Probabilistic Bisimilarity as Testing Equivalence

Yuxin Deng[a,*], Yuan Feng[b]

[a] *Shanghai Key Laboratory of Trustworthy Computing, MOE International Joint Lab of Trustworthy Software, and International Research Center of Trustworthy Software, East China Normal University*

[b] *University of Technology Sydney, Australia*

Abstract

Larsen and Skou initiated the study of probabilistic bisimilarity and its characterisation in terms of tests. Later on, van Breugel et al. showed that, for labelled Markov processes with continuous state spaces, probabilistic bisimilarity nicely coincides with a simple notion of testing equivalence. Their proof employs advanced machinery from topology. In the discrete case of finite-state reactive probabilistic processes, we prove that coincidence result with an elementary and more accessible proof.

Keywords: Probabilistic processes, bisimilarity, Testing equivalence, Modal logic

1. Introduction

Bisimulation [17, 18] is a central concept in concurrency theory. Bisimilarity, the largest bisimulation, admits beautiful characterisations in terms of fixed points, modal logics, coalgebras, games, algorithms, pseudometrics, etc. Its generalisation in the probabilistic setting is initiated by Larsen and Skou in [16] and has subsequently been widely investigated in probabilistic concurrency theory. Various characterisations of probabilistic bisimilarity by probabilistic extensions of Hennessy-Milner logic [11] have appeared in the literature [16, 8, 9, 19, 4, 12, 10, 7, 3]. Most characterisations employ some modalities indexed with numbers. A typical modal formula, dated back to [16], is $\langle a\rangle_p\phi$, where p is a probability value. A state s satisfies this formula if the probability that s can make an a-labelled transition to the set of states satisfying ϕ exceeds p. For reactive probabilistic processes [16, 22] with a minimal probability assumption, it is also shown in [16] that probabilistic bisimilarity can be characterised by a very simple testing framework. Two remarkable features of this framework are the *absence* of any modality indexed with numbers and the arithmetic interpretation

*Corresponding author

Email addresses: yxdeng@sei.ecnu.edu.cn (Yuxin Deng), Yuan.Feng@uts.edu.au (Yuan Feng)

of conjunction as multiplication. To prove the testing characterisation result, a notion of observation is introduced, every test is associated with a set of observations, and a testing equivalence is defined for states as the equality of probabilities on these observations. Furthermore, the coincidence proof of probabilistic bisimilarity with the testing equivalence crucially relies on the modal characterisation of probabilistic bisimilarity by a variant of Hennessy-Milner logic. In [21] van Breugel et al. avoid observations and directly define a testing equivalence for states as the equality of probabilities on tests. They generalise the testing characterisation of [16] to labelled Markov processes, i.e. reactive probabilistic processes [16, 22] with continuous state spaces, and surprisingly, without going through any modal logic. Usually, the simpler the logical or testing characterisation, the more difficult the completeness proof, since constructing distinguishing formulae or tests for non-bisimilar states with fewer modalities is more challenging. Van Breugel et al. prove such an elegant result by using advanced machinery such as the Lawson topology on probabilistic powerdomains [13] and Banach algebras. However, if we confine ourselves to discrete rather than continuous state spaces, as in e.g. [16], it is still unclear how to give a more direct and much simpler proof of the coincidence result given in [21], though intuitively it should be possible. The current work aims to provide a clear answer to this question. We consider finite-state reactive probabilistic processes and give an elementary proof of the coincidence of bisimilarity with the aforementioned testing equivalence while avoiding all the advanced machinery used in [21]. Our arguments only involve simple probability theory, ranks of matrices, and induction. It is worth mentioning that our proof is also constructive.

In the current work we focus on reactive probabilistic processes. Testing equivalences for other models have received a lot of attention. For example, Kwiatkowska and Norman [15] have generalised the testing framework of [16] to probabilistic systems with external and internal choice. Cleaveland et al. [1] have generalised the testing theory of [2] to generative probabilistic processes. Jonsson and Wang [14] have generalised [2] to nondeterministic probabilistic processes, which is further developed in [20, 6, 5, 3]. For the moment, we are not clear if our proof idea can be used in a setting with both probabilities and nondeterminism, which is left as an open problem.

2. Preliminaries

Let S be a finite set. A *(discrete) probability distribution* over set S is a function $\Delta : S \to [0,1]$ with $\sum_{s\in S} \Delta(s) = 1$. Its *support*, written $\lceil\Delta\rceil$, is the set $\{s \in S \mid \Delta(s) > 0\}$. Let $\mathcal{D}(S)$ denote the set of all distributions over S. We write $\overline{s}$ for the point distribution, satisfying $\overline{s}(t) = 1$ if $t = s$, and 0 otherwise. If $p_i \geq 0$ and Δ_i is a distribution for each i in some finite index set I, then $\sum_{i\in I} p_i \cdot \Delta_i$ is given by

$$(\sum_{i\in I} p_i \cdot \Delta_i)(s) \quad = \quad \sum_{i\in I} p_i \cdot \Delta_i(s)$$

If $\sum_{i\in I} p_i = 1$ then this is easily seen to be a distribution in $\mathcal{D}(S)$.

Definition 1. *A* reactive probabilistic labelled transition system *(rpLTS) is a triple* $(S, A, \rightarrow)$*, where* S *is a set of states,* A *is a set of actions, and the transition relation* $\rightarrow$ *is a partial function from* $S \times A$ *to* $\mathcal{D}(S)$*.*

We write $s \xrightarrow{a} \Delta$ for $\rightarrow (s, a) = \Delta$. In the current work, we focus on rpLTSs with finitely many states. Let us fix a rpLTS $(S, A, \rightarrow)$ for the rest of this note, with S and A being finite sets.

In the probabilistic setting, formal systems are usually modelled as distributions over states. To compare two systems involves the comparison of two distributions. So we need a way of lifting relations on states to relations on distributions. A few approaches have appeared in the literature. The following one is taken from [5],

Definition 2. *Given two sets* S, T *and a binary relation* $\mathcal{R} \subseteq S \times T$*, the lifted relation* $\mathcal{R}^\dagger \subseteq \mathcal{D}(S) \times \mathcal{D}(T)$ *is the smallest relation that satisfies:*

(i) $s\ \mathcal{R}\ t$ *implies* $\overline{s}\ \mathcal{R}^\dagger\ \overline{t}$*;*

(ii) $\Delta_i\ \mathcal{R}^\dagger\ \Theta_i$ *for all* $i \in I$ *implies* $(\sum_{i\in I} p_i \cdot \Delta_i)\ \mathcal{R}^\dagger\ (\sum_{i\in I} p_i \cdot \Theta_i)$*, where* I *is a finite index set and* $\sum_{i\in I} p_i = 1$*.*

There are alternative presentations of Definition 2; see [3, Chapter 3] for more detailed discussion. The following property is very useful.

Proposition 1. *Let* Δ *and* Θ *be distributions over* S *and* $\mathcal{R}$ *be an equivalence relation. Then* $\Delta\ \mathcal{R}^\dagger\ \Theta$ *if and only if* $\Delta(C) = \Theta(C)$ *for all equivalence classes* $C \in S/\mathcal{R}$*, where* $\Delta(C)$ *stands for the accumulation probability* $\sum_{s\in C} \Delta(s)$*.*

Bisimulation is a central notion in concurrency theory. Larsen and Skou [16] generalised it to the probabilistic setting and defined probabilistic bisimulation for rpLTSs.

Definition 3. *A binary relation* $\mathcal{R} \subseteq S \times S$ *is a* probabilistic simulation *if* $s\ \mathcal{R}\ t$ *and* $s \xrightarrow{a} \Delta$ *implies the existence of some transition* $t \xrightarrow{a} \Theta$ *with* $\Delta\ \mathcal{R}^\dagger\ \Theta$*.*

If both $\mathcal{R}$ *and* $\mathcal{R}^{-1}$ *are probabilistic simulations, then* $\mathcal{R}$ *is a* probabilistic bisimulation*. The largest probabilistic bisimulation is called* probabilistic bisimilarity*, denoted by* $\sim$*.*

Note that $\sim$ is an equivalence relation on S.

Various characterisations of probabilistic bisimilarity by probabilistic versions of Hennessy-Milner logic [11] have appeared in the literature. For example, Desharnais et al. [8] uses a logic with the grammar

$$\phi ::= \top \mid \phi \wedge \phi \mid \langle a \rangle_q \phi$$

where q is any rational number in the unit interval $[0, 1]$. Most other characterisations also employ modalities indexed with numbers. This fits with our intuition: if two states are not bisimilar, then they may satisfy a property with different probabilities, so by fiddling with the numbers we can form a formula or test that can tell apart the two states. The only exception is the one given in [21], which shows that, for reactive probabilistic processes, probabilistic bisimilarity can be characterised by a surprisingly simple testing framework. Let $\mathcal{T}$ be a testing language given by the grammar

$$t \ ::= \ \omega \mid a \cdot t \mid \langle t, t \rangle$$

where a ranges over the fixed set of labels A of our rpLTS.

Applying a test t to a state s in a reactive probabilistic process yields a probability $Pr(s, t)$ defined as follows:

$$\begin{array}{rcl} Pr(s, \omega) & = & 1 \\ Pr(s, a \cdot t) & = & \begin{cases} \sum_{s' \in S} \Delta(s') \cdot Pr(s', t) & \text{if } s \xrightarrow{a} \Delta \\ 0 & \text{otherwise} \end{cases} \\ Pr(s, \langle t_1, t_2 \rangle) & = & Pr(s, t_1) \cdot Pr(s, t_2) \end{array}$$

We call $\langle t_1, t_2 \rangle$ a *conjunction* of two tests, which models the copying capacity of probabilistic testing. Here, conjunction is given the arithmetic interpretation as multiplication, which differs from other logical characterisations of probabilistic bisimilarity. We often write t^2 for $\langle t, t \rangle$, and t^{m+1} for $\langle t, t^m \rangle$, where $m \geq 2$. That is, t^m is the conjunction of m copies of t. It is obvious that

$$Pr(s, t^m) = (Pr(s, t))^m \tag{1}$$

for any state s and test t.

Definition 4. *The testing language $\mathcal{T}$ induces a testing equivalence relation, written $=_T$, by letting $s_1 =_T s_2$ if $Pr(s_1, t) = Pr(s_2, t)$ for any $t \in \mathcal{T}$.*

In [21] van Breugel et al. consider labelled Markov processes with continuous state spaces and they show that probabilistic bisimilarity coincides with the above notion of testing equivalence. Their proof involves advanced machinery such as the Lawson topology on probabilistic powerdomains [13] and Banach algebras. In the current work we confine ourselves to finite-state rpLTSs. In this limited setting we avoid all the advanced machinery and give an elementary proof of the coincidence of $\sim$ with $=_T$. One direction of the coincidence result is easy.

Theorem 1. $\sim \subseteq =_T$.

Proof: We prove by induction on the structure of t that if $s_1 \sim s_2$ then $Pr(s_1, t) = Pr(s_2, t)$.

- $t \equiv \omega$. Then we have that $Pr(s_1, t) = 1 = Pr(s_2, t)$.

- $t \equiv a \cdot t'$. Since $s_1 \sim s_2$, then either (i) neither of the two states can perform action a, or (ii) both of them can perform action a. In the first case we have

$$Pr(s_1, a \cdot t') \;=\; 0 \;=\; Pr(s_2, a \cdot t') \;.$$

  In the second case, after performing action a, s_1 and s_2 can reach a unique distribution, say Δ_1 and Δ_2, respectively so that $\Delta_1 \sim^{\dagger} \Delta_2$. As an equivalence relation, $\sim$ partitions the state space S into a finite number of equivalence classes, e.g. $C_1, ..., C_n$ with $n \geq 1$. By Proposition 1, we have $\Delta_1(C_i) = \Delta_2(C_i)$ for all $i = 1..n$. Within each equivalence class C_i, any two states s_{i1}, s_{i2} are bisimilar. By induction we have $Pr(s_{i1}, t') = Pr(s_{i2}, t')$, so we can use $Pr(C_i, t')$ to stand for $Pr(s_{ij}, t')$, for any $s_{ij} \in C_i$. Then we can infer that

$$\begin{array}{rcl} Pr(s_1, a \cdot t') & = & \sum_{s' \in S} \Delta_1(s') \cdot Pr(s', t') \\ & = & \sum_{i=1}^{n} \Delta_1(C_i) \cdot Pr(C_i, t') \\ & = & \sum_{i=1}^{n} \Delta_2(C_i) \cdot Pr(C_i, t') \\ & = & \sum_{s' \in S} \Delta_2(s') \cdot Pr(s', t') \\ & = & Pr(s_2, a \cdot t') \;. \end{array}$$

- $t \equiv \langle t_1, t_2 \rangle$. By induction, we have $Pr(s_1, t_i) = Pr(s_2, t_i)$ for $i = 1, 2$. It follows that

$$Pr(s_1, \langle t_1, t_2 \rangle) \;=\; Pr(s_1, t_1) \cdot Pr(s_1, t_2) \;=\; Pr(s_2, t_1) \cdot Pr(s_2, t_2) \;=\; Pr(s_2, \langle t_1, t_2 \rangle) \;.$$

□

The converse of Theorem 1 also holds. However, the proof is far from being straightforward. The next section is devoted to it.

3. Testing equivalence as bisimulation

It is clear that $=_T$ is an equivalence relation. As in the proof of Theorem 1, we partition the state space S according to $=_T$. Let $C_1, \cdots, C_n$ be the equivalence classes induced by $=_T$, and write $Pr(C_i, t)$ for $Pr(s_{ij}, t)$, where s_{ij} is any state in C_i and t is any test. Note that for any two states in different equivalence classes, there exist some tests that can tell them apart. For each $1 \leq i < j \leq n$, let t_{ij} be a test that distinguishes C_i from C_j; that is, $Pr(C_i, t_{ij}) \neq Pr(C_j, t_{ij})$. Here t_{ij} is only a distinguishing test for C_i and C_j, and in general it says nothing about a third equivalence class C_k when $k \neq i, j$. For example, applying t_{ij} to C_i and then to C_k might yield the same outcome, that is, t_{ij} is not necessary a distinguishing test for C_i and C_k. However, we can construct an *enhanced test* that sharpens testing outcomes so that applying it to some equivalence classes gives either 0 or distinct positive values.

Lemma 1. *For any $I \subseteq \{1, \cdots, n\}$ with $I \neq \emptyset$, there exist a nonempty $I' \subseteq I$ and an enhanced test t such that*

(i) for all $k \in I$, $Pr(C_k, t) > 0$ iff $k \in I'$;
(ii) for any $i \neq j \in I'$, $Pr(C_i, t) \neq Pr(C_j, t)$.

Proof: We give a constructive proof by providing an algorithm to construct such a test.

Algorithm 1 initially sets I' to be I and t to be ω, then it gradually constructs more discriminating test t that, by removing the indices k with $Pr(C_k, t) = 0$ from I', tries to keep the indices i, j such that $Pr(C_i, t)$ and $Pr(C_j, t)$ are distinct and both positive. The outer loop uses a few auxiliary sets. For example, $\mathcal{I}_{pass}$ and $\mathcal{I}_{rem}$ form a partition of $\{(i, j) \in I' \times I' : i < j\}$. $\mathcal{I}_{pass}$ contains the pairs (i, j) such that the current test t can distinguish C_i from C_j. $\mathcal{I}_{rem}$ are the remaining pairs to be processed. Each iteration of the outer loop picks up any pair (i, j) in $\mathcal{I}_{rem}$, uses the distinguishing test t_{ij} to move some pairs from $\mathcal{I}_{rem}$ to $\mathcal{I}_{pass}$. The pairs being moved, e.g. (k, l) indicating that C_k and C_l can be differed by t_{ij}, are collected in $\mathcal{I}_{dis}$. However, just expanding $\mathcal{I}_{pass}$ with $\mathcal{I}_{dis}$ is insufficient. A newly added index k might conflict with another index l, which occurs either already in the old $\mathcal{I}_{pass}$ or in the set $\mathcal{I}_{dis}$, in the sense that C_k and C_l cannot be distinguished by t. To solve this problem, the inner loop tries to update t by padding it with enough copies of t_{ij} until it can distinguish all the equivalence classes indicated by the pairs in $\mathcal{I}_{pass}$. When all the pairs in $\mathcal{I}_{rem}$ are exhausted, the whole procedure terminates.

Let us give a more detailed analysis of the termination of this algorithm. We first look at the inner **while** loop. In each iteration, $\mathcal{I}$ is assigned a new value, which is a subset of $\mathcal{I}_{pass} \backslash \mathcal{I}_{term}$. If it becomes empty, the loop terminates immediately. Otherwise, the set $\mathcal{I}_{tem}$ is enlarged to include $\mathcal{I}$, so in the next iteration the set $\mathcal{I}_{pass} \backslash \mathcal{I}_{term}$ becomes smaller and so does $\mathcal{I}$. Eventually, $\mathcal{I}$ has to become $\emptyset$ and the loop terminates. For the outer **while** loop, in each iteration we choose a pair, say (i, j), from $\mathcal{I}_{rem}$, and then update $\mathcal{I}_{dis}$ and $\mathcal{I}_{rem}$. Since t_{ij} is a distinguishing test for (i, j), the two values $Pr(C_i, t_{ij})$ and $Pr(C_j, t_{ij})$ cannot be 0 at the same time. If both of them are positive, then $\mathcal{I}_{dis}$ contains at least the pair (i, j) and is not empty. If exactly one of them, say $Pr(C_i, t_{ij})$, is 0, then the corresponding index i is removed from I', which causes I' to shrink. In both cases, the assignment of $\mathcal{I}_{rem}$ by $(\mathcal{I}_{rem} \cap I' \times I') \backslash \mathcal{I}_{dis}$ makes $\mathcal{I}_{rem}$ strictly smaller. Eventually, $\mathcal{I}_{rem}$ becomes empty and the outer loop terminates. The number of iterations for the inner loop depends on the size of the set $\mathcal{I}_{pass}$, and for the outer loop depends on the size of $\mathcal{I}_{rem}$. It is easy to see that the maximal size for each of them is $\frac{n(n-1)}{2}$. So the overall time cost for the algorithm is $O(n^4)$.

To show the correctness, we only need to prove that at the beginning of each run of the outer **while** loop,

(a) $\mathcal{I}_{pass} \cup \mathcal{I}_{rem} = \{(i, j) \in I' \times I' : i < j\}$;

(b) $I' \neq \emptyset$;
(c) for all $k \in I$, $Pr(C_k, t) > 0$ iff $k \in I'$;
(d) for any $(i,j) \in \mathcal{I}_{pass}$, $Pr(C_i, t) \neq Pr(C_j, t)$.

Statement (a) is easy because each time $\mathcal{I}_{dis}$ is obtained, it is moved from $\mathcal{I}_{rem}$ to $\mathcal{I}_{pass}$. Statement (b) is also easy. Since t_{ij} is a distinguishing test for (i,j), at least one of the two values $Pr(C_i, t_{ij})$ and $Pr(C_j, t_{ij})$ is positive. Therefore, I' cannot be empty. Statement (c) can be proved by induction on the number of iterations of the outer **while** loop.

Now we prove Statement (d). The basis case when $\mathcal{I}_{pass} = \emptyset$ is trivial. For the induction step, suppose the outer **while** loop has been executed r times for some $r \geq 0$. Then in its $(r+1)$-th iteration before the inner loop, let (i,j) be chosen from $\mathcal{I}_{rem}$, I' and $\mathcal{I}_{dis}$ as defined in the algorithm, and $\mathcal{I}_{pass} = (\mathcal{I}'_{pass} \cap I' \times I') \cup \mathcal{I}_{dis}$ and $t = \langle t', t_{ij} \rangle$, where $\mathcal{I}'_{pass}$ and t' are the outcomes of the r-th iteration of the outer loop. By induction hypothesis, t' distinguishes any pair in $\mathcal{I}'_{pass}$. Suppose the inner loop terminates after M iterations in this outer loop. We show by induction on m the following claim.

Claim: For any m and m' with $0 \leq m \leq m' \leq M$, $t^{[m']}$ distinguishes any pair in $\mathcal{I}^{[m]}_{tem}$, where the superscripts, say $[m]$, indicate the outcomes obtained after m-iteration of the inner loop. Note that the notation $t^{[m]}$ differs from t^m that stands for the conjunction of m copies of t.

Obviously, $t^{[m]} = \langle t, t^m_{ij} \rangle$ when $m < M$ and $t^{[M]} = t^{[M-1]}$, where t^m_{ij} is the conjunction of m copies of t_{ij}, and

$$\mathcal{I}^{[m+1]}_{tem} = \mathcal{I}^{[m]}_{tem} \cup \{(k,l) \in \mathcal{I}_{pass} \backslash \mathcal{I}^{[m]}_{tem} : Pr(C_k, t^{[m]}) = Pr(C_l, t^{[m]})\}$$

for all $m < M$ with $\mathcal{I}^{[0]}_{tem} = \emptyset$. Note also that $\mathcal{I}^{[M]}_{tem} = \mathcal{I}^{[M-1]}_{tem}$. The basis case when $m = 0$ is trivial. Now suppose for any $m' \geq m \geq 0$, $t^{[m']}$ distinguishes any pair in $\mathcal{I}^{[m]}_{tem}$. Take any $(k,l) \in \mathcal{I}^{[m+1]}_{tem}$ and any $m' \geq m+1$. There are two cases to consider:

- $(k,l) \in \mathcal{I}^{[m]}_{tem}$. This case follows directly from the hypothesis.
- $(k,l) \in \mathcal{I}_{pass} \backslash \mathcal{I}^{[m]}_{tem}$ and $Pr(C_k, t^{[m]}) = Pr(C_l, t^{[m]})$. We now show that it must hold $Pr(C_k, t_{ij}) \neq Pr(C_l, t_{ij})$. When $(k,l) \in \mathcal{I}_{dis}$, this is from the definition of $\mathcal{I}_{dis}$; otherwise, we have $(k,l) \in \mathcal{I}'_{pass}$, and from the assumption that t' distinguishes any pair in $\mathcal{I}'_{pass}$, $Pr(C_k, t') \neq Pr(C_l, t')$. Note that

  $$Pr(C_k, t^{[m]}) = Pr(C_k, t') \cdot Pr(C_k, t_{ij})^{m+1},$$

  and similarly for $Pr(C_l, t^{[m]})$. From the assumption that $Pr(C_k, t_{ij}) > 0$ and $Pr(C_l, t_{ij}) > 0$, we derive $Pr(C_k, t_{ij}) \neq Pr(C_l, t_{ij})$, and then

  $$\begin{aligned} Pr(C_k, t^{[m']}) &= Pr(C_k, t^{[m]}) \cdot Pr(C_k, t_{ij})^{m'-m} \\ &\neq Pr(C_l, t^{[m]}) \cdot Pr(C_l, t_{ij})^{m'-m} \\ &= Pr(C_l, t^{[m']}). \end{aligned}$$

Here we have used the property that $Pr(C_k, t^{[m]}) > 0$ which is easy to observe by the construction of $t^{[m]}$.

With the claim we derive that $t^{[M]}$ distinguishes any pair in $\mathcal{I}_{tem}^{[M]}$. Furthermore, as $\mathcal{I} = \emptyset$, for any $(k,l) \in \mathcal{I}_{pass} \backslash \mathcal{I}_{tem}^{[m]}$, $Pr(C_k, t^{[M]}) \neq Pr(C_l, t^{[M]})$. In other words, $t^{[M]}$ actually distinguishes any pair in $\mathcal{I}_{pass}$. That completes the induction step of the outer loop. Moreover, when the outer loop is finished, we have $\mathcal{I}_{rem} = \emptyset$, which implies $\mathcal{I}_{pass} = \{(i,j) \in I' \times I' : i < j\}$ by (a). Then Clauses (i) and (ii) follow from (c) and (d), respectively, which concludes the proof of the lemma. □

Theorem 2. $=_T \subseteq \sim$.

Proof: We show that the relation $=_T$ is a bisimulation. Suppose $s_1 =_T s_2$ and $s_1 \xrightarrow{a} \Delta_1$. Observe that s_2 can also enable action a. Otherwise, applying the test $a \cdot \omega$ to s_1 would get outcome 1 while applying it to s_2 would get 0, which contradicts the assumption that $s_1 =_T s_2$. Therefore, there must exist some distribution Δ_2 such that $s_2 \xrightarrow{a} \Delta_2$. It remains to show that $\Delta_1 \, (=_T)^\dagger \, \Delta_2$. By Proposition 1, it suffices to check that $\Delta_1(C_i) = \Delta_2(C_i)$ for each equivalence class in $\{C_i \mid i \in I\} = S/_{=_T}$.

Let t be an arbitrary test in $\mathcal{T}$. Since $s_1 =_T s_2$, we have that

$$\begin{array}{rcl} 0 & = & Pr(s_1, a \cdot t) - Pr(s_2, a \cdot t) \\ & = & \sum_{s' \in S} \Delta_1(s') \cdot Pr(s', t) - \sum_{s' \in S} \Delta_2(s') \cdot Pr(s', t) \\ & = & \sum_{i \in I} \Delta_1(C_i) \cdot Pr(C_i, t) - \sum_{i \in I} \Delta_2(C_i) \cdot Pr(C_i, t) \\ & = & \sum_{i \in I} Pr(C_i, t) \cdot (\Delta_1(C_i) - \Delta_2(C_i)) \end{array} \tag{2}$$

Now we prove by induction on the size of the index set I, written $|I|$, that Eq. (2) implies $\Delta_1(C_i) = \Delta_2(C_i)$ for each $i \in I$.

- If $|I| = 1$ then the only equivalence class is S itself. So we clearly have $\Delta_1(S) = 1 = \Delta_2(S)$.

- Suppose the result holds for index sets of sizes less than $|I|$. Let

  $$x_i = \Delta_1(C_i) - \Delta_2(C_i).$$

  By the arbitrariness of t, each choice of t gives rise to an equation $\sum_{i \in I} Pr(C_i, t) \cdot x_i = 0$. By Lemma 1, there exist a nonempty set $I' \subseteq I$ and a test t_0 such that (i) $a_i \neq a_j$ for any $i \neq j \in I'$, and (ii) $a_k = 0$ for any $k \in I \backslash I'$, where we let $a_i = Pr(C_i, t_0)$ for any $i \in I$. Without loss of generality, we assume that $I' = \{1, 2, \cdots, n\}$, where $n \geq 1$. For the test t_0, we obtain the equation

  $$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = 0 \; . \tag{3}$$

  By (1), we see that $a_i^m = Pr(C_i, t_0^m)$ for any $m \geq 2$. So for any test t_0^m we obtain the equation

  $$a_1^m x_1 + a_2^m x_2 + \cdots + a_n^m x_n = 0 \; . \tag{4}$$

Putting these equations together, we obtain the linear equation system below

$$\begin{array}{rcl} a_1x_1 + a_2x_2 + \cdots + a_nx_n & = & 0 \\ a_1^2x_1 + a_2^2x_2 + \cdots + a_n^2x_n & = & 0 \\ & \vdots & \\ a_1^nx_1 + a_2^nx_2 + \cdots + a_n^nx_n & = & 0 \end{array}$$

Its coefficient matrix is the following

$$\begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ a_1^2 & a_2^2 & a_3^2 & \cdots & a_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^n & a_2^n & a_3^n & \cdots & a_n^n \end{bmatrix} \tag{5}$$

Note that $a_i > 0$ for each $i \in I'$. By dividing the ith column by a_i, we obtain the next matrix

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ a_1 & a_2 & a_3 & \cdots & a_n \\ a_1^2 & a_2^2 & a_3^2 & \cdots & a_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^{n-1} & a_2^{n-1} & a_3^{n-1} & \cdots & a_n^{n-1} \end{bmatrix} \tag{6}$$

This is the transpose of a Vandermonde matrix. Since all the a_i are distinct, the rank of the matrix is n. The original matrix in (5) must also have rank n. Then the only solution of the equation system is that $x_i = 0$ for all $i \in I'$. It follows that

$$\Delta_1(C_i) = \Delta_2(C_i) \qquad \text{for all } i \in I'. \tag{7}$$

Combining (2) and (7), we know that

$$\sum_{i \in I \backslash I'} Pr(C_i, t) \cdot (\Delta_1(C_i) - \Delta_2(C_i)) \;=\; 0 \;.$$

Note that $|I \backslash I'| < |I|$ and by induction hypothesis we obtain

$$\Delta_1(C_i) = \Delta_2(C_i) \qquad \text{for all } i \in I \backslash I'. \tag{8}$$

Hence, the desired result follows from (7) and (8).

□

Combining Theorems 1 and 2, we see that $\sim$ coincides with $=_T$, for finite-state reactive probabilistic processes. Our proof is very elementary and thus more accessible than the original proof presented in [21].

Acknowledgment We thank the anonymous referees for the helpful comments. Deng would like to acknowledge the support of the National Natural Science Foundation of China (61672229) and Shanghai Municipal Natural Science Foundation

(16ZR1409100). Feng was supported by Australian Research Council (Grant No. DP160101652), the CAS/SAFEA International Partnership Program for Creative Research Teams, and the Overseas Team Program of Academy of Mathematics and Systems Science, Chinese Academy of Sciences.

References

[1] R. Cleaveland, S. A. Smolka, and A. E. Zwarico. Testing preorders for probabilistic processes. In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, volume 623 of *Lecture Notes in Computer Science*, pages 708–719. Springer, 1992.

[2] R. De Nicola and M. Hennessy. Testing equivalences for processes. *Theoretic Computer Science*, 34:83–133, 1984.

[3] Y. Deng. *Semantics of Probabilistic Processes: An Operational Approach*. Springer, 2015.

[4] Y. Deng and R. van Glabbeek. Characterising probabilistic processes logically. In *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 6397 of *Lecture Notes in Computer Science*, pages 278–293. Springer, 2010.

[5] Y. Deng, R. van Glabbeek, M. Hennessy, and C. Morgan. Testing finitary probabilistic processes (extended abstract). In *Proceedings of the 20th International Conference on Concurrency Theory*, volume 5710 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2009.

[6] Y. Deng, R. van Glabbeek, M. Hennessy, C. Morgan, and C. Zhang. Characterising testing preorders for finite probabilistic processes. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science*, pages 313–325. IEEE Computer Society, 2007.

[7] Y. Deng and H. Wu. Modal characterisations of probabilistic and fuzzy bisimulations. In *Proceedings of the 16th International Conference on Formal Engineering Methods*, volume 8829 of *Lecture Notes in Computer Science*, pages 123–138. Springer, 2014.

[8] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. *Information and Computation*, 179(2):163–193, 2002.

[9] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating labelled Markov processes. *Information and Computation*, 184(1):160–200, 2003.

[10] M. Hennessy. Exploring probabilistic bisimulations, part I. *Formal Aspects of Computing*, 24(4-6):749–768, 2012.

[11] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[12] H. Hermanns, A. Parma, R. Segala, B. Wachter, and L. Zhang. Probabilistic logical characterization. *Information and Computation*, 209(2):154–172, 2011.

[13] C. Jones. *Probabilistic nondeterminism*. PhD thesis, University of Edinburgh, 1990.

[14] B. Jonsson and W. Yi. Compositional testing preorders for probabilistic processes. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science*, pages 431–441. IEEE Computer Society, 1995.

[15] M. Z. Kwiatkowska and G. Norman. A testing equivalence for reactive probabilistic processes. *Electronic Notes in Theoretical Computer Science*, 16(2):114–132, 1998.

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

[17] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[18] D. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981.

[19] A. Parma and R. Segala. Logical characterizations of bisimulations for discrete probabilistic systems. In *Proceedings of the 10th International Conference on Foundations of Software Science and Computational Structures*, volume 4423 of *Lecture Notes in Computer Science*, pages 287–301. Springer, 2007.

[20] R. Segala. Testing probabilistic automata. In *Proceedings of the 7th International Conference on Concurrency Theory*, volume 1119 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 1996.

[21] F. van Breugel, M. W. Mislove, J. Ouaknine, and J. Worrell. Domain theory, testing and simulation for labelled Markov processes. *Theoretical Computer Science*, 333(1-2):171–197, 2005.

[22] R. J. van Glabbeek, S. A. Smolka, B. Steffen, and C. M. N. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Proceedings of the 5th Annual Symposium on Logic in Computer Science*, pages 130–141. IEEE Computer Society, 1990.

Algorithm 1: Compute an enhanced test

input : A nonempty subset I of $\{1, \cdots, n\}$ with the distinguishing tests t_{ij} for all $i \neq j$.

output: A nonempty $I' \subseteq I$ and an enhanced test t satisfying (i) and (ii) in Lemma 1.

begin

$\mathcal{I}_{pass} \leftarrow \emptyset$;
$\mathcal{I}_{rem} \leftarrow \{(i,j) \in I \times I : i < j\}$;
$I' \leftarrow I$;
$t \leftarrow \omega$;
while $\mathcal{I}_{rem} \neq \emptyset$ **do**
Choose arbitrarily $(i,j) \in \mathcal{I}_{rem}$;
$I' \leftarrow \{k \in I' : Pr(C_k, t_{ij}) > 0\}$;
$\mathcal{I}_{dis} \leftarrow \{(k,l) \in \mathcal{I}_{rem} \cap I' \times I' : Pr(C_k, t_{ij}) \neq Pr(C_l, t_{ij})\}$;
$\mathcal{I}_{rem} \leftarrow (\mathcal{I}_{rem} \cap I' \times I') \backslash \mathcal{I}_{dis}$;
$\mathcal{I}_{pass} \leftarrow (\mathcal{I}_{pass} \cap I' \times I') \cup \mathcal{I}_{dis}$;
$t \leftarrow \langle t, t_{ij} \rangle$;
$\mathcal{I}_{tem} \leftarrow \emptyset$;
$\mathcal{I} \leftarrow \mathcal{I}_{pass}$;
while $\mathcal{I} \neq \emptyset$ **do**
$\mathcal{I} \leftarrow \{(k,l) \in \mathcal{I}_{pass} \backslash \mathcal{I}_{tem} : Pr(C_k, t) = Pr(C_l, t)\}$;
if $\mathcal{I} \neq \emptyset$ **then**
$t \leftarrow \langle t, t_{ij} \rangle$;
$\mathcal{I}_{tem} \leftarrow \mathcal{I}_{tem} \cup \mathcal{I}$;
end
end
end
return I', t;

end