoons_b is the maximal branching bisimulation as of [2].

The following lemma and Lemma 4.5 show that \sim_b is stronger than \rightleftharpoons_b . That it is strictly stronger follows directly from the fact that in Figure 2a we have $s \rightleftharpoons_b t$, whereas $s \not\sim_b t$.

Lemma 4.7 *Let $R \subseteq S \times S$ be a branching bisimulation. Let $(s, t) \in R$ be such that $s \in S_p$. Let σ be a scheduler such that $\sigma(c) = \perp$ for all paths c . Then $\mu_R(s, D) = Prob(\text{Paths}_m(\sigma)_{/t \Rightarrow D})$, for all $D \in S/R$ such that $D \neq [s]_R$.*

Proof. Let $D \neq [s]_R$. If $\Pi(s, [s]_R) = 0$, then $\mu_R(s, D) = \Pi(s, D) = \Pi(t, D) = Prob(\text{Paths}_m(\sigma)_{/t \Rightarrow D})$. If $\Pi(s, [s]_R) > 0$, then, since every Cesaro transition leads to a nondeterministic state, there exists an $s' \in S_n$, such that $\Pi(s, s') > 0$ and $(s, s') \in R$. Since $\Pi(s', s') = 1$, we have $\Pi(s, [s]_R) = 1$. From this, $\mu_R(s, D) = 0$ and $\Pi(t, D) = \Pi(s, D) = 0$. Then $\text{Paths}_m(\sigma)_{/t \Rightarrow D}$ is the empty set, from which it follows directly that $Prob(\text{Paths}_m(\sigma)_{/t \Rightarrow D}) = 0$. \square

The following lemma plays the central role in the coarsest congruence proof.

Lemma 4.8 *Let $s, t \in S$. Let x be an action label that does not appear in any path in $(S, \rightarrow, \rightsquigarrow)$. Let x be a new state such that $x \xrightarrow{x} \delta$, x has no other outgoing transitions, and δ has no transitions at all. If $s \parallel x \rightleftharpoons_b t \parallel x$, then $s \sim_b t$.*

Proof. Define the equivalence relation $R \subseteq S \times S$ by $R = \{(p, q) \mid p \parallel x \rightleftharpoons_b q \parallel x\}$. We prove that R is a branching bisimulation according to our definition. Let $(s, t) \in R$.

- (i) Suppose that $s \xrightarrow{a} s'$ for some $s' \in S$ and $a \in A$. Then $s \parallel x \xrightarrow{a} s' \parallel x$. Since $s \parallel x \rightleftharpoons_b t \parallel x$, there exists a scheduler σ such that $Prob(\text{Paths}_m(\sigma)_{/t \parallel x \xrightarrow{a} [s']_R \rightleftharpoons_b}) = 1$. From this it easily follows that there exists a path of type $t \xrightarrow{a} [s']_R$, which directly implies the first condition of Definition 2.3.

- (ii) Suppose now that $s \in S_p$. We show that for all $D \in S/R$, $\Pi(s, D) = \Pi(t, D)$.

Suppose first that $t \in S_n$. Since $\Pi(t, [t]_R) = 1$, it is sufficient to show that $\Pi(s, [s]_R) = 1$. We have $t \parallel x \xrightarrow{x} t$. From $s \parallel x \rightleftharpoons_b t \parallel x$ it follows that there exists a scheduler σ such that $Prob(\text{Paths}_m(\sigma)_{/s \parallel x \xrightarrow{x} [t]_R \rightleftharpoons_b}) = 1$. Assume that $\Pi(s, [s]_R) \neq 1$, i.e. that there exist $s' \in S$ and $\pi \in (0, 1)$ such that $\Pi(s \parallel x, s' \parallel x) = \pi$ and $s \parallel x \not\rightleftharpoons_b s' \parallel x$. But then $Prob(\text{Paths}_m(\sigma)_{/s \parallel x \xrightarrow{x} [t]_R \rightleftharpoons_b}) \leq 1 - \pi$ for any scheduler σ , which leads to a contradiction. We conclude that $\Pi(s, [s]_R) = 1$.

Suppose now that $t \in S_p$. Suppose first that $\Pi(s, [s]_R) > 0$. Then there exists an $s' \in S_n$ such that $\Pi(s \parallel x, s' \parallel x) > 0$ and $s \parallel x \rightleftharpoons_b s' \parallel x$. As s is related to a

non-deterministic state, similarly to the previous case, it follows that $\Pi(s \parallel x, [s \parallel x] \Leftrightarrow_b) = \Pi(s, [s]_R) = 1$. We show that $\Pi(t, [s]_R) = 1$. Assume the opposite, i.e. that $\Pi(t, [s]_R) < 1$. Then $\Pi(t \parallel x, [s \parallel x] \Leftrightarrow_b) < 1$. This means that there exists a class $D \neq [s \parallel x] \Leftrightarrow_b$ such that $\Pi(t \parallel x, D) > 0$. This contradicts the fact that $\text{Prob}(\text{Paths}_m(\sigma)_{/t \parallel x \Rightarrow D}) = \mu_R(s, D) = 0$ for some scheduler σ .

Suppose now that $\Pi(s, [s]_R) = 0$. Assume that $\Pi(t, [s]_R) > 0$. Then $\Pi(t \parallel x, [s \parallel x] \Leftrightarrow_b) > 0$. This means that there exists a $t' \in S$, $t' \neq t$, such that $\Pi(t \parallel x, t' \parallel x) > 0$ and $t \parallel x \Leftrightarrow_b t' \parallel x$. Since $t' \parallel x \in S_n$, it follows that $t' \parallel x \xrightarrow{x} t'$. This means that $\text{Prob}(\text{Paths}_m(\sigma)_{/s \parallel x \Rightarrow [t'] \Leftrightarrow_b}) = 1$ for some scheduler σ . But this implies that there is an $s' \in S$, $s' \neq s$, such that $\Pi(s \parallel x, s' \parallel x) > 0$ and $s \parallel x \Leftrightarrow_b s' \parallel x$, which contradicts the fact that $\Pi(s, [s]_R) = 0$. We conclude that $\Pi(t, [s]_R) = 0$. Let $D \in S_{/R}$ be such that $D \neq [s]_R$. Then $\mu_R(s, D) = \Pi(s, D)$ and there exists a scheduler σ such that $\text{Prob}(\text{Paths}_m(\sigma)_{/t \Rightarrow D}) = \Pi(s, D)$. Since $\Pi(t, [t]_R) = 0$, it must hold that $\text{Prob}(\text{Paths}_m(\sigma)_{/t \Rightarrow D}) = \Pi(t, D)$.

It is left to show that $\Pi(s, s) > 0$ implies $t \in S_p$. This is, however, always true because, in the alternating model, $\Pi(s, s) = 0$ for all $s \in S_p$. \square

We now prove that, when applied in the alternating model, our branching bisimulation is the largest equivalence included in \Leftrightarrow_b that is compatible with the merge operator.

Theorem 4.9 *Let $R \subseteq S \times S$ be an equivalence relation that is a congruence for the merge operator. If $R \subseteq \Leftrightarrow_b$, then $R \subseteq \sim_b$.*

Proof. Suppose that $(s, t) \in R$ and let x be as in Lemma 4.8. Then $(s \parallel x, t \parallel x) \in R$ because R is a congruence relation. Since $R \subseteq \Leftrightarrow_b$, we have $s \parallel x \Leftrightarrow_b t \parallel x$. From Lemma 4.8, it follows that $s \sim_b t$. \square

5 Conclusions and Future work

We have defined a notion of branching bisimulation for a general alternating model of probabilistic systems that allows successive probabilistic transitions. We have proved our notion to be compatible with parallel composition without synchronization. We have also shown that our bisimulation is the coarsest congruence included in the branching bisimulation of [2] (when restricted to the alternating model), thus properly solving the congruence problem of [2].

Although we have only considered the merge operator, it is easy to extend the theory with the rest of the standard operators. The only non-standard modification involves the sequential composition operator, because we have to forbid processes $x \cdot \tau$ and x to be bisimilar. This, however, can be easily solved, for example by using the same termination condition as in [7].

In this paper we worked with the (generalized) alternating model of probabilistic systems, and we did not consider e.g. the non-alternating model of [17], for which branching bisimulation exists and is a congruence [18]. In [19] an embedding from the (general) alternating to the non-alternating model is proposed, that

uses a special internal action for the purpose of translating the probabilistic transitions. This embedding, however, does not preserve parallel composition, because the parallel composition of two alternating systems does not map to the parallel composition of their individual images. One way to preserve parallel composition is to adapt the standard parallel operator from the non-alternating setting to give special treatment to the newly introduced action, i.e. to give this action a priority. This would, however, either introduce the same congruence problems as in the alternating model, or would require that the special action is the τ action, thus not obtaining a conservative extension.

In [19] there is also another type of mapping proposed, that does preserve parallel composition. This transformation can be applied only in one version of the alternating model, and is simply based on combining probabilistic transitions with action transitions that pass through them (if any). For future work we schedule to investigate how our branching bisimulation is related to the notion that is naturally obtained by propagating the notion of [18] to our setting.

In future we also plan to develop an algorithm for deciding our bisimulation, and to find a sound and complete axiomatization and study application.

Acknowledgement

We thank Suzana Andova and Jos Baeten for the useful suggestions on an earlier version of this work. We thank also the anonymous referees for their suggestions and to the participants of the QAPL'08 workshop for the interesting discussions.

References

- [1] Andova, S., J. Baeten, P. D'Argenio and T. Willemse, *A compositional merge of probabilistic processes in the alternating model*, in: *18th Nordic Workshop on Programming Theory*, Reykjavik, Iceland, 2006.
- [2] Andova, S. and T. Willemse, *Branching bisimulation for probabilistic systems: characteristics and decidability*, Theor. Comput. Sci. **356** (2006), pp. 325–355.
- [3] Baeten, J. and W. Weijland, “Process Algebra,” Number 18 in Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1990.
- [4] Baier, C. and H. Hermanns, *Weak bisimulation for fully probabilistic processes*, in: *CAV '97: Proceedings of the 9th International Conference on Computer Aided Verification* (1997), pp. 119–130.
- [5] Bartels, F., A. Sokolova and E. de Vink, *A hierarchy of probabilistic system types*, Theor. Comput. Sci. **327** (2004), pp. 3–22.
- [6] Basten, T., *Branching bisimilarity is an equivalence indeed!*, Information Processing Letters **58** (1996), pp. 141–147.
- [7] Bergstra, J., A. Ponse and M. van der Zwaag, *Branching time and orthogonal bisimulation equivalence*, Theoretical Computer Science **309** (2003), pp. 313–355.
- [8] Glabbeek, R. v. and W. Weijland, *Branching time and abstraction in bisimulation semantics*, JACM **43** (1996), pp. 555–600.
- [9] Graham, A., “Kronecker Products and Matrix Calculus With Applications,” Ellis Horwood Lim., Chichester, 1981.
- [10] Hansson, H. A., “Time and Probability in Formal Design of Distributed Systems,” Elsevier, 1994.
- [11] Hoare, C., “Communicating Sequential Processes,” Prentice Hall, Englewood Cliffs, 1985.

- [12] Kemeny, J. and J. Snell, “Finite Markov chains,” Springer, 1976.
- [13] Milner, R., “A Calculus of Communicating Systems,” Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.
- [14] Milner, R., *Operational and algebraic semantics of concurrent processes*, in: *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, Elsevier and MIT Press, 1990 pp. 1201–1242.
- [15] Park, D., *Concurrency and automata on infinite sequences*, in: *Proceedings of the 5th GI-Conference on Theoretical Computer Science*, LNCS **104** (1981), pp. 167–183.
- [16] Philippou, A., I. Lee and O. Sokolsky, *Weak bisimulation for probabilistic systems*, Lecture Notes in Computer Science **1877** (2000), pp. 334–349.
- [17] Segala, R., “Modeling and verification of randomized distributed real-time systems,” Ph.D. thesis, MIT (1995).
- [18] Segala, R. and N. Lynch, *Probabilistic simulations for probabilistic processes*, Nordic J. of Computing **2** (1995), pp. 250–273.
- [19] Segala, R. and A. Turrini, *Comparative analysis of bisimulation relations on alternating and non-alternating probabilistic models*, in: *QEST* (2005), pp. 44–53.
- [20] Trčka, N., “Silent Steps in Transition Systems and Markov Chains,” Ph.D. thesis, Eindhoven University of Technology (2007).
- [21] Vardi, M., *Automatic verification of probabilistic concurrent finite-state programs*, in: *26th FOCS* (1985), pp. 327–338.