



Alternative Characterizations of Probabilistic Trace Equivalences on Coherent Resolutions of Nondeterminism

Marco Bernardo^(✉)

Dipartimento di Scienze Pure e Applicate, Università di Urbino, Urbino, Italy
marco.bernardo@uniurb.it

Abstract. For nondeterministic and probabilistic processes, the validity of some desirable properties of probabilistic trace semantics depends both on the class of schedulers used to resolve nondeterminism and on the capability of suitably limiting the power of the considered schedulers. Inclusion of probabilistic bisimilarity, compositionality with respect to typical process operators, and backward compatibility with trace semantics over fully nondeterministic or fully probabilistic processes, can all be achieved by restricting to coherent resolutions of nondeterminism. Here we provide alternative characterizations of probabilistic trace post-equivalence and pre-equivalence in the case of coherent resolutions. The characterization of the former is based on fully coherent trace distributions, whereas the characterization of the latter relies on coherent weighted trace sets.

1 Introduction

Quantitative models of computer, communication, and software systems combine, among others, functional and extra-functional aspects of system behavior. On the one hand, these models describe system activities and their execution order, possibly admitting nondeterminism in case of concurrency phenomena or to support implementation freedom. On the other hand, they include some information about the probabilities or the timing of activities and events in which the system is involved.

In the probabilistic setting, a particularly expressive model is given by *probabilistic automata* [24], because they encompass as special cases fully nondeterministic models like labeled transition systems [21], fully probabilistic models like action-labeled variants of discrete-time Markov chains [22], and reactive probabilistic models like Markov decision processes [13]. In a probabilistic automaton, the choice among the transitions departing from the current state is nondeterministic and can be influenced by the external environment, while the choice of the next state reached by the selected transition is probabilistic and made internally by the process.

Behavioral relations [1, 4, 16, 20, 28] play a fundamental role in the analysis of probabilistic models. They formalize observational mechanisms that permit

relating models that, despite their different representations in the same mathematical domain, cannot be distinguished by external entities when abstracting from certain internal details. Moreover, they support system modeling and verification by providing a means to relate system descriptions expressed at different levels of abstraction, as well as to reduce the size of a system representation while preserving specific properties to be assessed later.

In this paper, we focus on *trace semantics* for nondeterministic and probabilistic processes represented through a variant of simple probabilistic automata. A trace is a sequence of activities labeling a sequence of transitions performed by a process, thus abstracting from branching points in the process behavior. Several execution probabilities may be associated with the same trace, each corresponding to a different resolution of nondeterminism. The discriminating power of probabilistic trace semantics thus depends on the class of schedulers used to resolve nondeterminism, but in general it turns out to be excessive. This may hamper the achievement of a number of desirable properties.

The problem with almighty schedulers yielding a demonic view of nondeterminism is well known, both for trace semantics and for testing semantics. In the case of a process given by the parallel composition of several subprocesses, or in a testing scenario where a process is composed in parallel with a test, schedulers come into play after assembling the various components. As a consequence, schedulers can solve both choices local to the individual subprocesses and choices arising from their interleaving execution. This centralized approach thus gives the possibility to make decisions in one component on the basis of those made in other components, especially in the case of history-dependent schedulers [30].

To cope with the aforementioned information leakage, the idea of distributed scheduling was proposed in [10]. Given a number of modules, i.e., of variable-based versions of automata, that interact synchronously by updating all variables during every round, for each module there are several schedulers. One of them chooses the initial and updated values for the module external variables; for each atom, intended as a cluster of variables of the module, a further scheduler chooses the initial and updated values for the private and interface variables controlled by that atom. Compose-and-schedule is thus replaced by schedule-and-compose.

Distributed scheduling was then applied in [9] to the asynchronous model of switched probabilistic input/output automata. Following the terminology of [29], given a reactive interpretation to input actions and a generative interpretation to output actions, an input scheduler and an output scheduler are considered for each automaton occurring in a system. A token passing mechanism among the automata eliminates global choices by ensuring that a single automaton at a time can select a generative output action, to which the other automata can respond with reactive input actions having the same name.

Both [10] and [9] guarantee the compositionality of the probabilistic trace-distribution equivalence of [25]. This is not a congruence with respect to parallel composition under centralized scheduling; as shown in [23], the coarsest congruence contained in that linear-time equivalence turns out to be a variant of the simulation equivalence of [27], which is a branching-time equivalence. Distributed scheduling was further investigated in [15] for interleaved probabilistic

input/output automata, a variant of switched ones in which an interleaving scheduler replaces the token passing mechanism. The examined problem was the attainment of the extremal probabilities of satisfying reachability properties under different classes of distributed schedulers (memoryless vs. history-dependent, deterministic vs. randomized), knowing that in the centralized case those probabilities are obtained when using memoryless deterministic schedulers [6].

Indeed, the overwhelming power of schedulers already shows up in the memoryless case, i.e., when neglecting the path followed to reach the current state. Under centralized scheduling, in [14] additional labels were used so that the same decisions are made by schedulers in distinct copies of the same state of a testing system, thus weakening the discriminating power of the probabilistic testing equivalences of [18, 26, 31] that, as shown in [11, 19], can be characterized in terms of branching-time, simulation-like relations. An analogous weakening result under the same class of schedulers was obtained in [3] by means of a different definition of probabilistic testing equivalence, in which success probabilities are compared in a trace-by-trace fashion rather than cumulatively.

Likewise, under memoryless schedulers, a different definition of probabilistic trace equivalence allows compositionality to be recovered without resorting to distributed scheduling. In the probabilistic trace-distribution equivalence of [25], for each resolution of either process there must exist a resolution of the other process such that the two resolutions are fully matching, in the sense that, for every trace, both resolutions feature the same probability of executing that trace. We call it *probabilistic trace post-equivalence* as the quantification over traces occurs after the quantifications over resolutions. In [3] it was proposed to exchange the order of those quantifications, which avoids hardly justifiable process distinctions and regains compositionality. Given an arbitrary trace, for each resolution of either process there must exist a resolution of the other process such that both of them exhibit the same probability of executing that trace. In this case, we speak of partially matching resolutions, as a resolution of either process can be matched by different resolutions of the other process with respect to different traces. We call the resulting relation *probabilistic trace pre-equivalence*, because the quantification over traces occurs before the quantifications over resolutions.

Congruence with respect to parallel composition, which is ensured by distributed scheduling, is not the only desirable property of probabilistic trace equivalences. In addition to compositionality with respect to other typical process operators, it is necessary to address inclusion of probabilistic bisimilarity [27] as well as backward compatibility with trace equivalences over less expressive processes such as fully nondeterministic ones [8] and fully probabilistic ones [20]. As recently shown in [2], the validity of all these properties critically depends on the capability of limiting the freedom of schedulers and can be achieved if we restrict ourselves to *coherent resolutions* of nondeterminism. Similar to [14], the basic idea is that schedulers cannot select different continuations in states of a process that are equivalent to each other, so that also the states to which they correspond in any resolution of the process have equivalent continuations.

The focus of this paper is on alternative characterizations of trace semantics. In a fully nondeterministic setting, two processes are trace equivalent iff, for each

trace α , both processes can perform α or neither can. An immediate alternative characterization is that two trace equivalent processes possess the same trace set [8], where this set can be viewed as the language accepted by the automata underlying those processes. Likewise, two fully probabilistic processes are trace equivalent iff, for each trace α , both processes can perform α with the same probability, which amounts to possessing the same set of traces each weighted with its execution probability [20], i.e., the same probabilistic language. In either case, process equivalence reduces to (possibly weighted) trace set equality.

Straightforward characterizations of that form are not possible in the case of nondeterministic and probabilistic processes, because (i) traces can have different execution probabilities in different coherent resolutions and (ii) trace semantics can be defined according to different approaches leading to probabilistic trace post-/pre-equivalences. This motivates the investigation of alternative characterizations for the two aforementioned equivalences under coherent resolutions arising from centralized, memoryless schedulers – i.e., as they were defined in [2] – which is the subject of this paper.

The construction developed in [2] to formalize the coherency constraints relies on *coherent trace distributions*, i.e., suitable families of sets of traces weighted with their execution probabilities in a given resolution. Therefore, one may expect that the coherency-based variant of the probabilistic trace-distribution equivalence of [25], i.e., probabilistic trace post-equivalence, can be characterized in terms of coherent trace distribution equality. We will show by means of an example that this is not the case. The characterization of the coherency-based variant of probabilistic trace post-equivalence relies on the equality of something stronger, which we will call *fully coherent trace distributions* and could also replace coherent trace distributions in the coherency constraints.

The coherency-based variant of the probabilistic trace pre-equivalence of [3] is less discriminating because it treats traces individually without keeping track of the resolutions in which they can be executed. We will show that it can thus be characterized through the equality of something weaker than coherent trace distributions, which we will call *coherent weighted trace sets* and is constituted by suitable sets of traces weighted with their execution probabilities. We will also illustrate by means of an example that we cannot use them to set up adequate coherency constraints. In conclusion, fully coherent trace distributions, coherent trace distributions, and coherent weighted trace sets form a hierarchy in which every layer serves a different purpose.

This paper is organized as follows. In Sect. 2 we recall simple probabilistic automata and resolutions of nondeterminism, while in Sect. 3 we recall the two probabilistic trace equivalences together with three anomalies that can be avoided by resorting to coherent resolutions. In Sect. 4 we show some properties of coherent trace distributions and coherent resolutions, which are then exploited in Sects. 5 and 6 to develop the alternative characterizations of the coherency-based variants of the two equivalences, respectively relying on the equality of fully coherent trace distributions and on the equality of coherent weighted trace sets. Finally, in Sect. 7 we provide some concluding remarks.

2 Nondeterministic and Probabilistic Models

We formalize systems featuring nondeterminism and probabilities through a variant of simple probabilistic automata [24], in which we do not distinguish between external and internal actions.

Definition 1. A nondeterministic and probabilistic labeled transition system, *NPLTS* for short, is a triple (S, A, \longrightarrow) where $S \neq \emptyset$ is an at most countable set of states, $A \neq \emptyset$ is a countable set of transition-labeling actions, and $\longrightarrow \subseteq S \times A \times \text{Distr}(S)$ is a transition relation, with $\text{Distr}(S)$ being the set of discrete probability distributions over S . ■

A transition (s, a, Δ) is written $s \xrightarrow{a} \Delta$. We say that $s' \in S$ is not reachable from s via that a -transition if $\Delta(s') = 0$, otherwise we say that it is reachable with probability $p = \Delta(s')$. The reachable states form the support of the target distribution Δ , i.e., $\text{supp}(\Delta) = \{s' \in S \mid \Delta(s') > 0\}$. An NPLTS can be depicted as a directed graph in which vertices represent states and action-labeled edges represent transitions, with states in the support of the same target distribution being linked by a dashed line and decorated with the respective probabilities when these are different from 1 (see the forthcoming Figs. 1, 2, 3, 4 and 5).

An NPLTS represents (i) a *fully nondeterministic* process when every transition has a target distribution with a singleton support, (ii) a *fully probabilistic* process when every state has at most one outgoing transition, or (iii) a Markov decision process when for each action any state has at most one outgoing transition labeled with that action implying the absence of *internal nondeterminism*.

Definition 2. Let $\mathcal{L} = (S, A, \longrightarrow)$ be an NPLTS and $s, s' \in S$. We say that the finite sequence of steps:

$$c \equiv s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \dots s_{n-1} \xrightarrow{a_n} s_n$$

is a computation of \mathcal{L} of length $n \in \mathbb{N}$ from $s = s_0$ to $s' = s_n$ compatible with trace $\alpha = a_1 a_2 \dots a_n \in A^*$, written $c \in \mathcal{CC}(s, \alpha)$, iff for each step $s_{i-1} \xrightarrow{a_i} s_i$ in c there is a transition $s_{i-1} \xrightarrow{a_i} \Delta_i$ in \mathcal{L} such that $s_i \in \text{supp}(\Delta_i)$, $1 \leq i \leq n$, where:

- $\Delta_i(s_i)$ is the execution probability of step $s_{i-1} \xrightarrow{a_i} s_i$ conditioned on the selection of transition $s_{i-1} \xrightarrow{a_i} \Delta_i$ at state s_{i-1} , or simply the execution probability of that step if \mathcal{L} is fully probabilistic.
- $\text{prob}(c) = \prod_{1 \leq i \leq n} \Delta_i(s_i)$ is the execution probability of c if \mathcal{L} is fully probabilistic, assuming that $\text{prob}(c) = 1$ when $n = 0$.
- For $C \subseteq \mathcal{CC}(s, \alpha)$, we let $\text{prob}(C) = \sum_{c \in C} \text{prob}(c)$ if \mathcal{L} is fully probabilistic, provided that no computation in C is a proper prefix of one of the others. ■

When several transitions depart from the same state s of an NPLTS \mathcal{L} , they describe a nondeterministic choice among different behaviors. A *resolution* of s is the result of a possible way of resolving nondeterministic choices starting from s , as if a *scheduler* were applied that decides which activity has to be

performed next. A resolution of nondeterminism can thus be formalized as a fully probabilistic NPLTS \mathcal{Z} with a tree-like structure, whose branching points correspond to target distributions of transitions deriving from those of \mathcal{L} .

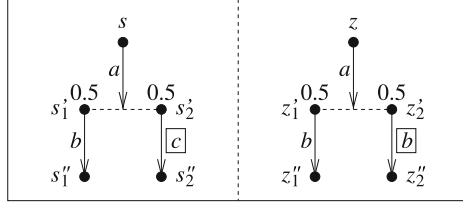


Fig. 1. Lack of injectivity breaks structure preservation

In [2] we examined two ways of resolving nondeterminism. The structure-preserving approach constructs a resolution by importing states and transitions from the original NPLTS via a *deterministic scheduler*. In a resolution of the structure-modifying approach (i) a transition can be produced by probabilistically combining transitions of the original model via a *randomized scheduler* [24], or (ii) a state can be obtained by probabilistically splitting states of the original model via an *interpolating scheduler* [12], or (iii) a combination thereof [7].

As in [2], we focus on structure-preserving resolutions arising from centralized, memoryless, deterministic schedulers. At each step, a scheduler of this kind selects one of the transitions departing from the current state, or no transitions at all thus stopping the execution. As a consequence, the resulting resolution is isomorphic to a submodel of the original model (or of its unfolding, should cycles be present), thereby preserving the structure of the original model (or of its unfolding). If the model is fully nondeterministic, each of its resolutions coincides with a computation of the model; if the model is fully probabilistic, its maximal resolution coincides with (the unfolding of) the entire model.

Following [5, 17] we introduce a *correspondence function* $corr_{\mathcal{Z}} : Z \rightarrow S$ from the acyclic state space of the resolution $\mathcal{Z} = (Z, A, \longrightarrow_{\mathcal{Z}})$ being built, to the possibly cyclic state space of the considered model $\mathcal{L} = (S, A, \longrightarrow_{\mathcal{L}})$. For each transition $z \xrightarrow{a}_{\mathcal{Z}} \Delta$, the function $corr_{\mathcal{Z}}$ must preserve the probabilities of all the states corresponding to those in $supp(\Delta)$ and must be injective over $supp(\Delta)$. In the absence of injectivity, the original structure may not be preserved in the case that the target distribution of a transition assigns the same probability to several inequivalent states. This is exemplified in Fig. 1. The correspondence function that maps z to s , z'_1 and z'_2 to s'_1 , and z''_1 and z''_2 to s''_1 would cause the rightmost NPLTS to be considered a legal resolution of the leftmost NPLTS, which is not correct as the former is not isomorphic to any submodel of the latter.

Definition 3. Let $\mathcal{L} = (S, A, \longrightarrow_{\mathcal{L}})$ be an NPLTS and $s \in S$. An acyclic NPLTS $\mathcal{Z} = (Z, A, \longrightarrow_{\mathcal{Z}})$ is a structure-preserving resolution of s , written $\mathcal{Z} \in Res_{sp}(s)$, iff there exists a correspondence function $corr_{\mathcal{Z}} : Z \rightarrow S$ such that $s = corr_{\mathcal{Z}}(z_s)$, for some $z_s \in Z$ acting as the initial state of \mathcal{Z} , and for all $z \in Z$ it holds that:

- If $z \xrightarrow{a}_{\mathcal{Z}} \Delta$ then $\text{corr}_{\mathcal{Z}}(z) \xrightarrow{a}_{\mathcal{L}} \Gamma$, with $\text{corr}_{\mathcal{Z}}$ being injective over $\text{supp}(\Delta)$ and satisfying $\Delta(z') = \Gamma(\text{corr}_{\mathcal{Z}}(z'))$ for all $z' \in \text{supp}(\Delta)$.
- At most one transition departs from z . ■

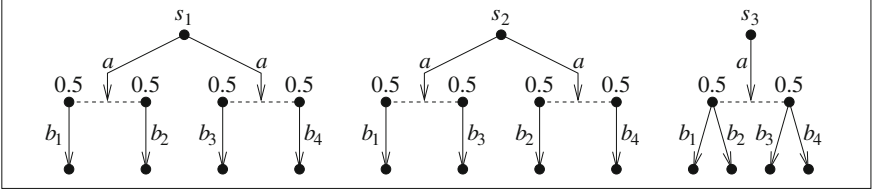


Fig. 2. $\sim_{\text{PTTr}}^{\text{post}}$ is strictly finer than $\sim_{\text{PTTr}}^{\text{pre}}: s_1 \not\sim_{\text{PTTr}}^{\text{post}} s_2 \not\sim_{\text{PTTr}}^{\text{post}} s_3, s_1 \sim_{\text{PTTr}}^{\text{pre}} s_2 \sim_{\text{PTTr}}^{\text{pre}} s_3$

3 Probabilistic Trace Equivalences and Their Anomalies

There is only one way of defining trace semantics for fully nondeterministic processes [8] and for fully probabilistic processes [20]. In contrast, this is not the case with processes featuring both nondeterminism and probabilities, as shown in the spectrum of behavioral equivalences for NPLTS models studied in [4].

The first probabilistic trace equivalence that we consider is the one of [25]. Two states are deemed equivalent when every resolution of either state is matched by a resolution of the other, in the sense that for each trace both resolutions execute that trace with the same probability. We call it probabilistic trace *post*-equivalence because the quantification over traces occurs *after* selecting the two fully matching resolutions as underlined in the definition below, where z_{s_i} denotes both the initial state of \mathcal{Z}_i and the state to which s_i corresponds.

Definition 4. Let (S, A, \longrightarrow) be an NPLTS and $s_1, s_2 \in S$. We let $s_1 \sim_{\text{PTTr}}^{\text{post}} s_2$ iff for each $\mathcal{Z}_1 \in \text{Res}_{\text{sp}}(s_1)$ there exists $\mathcal{Z}_2 \in \text{Res}_{\text{sp}}(s_2)$ such that for all $\alpha \in A^*$:

$$\text{prob}(\text{CC}(z_{s_1}, \alpha)) = \text{prob}(\text{CC}(z_{s_2}, \alpha))$$

and also the condition obtained by exchanging \mathcal{Z}_1 with \mathcal{Z}_2 is satisfied. ■

The second probabilistic trace equivalence is the one of [3], which is a congruence with respect to parallel composition. It is less restrictive than the previous equivalence because, given two states, a resolution of either state can be matched by different resolutions of the other with respect to different traces. We call it probabilistic trace *pre*-equivalence because traces are fixed *before* selecting the two partially matching resolutions.

Definition 5. Let (S, A, \longrightarrow) be an NPLTS and $s_1, s_2 \in S$. We let $s_1 \sim_{\text{PTTr}}^{\text{pre}} s_2$ iff, for all $\alpha \in A^*$, for each $\mathcal{Z}_1 \in \text{Res}_{\text{sp}}(s_1)$ there is $\mathcal{Z}_2 \in \text{Res}_{\text{sp}}(s_2)$ such that:

$$\text{prob}(\text{CC}(z_{s_1}, \alpha)) = \text{prob}(\text{CC}(z_{s_2}, \alpha))$$

and also the condition obtained by exchanging \mathcal{Z}_1 with \mathcal{Z}_2 is satisfied. ■

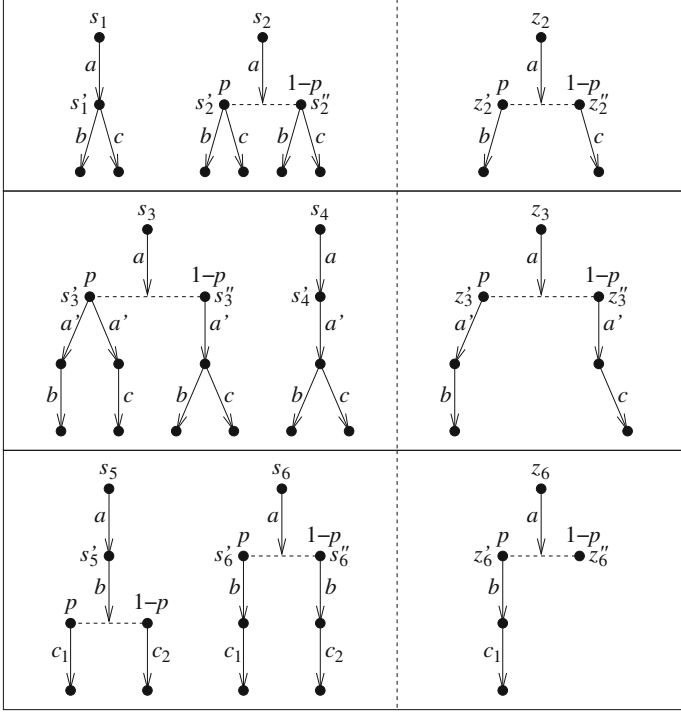


Fig. 3. Three anomalies of the probabilistic trace equivalences $\sim_{\text{PTr}}^{\text{post}}$ and $\sim_{\text{PTr}}^{\text{pre}}$

In Fig. 2 we show three NPLTS models whose initial states s_1, s_2, s_3 are pairwise distinguished by $\sim_{\text{PTr}}^{\text{post}}$ but identified by $\sim_{\text{PTr}}^{\text{pre}}$, because for all $i = 1, \dots, 4$ the probability of executing trace $a b_i$ is the same in the three models.

Although deterministic schedulers are very intuitive, the rigid preservation they ensure about the structure of the original model, together with their freedom of performing choices inconsistent with each other in states with equivalent continuations, causes the two considered probabilistic trace equivalences to be overdiscriminating. This results in the violation of a number of desirable properties (a fact that also happens with structure-modifying schedulers, but to a much lesser extent). More precisely, in [2] we showed that $\sim_{\text{PTr}}^{\text{post}}$ and $\sim_{\text{PTr}}^{\text{pre}}$:

- are not coarser than probabilistic bisimilarity under deterministic schedulers;
- are not congruences w.r.t. action prefix under deterministic schedulers;
- are not compatible with their version for fully probabilistic processes.

The first anomaly is illustrated by the two NPLTS models in Fig. 3 whose initial states are s_1 and s_2 . They are probabilistic bisimilar in the sense of [27] but $s_1 \not\sim_{\text{PTr}}^{\text{post}} s_2$ and $s_1 \not\sim_{\text{PTr}}^{\text{pre}} s_2$ because of the resolution whose initial state is z_2 , where trace $a b$ is executable with probability p instead of 1. This resolution belongs to $\text{Res}_{\text{sp}}(s_2) \setminus \text{Res}_{\text{sp}}(s_1)$ as it does not preserve the structure of the NPLTS

whose initial state is s_1 . Therefore, the two probabilistic trace equivalences do not include probabilistic bisimilarity.

The second anomaly is illustrated by the two NPLTS models in Fig. 3 whose initial states are s_3 and s_4 . After the two a -transitions, two distributions are reached that are probabilistic trace equivalent, in the sense that for each class of equivalent states they both assign the same probability to that class. However, it holds that $s_3 \not\sim_{\text{PTr}}^{\text{post}} s_4$ and $s_3 \not\sim_{\text{PTr}}^{\text{pre}} s_4$ due to the resolution whose initial state is z_3 , where trace $a a' b$ is executable with probability p instead of 1. This resolution belongs to $\text{Res}_{\text{sp}}(s_3) \setminus \text{Res}_{\text{sp}}(s_4)$ as it does not preserve the structure of the NPLTS whose initial state is s_4 . Therefore, the two probabilistic trace equivalences are not congruences with respect to the action prefix operator, which concatenates the execution of an action with a process distribution.

The third anomaly is illustrated by the two NPLTS models in Fig. 3 whose initial states are s_5 and s_6 . They are identified by the trace equivalence for fully probabilistic processes of [20], which does not use schedulers at all as in those processes there are no nondeterministic choices to be solved. However, it turns out that $s_5 \not\sim_{\text{PTr}}^{\text{post}} s_6$ and $s_5 \not\sim_{\text{PTr}}^{\text{pre}} s_6$ because $\sim_{\text{PTr}}^{\text{post}}$ and $\sim_{\text{PTr}}^{\text{pre}}$ do make use of schedulers, and schedulers may decide of stopping the execution. This is witnessed by the resolution whose initial state is z_6 – notice that the scheduler has decided to stop the execution at z_6'' – where not only trace $a b c_1$ but also trace $a b$ is executable with probability p . This resolution belongs to $\text{Res}_{\text{sp}}(s_6) \setminus \text{Res}_{\text{sp}}(s_5)$ as it does not preserve the structure of the NPLTS whose initial state is s_5 . Therefore, the two probabilistic trace equivalences are not backward compatible with the one for fully probabilistic processes.

4 Properties of Coherency

The anomalies shown in Fig. 3 are due to the freedom of schedulers of making different decisions in states enabling the same actions. In [2] we proposed to limit the excessive power of schedulers by restricting them to yield *coherent resolutions*. Intuitively, this means that, if several states in the support of the target distribution of a transition are equivalent, then the decisions made by the scheduler in those states have to be coherent with each other, so that the states to which they correspond in any resolution are equivalent as well.

The *coherency constraints* implementing this idea have been expressed in [2] by reasoning on *coherent trace distributions*, i.e., families of sets of traces weighted with their execution probabilities in a given resolution, built through the following operations.

Definition 6. Let $A \neq \emptyset$ be a countable set. For $a \in A$, $p \in \mathbb{R}$, $TD \subseteq 2^{A^* \times \mathbb{R}}$, and $T \subseteq A^* \times \mathbb{R}$ we define:

$$\begin{aligned}
a \cdot TD &= \{a \cdot T \mid T \in TD\} & a \cdot T &= \{(a\alpha, p') \mid (\alpha, p') \in T\} \\
p \cdot TD &= \{p \cdot T \mid T \in TD\} & p \cdot T &= \{(\alpha, p \cdot p') \mid (\alpha, p') \in T\} \\
tr(TD) &= \{tr(T) \mid T \in TD\} & tr(T) &= \{\alpha \in A^* \mid \exists p' \in \mathbb{R}. (\alpha, p') \in T\}
\end{aligned}$$

while for $TD_1, TD_2 \subseteq 2^{A^* \times \mathbb{R}}$ we define:

$$TD_1 + TD_2 = \begin{cases} \{T_1 + T_2 \mid T_1 \in TD_1 \wedge T_2 \in TD_2 \wedge tr(T_1) = tr(T_2)\} & \text{if } tr(TD_1) = tr(TD_2) \\ \{T_1 + T_2 \mid T_1 \in TD_1 \wedge T_2 \in TD_2\} & \text{otherwise} \end{cases}$$

where for $T_1, T_2 \subseteq A^* \times \mathbb{R}$ we define:

$$\begin{aligned}
T_1 + T_2 &= \{(\alpha, p_1 + p_2) \mid (\alpha, p_1) \in T_1 \wedge (\alpha, p_2) \in T_2\} \cup \\
&\quad \{(\alpha, p) \in T_1 \cup T_2 \mid \alpha \notin tr(T_1) \cap tr(T_2)\}
\end{aligned}$$

■

Weighted trace set addition $T_1 + T_2$ is commutative and associative, with probabilities of identical traces in the two summands being always added up for coherency purposes. In contrast, trace distribution addition is only commutative. Essentially, the two summands in $TD_1 + TD_2$ represent two families of sets of weighted traces executable in the resolutions of two states in the support of a target distribution. Every weighted trace set $T_1 \in TD_1$ is summed with every weighted trace set $T_2 \in TD_2$ – so to characterize an overall resolution – unless TD_1 and TD_2 have the same family of trace sets, in which case summation is restricted to weighted trace sets featuring the same traces for the sake of coherency. Due to the lack of associativity, in the definition below all trace distributions $\Delta(s') \cdot TD_{n-1}^c(s')$ exhibiting the same family Θ of trace sets have to be summed up first, which is ensured by the presence of a double summation.

Definition 7. Let (S, A, \longrightarrow) be an NPLTS and $s \in S$. The coherent trace distribution of s is the subset of $2^{A^* \times \mathbb{R}_{]0,1]}$ defined as follows:

$$TD^c(s) = \bigcup_{n \in \mathbb{N}} TD_n^c(s)$$

with the coherent trace distribution of s whose traces have length at most n being defined as:

$$TD_n^c(s) = \begin{cases} (\varepsilon, 1) \dagger \bigcup_{s \xrightarrow{a} \Delta} a \cdot \left(\sum_{\Theta \in tr(\Delta, n-1)} \sum_{s' \in supp(\Delta)}^{tr(TD_{n-1}^c(s')) = \Theta} \Delta(s') \cdot TD_{n-1}^c(s') \right) & \text{if } n > 0 \text{ and } s \text{ has outgoing transitions} \\ \{(\varepsilon, 1)\} & \text{otherwise} \end{cases}$$

where $tr(\Delta, n-1) = \{tr(TD_{n-1}^c(s')) \mid s' \in supp(\Delta)\}$ and the operator $(\varepsilon, 1) \dagger -$ is such that $(\varepsilon, 1) \dagger TD = \{(\varepsilon, 1)\} \cup T \mid T \in TD$. ■

In the case of a fully probabilistic NPLTS, due to the absence of nondeterminism any coherent trace distribution $TD_n^c(s)$ contains a single weighted trace set. This holds in particular for resolutions.

Proposition 1. Let (S, A, \longrightarrow) be a fully probabilistic NPLTS, $s \in S$, $n \in \mathbb{N}$. Let $A^{\leq n} = \{\alpha \in A^* \mid |\alpha| \leq n\}$. Then $TD_n^c(s) = \{(\alpha, p) \in A^{\leq n} \times \mathbb{R}_{]0,1]} \mid prob(CC(s, \alpha)) = p\}$. ■

As for the relationship between $TD_n^c(s)$ and $TD_{n-1}^c(s)$, it turns out that every element of the former contains the same traces as an element of the latter. As we will see in the next section, their probabilities may differ.

Proposition 2. *Let (S, A, \longrightarrow) be an NPLTS, $s \in S$, $n \in \mathbb{N}_{\geq 1}$. Then for all $T \in TD_n^c(s)$ there exists $T' \in TD_{n-1}^c(s)$ such that $tr(T') \subseteq tr(T)$. ■*

For the NPLTS models in Fig. 3 we have that:

- $TD^c(s'_2) = \{ \{(\varepsilon, 1)\}, \{(\varepsilon, 1), (b, 1)\}, \{(\varepsilon, 1), (c, 1)\} \} = TD^c(s''_2)$ while in the related resolution states it holds that $TD^c(z'_2) = \{ \{(\varepsilon, 1)\}, \{(\varepsilon, 1), (b, 1)\} \} \neq \{ \{(\varepsilon, 1)\}, \{(\varepsilon, 1), (c, 1)\} \} = TD^c(z''_2)$.
- $TD^c(s'_3) = \{ \{(\varepsilon, 1)\}, \{(\varepsilon, 1), (a', 1)\}, \{(\varepsilon, 1), (a', 1), (a' b, 1)\}, \{(\varepsilon, 1), (a', 1), (a' c, 1)\} \} = TD^c(s''_3)$ but $TD^c(z'_3) = \{ \{(\varepsilon, 1)\}, \{(\varepsilon, 1), (a', 1)\}, \{(\varepsilon, 1), (a', 1), (a' b, 1)\} \} \neq \{ \{(\varepsilon, 1)\}, \{(\varepsilon, 1), (a', 1)\}, \{(\varepsilon, 1), (a', 1), (a' c, 1)\} \} = TD^c(z''_3)$.
- $TD_1^c(s'_6) = \{ \{(\varepsilon, 1), (b, 1)\} \} = TD_1^c(s''_6)$ but $TD_1^c(z'_6) = \{ \{(\varepsilon, 1), (b, 1)\} \} \neq \{ \{(\varepsilon, 1)\} \} = TD_1^c(z''_6)$, which indicates that separate coherency constraints are needed relying on TD_n^c sets for every $n \in \mathbb{N}$.

Further examples in [2] show that the coherency constraints should be based on TD_n^c sets up to the probabilities they contain, i.e., the constraints should rely on $tr(TD_n^c)$ sets. Moreover, for every $n \in \mathbb{N}$, those examples call for a complete presence in each resolution of computations of length n if any, including possible shorter maximal computations. Note that trace completeness up to length n is looser than requiring resolution maximality.

Definition 8. *Let $\mathcal{L} = (S, A, \longrightarrow_{\mathcal{L}})$ be an NPLTS, $s \in S$, and $\mathcal{Z} = (Z, A, \longrightarrow_{\mathcal{Z}}) \in Res_{sp}(s)$ with correspondence function $corr_{\mathcal{Z}} : Z \rightarrow S$. We say that \mathcal{Z} is a coherent resolution of s , written $\mathcal{Z} \in Res_{sp}^c(s)$, iff for all $z \in Z$, whenever $z \xrightarrow{a}_{\mathcal{Z}} \Delta$, then for all $n \in \mathbb{N}$:*

1. $tr(TD_n^c(corr_{\mathcal{Z}}(z'))) = tr(TD_n^c(corr_{\mathcal{Z}}(z''))) \implies tr(TD_n^c(z')) = tr(TD_n^c(z''))$ for all $z', z'' \in supp(\Delta)$.
2. For all $z' \in supp(\Delta)$, the only $T \in TD_n^c(z')$ admits $\bar{T} \in TD_n^c(corr_{\mathcal{Z}}(z'))$ such that $tr(T) = tr(\bar{T})$. ■

Any complete submodel rooted at a state z of a coherent resolution turns out to be coherent too, where complete means that no state reachable from z in the resolution is cut off in the resolution submodel. Completeness is important for satisfying in particular the second coherency constraint of Definition 8.

Proposition 3. *Let $\mathcal{L} = (S, A, \longrightarrow_{\mathcal{L}})$ be an NPLTS, $s \in S$, $\mathcal{Z} = (Z, A, \longrightarrow_{\mathcal{Z}}) \in Res_{sp}^c(s)$ with correspondence function $corr_{\mathcal{Z}} : Z \rightarrow S$. Let $\mathcal{Z}'_z = (Z', A, \longrightarrow_{\mathcal{Z}'}) \in Res_{sp}^c(s)$ be the complete submodel of \mathcal{Z} rooted at $z \in Z$. Then $\mathcal{Z}'_z \in Res_{sp}^c(corr_{\mathcal{Z}}(z))$. ■*

The resolutions in Fig. 3 do *not* respectively belong to $Res_{sp}^c(s_2)$, $Res_{sp}^c(s_3)$, $Res_{sp}^c(s_6)$. We proved in Thm. 1 of [2] that the examined anomalies disappear by substituting Res_{sp}^c for Res_{sp} in Definitions 4 and 5. This replacement yields the two coherency-based probabilistic trace equivalences $\sim_{PTr}^{post,c}$ and $\sim_{PTr}^{pre,c}$ for which we will investigate alternative characterizations in the next two sections by exploiting the properties shown in Propositions 1, 2, and 3.

5 Alternative Characterization of $\sim_{\text{PTr}}^{\text{post},c}$

The definition of $\sim_{\text{PTr}}^{\text{post},c}$ essentially requires that two states have the same trace distributions. Therefore, it is natural to expect an alternative characterization of $\sim_{\text{PTr}}^{\text{post},c}$ based on the construction of Definition 7. Incidentally, this would fully justify the construction itself, given that the probabilities contained in the TD_n^c sets have not been exploited in the coherency constraints of Definition 8. However, for an NPLTS (S, A, \longrightarrow) and $s \in S$, the set $TD^c(s)$ may contain weighted traces that break coherency, hence that set cannot be used for characterization purposes.

For example, consider the NPLTS in Fig. 4. We have that:

$$TD_1^c(s_1) = \{(\varepsilon, 1), (b, 1)\} = TD_1^c(s_2)$$

and also:

$$TD_2^c(s_1) = \{(\varepsilon, 1), (b, 1), (bc, 1)\}, \{(\varepsilon, 1), (b, 1), (bd, 1)\} = TD_2^c(s_2)$$

because in the complete submodel rooted at s_1 it holds that:

$$TD_1^c(s'_1) = \{(\varepsilon, 1), (c, 1)\}, \{(\varepsilon, 1), (d, 1)\} = TD_1^c(s''_1)$$

and hence, when applying Definition 7 to compute $TD_2^c(s_1)$, according to Definition 6 the summation is restricted to weighted trace sets featuring the same traces as:

$$tr(TD_1^c(s'_1)) = \{\{\varepsilon, c\}, \{\varepsilon, d\}\} = tr(TD_1^c(s''_1))$$

Nevertheless, since:

$$\begin{aligned} TD_2^c(s'_1) &= \{(\varepsilon, 1), (c, 1), (ce_1, 1)\}, \{(\varepsilon, 1), (d, 1), (de_2, 1)\} \\ TD_2^c(s''_1) &= \{(\varepsilon, 1), (c, 1), (ce_3, 1)\}, \{(\varepsilon, 1), (d, 1), (de_4, 1)\} \end{aligned}$$

where:

$$tr(TD_2^c(s'_1)) = \{\{\varepsilon, c, ce_1\}, \{\varepsilon, d, de_2\}\} \neq \{\{\varepsilon, c, ce_3\}, \{\varepsilon, d, de_4\}\} = tr(TD_2^c(s''_1))$$

we subsequently derive that:

$$\begin{aligned} TD_3^c(s_1) &= (\varepsilon, 1) \dagger \\ &\quad (\{(b, p), (bc, p), (bce_1, p)\}, \\ &\quad \{(b, p), (bd, p), (bde_2, p)\}) + \\ &\quad \{(b, 1-p), (bc, 1-p), (bce_3, 1-p)\}, \\ &\quad \{(b, 1-p), (bd, 1-p), (bde_4, 1-p)\}) \\ &= \{(\varepsilon, 1), (b, 1), (bc, 1), (bce_1, p), (bce_3, 1-p)\}, \\ &\quad \{(\varepsilon, 1), (b, 1), (bc, p), (bd, 1-p), (bce_1, p), (bde_4, 1-p)\}, \\ &\quad \{(\varepsilon, 1), (b, 1), (bd, p), (bc, 1-p), (bde_2, p), (bce_3, 1-p)\}, \\ &\quad \{(\varepsilon, 1), (b, 1), (bd, 1), (bde_2, p), (bde_4, 1-p)\} \end{aligned}$$

whereas:

$$\begin{aligned} TD_3^c(s_2) &= \{(\varepsilon, 1), (b, 1), (bc, 1), (bce_1, p), (bce_3, 1-p)\}, \\ &\quad \{(\varepsilon, 1), (b, 1), (bd, 1), (bde_2, p), (bde_4, 1-p)\} \end{aligned}$$

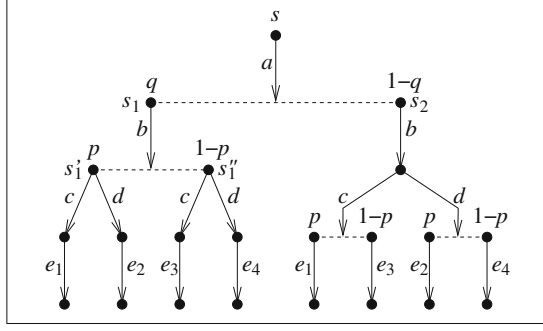


Fig. 4. Full coherency is necessary to reconcile $TD_3^c(s_1)$ and $TD_3^c(s_2)$

Therefore, in the calculation of $TD_4^c(s)$ we cannot simply sum up weighted trace sets in $TD_3^c(s_1)$ and in $TD_3^c(s_2)$ that exhibit the same traces. This is due to the presence in $TD_3^c(s_1)$ of the following two weighted trace sets:

$$\begin{aligned} &\{(\varepsilon, 1), (b, 1), (bc, p), (bd, 1-p), (bce_1, p), (bde_4, 1-p)\} \\ &\{(\varepsilon, 1), (b, 1), (bd, p), (bc, 1-p), (bde_2, p), (bce_3, 1-p)\} \end{aligned}$$

which cannot be exposed by any coherent resolution. The key observation is that coherency constraints on traces like bc and bd are ignored, hence those two weighted trace sets in $TD_3^c(s_1)$ are *not extensions* of weighted trace sets in $TD_2^c(s_1)$. Indeed, neither of those weighted trace sets in $TD_3^c(s_1)$ includes as a subset a weighted trace set in $TD_2^c(s_1)$ because of the different probabilities of the aforementioned traces in the considered sets (see the sentence before Proposition 2).

This example reveals that the construction of Definition 7, together with weighted trace set addition and trace distribution addition as provided in Definition 6, are appropriate to set up the coherency constraints in Definition 8, but not to characterize the trace distributions of coherent resolutions. To achieve this, every set $TD_n^c(s)$, with $n > 0$ and s having outgoing transitions, should *incrementally* build on $TD_{n-1}^c(s)$, in the sense that every weighted trace set in the former should include as a subset a weighted trace set in the latter (a *monotonicity*-like property stronger than the one of Proposition 2). We thus introduce a variant of coherent trace distribution, which we call *fully* coherent trace distribution.

Definition 9. Let (S, A, \longrightarrow) be an NPLTS and $s \in S$. The fully coherent trace distribution of s is the subset of $2^{A^* \times \mathbb{R}_{[0,1]}}$ defined as follows:

$$TD^{fc}(s) = \bigcup_{n \in \mathbb{N}} TD_n^{fc}(s)$$

with the fully coherent trace distribution of s whose traces have length at most n being the subset of $TD_n^c(s)$ defined as:

$$TD_n^{\text{fc}}(s) = \begin{cases} \left\{ \begin{array}{l} \{T \in TD_n^c(s) \mid \exists T' \in TD_{n-1}^{\text{fc}}(s). T' \subseteq T\} \\ \text{if } n > 0 \text{ and } s \text{ has outgoing transitions} \\ \{ \{(\varepsilon, 1)\} \} \\ \text{otherwise} \end{array} \right. \end{cases}$$

■

For the NPLTS in Fig. 4 we have that:

$$TD_3^{\text{fc}}(s_1) = \{ \{(\varepsilon, 1), (b, 1), (bc, 1), (bce_1, p), (bce_3, 1-p)\}, \\ \{(\varepsilon, 1), (b, 1), (bd, 1), (bde_2, p), (bde_4, 1-p)\} \} = TD_3^{\text{fc}}(s_2)$$

and overall $TD^{\text{fc}}(s) = \bigcup_{0 \leq n \leq 4} TD_n^{\text{fc}}(s)$ where:

$$\begin{aligned} TD_0^{\text{fc}}(s) &= \{ \{(\varepsilon, 1)\} \} \\ TD_1^{\text{fc}}(s) &= \{ \{(\varepsilon, 1), (a, 1)\} \} \\ TD_2^{\text{fc}}(s) &= \{ \{(\varepsilon, 1), (a, 1), (ab, 1)\} \} \\ TD_3^{\text{fc}}(s) &= \{ \{(\varepsilon, 1), (a, 1), (ab, 1), (abc, 1)\}, \\ &\quad \{(\varepsilon, 1), (a, 1), (ab, 1), (abd, 1)\} \} \\ TD_4^{\text{fc}}(s) &= \{ \{(\varepsilon, 1), (a, 1), (ab, 1), (abc, 1), (abce_1, p), (abce_3, 1-p)\}, \\ &\quad \{(\varepsilon, 1), (a, 1), (ab, 1), (abd, 1), (abde_2, p), (abde_4, 1-p)\} \} \end{aligned}$$

with the various sets $TD_n^{\text{fc}}(s)$ precisely capturing the trace distributions of the coherent resolutions of s .

Fully coherent trace distributions $TD_n^{\text{fc}}(s)$ coincide with coherent ones $TD_n^c(s)$ when $n \leq 2$ as a consequence of Definition 6. The example in Fig. 4 shows that, when $n \geq 3$, in general $TD_n^{\text{fc}}(s)$ cannot be recursively characterized as $TD_n^c(s)$ in Definition 7 even though each element of a fully coherent trace distribution can be expressed as a sum of elements of other fully coherent trace distributions.

Proposition 4. *Let (S, A, \longrightarrow) be an NPLTS, $s \in S$, $n \in \mathbb{N}$. If $n \leq 2$ or s has no outgoing transitions, then $TD_n^{\text{fc}}(s) = TD_n^c(s)$, otherwise each element of $TD_n^{\text{fc}}(s)$ is anyhow obtained by summing up a suitable element of $TD_{n-1}^{\text{fc}}(s')$ for every s' in the support of the target distribution of a transition of s .* ■

By virtue of Proposition 1, the equality $TD_n^{\text{fc}}(s) = TD_n^c(s)$ extends to all $n \in \mathbb{N}$, i.e., fully coherent trace distributions boil down to coherent ones, in the case of a fully probabilistic NPLTS. This holds in particular for resolutions.

Proposition 5. *Let (S, A, \longrightarrow) be a fully probabilistic NPLTS, $s \in S$, $n \in \mathbb{N}$. Then $TD_n^{\text{fc}}(s) = TD_n^c(s)$.* ■

Before presenting the characterization result, we need to revisit the coherency constraints of Definition 8 in the light of the example of Fig. 4. Suppose that each of the two terminal states reached by an e_1 -transition is replaced by a distribution with two states in its target, reached with probabilities r and $1-r$ and both featuring a nondeterministic choice between an f -transition to a terminal state and a g -transition to a terminal state. According to Definition 8, after the leftmost e_1 -transition either both f -transitions are selected or both g -transitions are selected, and the same holds after the rightmost e_1 -transition.

However, from $TD_3^c(s_1) \neq TD_3^c(s_2)$ it follows that $TD_4^c(s_1) \neq TD_4^c(s_2)$, so that on s_1 side both f -transitions may be selected while on s_2 side both g -transitions may be selected instead, or vice versa. If we consider an NPLTS whose initial state s' has a single outgoing transition, which is labeled with a and reaches a model isomorphic to the complete submodel rooted at s_2 in Fig. 4 (and extended in the aforementioned way after its only e_1 -transition), s' would be distinguished from s instead of being identified with it. The coherency constraints of Definition 8 thus need to be strengthened by substituting TD_n^{fc} for TD_n^c , in which case Thm. 1 of [2] is still valid and $\sim_{\text{PTr}}^{\text{post},c}$ and $\sim_{\text{PTr}}^{\text{pre},c}$ are modified accordingly.

The following lemma, where Proposition 1 is exploited again together with Propositions 3, 4, and 5, lays the basis for a characterization of $\sim_{\text{PTr}}^{\text{post},c}$ in terms of fully coherent trace distribution equality. In the lemma, z_s denotes both the initial state of \mathcal{Z} and the state to which s corresponds.

Lemma 1. *Let (S, A, \longrightarrow) be an NPLTS, $s \in S$, $n \in \mathbb{N}$, and $T \subseteq A^* \times \mathbb{R}_{]0,1]}$. Then $T \in TD_n^{fc}(s)$ iff there exists $\mathcal{Z} \in \text{Res}_{\text{sp}}^c(s)$ such that $TD_n^{fc}(z_s) = \{T\}$. ■*

Theorem 1. *Let (S, A, \longrightarrow) be an NPLTS and $s_1, s_2 \in S$. Then $s_1 \sim_{\text{PTr}}^{\text{post},c} s_2$ iff $TD^{fc}(s_1) = TD^{fc}(s_2)$. ■*

6 Alternative Characterization of $\sim_{\text{PTr}}^{\text{pre},c}$

As far as $\sim_{\text{PTr}}^{\text{pre},c}$ is concerned, similar to [3] we can provide an alternative characterization based on sets $T^c(s)$ built by considering all weighted traces executable from state s at once, i.e., without keeping track of the resolutions of s in which they are feasible. This is consistent with the focus of $\sim_{\text{PTr}}^{\text{pre},c}$ on individual traces rather than on trace distributions. In the definition below, the double summation used in Definition 7 in the case that $n > 0$ and s has outgoing transitions is not needed thanks to the commutativity and associativity of weighted trace set addition deriving from Definition 6.

Definition 10. *Let (S, A, \longrightarrow) be an NPLTS and $s \in S$. The coherent weighted trace set of s is the subset of $A^* \times \mathbb{R}_{]0,1]}$ defined as follows:*

$$T^c(s) = \bigcup_{n \in \mathbb{N}} T_n^c(s)$$

with the coherent weighted trace set of s whose traces have length at most n being defined as:

$$T_n^c(s) = \begin{cases} \{(\varepsilon, 1)\} \cup \bigcup_{s \xrightarrow{a} \Delta} a \cdot \left(\sum_{s' \in \text{supp}(\Delta)} \Delta(s') \cdot T_{n-1}^c(s') \right) & \text{if } n > 0 \text{ and } s \text{ has outgoing transitions} \\ \{(\varepsilon, 1)\} & \text{otherwise} \end{cases} \quad \blacksquare$$

For the NPLTS in Fig. 4 we have that $T^c(s) = \bigcup_{0 \leq n \leq 4} T_n^c(s)$ where:

$$\begin{aligned} T_0^c(s) &= \{(\varepsilon, 1)\} \\ T_1^c(s) &= \{(\varepsilon, 1), (a, 1)\} \\ T_2^c(s) &= \{(\varepsilon, 1), (a, 1), (ab, 1)\} \\ T_3^c(s) &= \{(\varepsilon, 1), (a, 1), (ab, 1), (abc, 1), (abd, 1)\} \\ T_4^c(s) &= \{(\varepsilon, 1), (a, 1), (ab, 1), (abc, 1), (abd, 1), \\ &\quad (abce_1, p), (abce_3, 1-p), (abde_2, p), (abde_4, 1-p)\} \end{aligned}$$

with the various sets $T_n^c(s)$ precisely capturing the weighted traces of the coherent resolutions of s .

It is easy to characterize $T_n^c(s)$ in the case of a fully probabilistic NPLTS. This holds in particular for resolutions.

Proposition 6. *Let (S, A, \longrightarrow) be a fully probabilistic NPLTS, $s \in S$, $n \in \mathbb{N}$. Let $A^{\leq n} = \{\alpha \in A^* \mid |\alpha| \leq n\}$. Then $T_n^c(s) = \{(\alpha, p) \in A^{\leq n} \times \mathbb{R}_{[0,1]} \mid \text{prob}(\mathcal{CC}(s, \alpha)) = p\}$. ■*

The construction in Definition 10 turns out to be monotonic, in the sense that $T_n^c(s)$ includes as a subset $T_{n-1}^c(s)$.

Proposition 7. *Let (S, A, \longrightarrow) be an NPLTS, $s \in S$, $(\alpha, p) \in A^* \times \mathbb{R}_{[0,1]}$, and $n \in \mathbb{N}_{\geq |\alpha|}$. Then $(\alpha, p) \in T_n^c(s)$ implies $(\alpha, p) \in T_{n+1}^c(s)$. ■*

The following lemma, which exploits Proposition 6 and 7, provides the basis for a characterization of $\sim_{\text{PTr}}^{\text{pre},c}$ in terms of coherent weighted trace set equality. In the lemma, z_s denotes both the initial state of \mathcal{Z} and the state to which s corresponds.

Lemma 2. *Let (S, A, \longrightarrow) be an NPLTS, $s \in S$, and $(\alpha, p) \in A^* \times \mathbb{R}_{[0,1]}$. Then $(\alpha, p) \in T^c(s)$ iff there exists $\mathcal{Z} \in \text{Res}_{\text{sp}}^c(s)$ such that $\text{prob}(\mathcal{CC}(z_s, \alpha)) = p$. ■*

Theorem 2. *Let (S, A, \longrightarrow) be an NPLTS and $s_1, s_2 \in S$. Then $s_1 \sim_{\text{PTr}}^{\text{pre},c} s_2$ iff $T^c(s_1) = T^c(s_2)$. ■*

We conclude with two remarks. The first one is that the construction in Definition 10 is identical to the one in Definition 3.5 of [3], but this should not be the case as coherency was neglected in [3]. Indeed, before Definition 3.5 of [3] the definition of $X + Y$ – i.e., $T_1 + T_2$ using the notation of this paper – should have included also $(\alpha, q_1) \in X$ and $(\alpha, q_2) \in Y$ without summing them up, otherwise the right-to-left implication in Lemma 3.7 of [3] does not hold as can be seen from trace ab of the (incoherent) resolution in Fig. 3 of this paper whose initial state is z_2 . That definition of $X + Y$ works here instead because the focus on coherency requires to always sum up the probabilities of weighted traces sharing the same trace.

The second remark is that looser coherency constraints, based on weighted trace sets rather than on trace distributions as in Definition 8, would not work. As anticipated in [2], if we used T_n^c sets instead of TD_n^c sets, then probabilistic trace equivalent NPLTS models like the ones in Fig. 5 would be told apart. Indeed, we would have $\text{tr}(T^c(s'_1)) = \{\varepsilon, b, bc_1, bc_2, bc\} = \text{tr}(T^c(s'_2))$ – whereas

$tr(TD^c(s'_1)) \neq tr(TD^c(s'_2))$ – hence in any coherent resolution of s' traces $a b c_1$, $a b c_2$, $a b c$ could only be executed with probability 0.5 if present, while s'' admits coherent resolutions in which those traces have execution probability 0.25.

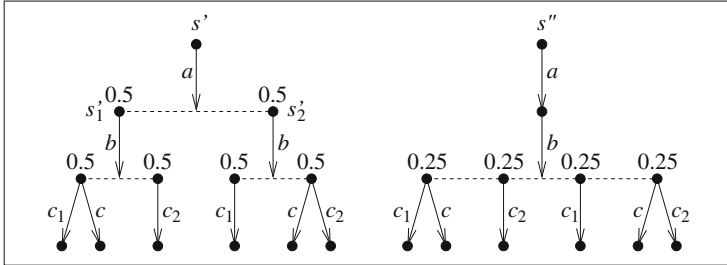


Fig. 5. Using weighted trace sets for coherency breaks probabilistic trace equivalence

7 Conclusions

Based on the notion of coherent resolution of nondeterminism, presented in [2] to avoid the anomalies of probabilistic trace semantics depicted in Fig. 3, in this paper we have provided alternative characterizations of $\sim_{PTTr}^{post,c}$ [25] and $\sim_{PTTr}^{pre,c}$ [3], respectively relying on the equality of fully coherent trace distributions and on the equality of coherent weighted trace sets. Both fully coherent trace distributions and coherent weighted trace sets are different from coherent trace distributions, introduced in [2] for defining coherency constraints on resolutions.

We plan to exploit the aforementioned alternative characterizations for studying properties and decision procedures of the two examined coherency-based probabilistic trace equivalences over nondeterministic and probabilistic processes.

References

1. Baier, C., Katoen, J.P., Hermanns, H., Wolf, V.: Comparative branching-time semantics for Markov chains. *Inf. Comput.* **200**, 149–214 (2005)
2. Bernardo, M.: Coherent resolutions of nondeterminism. In: Gribaudo, M., Iacono, M., Phung-Duc, T., Razumchik, R. (eds.) *EPEW 2019*. LNCS, vol. 12039, pp. 16–32. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44411-2_2
3. Bernardo, M., De Nicola, R., Loreti, M.: Revisiting trace and testing equivalences for nondeterministic and probabilistic processes. *Logical Methods Comput. Sci.* **10**(116), 1–42 (2014)
4. Bernardo, M., De Nicola, R., Loreti, M.: Relating strong behavioral equivalences for processes with nondeterminism and probabilities. *Theoret. Comput. Sci.* **546**, 63–92 (2014)

5. Bernardo, M., Sangiorgi, D., Vignudelli, V.: On the discriminating power of testing equivalences for reactive probabilistic systems: results and open problems. In: Norman, G., Sanders, W. (eds.) QEST 2014. LNCS, vol. 8657, pp. 281–296. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10696-0_23
6. Bianco, A., de Alfaro, L.: Model checking of probabilistic and nondeterministic systems. In: Thiagarajan, P.S. (ed.) FSTTCS 1995. LNCS, vol. 1026, pp. 499–513. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60692-0_70
7. Bonchi, F., Sokolova, A., Vignudelli, V.: The theory of traces for systems with nondeterminism and probability. In: Proceedings of the 34th ACM/IEEE Symposium on Logic in Computer Science (LICS 2019), pp. (19:62)1–14. IEEE-CS Press (2019)
8. Brookes, S.D., Hoare, C.A.R., Roscoe, A.W.: A theory of communicating sequential processes. *J. ACM* **31**, 560–599 (1984)
9. Cheung, L., Lynch, N.A., Segala, R., Vaandrager, F.: Switched PIOA: parallel composition via distributed scheduling. *Theoret. Comput. Sci.* **365**, 83–108 (2006)
10. de Alfaro, L., Henzinger, T.A., Jhala, R.: Compositional methods for probabilistic systems. In: Larsen, K.G., Nielsen, M. (eds.) CONCUR 2001. LNCS, vol. 2154, pp. 351–365. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44685-0_24
11. Deng, Y., van Glabbeek, R.J., Hennessy, M., Morgan, C.: Characterising testing preorders for finite probabilistic processes. *Logical Methods Comput. Sci.* **4**(4:4), 1–33 (2008)
12. Deng, Y., van Glabbeek, R., Morgan, C., Zhang, C.: Scalar outcomes suffice for finitary probabilistic testing. In: De Nicola, R. (ed.) ESOP 2007. LNCS, vol. 4421, pp. 363–378. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71316-6_25
13. Derman, C.: *Finite State Markovian Decision Processes*. Academic Press, Cambridge (1970)
14. Georgievska, S., Andova, S.: Probabilistic may/must testing: retaining probabilities by restricted schedulers. *Formal Aspects Comput.* **24**, 727–748 (2012). <https://doi.org/10.1007/s00165-012-0236-5>
15. Giro, S., D’Argenio, P.R.: On the expressive power of schedulers in distributed probabilistic systems. In: Proceedings of the 7th International Workshop on Quantitative Aspects of Programming Languages (QAPL 2009), ENTCS, vol. 253(3), pp. 45–71. Elsevier (2009)
16. Huynh, D.T., Tian, L.: On some equivalence relations for probabilistic processes. *Fundamenta Informaticae* **17**, 211–234 (1992)
17. Jonsson, B., Ho-Stuart, C., Yi, W.: Testing and refinement for nondeterministic and probabilistic processes. In: Langmaack, H., de Roever, W.-P., Vytupil, J. (eds.) FTRTFT 1994. LNCS, vol. 863, pp. 418–430. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58468-4_176
18. Jonsson, B., Yi, W.: Compositional testing preorders for probabilistic processes. In: Proceedings of the 10th IEEE Symposium on Logic in Computer Science (LICS 1995), pp. 431–441. IEEE-CS Press (1995)
19. Jonsson, B., Yi, W.: Testing preorders for probabilistic processes can be characterized by simulations. *Theoret. Comput. Sci.* **282**, 33–51 (2002)
20. Jou, C.-C., Smolka, S.A.: Equivalences, congruences, and complete axiomatizations for probabilistic processes. In: Baeten, J.C.M., Klop, J.W. (eds.) CONCUR 1990. LNCS, vol. 458, pp. 367–383. Springer, Heidelberg (1990). <https://doi.org/10.1007/BFb0039071>
21. Keller, R.M.: Formal verification of parallel programs. *Commun. ACM* **19**, 371–384 (1976)

22. Kemeny, J.G., Snell, J.L.: *Finite Markov Chains*. Van Nostrand, London (1960)
23. Lynch, N., Segala, R., Vaandrager, F.: Compositionality for probabilistic automata. In: Amadio, R., Lugiez, D. (eds.) *CONCUR 2003*. LNCS, vol. 2761, pp. 208–221. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45187-7_14
24. Segala, R.: *Modeling and Verification of Randomized Distributed Real-Time Systems*. Ph.D. thesis (1995)
25. Segala, R.: A compositional trace-based semantics for probabilistic automata. In: Lee, I., Smolka, S.A. (eds.) *CONCUR 1995*. LNCS, vol. 962, pp. 234–248. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60218-6_17
26. Segala, R.: Testing probabilistic automata. In: Montanari, U., Sassone, V. (eds.) *CONCUR 1996*. LNCS, vol. 1119, pp. 299–314. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61604-7_62
27. Segala, R., Lynch, N.: Probabilistic simulations for probabilistic processes. In: Jonsson, B., Parrow, J. (eds.) *CONCUR 1994*. LNCS, vol. 836, pp. 481–496. Springer, Heidelberg (1994). https://doi.org/10.1007/978-3-540-48654-1_35
28. van Glabbeek, R.J.: The linear time - branching time spectrum I. In: *Handbook of Process Algebra*, pp. 3–99. Elsevier (2001)
29. van Glabbeek, R.J., Smolka, S.A., Steffen, B.: Reactive, generative and stratified models of probabilistic processes. *Inf. Comput.* **121**, 59–80 (1995)
30. Vardi, M.Y.: Automatic verification of probabilistic concurrent finite-state programs. In: *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science (FOCS 1985)*, pp. 327–338. IEEE-CS Press (1985)
31. Yi, W., Larsen, K.G.: Testing probabilistic and nondeterministic processes. In: *Proceedings of the 12th International Symposium on Protocol Specification, Testing and Verification (PSTV 1992)*, pp. 47–61. North-Holland (1992)