

Deciding Probabilistic Bisimilarity Distance One for Probabilistic Automata^{*,**}

Qiyi Tang

*Department of Computer Science, University of Oxford
Oxford, United Kingdom*

Franck van Breugel

*Department of Electrical Engineering and Computer Science, York University
Toronto, Canada*

Abstract

Probabilistic bisimilarity, due to Segala and Lynch, is an equivalence relation that captures which states of a probabilistic automaton behave exactly the same. Deng et al. proposed a robust quantitative generalization of probabilistic bisimilarity. Their probabilistic bisimilarity distances of states of a probabilistic automaton capture the similarity of their behaviour. The smaller the distance, the more alike the states behave. In particular, states are probabilistic bisimilar if and only if their distance is zero.

Although the complexity of computing probabilistic bisimilarity distances for probabilistic automata has already been studied, we are not aware of any practical algorithms to compute those distances. In this paper, we provide several key results towards algorithms to compute probabilistic bisimilarity

^{*}This work was supported by the Natural Sciences and Engineering Research Council of Canada.

^{**}This paper is an extended version of [1].

Email addresses: `qiyi.tang@cs.ox.ac.uk` (Qiyi Tang), `franck@eecs.yorku.ca` (Franck van Breugel)

distances for probabilistic automata. In particular, we present a polynomial time algorithm that decides distance one. Furthermore, we give an alternative characterization of the probabilistic bisimilarity distances as a basis for a policy iteration algorithm.

Keywords: probabilistic automaton, probabilistic bisimilarity, probabilistic bisimilarity distance

1. Introduction

Behavioural equivalences, such as bisimilarity due to Milner [2] and Park [3], are one of the cornerstones of concurrency theory. Recall that a behavioural equivalence $\sim \subseteq S \times S$, where S is the set of states of the model, satisfies

$$s \sim s$$

$$\text{if } s \sim t \text{ then } t \sim s$$

$$\text{if } s \sim t \text{ and } t \sim u \text{ then } s \sim u$$

for all $s, t, u \in S$. The fact that states s and t behave the same is captured by $s \sim t$.

As first observed by Giacalone, Jou and Smolka [4], behavioural equivalences are *not robust* for models that contain quantitative information such as probabilities. This lack of robustness is caused by the discrepancy between the discrete nature of behavioural equivalence and the continuous nature of the quantitative information on which the behavioural equivalence relies. In particular, even small changes to the quantitative information may cause behaviourally equivalent states become inequivalent or vice versa.

Giacalone et al. proposed *behavioural pseudometrics* as a robust quantitative generalization of behavioural equivalences. A behavioural pseudometric

$d : S \times S \rightarrow [0, 1]$ satisfies

$$d(s, t) = 0 \text{ if and only if } s \sim t$$

$$d(s, t) = d(t, s)$$

$$d(s, u) \leq d(s, t) + d(t, u)$$

for all $s, t, u \in S$. The distance $d(s, t)$ measures the similarity of the behaviour of states s and t . The smaller this distance, the more alike the states behave. Distance zero captures that states are behaviourally equivalent.

In this paper, we focus on *probabilistic automata*. This model was first studied in the context of concurrency by Segala in [5]. It captures both nondeterminism (and, hence, concurrency) and probabilities. Consider the probabilistic automaton depicted in Figure 1. The states of a probabilistic automaton are labelled. These labels provide a partition of the states so that states satisfying the same basic properties of interest are in the same partition. In Figure 1, the labels are represented by colours. Each state has one or more probabilistic transitions. For example, the state t has a single probabilistic transition that takes state t to itself with probability one. State f has two probabilistic transitions. The one takes state f to state h with probability one. The other represents a fair coin toss, that is, it transitions to state h with probability $\frac{1}{2}$ and to state t with probability $\frac{1}{2}$. Also state b has two transitions, one of which represents a biased coin toss.

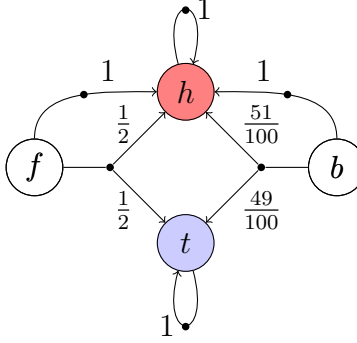


Figure 1: an example of a probabilistic automaton.

Segala and Lynch [6] introduced *probabilistic bisimilarity*. This behavioural equivalence for probabilistic automata generalizes the one introduced by Larsen and Skou [7]. The latter is applicable to models without nondeterminism, known as *labelled Markov chains*. These can be viewed as probabilistic automata where each state has exactly one probabilistic transition. States s and t of a probabilistic automaton are probabilistic bisimilar if the states have the same label and for each outgoing probabilistic transition of state s there exists a matching outgoing probabilistic transition of state t , and vice versa. Two probabilistic transitions match if they both transition to each probabilistic bisimilarity equivalence class with the exact same probability. States f and b in the above example are not probabilistic bisimilar. Although the transition from state f to state h can be matched by the transition from state b to state h , the probabilistic transitions representing a fair and biased coin toss do not match since the probabilities are slightly different.

Deng, Chothia, Palamidessi and Pang [8] introduced a behavioural pseudometric for probabilistic automata that generalizes probabilistic bisimilarity.

Their pseudometric also generalizes the one introduced for labelled Markov chains by Desharnais, Gupta, Jagadeesan and Panangaden in [9]. The *Hausdorff metric* [10] and the *Kantorovich metric* [11] are key ingredients of the definition of the pseudometric of Deng et al. The former is used to capture nondeterminism. This idea dates back to the work of De Bakker and Zucker [12]. The latter was first used by Van Breugel and Worrell [13] to capture probabilistic behaviour. On the one hand, the behaviours of the states h and t of the above example are very different since their labels are different. As a result, their probabilistic bisimilarity distance is one. On the other hand, the behaviours of the states f and b are very similar, which is reflected by the fact that these states have probabilistic bisimilarity distance $\frac{1}{100}$.

Tracol, Desharnais and Zhioua [14] also introduced a behavioural pseudometric for probabilistic automata. Their probabilistic bisimilarity distances generalize probabilistic bisimilarity as well, but are different from the ones introduced by Deng et al. An example showing the difference can be found in [14, Example 5]. To compute their probabilistic bisimilarity distances, they developed an iterative algorithm. In each iteration, a maximum flow problem needs to be solved. The resulting algorithm runs in polynomial time.

Before discussing other algorithms to approximate or compute probabilistic bisimilarity distances à la Deng et al. for probabilistic automata, let us first review some of the algorithms for the special case of labelled Markov chains. Van Breugel, Sharma and Worrell [15] presented an algorithm that approximates the probabilistic bisimilarity distances of a labelled Markov chain. Since the statement that the distance of states s and t is less than some rational q can be expressed in the *existential fragment of the first order*

theory over the reals, and this theory is decidable as shown by Tarski [16], one can use binary search to approximate the distance of s and t . The satisfiability problem for the existential fragment of the first order theory over the reals has been shown to be in **PSPACE** by Canny [17] (see also [18]). The algorithm of Van Breugel et al. is impractical. For example, consider a labelled Markov chain that captures randomized quicksort applied to the list [1,3,2,4,6,5]. This labelled Markov chain has 82 states and 203 transitions.¹ Their algorithm to approximate the probabilistic bisimilarity distances up to an accuracy of 0.0001 takes 66 hours for this labelled Markov chain.

Behavioural pseudometrics are usually defined as a *least fixed point*. This approach was first employed by Desharnais et al. [20] and is discussed in detail in Section 2 and 3. In [21], Chen, Van Breugel and Worrell provided an alternative characterization of the probabilistic bisimilarity distances for labelled Markov chains. This characterization provides a bridge to *reinforcement learning* (see, for example, [22] for an introduction to reinforcement learning) that we will discuss in more detail below. Bacci, Bacci, Larsen and Mardare [23] used this characterization as the basis for an algorithm to compute the probabilistic bisimilarity distances. This algorithm is much more efficient than the algorithm of Van Breugel et al. [15]. For example, for the labelled Markov chain representing an instance of randomized quicksort, the

¹The labelled Markov chain is obtained from an implementation of randomized quicksort in Java by means of the model checker Java PathFinder [19] and its extension jpf-probabilistic. The latter can be found at <https://bitbucket.org/discoveri/jpf-probabilistic>. An implementation of the algorithm of Van Breugel, Sharma, and Worrell can be found at <https://bitbucket.org/discoveri/first-order>.

algorithm of Van Breugel et al. took 66 hours. The algorithm of Bacci et al. took only 42 seconds for this example.

As we have shown in [24], in order to compute the probabilistic bisimilarity distances correctly, one has to decide *distance zero* before running the algorithm of Bacci et al. [23]. As shown by Desharnais et al. [9], states of a labelled Markov chain are probabilistic bisimilar if and only if they have distance zero. Since probabilistic bisimilarity can be decided in polynomial time, as shown by Baier [25], distance zero can be decided in polynomial time as well.

In [26], we have shown that for labelled Markov chains *distance one*, the largest distance, can be decided in polynomial time as well. We have also demonstrated experimentally that by first deciding distance zero, subsequently deciding distance one, and finally running the algorithm of Bacci et al. to compute the remaining distances, we can compute the probabilistic bisimilarity distances significantly faster and we can handle much larger labelled Markov chains. For example, for the randomized quicksort labelled Markov chain, the algorithm of Bacci et al. takes 42 seconds, whereas our algorithm takes 3 seconds.² Furthermore, whereas the algorithm of Bacci et al. can handle labelled Markov chains up to 150 states, our algorithm can handle a labelled Markov chain with more than 12,000 states in less than 50 minutes [27, Chapter 9]. As these examples show, our addition of deciding distance one moved the computation of probabilistic bisimilarity distances from a theoretical curiosity to a potential ingredient for practical verification

²Implementations of these algorithms can be found at <https://bitbucket.org/discoveri/probabilistic-bisimilarity-distances>.

tools.

As we already mentioned above, the alternative characterization of the probabilistic bisimilarity distances of labelled Markov chains due to Chen et al. [21] provides a bridge to reinforcement learning. Each labelled Markov chain can be mapped to a *Markov decision process* such that the states of the Markov decision process are pairs of states of the original labelled Markov chain. Furthermore, this mapping is such that the *value* of the state (s, t) of the constructed Markov decision process is the probabilistic bisimilarity distance of the states s and t of the labelled Markov chain. The details can be found in [27, Section 5.3]. The values of a Markov decision process can be computed by *policy iteration*, an algorithm due to Howard [28]. In [24] we have shown that the algorithm of Bacci et al. can be seen as policy iteration (see also [27, Chapter 6] for more details). The correspondence with reinforcement learning is summarized in Table 1.

behavioural pseudometrics	reinforcement learning
labelled Markov chain	Markov decision process
distances	values
policy iteration	

Table 1: correspondence between ingredients of behavioural pseudometrics and reinforcement learning.

The complexity of computing the probabilistic bisimilarity distances for probabilistic automata à la Deng et al. was first studied by Fu [29]. He showed that these probabilistic bisimilarity distances are rational. Furthermore, he proved that the problem of deciding whether the distance of two states is

smaller than a given rational is in $\mathbf{NP} \cap \mathbf{coNP}$. The proof has been adapted to show that the decision problem is in $\mathbf{UP} \cap \mathbf{coUP}$ [30]. Recall that \mathbf{UP} contains those problems in \mathbf{NP} with a unique accepting computation. Note that the complexity of computing the probabilistic bisimilarity distances for labelled Markov chains has shown to be in \mathbf{P} by Chen et al. [21].

Van Breugel and Worrell [31] have shown that the problem of computing the probabilistic bisimilarity distances of probabilistic automata is in \mathbf{PPAD} . This complexity class, which is short for *polynomial parity argument in a directed graph*, was introduced by Papadimitriou in [32]. It lies between the search problem versions of \mathbf{P} and \mathbf{NP} . The class captures the basic principles of path-following algorithms like those of Lemke and Howson [33] and Scarf [34]. Finding Nash equilibria of two player games is \mathbf{PPAD} -complete, as shown by Chen and Deng in [35]. Kintali et al. [36] present several other \mathbf{PPAD} -complete problems. Etessami and Yannakakis [37] have shown that computing the value of a simple stochastic game is in \mathbf{PPAD} . The relationship of \mathbf{PPAD} with the above mentioned complexity classes is given in Figure 2. It is not known what the relation between \mathbf{PPAD} and $\mathbf{UP} \cap \mathbf{coUP}$ is.

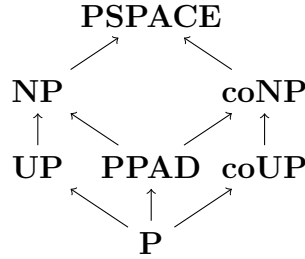


Figure 2: relationships of complexity classes.

The algorithm to approximate the probabilistic bisimilarity distances for probabilistic automata by Chen, Han and Lu [38] generalizes the algorithm of Van Breugel et al. [15] and uses the first order theory over the reals. As we already mentioned above, such algorithms are not practical. We have not implemented the algorithms underlying the above mentioned complexity results from [29, 31]. However, we anticipate that these are not practical either.

As shown by Deng et al. [8], states of a probabilistic automaton are probabilistic bisimilar if and only if they have distance zero. Since probabilistic bisimilarity can be decided in polynomial time, as shown by Baier [25], distance zero for probabilistic automata can be decided in polynomial time as well. As we have already discussed above, being able to decide distance one in polynomial time has a significant impact on computing probabilistic bisimilarity distances for labelled Markov chains. In Section 5 we present a polynomial time algorithm that decides distance one for probabilistic automata. This is the main contribution of this paper. We anticipate that this decision procedure will also impact the computation of probabilistic bisimilarity distances for probabilistic automata. We have implemented our algorithm in Java.³ In Section 7 we discuss some initial experimental results.

To prove our decision procedure correct, we use an alternative characterization of the probabilistic bisimilarity distances for probabilistic automata. This characterization generalizes the alternative characterization of the probabilistic bisimilarity distances for labelled Markov chains that we discussed

³The source code is available at <https://github.com/qiyitang71/distance-one-probabilistic-automata>.

above. In [31], Van Breugel and Worrell presented a mapping from probabilistic automata to *simple stochastic games*. Those games were introduced by Condon [39]. The vertices of the resulting simple stochastic game are pairs of states of the original probabilistic automaton. Their mapping is such that the value of a vertex (s, t) is the probabilistic bisimilarity distance of the states s and t . Our alternative characterization in Section 4 is based on a very similar mapping. As we will see, the size of the resulting game may be exponential in the size of the probabilistic automaton. Despite this potential exponential blow up, this correspondence between behavioural pseudometrics and game theory seems a promising avenue for further research.

2. Order and Distances

In this section, we provide some definitions and results from the literature about orders and distances that we will use in the remainder of this paper. For more details we refer the reader to, for example, [40] and [41].

2.1. Ordered sets

Given a set S , the set of *distance functions* on S , that is, functions from $S \times S$ to $[0, 1]$, are denoted by $[0, 1]^{S \times S}$. As in the work of Desharnais et al. [20], we endow the set $[0, 1]^{S \times S}$ with the following natural order.

Definition 1. The relation $\sqsubseteq \subseteq [0, 1]^{S \times S} \times [0, 1]^{S \times S}$ is defined by

$$d \sqsubseteq e \text{ if } d(s, t) \leq e(s, t) \text{ for all } s, t \in S.$$

Proposition 1. $\langle [0, 1]^{S \times S}, \sqsubseteq \rangle$ is a complete lattice.

PROOF. See, for example, [20, Lemma 3.2]. □

Let $\langle X, \leq \rangle$ be an ordered set. Let $f : X \rightarrow X$. Following [40, Definition 8.14], we define the following three notions:

- $x \in X$ is a *fixed point* of f if $f(x) = x$,
- $x \in X$ is a *pre-fixed point* of f if $f(x) \leq x$, and
- $x \in X$ is a *post-fixed point* of f if $x \leq f(x)$.

A function $f : X \rightarrow X$ is *monotone* if for all $x, y \in X$, $x \leq y$ implies $f(x) \leq f(y)$. The following result is known as the Knaster-Tarski fixed point theorem [42, 43].

Theorem 2. *Let X be a complete lattice and let $f : X \rightarrow X$ be a monotone function.*

- (a) *f has a greatest fixed point.*
- (b) *The greatest fixed point of f is the greatest post-fixed point of f .*
- (c) *f has a least fixed point.*
- (d) *The least fixed point of f is the least pre-fixed point of f .*

PROOF. See, for example, [40, Theorem 2.35] and [40, Theorem 8.20]. \square

We denote the greatest and least fixed point of a function f by νf and μf , respectively. Given a set X , we denote the set of subsets of X by 2^X .

Proposition 3. *$\langle 2^X, \subseteq \rangle$ is a complete lattice.*

PROOF. See, for example, [40, Example 2.6(2)]. \square

The correctness of our iterative algorithm to decide distance one relies on the following theorem.

Theorem 4. *Let X be a finite set and let $\Phi : 2^X \rightarrow 2^X$ be a monotone function.*

- (a) $\mu\Phi = \Phi^n(\emptyset)$ for some $n \in \mathbb{N}$.
- (b) $\nu\Phi = \Phi^n(X)$ for some $n \in \mathbb{N}$.
- (c) If $Y \subseteq \mu\Phi$ then $\mu\Phi = \Phi^n(Y)$ for some $n \in \mathbb{N}$.

PROOF. See, for example, [44, Lemma 8]. \square

2.2. Metric spaces

The set $[0, 1]^{S \times S}$ of distance functions on S also carries the following natural metric.

Definition 2. The function $\|\cdot - \cdot\| : [0, 1]^{S \times S} \times [0, 1]^{S \times S} \rightarrow [0, 1]$ is defined by

$$\|d - e\| = \sup_{s, t \in S} |d(s, t) - e(s, t)|.$$

Proposition 5. $\langle [0, 1]^{S \times S}, \|\cdot - \cdot\| \rangle$ is a nonempty complete metric space.

PROOF. See, for example, [41, Section 1.1.2]. \square

Let $\langle X, d \rangle$ be a metric space and $c \in (0, 1]$. A function $f : X \rightarrow X$ is *c-Lipschitz* if for all $x, y \in X$, $d(f(x), f(y)) \leq c d(x, y)$. A 1-Lipschitz function is also called *nonexpansive*. A function is *contractive* if it is *c-Lipschitz* for some $c \in (0, 1)$. The following result is known as Banach's fixed point theorem [45].

Theorem 6. *Let X be a nonempty complete metric space and $f : X \rightarrow X$ a contractive function. Then f has a unique fixed point.*

PROOF. See, for example, [41, Theorem 1.34]. \square

The least element of the complete lattice $[0, 1]^{S \times S}$ is the function $\vec{0} : S \times S \rightarrow [0, 1]$ which maps each pair of states to zero.

Theorem 7. *Let S be a finite set. If $\Psi : [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$ is monotone and nonexpansive then*

$$\mu\Psi = \sup_{n \in \mathbb{N}} \Psi^n(\vec{0}).$$

PROOF. See [46, Corollary 1]. \square

The following construction, due to Hausdorff [10], lifts a distance function on a set X to a distance function on the set 2^X as follows.

Definition 3. The function $H : [0, 1]^{X \times X} \rightarrow [0, 1]^{2^X \times 2^X}$ is defined by

$$H(d)(M, N) = \max \left\{ \max_{\mu \in M} \min_{\nu \in N} d(\mu, \nu), \max_{\nu \in N} \min_{\mu \in M} d(\mu, \nu) \right\}.$$

Given a nonempty finite set X , we denote the set of probability distributions on X by $\text{Distr}(X)$. For $\mu \in \text{Distr}(X)$, we define its support by $\text{support}(\mu) = \{x \in X \mid \mu(x) > 0\}$. A construction due to Kantorovich [11] lifts a distance function on X to a distance function on $\text{Distr}(X)$. To define this lifting, we need the notion of a coupling due to Doeblin [47].

Definition 4. Let $\mu, \nu \in \text{Distr}(X)$. The set $\Omega(\mu, \nu)$ of *couplings* of μ and ν is defined by

$$\Omega(\mu, \nu) = \left\{ \omega \in \text{Distr}(X \times X) \left| \sum_{x \in X} \omega(x, y) = \mu(y) \text{ and } \sum_{y \in X} \omega(x, y) = \nu(x) \right. \right\}.$$

The Kantorovich lifting [11] is defined as follows.

Definition 5. The function $K : [0, 1]^{X \times X} \rightarrow [0, 1]^{Distr(X) \times Distr(X)}$ is defined by

$$K(d)(\mu, \nu) = \inf_{\omega \in \Omega(\mu, \nu)} \sum_{u, v \in X} \omega(u, v) d(u, v).$$

In general, the set $\Omega(\mu, \nu)$ is infinite. The set of vertices of the convex polytope $\Omega(\mu, \nu)$ is denoted by $V(\Omega(\mu, \nu))$. The latter set is finite (see, for example, [48, page 259]). In the lifting, we can restrict to the vertices. This fact will be crucial in the proof of Theorem 12.

Proposition 8. For all $d \in [0, 1]^{X \times X}$ and $\mu, \nu \in Distr(X)$,

$$K(d)(\mu, \nu) = \min_{\omega \in V(\Omega(\mu, \nu))} \sum_{u, v \in X} \omega(u, v) d(u, v).$$

PROOF. See, for example, [27, Proposition 2.1.12].

The above described liftings due to Hausdorff and Kantorovich are key ingredients of the definition of the probabilistic bisimilarity distances, as we will see in the next section.

3. Probabilistic Automata

Also in this section, we recall some definitions and results from the literature. In particular, we introduce the model of interest, probabilistic automata, its best known behavioural equivalence, probabilistic bisimilarity, and its quantitative generalization. Probabilistic automata were first studied in the context of concurrency by Segala [5].

Definition 6. A *probabilistic automaton* is a tuple $\langle S, L, \rightarrow, \ell \rangle$ consisting of

- a nonempty finite set S of *states*,
- a nonempty finite set L of *labels*,
- a finitely branching *transition relation* $\rightarrow \subseteq S \times \text{Distr}(S)$, and
- a *labelling function* $\ell : S \rightarrow L$.

Instead of $(s, \mu) \in \rightarrow$, we write $s \rightarrow \mu$. A transition relation is finitely branching if for all $s \in S$, the set $\{\mu \in \text{Distr}(S) \mid s \rightarrow \mu\}$ is *nonempty* and finite. For the remainder of this paper we fix a probabilistic automaton $\langle S, L, \rightarrow, \ell \rangle$.

In order to define probabilistic bisimilarity, we first show how a relation on states can be lifted to a relation on probability distributions over states. This notion of lifting is due to Jonsson and Larsen [49, Definition 4.3].

Definition 7. The *lifting* of a relation $\mathcal{R} \subseteq S \times S$ is the relation $\mathcal{R}\uparrow \subseteq \text{Distr}(S) \times \text{Distr}(S)$ defined by $(\mu, \nu) \in \mathcal{R}\uparrow$ if there exists $\omega \in \Omega(\mu, \nu)$ such that $\text{support}(\omega) \subseteq \mathcal{R}$.

Probabilistic bisimilarity, a notion due to Segala and Lynch [6], is introduced next. States are probabilistic bisimilar if they have the same label and each probabilistic transition of the one state can be matched by a probabilistic transition of the other state, and vice versa. Two probabilistic transitions match if they transition with *exactly* the same probability to states that behave *exactly* the same.

Definition 8. An equivalence relation $\mathcal{R} \subseteq S \times S$ is a *probabilistic bisimulation* if for all $(s, t) \in \mathcal{R}$,

- $\ell(s) = \ell(t)$,
- for all $s \rightarrow \mu$ there exists $t \rightarrow \nu$ such that $(\mu, \nu) \in \mathcal{R}\uparrow$ and
- for all $t \rightarrow \nu$ there exists $s \rightarrow \mu$ such that $(\nu, \mu) \in \mathcal{R}\uparrow$.

Probabilistic bisimilarity, denoted \sim , is the largest probabilistic bisimulation.

For a proof that a largest probabilistic bisimulation exists, we refer the reader to, for example, [50, Proposition 4.3]. Relying on exact matching is the cause for a lack of robustness. To address this shortcoming, we define a quantitative generalization of probabilistic bisimilarity, the probabilistic bisimilarity distances, as the least fixed point of the function Δ_1 . To prove an alternative characterization of the probabilistic bisimilarity distances in the next section, we also introduce a family of discounted versions of Δ_1 , namely Δ_c with $c \in (0, 1)$.

Definition 9. Let $c \in (0, 1]$. The function $\Delta_c : [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$ is defined by

$$\Delta_c(d)(s, t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ c H(K(d))(\{\mu \mid s \rightarrow \mu\}, \{\nu \mid t \rightarrow \nu\}) & \text{otherwise.} \end{cases}$$

Probabilistic automata combine probabilistic and nondeterministic choices. The Hausdorff distance is usually employed to model nondeterminism (see, for example, [12]). For a discussion of the suitability of the Kantorovich distance to handle probabilistic choices, we refer the reader to [50].

Proposition 9. *For all $c \in (0, 1]$, the function Δ_c is monotone.*

PROOF. See [8, Lemma 2.10]. \square

Since $\langle [0, 1]^{S \times S}, \sqsubseteq \rangle$ is a complete lattice according to Proposition 1 and Δ_c is a monotone function by Proposition 9, we can conclude from Theorem 2(c) that Δ_c has a least fixed point $\mu\Delta_c$. The probabilistic bisimilarity distances $\mu\Delta_1$ form a behavioural pseudometric and the fact that they provide a quantitative generalization of probabilistic bisimilarity is captured by the following theorem due to Deng et al. [8].

Theorem 10. *For all $s, t \in S$, $\mu\Delta_1(s, t) = 0$ if and only if $s \sim t$.*

PROOF. See [8, Corollary 2.14]. \square

4. An Alternative Characterization

In the previous section, we defined the probabilistic bisimilarity distances as a least fixed point. Next, we present an alternative characterization. This generalizes the characterization of probabilistic bisimilarity distances for labelled Markov chains due to Chen et al. [21, Theorem 8]. First, we partition the set of state pairs as follows.

$$\begin{aligned} S_0^2 &= \{ (s, t) \in S \times S \mid s \sim t \} \\ S_1^2 &= \{ (s, t) \in S \times S \mid \ell(s) \neq \ell(t) \} \\ S_?^2 &= (S \times S) \setminus (S_0^2 \cup S_1^2) \end{aligned}$$

Note that, due to Theorem 10 the state pairs in S_0^2 have distance zero. From Definition 9 we can infer that the state pairs in S_1^2 have distance one. The

state pairs in S_7^2 cannot have distance zero, again due to Theorem 10, but can have any distance in the interval $(0, 1]$, including distance one.

As we will discuss below, the alternative characterization of the probabilistic bisimilarity distances can be viewed in terms of a *stochastic game*. These games were introduced by Shapley [51]. We focus here on a simplified version of these games, called *simple stochastic games*, which were first studied by Condon [39]. We use the more general definition of Zwick and Paterson [52]. The more general simple stochastic game can be converted to an ordinary simple stochastic game as defined in [39] in polynomial time [52, page 355].

A simple stochastic game is played with a single token by two players, called min and max, on a finite directed graph. The graph has five types of vertices: min, max and random vertices, 0-sinks and 1-sinks. The min, max, and random vertices have several outgoing edges, whereas the 0-sinks and 1-sinks have no outgoing edges. Whenever the token is in a min (max) vertex, the token is moved to one of the successors of the vertex, chosen by the min (max) player. If the token is in a random vertex, the successor is chosen randomly. The min (max) player's objective is to minimize (maximize) the probability of reaching a 1-sink.

Our alternative characterization of probabilistic bisimilarity distances can be viewed in terms of a simple stochastic game, similar to the one presented in [31]. The game can be considered a quantitative generalization of the game that characterizes bisimilarity (see [53]). In this turn based game, starting in a pair of states (s, t) , the max player chooses a probabilistic transition from either s or t . Subsequently, the min player chooses a probabilistic transition

from the other state and also chooses a coupling. In [31], the latter move by the min player is split into two moves by the min player. For example, if the max player picks $s \rightarrow \mu$ and the min player picks $t \rightarrow \nu$, then the min player also has to choose $\omega \in V(\Omega(\mu, \nu))$. This will be formalized in Definition 10. Recall that such a coupling ω is a probability distribution on $S \times S$. From a coupling ω the stochastic game moves to state pair (u, v) with probability $\omega(u, v)$.

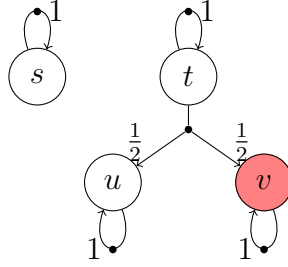


Figure 3: a probabilistic automaton

Consider, for example, the probabilistic automaton in Figure 3. Note that the states s and u are probabilistic bisimilar. The corresponding game graph is depicted in Figure 4. Since the game will be used to characterize the probabilistic bisimilarity distances, the state pairs for which we can easily determine their distance have no outgoing edges in the game graph. In particular, state pairs with different labels such as (s, v) , which have distance one, and state pairs that are probabilistic bisimilar such as (s, u) , which have distance zero, have no outgoing edges. These state pairs correspond to 1-sinks and 0-sinks, respectively.

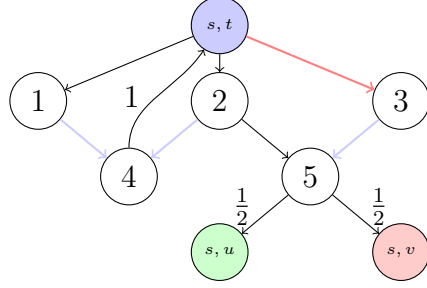


Figure 4: the game graph corresponding to the probabilistic automaton of Figure 3. The vertices labelled (s, t) , (s, u) and (s, v) are max vertices and belong to S_7^2 , S_0^2 and S_1^2 , respectively. The vertices labelled 1, 2 and 3 are min vertices and correspond to (s, Dir_t) , (t, Dir_s) and $(s, \frac{1}{2}\text{Dir}_u + \frac{1}{2}\text{Dir}_v)$, where Dir_x denotes the Dirac distribution concentrated on x . The vertices 4 and 5 are random vertices and correspond to $\text{Dir}_{(s, t)}$ and $\frac{1}{2}\text{Dir}_{(s, u)} + \frac{1}{2}\text{Dir}_{(s, v)}$, respectively.

The objective of the max player is to maximize the probability of reaching a state pair with different labels. The min player tries to minimize this probability. In the above example, the max player tries to reach the state pair (s, v) , whereas the min player tries to avoid that from happening. The *policies*, also known as *strategies*, for the max and min player are introduced next.

Definition 10. The set \mathcal{A} of *max policies* is defined by

$$\mathcal{A} = \left\{ A \in (S_7^2 \rightarrow (S \times \text{Distr}(S))) \left| \begin{array}{l} \forall (s, t) \in S_7^2 : \\ (\exists \nu \in \text{Distr}(S) : A(s, t) = (s, \nu) \wedge t \rightarrow \nu) \vee \\ (\exists \mu \in \text{Distr}(S) : A(s, t) = (t, \mu) \wedge s \rightarrow \mu) \end{array} \right. \right\}.$$

The set \mathcal{I} of *min policies* is defined by

$$\mathcal{I} = \left\{ I \in ((S \times \text{Distr}(S)) \rightarrow \text{Distr}(S \times S)) \left| \begin{array}{l} \forall (s, \nu) \in S \times \text{Distr}(S) : \exists \mu \in \text{Distr}(S) : \\ I(s, \nu) \in V(\Omega(\mu, \nu)) \wedge s \rightarrow \mu \end{array} \right. \right\}.$$

Given a policy A for the max player and a policy I for the min player, we define the *value function* as the least fixed point of the function $\Gamma_1^{A,I}$. This least fixed point captures the probability of reaching a state pair with different labels if both players use the given policies. We also introduce a family of discounted versions of $\Gamma_1^{A,I}$, namely $\Gamma_c^{A,I}$ with $c \in (0, 1)$, that we will use later in this section.

Definition 11. Let $A \in \mathcal{A}$, $I \in \mathcal{I}$ and $c \in (0, 1]$. The function $\Gamma_c^{A,I} : [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$ is defined by

$$\Gamma_c^{A,I}(d)(s, t) = \begin{cases} 0 & \text{if } (s, t) \in S_0^2 \\ 1 & \text{if } (s, t) \in S_1^2 \\ c \sum_{u, v \in S} I(A(s, t))(u, v) d(u, v) & \text{otherwise.} \end{cases}$$

We collect some properties of $\Gamma_c^{A,I}$ in the following proposition.

Proposition 11. For all $A \in \mathcal{A}$, $I \in \mathcal{I}$ and $c \in (0, 1]$,

- (a) the function $\Gamma_c^{A,I}$ is monotone and
- (b) the function $\Gamma_c^{A,I}$ is c -Lipschitz.

PROOF. Let $A \in \mathcal{A}$, $I \in \mathcal{I}$ and $c \in (0, 1]$.

- (a) Let $d, e \in [0, 1]^{S \times S}$ with $d \sqsubseteq e$. We distinguish three cases.

- If $(s, t) \in S_0^2$ then

$$\Gamma_c^{A,I}(d)(s, t) = 0 = \Gamma_c^{A,I}(e)(s, t).$$

– If $(s, t) \in S_1^2$ then

$$\Gamma_c^{A,I}(d)(s, t) = 1 = \Gamma_c^{A,I}(e)(s, t).$$

– Otherwise, $(s, t) \in S_7^2$. Then

$$\begin{aligned} \Gamma_c^{A,I}(d)(s, t) &= c \sum_{u, v \in S} I(A(s, t))(u, v) d(u, v) \\ &\leq c \sum_{u, v \in S} I(A(s, t))(u, v) e(u, v) \quad [d \sqsubseteq e] \\ &= \Gamma_c^{A,I}(e)(s, t). \end{aligned}$$

(b) Let $d, e \in [0, 1]^{S \times S}$. Let $s, t \in S$. We distinguish three cases.

– If $(s, t) \in S_0^2$ then

$$|\Gamma_c^{A,I}(d)(s, t) - \Gamma_c^{A,I}(e)(s, t)| = |0 - 0| = 0 \leq c \|d - e\|.$$

– If $(s, t) \in S_1^2$ then

$$|\Gamma_c^{A,I}(d)(s, t) - \Gamma_c^{A,I}(e)(s, t)| = |1 - 1| = 0 \leq c \|d - e\|.$$

– Otherwise, $(s, t) \in S_7^2$. Then

$$\begin{aligned}
& |\Gamma_c^{A,I}(d)(s, t) - \Gamma_c^{A,I}(e)(s, t)| \\
&= \left| c \sum_{u,v \in S} I(A(s, t))(u, v) d(u, v) - c \sum_{u,v \in S} I(A(s, t))(u, v) e(u, v) \right| \\
&= c \left| \sum_{u,v \in S} I(A(s, t))(u, v) d(u, v) - \sum_{u,v \in S} I(A(s, t))(u, v) e(u, v) \right| \\
&\leq c \sum_{u,v \in S} I(A(s, t))(u, v) |d(u, v) - e(u, v)| \\
&\leq c \sum_{u,v \in S} I(A(s, t))(u, v) \|d - e\| \\
&= c \|d - e\|.
\end{aligned}$$

□

From Theorem 2(c) we can conclude that $\Gamma_c^{A,I}$ has a least fixed point, which we denote by $\boldsymbol{\mu}\Gamma_c^{A,I}$. In the remainder of this section we will show that there exist an optimal max policy A^* and an optimal min policy I^* such that the corresponding value function captures the probabilistic bisimilarity distances. In the game graph of Figure 4, the red edge represents the optimal max policy and the blue edges represent the optimal min policy. The proof of $\boldsymbol{\mu}\Delta_1 = \boldsymbol{\mu}\Gamma_1^{A^*, I^*}$ consists of two parts. First, we prove that there exists an optimal min policy.

Theorem 12. $\exists I \in \mathcal{I} : \forall A \in \mathcal{A} : \boldsymbol{\mu}\Gamma_1^{A,I} \sqsubseteq \boldsymbol{\mu}\Delta_1$.

PROOF. Towards the construction of $I^* \in \mathcal{I}$, let $s \in S$ and $\nu \in \text{Distr}(S)$. Since we restrict our attention to finitely branching probabilistic automata,

$$\mu_{s,\nu} = \operatorname{argmin}_{s \rightarrow \mu} K(\boldsymbol{\mu}\Delta_1)(\mu, \nu) \quad (1)$$

exists. Because the set $V(\Omega(\mu_{s,\nu}, \nu))$ is nonempty and finite, we can define

$$I^*(s, \nu) = \underset{\omega \in V(\Omega(\mu_{s,\nu}, \nu))}{\operatorname{argmin}} \sum_{u,v \in S} \omega(u, v) \boldsymbol{\mu} \Delta_1(u, v). \quad (2)$$

By construction $I^* \in \mathcal{I}$.

Let $A \in \mathcal{A}$. Since $\boldsymbol{\mu} \Gamma_1^{A, I^*}$ is the least pre-fixed point of Γ_1^{A, I^*} according to Theorem 2(d), to conclude that $\boldsymbol{\mu} \Gamma_1^{A, I^*} \sqsubseteq \boldsymbol{\mu} \Delta_1$ it suffices to show that $\boldsymbol{\mu} \Delta_1$ is a pre-fixed point of Γ_1^{A, I^*} , that is, $\Gamma_1^{A, I^*}(\boldsymbol{\mu} \Delta_1) \sqsubseteq \boldsymbol{\mu} \Delta_1$. Let $s, t \in S$. We distinguish three cases.

- If $(s, t) \in S_0^2$, then

$$\begin{aligned} \Gamma_1^{A, I^*}(\boldsymbol{\mu} \Delta_1)(s, t) &= 0 \\ &= \boldsymbol{\mu} \Delta_1(s, t) \quad [\text{Theorem 10}] \end{aligned}$$

- If $(s, t) \in S_1^2$, then

$$\begin{aligned} \Gamma_1^{A, I^*}(\boldsymbol{\mu} \Delta_1)(s, t) &= 1 \\ &= \Delta_1(\boldsymbol{\mu} \Delta_1)(s, t) \\ &= \boldsymbol{\mu} \Delta_1(s, t). \end{aligned}$$

- Otherwise, $(s, t) \in S_7^2$. Without any loss of generality, we assume that

$A(s, t) = (s, \nu)$ with $t \rightarrow \nu$. Then

$$\begin{aligned}
\Gamma_1^{A, I^*}(\boldsymbol{\mu}\Delta_1)(s, t) &= \sum_{u, v \in S} I^*(A(s, t))(u, v) \boldsymbol{\mu}\Delta_1(u, v) \\
&= \sum_{u, v \in S} I^*(s, \nu)(u, v) \boldsymbol{\mu}\Delta_1(u, v) \quad [A(s, t) = (s, \nu)] \\
&= \min_{\omega \in V(\Omega(\mu_{s, \nu}, \nu))} \sum_{u, v \in S} \omega(u, v) \boldsymbol{\mu}\Delta_1(u, v) \quad [(2)] \\
&= K(\boldsymbol{\mu}\Delta_1)(\mu_{s, \nu}, \nu) \quad [\text{Proposition 8}] \\
&= \min_{s \rightarrow \mu} K(\boldsymbol{\mu}\Delta_1)(\mu, \nu) \quad [(1)] \\
&\leq \max_{t \rightarrow \nu} \min_{s \rightarrow \mu} K(\boldsymbol{\mu}\Delta_1)(\mu, \nu) \\
&\leq H(K(\boldsymbol{\mu}\Delta_1))(\{ \mu \mid s \rightarrow \mu \}, \{ \nu \mid t \rightarrow \nu \}) \\
&= \Delta_1(\boldsymbol{\mu}\Delta_1)(s, t) \\
&= \boldsymbol{\mu}\Delta_1(s, t).
\end{aligned}$$

□

In the remainder of this paper, we denote the optimal min policy constructed in the above proof by I^* . It remains to prove that there exists an optimal max policy. The proof of this second part turns out to be more involved than the proof of the first part contained in the above theorem. The proof has the following three major components.

- For all $A \in \mathcal{A}$ and $I \in \mathcal{I}$, the value function $\boldsymbol{\mu}\Gamma_1^{A, I}$ is the limit of the discounted value functions $\boldsymbol{\mu}\Gamma_c^{A, I}$. This result is inspired by [54, Theorem 4.4.1].
- Similarly, the probabilistic bisimilarity distances captured by $\boldsymbol{\mu}\Delta_1$ are the limit of their discounted counterparts represented by $\boldsymbol{\mu}\Delta_c$.

- There exists an optimal max policy in the discounted setting.

Combining the above three components, we arrive at an optimal max policy.

The next few propositions lead up to the proof that for all $A \in \mathcal{A}$ and $I \in \mathcal{I}$, the value function $\boldsymbol{\mu}(\Gamma_1^{A,I})$ is the limit of the discounted value functions $\boldsymbol{\mu}(\Gamma_c^{A,I})$, that is, $\lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I}) = \boldsymbol{\mu}(\Gamma_1^{A,I})$.

Proposition 13. *For all $A \in \mathcal{A}$, $I \in \mathcal{I}$ and $b, c \in (0, 1]$, if $b \leq c$ then $\boldsymbol{\mu}(\Gamma_b^{A,I}) \subseteq \boldsymbol{\mu}(\Gamma_c^{A,I})$.*

PROOF. Let $A \in \mathcal{A}$ and $I \in \mathcal{I}$. Let $b, c \in (0, 1]$ with $b \leq c$. To conclude that $\boldsymbol{\mu}(\Gamma_b^{A,I}) \subseteq \boldsymbol{\mu}(\Gamma_c^{A,I})$ it suffices to show $\Gamma_b^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I})) \subseteq \boldsymbol{\mu}(\Gamma_c^{A,I})$ according to Theorem 2(d). Let $s, t \in S$. We distinguish three cases.

- If $(s, t) \in S_0^2$ then

$$\Gamma_b^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) = 0 = \Gamma_c^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) = \boldsymbol{\mu}(\Gamma_c^{A,I})(s, t).$$

- If $(s, t) \in S_1^2$ then

$$\Gamma_b^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) = 1 = \Gamma_c^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) = \boldsymbol{\mu}(\Gamma_c^{A,I})(s, t).$$

- Otherwise, $(s, t) \in S_7^2$. Then

$$\begin{aligned} & \Gamma_b^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \\ &= b \Gamma_1^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \\ &\leq c \Gamma_1^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \quad [b \leq c] \\ &= \Gamma_c^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \\ &= \boldsymbol{\mu}(\Gamma_c^{A,I})(s, t). \end{aligned}$$

□

To prove $\lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I}) = \boldsymbol{\mu}(\Gamma_1^{A,I})$, we define a function $\gamma_1^{A,I}$.

Definition 12. Let $A \in \mathcal{A}$ and $I \in \mathcal{I}$. The function $\gamma_1^{A,I} : S \times S \rightarrow [0, 1]$ is defined by

$$\gamma_1^{A,I}(s, t) = \sup_{c \in (0,1)} \boldsymbol{\mu}(\Gamma_c^{A,I})(s, t).$$

Proposition 14. For all $A \in \mathcal{A}$ and $I \in \mathcal{I}$, $\lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I}) = \gamma_1^{A,I}$.

PROOF. Let $A \in \mathcal{A}$ and $I \in \mathcal{I}$. It suffices to prove that

$$\forall \epsilon > 0 : \exists b \in (0, 1) : \forall c \in [b, 1) : \|\boldsymbol{\mu}(\Gamma_c^{A,I}) - \gamma_1^{A,I}\| < \epsilon.$$

From Proposition 13 and the definition of $\gamma_1^{A,I}$ we can conclude that for all $s, t \in S$ there exists $b_{(s,t)} \in (0, 1)$ such that $|\boldsymbol{\mu}(\Gamma_c^{A,I})(s, t) - \gamma_1^{A,I}(s, t)| < \epsilon$ for all $b_{(s,t)} \leq c < 1$. Let $b = \max_{s,t \in S} b_{(s,t)}$. Note that b exists since S is a finite set. Then $\|\boldsymbol{\mu}(\Gamma_c^{A,I}) - \gamma_1^{A,I}\| < \epsilon$ for all $b \leq c < 1$. \square

Proposition 15. For all $A \in \mathcal{A}$ and $I \in \mathcal{I}$, $\gamma_1^{A,I} \sqsubseteq \boldsymbol{\mu}(\Gamma_1^{A,I})$.

PROOF. Let $A \in \mathcal{A}$ and $I \in \mathcal{I}$. From Proposition 13 we can conclude that for all $c \in (0, 1)$, $\boldsymbol{\mu}(\Gamma_c^{A,I}) \sqsubseteq \boldsymbol{\mu}(\Gamma_1^{A,I})$. From this we can immediately deduce that $\gamma_1^{A,I} \sqsubseteq \boldsymbol{\mu}(\Gamma_1^{A,I})$. \square

Proposition 16. For all $A \in \mathcal{A}$ and $I \in \mathcal{I}$, $\boldsymbol{\mu}(\Gamma_1^{A,I}) \sqsubseteq \gamma_1^{A,I}$.

PROOF. Let $A \in \mathcal{A}$ and $I \in \mathcal{I}$. To conclude that $\boldsymbol{\mu}(\Gamma_1^{A,I}) \sqsubseteq \gamma_1^{A,I}$, it suffices to show that $\Gamma_1^{A,I}(\gamma_1^{A,I}) = \gamma_1^{A,I}$ according to Theorem 2(d). Let $s, t \in S$. We distinguish three cases.

- If $(s, t) \in S_0^2$ then

$$\begin{aligned}
& \Gamma_1^{A,I}(\gamma_1^{A,I})(s, t) \\
&= 0 \\
&= \lim_{c \uparrow 1} \Gamma_c^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \\
&= \lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I})(s, t) \\
&= \gamma_1^{A,I}(s, t) \quad [\text{Proposition 14}]
\end{aligned}$$

- If $(s, t) \in S_1^2$ then

$$\begin{aligned}
& \Gamma_1^{A,I}(\gamma_1^{A,I})(s, t) \\
&= 1 \\
&= \lim_{c \uparrow 1} \Gamma_c^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \\
&= \lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I})(s, t) \\
&= \gamma_1^{A,I}(s, t) \quad [\text{Proposition 14}]
\end{aligned}$$

- Otherwise, $(s, t) \in S_7^2$. Then

$$\begin{aligned}
& \Gamma_1^{A,I}(\gamma_1^{A,I})(s, t) \\
&= \Gamma_1^{A,I}(\lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \quad [\text{Proposition 14}] \\
&= \lim_{c \uparrow 1} \Gamma_1^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \\
&\quad [\text{by Proposition 11(b), } \Gamma_1^{A,I} \text{ is 1-Lipschitz and, hence, continuous}] \\
&= \lim_{c \uparrow 1} \frac{1}{c} \Gamma_c^{A,I}(\boldsymbol{\mu}(\Gamma_c^{A,I}))(s, t) \\
&= \lim_{c \uparrow 1} \frac{1}{c} \boldsymbol{\mu}(\Gamma_c^{A,I})(s, t) \\
&= \lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I})(s, t) \\
&= \gamma_1^{A,I}(s, t) \quad [\text{Proposition 14}]
\end{aligned}$$

□

Combining the above propositions, we can conclude $\lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I}) = \boldsymbol{\mu}(\Gamma_1^{A,I})$, which completes the first component of the proof.

Lemma 17. *For all $A \in \mathcal{A}$ and $I \in \mathcal{I}$, $\lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I}) = \boldsymbol{\mu}(\Gamma_1^{A,I})$.*

PROOF. Let $A \in \mathcal{A}$ and $I \in \mathcal{I}$.

$$\begin{aligned}
\lim_{c \uparrow 1} \boldsymbol{\mu}(\Gamma_c^{A,I}) &= \gamma_1^{A,I} \quad [\text{Proposition 14}] \\
&= \boldsymbol{\mu}(\Gamma_1^{A,I}) \quad [\text{Proposition 15 and 16}]
\end{aligned}$$

□

Similarly, we can prove the second component. For the detailed proofs, we refer the reader to [27, Section 10.3].

Lemma 18. $\lim_{c \uparrow 1} \boldsymbol{\mu} \Delta_c = \boldsymbol{\mu} \Delta_1$.

The third and final part of the proof consists of showing that there exists an optimal max policy in the discounted setting.

Lemma 19. *For all $c \in (0, 1)$, $\exists A \in \mathcal{A} : \forall I \in \mathcal{I} : \boldsymbol{\mu} \Delta_c \sqsubseteq \boldsymbol{\mu} \Gamma_c^{A, I}$.*

PROOF. Let $c \in (0, 1)$. Let $s, t \in S$. If

$$\max_{s \rightarrow \mu} \min_{t \rightarrow \nu} K(\boldsymbol{\mu} \Delta_c)(\mu, \nu) \geq \max_{t \rightarrow \nu} \min_{s \rightarrow \mu} K(\boldsymbol{\mu} \Delta_c)(\mu, \nu) \quad (3)$$

then we define $A_c^*(s, t)$ by

$$A_c^*(s, t) = \left(t, \operatorname{argmax}_{s \rightarrow \mu} \min_{t \rightarrow \nu} K(\boldsymbol{\mu} \Delta_c)(\mu, \nu) \right).$$

Because the probabilistic automaton is finitely branching, the above exists.

Otherwise, we define $A_c^*(s, t)$ by

$$A_c^*(s, t) = \left(s, \operatorname{argmax}_{t \rightarrow \nu} \min_{s \rightarrow \mu} K(\boldsymbol{\mu} \Delta_c)(\mu, \nu) \right).$$

By construction, $A_c^* \in \mathcal{A}$.

Let $I \in \mathcal{I}$. Since $\langle [0, 1]^{S \times S}, \|\cdot - \cdot\| \rangle$ is a nonempty complete metric space according to Proposition 5 and the function $\Gamma_c^{A_c^*, I}$ is contractive by Proposition 11(b), we can conclude from Theorem 6 that $\Gamma_c^{A_c^*, I}$ has a unique fixed point. Therefore, $\boldsymbol{\mu} \Gamma_c^{A_c^*, I}$ is not only the least fixed point but also the greatest fixed point of $\Gamma_c^{A_c^*, I}$. According to Theorem 2(b), $\boldsymbol{\mu} \Gamma_c^{A_c^*, I}$ is the greatest post-fixed point of $\Gamma_c^{A_c^*, I}$. Hence, to conclude that $\boldsymbol{\mu} \Delta_c \sqsubseteq \boldsymbol{\mu} \Gamma_c^{A_c^*, I}$ it suffices to show that $\boldsymbol{\mu} \Delta_c$ is a post-fixed point of $\Gamma_c^{A_c^*, I}$, that is, $\boldsymbol{\mu} \Delta_c \sqsubseteq \Gamma_c^{A_c^*, I}(\boldsymbol{\mu} \Delta_c)$. Let $s, t \in S$. We distinguish three cases.

- If $(s, t) \in S_0^2$, then

$$\begin{aligned}
\boldsymbol{\mu}\Delta_c(s, t) &\leq \boldsymbol{\mu}\Delta_1(s, t) && [\Delta\text{-analogue of Proposition 13}] \\
&= 0 && [\text{Theorem 10}] \\
&= \Gamma_c^{A_c^*, I}(\boldsymbol{\mu}\Delta_c)(s, t).
\end{aligned}$$

- If $(s, t) \in S_1^2$, then

$$\begin{aligned}
\boldsymbol{\mu}\Delta_c(s, t) &= \Delta_c(\boldsymbol{\mu}\Delta_c)(s, t) \\
&= 1 \\
&= \Gamma_c^{A_c^*, I}(\boldsymbol{\mu}\Delta_c)(s, t).
\end{aligned}$$

- Otherwise, $(s, t) \in S_7^2$. Without loss of any generality, assume that $A_c^*(s, t) = (t, \mu)$. This assumption implies that (3) and

$$\Delta_1(\boldsymbol{\mu}\Delta_c)(s, t) = \min_{t \rightarrow \nu} K(\boldsymbol{\mu}\Delta_c)(\mu, \nu). \quad (4)$$

Hence,

$$\begin{aligned}
\boldsymbol{\mu}\Delta_c(s, t) &= \Delta_c(\boldsymbol{\mu}\Delta_c)(s, t) \\
&= c \Delta_1(\boldsymbol{\mu}\Delta_c)(s, t) \\
&= c \min_{t \rightarrow \nu} K(\boldsymbol{\mu}\Delta_c)(\mu, \nu) && [(4)] \\
&= c \min_{t \rightarrow \nu} \min_{\omega \in V(\Omega(\mu, \nu))} \sum_{u, v \in S} \omega(u, v) \boldsymbol{\mu}\Delta_c(u, v) && [\text{Proposition 8}] \\
&\leq c \sum_{u, v \in S} I(A_c^*(s, t))(u, v) \boldsymbol{\mu}\Delta_c(u, v) \\
&\quad [I(A_c^*(s, t)) \in V(\Omega(\mu, \nu)) \text{ for some } t \rightarrow \nu] \\
&= \Gamma_c^{A_c^*, I}(\boldsymbol{\mu}\Delta_c)(s, t).
\end{aligned}$$

□

Combining the above three components, we obtain the second part of the proof of the alternative characterization.

Theorem 20. $\exists A \in \mathcal{A} : \forall I \in \mathcal{I} : \mu\Delta_1 \sqsubseteq \mu\Gamma_1^{A,I}.$

PROOF. According to Lemma 19,

$$\forall n \in \mathbb{N} : \exists A_n \in \mathcal{A} : \forall I \in \mathcal{I} : \mu\Delta_{\frac{n}{n+1}} \sqsubseteq \mu\Gamma_{\frac{n}{n+1}}^{A_n,I}. \quad (5)$$

Since the set \mathcal{A} is finite, the sequence $(A_n)_{n \in \mathbb{N}}$ has a subsequence $(A_{\sigma(n)})_{n \in \mathbb{N}}$ that is constant, that is, there exists $A^* \in \mathcal{A}$ such that for all $n \in \mathbb{N}$, $A_{\sigma(n)} = A^*$. From Lemma 17 and 18 we can conclude that

$$\lim_{n \in \mathbb{N}} \mu\Delta_{\frac{\sigma(n)}{\sigma(n)+1}} = \mu\Delta_1 \text{ and } \lim_{n \in \mathbb{N}} \mu\Gamma_{\frac{\sigma(n)}{\sigma(n)+1}}^{A^*,I} = \mu\Gamma_1^{A^*,I}.$$

From (5) we can deduce that $\forall I \in \mathcal{I} : \mu\Delta_1 \sqsubseteq \mu\Gamma_1^{A^*,I}.$ \square

In the remainder of this paper, we use A^* to denote an optimal max policy, which exists according to Theorem 20. Combining the above results, we arrive at the following alternative characterization of the probabilistic bisimilarity distances.

Corollary 21. $\mu\Delta_1 = \mu\Gamma_1^{A^*,I^*}.$

PROOF.

$$\begin{aligned} \mu\Delta_1 &\sqsubseteq \mu\Gamma_1^{A^*,I^*} && [\text{Theorem 20}] \\ &\sqsubseteq \mu\Delta_1 && [\text{Theorem 12}] \end{aligned}$$

\square

5. Deciding Distance One

In this section, we present an algorithm to compute the set D_1 of state pairs that have distance one, that is

$$D_1 = \{ (s, t) \in S \times S \mid \mu\Delta_1(s, t) = 1 \}.$$

The key ingredient of our algorithm is the following function.

Definition 13. The function $\Lambda : 2^{S \times S} \times 2^{S \times S} \rightarrow 2^{S \times S}$ is defined by

$$\Lambda(X, Y) = S_1^2 \cup \left\{ (s, t) \in S_?^2 \left| \begin{array}{l} \exists s \rightarrow \mu : \forall t \rightarrow \nu : \forall \omega \in V(\Omega(\mu, \nu)) : \\ \text{support}(\omega) \subseteq X \wedge \text{support}(\omega) \cap Y \neq \emptyset \vee \\ \exists t \rightarrow \nu : \forall s \rightarrow \mu : \forall \omega \in V(\Omega(\mu, \nu)) : \\ \text{support}(\omega) \subseteq X \wedge \text{support}(\omega) \cap Y \neq \emptyset \end{array} \right. \right\}.$$

The set $\Lambda(X, Y)$ contains all state pairs with different labels and those state pairs for which there exists a move by the max player so that every subsequent move of the min player always ends up in X and with some positive probability in Y . We use the notation $\lambda Z.\Lambda(X, Z)$ to denote the function that maps the set Z to the set $\Lambda(X, Z)$. We denote the least and greatest fixed point of this function by $\mu Z.\Lambda(X, Z)$ and $\nu Z.\Lambda(X, Z)$, respectively. The function Λ has the following monotonicity properties.

Proposition 22. *For all $X, Y, Z \subseteq S \times S$ with $X \subseteq Y$,*

- (a) $\Lambda(X, Z) \subseteq \Lambda(Y, Z)$.
- (b) $\Lambda(Z, X) \subseteq \Lambda(Z, Y)$.
- (c) $\mu Z.\Lambda(X, Z) \subseteq \mu Z.\Lambda(Y, Z)$.

PROOF. Let $X, Y, Z \subseteq S \times S$ with $X \subseteq Y$.

- (a) Follows immediately from the definition of Λ and the fact that $\text{support}(\omega) \subseteq X$ and $X \subseteq Y$ imply $\text{support}(\omega) \subseteq Y$.
- (b) Follows immediately from the definition of Λ and the fact that $\text{support}(\omega) \cap X \neq \emptyset$ and $X \subseteq Y$ imply $\text{support}(\omega) \cap Y \neq \emptyset$.
- (c) We have that

$$\begin{aligned} \Lambda(X, \mu Z. \Lambda(Y, Z)) &\subseteq \Lambda(Y, \mu Z. \Lambda(Y, Z)) && [\text{part (a)}] \\ &= \mu Z. \Lambda(Y, Z) \\ &[\mu Z. \Lambda(Y, Z) \text{ is a fixed point of } \lambda Z. \Lambda(Y, Z)] \end{aligned}$$

that is, $\mu Z. \Lambda(Y, Z)$ is a pre-fixed point of $\lambda Z. \Lambda(X, Z)$. Since $\mu Z. \Lambda(X, Z)$ is the least pre-fixed point of $\lambda Z. \Lambda(X, Z)$ according to Theorem 2(d), we can conclude that $\mu Z. \Lambda(X, Z) \subseteq \mu Z. \Lambda(Y, Z)$. \square

Since $\langle 2^{S \times S}, \subseteq \rangle$ is a complete lattice and for each $X \subseteq S \times S$ the function $\lambda Y. \Lambda(X, Y)$ is monotone (Proposition 22(b)), its least fixed point $\mu Y. \Lambda(X, Y)$ exists according to Theorem 2(c). The set $\mu Y. \Lambda(X, Y)$ contains all state pairs (s, t) for which there exists a max policy such that for all min policies, (s, t) can reach a state pair with different labels and all state pairs reachable from (s, t) are an element of X .

Since the function $\lambda X. \mu Y. \Lambda(X, Y)$, that maps the set X to the set $\mu Y. \Lambda(X, Y)$, is monotone as well (Proposition 22(c)), we can conclude from Theorem 2(a) that its greatest fixed point denoted $\nu X. \mu Y. \Lambda(X, Y)$ exists. The set $\nu X. \mu Y. \Lambda(X, Y)$ contains all state pairs (s, t) for which there exists

a max policy such that for all min policies, all state pairs reachable from (s, t) can reach a state pair with different labels. In the next section, we will prove that $\nu X. \mu Y. \Lambda(X, Y)$ captures the set D_1 . According to Theorem 4(a) and (b), these greatest and least fixed points can be obtained iteratively as follows.

```

1   $X_c = S \times S$ 
2  do
3     $Y_c = \emptyset$ 
4  do
5     $Y_p = Y_c$ 
6     $Y_c = \Lambda(X_c, Y_p)$ 
7  while  $Y_p \neq Y_c$ 
8     $X_p = X_c$ 
9     $X_c = Y_c$ 
10 while  $X_p \neq X_c$ 

```

The inner loop (line 3–7) computes the least fixed point $\mu Y. \Lambda(X_c, Y)$. The outer loop (line 1–10) computes the greatest fixed point $\nu X. \mu Y. \Lambda(X, Y)$, which equals D_1 as we will prove in the next section. Due to the monotonicity of Λ we can conclude that both the inner and outer loop terminate after at most $|S|^2$ iterations. To conclude that the above algorithm is polynomial time, it remains to show that $\Lambda(X_c, Y_p)$ in line 6 can be computed in polynomial time.

Proposition 23. *For all $\mu, \nu \in \text{Distr}(S)$ and $X \subseteq S \times S$,*

(a)

$$\forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq X$$

$$\text{iff } K(d)(\mu, \nu) = 1$$

$$\text{iff } \forall \omega \in \Omega(\mu, \nu) : \text{support}(\omega) \subseteq X$$

$$\text{iff } \text{support}(\mu) \times \text{support}(\nu) \subseteq X$$

(b)

$$\forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \cap X \neq \emptyset$$

$$\text{iff } K(d)(\mu, \nu) > 0$$

$$\text{iff } \forall \omega \in \Omega(\mu, \nu) : \text{support}(\omega) \cap X \neq \emptyset$$

where

$$d(s, t) = \begin{cases} 1 & \text{if } (s, t) \in X \\ 0 & \text{otherwise.} \end{cases}$$

PROOF. Let $\mu, \nu \in \text{Distr}(S)$ and $X \subseteq S \times S$. For all $\omega \in \Omega(\mu, \nu)$,

$$\begin{aligned} K(d)(\mu, \nu) &\leq \sum_{u, v \in S} \omega(u, v) d(u, v) \\ &= \sum_{(u, v) \in X} \omega(u, v) d(u, v) + \sum_{(u, v) \in S \times S \setminus X} \omega(u, v) d(u, v) \\ &= \sum_{(u, v) \in X} \omega(u, v) \quad [\text{definition of } d] \end{aligned} \tag{6}$$

Furthermore, according to Proposition 8, there exists $\pi \in V(\Omega(\mu, \nu))$ such that

$$K(d)(\mu, \nu) = \sum_{u, v \in S} \pi(u, v) d(u, v) = \sum_{(u, v) \in X} \pi(u, v). \tag{7}$$

(a) The proof consists of several parts.

$$\begin{aligned}
& \forall \omega \in \Omega(\mu, \nu) : \text{support}(\omega) \subseteq X \\
& \text{iff } \forall \omega \in \Omega(\mu, \nu) : \sum_{(u,v) \in X} \omega(u, v) = 1 \\
& \text{iff } K(d)(\mu, \nu) = 1 \quad [(6) \text{ and } (7)]
\end{aligned}$$

According to Proposition 8, we can restrict to the vertices of $\Omega(\mu, \nu)$ in the definition of K , that is,

$$K(d)(\mu, \nu) = \min_{\omega \in V(\Omega(\mu, \nu))} \sum_{u,v \in S} \omega(u, v) d(u, v). \quad (8)$$

Hence, the proof that $K(d)(\mu, \nu) = 1$ if and only if $\forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq X$ is similar to the above proof.

Next, we show

$$\forall \omega \in \Omega(\mu, \nu) : \text{support}(\omega) \subseteq X \quad (9)$$

if and only if

$$\text{support}(\mu) \times \text{support}(\nu) \subseteq X \quad (10)$$

by proving two implications. Assume (9). Let $(u, v) \in \text{support}(\mu) \times \text{support}(\nu)$, that is $\mu(u) > 0$ and $\nu(v) > 0$. To conclude (10), by (9) it suffices to show that $(u, v) \in \text{support}(\omega)$ for some $\omega \in \Omega(\mu, \nu)$. Let $\pi \in \Omega(\mu, \nu)$. Since

$$\begin{aligned}
\sum_{y \in S} \pi(u, y) &= \mu(u) > 0 \\
\sum_{x \in S} \pi(x, v) &= \nu(v) > 0
\end{aligned}$$

there exist $x, y \in S$ such that $\pi(u, y) > 0$ and $\pi(x, v) > 0$. If $x = u$ or $y = v$, then $(u, v) \in \text{support}(\pi)$. Otherwise, we define

$$\omega(s, t) = \begin{cases} \pi(s, t) - m & \text{if } (s, t) = (u, y) \text{ or } (s, t) = (x, v) \\ \pi(s, t) + m & \text{if } (s, t) = (x, y) \text{ or } (s, t) = (u, v) \\ \pi(s, t) & \text{otherwise,} \end{cases}$$

where $m = \min\{\pi(u, y), \pi(x, v)\}$. By definition, $\omega(u, v) > 0$. We leave it to the reader to check that $\omega \in \Omega(\mu, \nu)$.

	v		y	
u	$+m$		$-m$	
x	$-m$		$+m$	

Figure 5: difference between π and ω .

Assume (10). Let $\omega \in \Omega(\mu, \nu)$ and $(u, v) \in \text{support}(\omega)$, that is, $\omega(u, v) > 0$. To conclude (9), it suffices to show that $(u, v) \in X$. Since

$$\mu(u) = \sum_{w \in S} \omega(u, w) \geq \omega(u, v) > 0,$$

we can conclude that $u \in \text{support}(\mu)$. Similarly, we can show that $v \in \text{support}(\nu)$. Hence, $(u, v) \in \text{support}(\mu) \times \text{support}(\nu)$. By (10), we can conclude that $(u, v) \in X$.

(b) Also this proof consists of several parts. For all $\omega \in \Omega(\mu, \nu)$,

$$\begin{aligned}
& \text{support}(\omega) \cap X \neq \emptyset \\
& \text{iff } \sum_{(u,v) \in X} \omega(u, v) > 0 \\
& \text{iff } \sum_{u,v \in S} \omega(u, v) d(u, v) > 0 \quad [(6)]
\end{aligned}$$

This is equivalent to $K(d)(\mu, \nu) > 0$. The proof that $K(d)(\mu, \nu) > 0$ if and only if $\forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \cap X \neq \emptyset$ is similar, again relying on (8). \square

Due to Proposition 23(a), the condition $\forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq X$ can be simplified considerably, namely to $\text{support}(\mu) \times \text{support}(\nu) \subseteq X$, which can obviously be checked in polynomial time. As discussed in, for example, [50, Section 5], computing $K(d)(\mu, \nu)$ boils down to solving a transportation problem, which is a special case of the minimum cost network flow problem, where d captures the cost. This problem can be solved in polynomial time using, for example, Orlin's network simplex algorithm [55]. Hence, $\Lambda(X_c, Y_p)$ can be computed in polynomial time. No further simplification for the condition $\forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \cap X \neq \emptyset$ is known to us. This seems to be related to the fact that no non-iterative procedure for finding a feasible solution of a capacitated transportation problem is known (see, for example, [56, page 378]).

6. Correctness Proof

In this section, we will prove the algorithm presented in the previous section correct. That is, we will show that $D_1 = \nu X. \mu Y. \Lambda(X, Y)$. Intuitively,

a pair of states in D_1 has the property that either the states have a different label or there exists a max policy that always ends up, no matter how the min player plays, in state pairs for which the same particular property holds. Let us first provide a high level overview of the proof. We will define a sequence of sets of state pairs X_0, X_1, \dots, X_m in terms of Λ in Definition 14. We will show that $X_m = \nu X. \mu Y. \Lambda(X, Y)$ in Lemma 29 and $D_1 \subseteq X_m$ in Proposition 27. To conclude that $D_1 = \nu X. \mu Y. \Lambda(X, Y)$, it remains to show that $X_m \setminus D_1 = \emptyset$. In Definition 15, we will partition $X_m \setminus D_1$ into the sets of state pairs Z_0, Z_1, \dots, Z_{n-1} as depicted in Figure 6. In Definition 16, we will construct a max policy A' that only differs from the optimal max policy A^* on $X_m \setminus D_1$. This max policy A' will be such that if the min player plays according to the optimal min policy I^* and the max player uses A' starting from $(s, t) \in Z_i$, then the game will stay within X_m according to Proposition 32 and the game will reach some $(u, v) \in D_1 \cup \bigcup_{0 \leq j < i} Z_j$ with positive probability. In the proof of Theorem 34, towards a contradiction, we will assume that $X_m \setminus D_1$ is nonempty. Let Z_i contain a state pair with minimal value, for the policies I^* and A' , among the state pairs in $X_m \setminus D_1$. We will prove that this minimal value is smaller than one. As we will show, Z_j , for some $0 \leq j < i$, contains a state pair with minimal value as well. As a result, Z_0 contains a state pair with minimal value, which will lead to a contradiction.

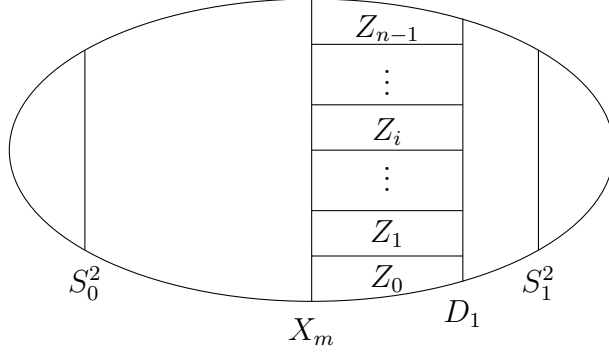


Figure 6: partition of $X_m \setminus D_1$.

6.1. Fixed Point Properties of Λ

Towards proving that D_1 is the greatest fixed point of the function $\lambda X. \mu Y. \Lambda(X, Y)$, we first show that D_1 is a fixed point of the functions $\lambda X. \Lambda(X, X)$ and $\lambda X. \mu Y. \Lambda(X, Y)$.

Proposition 24. $D_1 = \Lambda(D_1, D_1)$.

PROOF. For all $s, t \in S$,

$$(s, t) \in D_1$$

$$\text{iff } \boldsymbol{\mu}(\Delta_1)(s, t) = 1$$

$$\text{iff } \Delta_1(\boldsymbol{\mu}(\Delta_1))(s, t) = 1$$

$$\text{iff } (s, t) \in S_1^2 \vee$$

$$((s, t) \in S_?^2 \wedge \exists s \rightarrow \mu : \forall t \rightarrow \nu : \forall \omega \in V(\Omega(\mu, \nu)) : \sum_{u, v \in S} \omega(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) = 1) \vee$$

$$((s, t) \in S_?^2 \wedge \exists t \rightarrow \nu : \forall s \rightarrow \mu : \forall \omega \in V(\Omega(\mu, \nu)) : \sum_{u, v \in S} \omega(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) = 1)$$

[Proposition 8]

$$\text{iff } (s, t) \in S_1^2 \vee$$

$$((s, t) \in S_?^2 \wedge \exists s \rightarrow \mu : \forall t \rightarrow \nu : \forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq D_1) \vee$$

$$((s, t) \in S_?^2 \wedge \exists t \rightarrow \nu : \forall s \rightarrow \mu : \forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq D_1)$$

$$\text{iff } (s, t) \in S_1^2 \vee$$

$$((s, t) \in S_?^2 \wedge \exists s \rightarrow \mu : \forall t \rightarrow \nu : \forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq D_1 \wedge \text{support}(\omega) \cap D_1 \neq \emptyset) \vee$$

$$((s, t) \in S_?^2 \wedge \exists t \rightarrow \nu : \forall s \rightarrow \mu : \forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq D_1 \wedge \text{support}(\omega) \cap D_1 \neq \emptyset)$$

$$\text{iff } (s, t) \in \Lambda(D_1, D_1).$$

□

To prove the second fixed point property of Λ , we first show that if the game starts in D_1 and the max player uses the optimal max policy A^* , then the game stays in D_1 .

Proposition 25. *For all $(s, t) \in D_1 \setminus S_1^2$ and $I \in \mathcal{I}$, $\text{support}(I(A^*(s, t))) \subseteq D_1$.*

PROOF. Let $(s, t) \in D_1 \setminus S_1^2$. Towards a contradiction, assume there exist $I \in \mathcal{I}$ and $(x, y) \in \text{support}(I(A^*(s, t)))$ such that $\boldsymbol{\mu}(\Delta_1)(x, y) < 1$. Without any loss of generality, we may assume that $A^*(s, t) = (s, \nu)$. Let $I(s, \nu) = \pi$. Then $s \rightarrow \mu$, $t \rightarrow \nu$, and $\pi \in V(\Omega(\mu, \nu))$. Hence,

$$\begin{aligned}
\boldsymbol{\mu}(\Delta_1)(s, t) &= \boldsymbol{\mu}(\Gamma_1^{A^*, I^*})(s, t) \quad [\text{Corollary 21}] \\
&= \Gamma_1^{A^*, I^*}(\boldsymbol{\mu}(\Gamma_1^{A^*, I^*}))(s, t) \\
&= \Gamma_1^{A^*, I^*}(\boldsymbol{\mu}(\Delta_1))(s, t) \quad [\text{Corollary 21}] \\
&= \sum_{u, v \in S} I^*(A^*(s, t))(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) \\
&= \sum_{u, v \in S} I^*(s, \nu)(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) \quad [A^*(s, t) = (s, \nu)] \\
&= \min_{\omega \in V(\Omega(\mu, \nu))} \sum_{u, v \in S} \omega(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) \quad [(2)] \\
&= \min_{s \rightarrow \mu} \min_{\omega \in V(\Omega(\mu, \nu))} \sum_{u, v \in S} \omega(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) \quad [(1)] \\
&\leq \min_{\omega \in V(\Omega(\mu, \nu))} \sum_{u, v \in S} \omega(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) \quad [s \rightarrow \mu] \\
&\leq \sum_{u, v \in S} \pi(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) \quad [\pi \in V(\Omega(\mu, \nu))] \\
&= \sum_{u, v \in S} I(s, \nu)(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) \quad [I(s, \nu) = \pi] \\
&= \sum_{u, v \in S} I(A^*(s, t))(u, v) \boldsymbol{\mu}(\Delta_1)(u, v) \quad [A^*(s, t) = (s, \nu)] \\
&< 1 \quad [(x, y) \in \text{support}(I(A^*(s, t))) \text{ and } \boldsymbol{\mu}(\Delta_1)(x, y) < 1]
\end{aligned}$$

This contradicts $(s, t) \in D_1$. □

Secondly, we show that D_1 is a fixed point of $\boldsymbol{\lambda}X.\boldsymbol{\mu}Y.\Lambda(X, Y)$.

Proposition 26. $D_1 = \mu Y. \Lambda(D_1, Y)$.

PROOF. Let $\mathbf{Y} = \mu Y. \Lambda(D_1, Y)$. From Proposition 24, we can conclude $\mathbf{Y} \subseteq D_1$. It remains to prove that $D_1 \subseteq \mathbf{Y}$. Towards a contradiction, assume $D_1 \setminus \mathbf{Y} \neq \emptyset$. First, we show that there exists $I \in \mathcal{I}$ such that

$$\forall (s, t) \in D_1 \setminus \mathbf{Y} : \text{support}(I(A^*(s, t))) \subseteq D_1 \setminus \mathbf{Y}.$$

Next, we prove

$$\forall (s, t) \in D_1 \setminus \mathbf{Y} : \mu(\Gamma_1^{A^*, I})(s, t) = 0. \quad (11)$$

Finally, we show that there exists $(s, t) \in D_1 \setminus \mathbf{Y}$ such that $\mu(\Delta_1)(s, t) = 0$, which is the desired contradiction.

Let $(s, t) \in D_1 \setminus \mathbf{Y}$. Because $(s, t) \in D_1$ we can deduce that $(s, t) \notin S_0^2$. Since $(s, t) \notin \mathbf{Y} = \Lambda(D_1, \mathbf{Y})$, we have that $(s, t) \notin S_1^2$. Hence, $(s, t) \in S_7^2$.

Without loss of generality, assume that $A^*(s, t) = (t, \mu)$ with $s \rightarrow \mu$. Because $(s, t) \in S_7^2$ and $(s, t) \notin \Lambda(D_1, \mathbf{Y})$, we can conclude that

$$\exists t \rightarrow \nu : \exists \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \not\subseteq D_1 \vee \text{support}(\omega) \cap \mathbf{Y} = \emptyset. \quad (12)$$

In the remainder of this proof we denote the coupling ω satisfying the above by $I(t, \mu)$. Note that (12) depends on t and μ , but not on s . Hence,

$$\text{support}(I(A^*(s, t))) \not\subseteq D_1 \vee \text{support}(I(A^*(s, t))) \cap \mathbf{Y} = \emptyset.$$

Since $(s, t) \in D_1$ and $(s, t) \notin S_1^2$, we can conclude from Proposition 25 that

$$\text{support}(I(A^*(s, t))) \subseteq D_1.$$

Combining the above, we obtain

$$\text{support}(I(A^*(s, t))) \subseteq D_1 \setminus \mathbf{Y}. \quad (13)$$

Next, we prove for all $(s, t) \in D_1 \setminus \mathbf{Y}$, $\boldsymbol{\mu}(\Gamma_1^{A^*, I})(s, t) = 0$. To prove this, it suffices to show that for all $n \in \mathbb{N}$, $(\Gamma_1^{A^*, I})^n(\vec{0})(s, t) = 0$ according to Proposition 11 and a minor variation on Theorem 7. We prove this by induction on n . The base case $n = 0$ is immediate. Let $n > 0$. Then

$$\begin{aligned}
& (\Gamma_1^{A^*, I})^n(\vec{0})(s, t) \\
&= \Gamma_1^{A^*, I}((\Gamma_1^{A^*, I})^{n-1}(\vec{0}))(s, t) \\
&= \sum_{u, v \in S} I(A^*(s, t))(u, v) (\Gamma_1^{A^*, I})^{n-1}(\vec{0})(u, v) \\
&= \sum_{(u, v) \in D_1 \setminus \mathbf{Y}} I(A^*(s, t))(u, v) (\Gamma_1^{A^*, I})^{n-1}(\vec{0})(u, v) \quad [(13)] \\
&= 0 \quad [\text{by induction, } (\Gamma_1^{A^*, I})^{n-1}(\vec{0})(u, v) = 0 \text{ for all } (u, v) \in D_1 \setminus \mathbf{Y}]
\end{aligned}$$

By assumption, $D_1 \setminus \mathbf{Y} \neq \emptyset$. Let $(s, t) \in D_1 \setminus \mathbf{Y}$. Then

$$\begin{aligned}
\boldsymbol{\mu}(\Delta_1)(s, t) &\leq \boldsymbol{\mu}(\Gamma_1^{A^*, I})(s, t) \quad [\text{Theorem 20}] \\
&= 0 \quad [(11)]
\end{aligned}$$

Hence, $(s, t) \notin D_1$, which contradicts $(s, t) \in D_1 \setminus \mathbf{Y}$. □

6.2. Iterative Characterization

To conclude that the algorithm presented in the previous section is correct, it remains to show that $\boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X, Y)$ equals D_1 . We start by providing an iterative characterization of $\boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X, Y)$.

Definition 14. For each $i \in \mathbb{N}$, the set $X_i \subseteq S \times S$ is defined by

$$X_i = \begin{cases} S \times S & \text{if } i = 0 \\ \boldsymbol{\mu}Y.\Lambda(X_{i-1}, Y) & \text{otherwise.} \end{cases}$$

For each $i, j \in \mathbb{N}$, the set $Y_i^j \subseteq S \times S$ is defined by

$$Y_i^j = \begin{cases} D_1 & \text{if } j = 0 \\ \Lambda(X_i, Y_i^{j-1}) & \text{otherwise.} \end{cases}$$

The above definition differs from the iterative algorithm presented in the previous section in that $Y_i^0 = D_1$, whereas the algorithm starts its iteration towards the least fixed point from \emptyset .

Next, we prove a key property of the sets X_i .

Lemma 27. *For all $i \in \mathbb{N}$, $D_1 \subseteq X_i$.*

PROOF. We prove this proposition by induction on i . We distinguish the following two cases.

- If $i = 0$ then

$$D_1 \subseteq S \times S = X_0.$$

- If $i > 0$ then

$$\begin{aligned} D_1 &= \mu Y. \Lambda(D_1, Y) && [\text{Proposition 26}] \\ &\subseteq \mu Y. \Lambda(X_{i-1}, Y) && [\text{by induction } D_1 \subseteq X_{i-1} \text{ and Proposition 22(c)}] \\ &= X_i. \end{aligned}$$

□

The proposition below collects two properties of Y_i^j , which will be used later.

Proposition 28.

(a) For all $i, j \in \mathbb{N}$, $D_1 \subseteq Y_i^j$.

(b) For all $i, j \in \mathbb{N}$, $Y_i^j \subseteq Y_i^{j+1}$.

PROOF.

(a) Let $i \in \mathbb{N}$. We prove this proposition by induction on j . The base case, $j = 0$, is vacuously true. Let $j > 0$. Then

$$\begin{aligned}
 D_1 &= \Lambda(D_1, D_1) && [\text{Proposition 24}] \\
 &\subseteq \Lambda(X_i, D_1) && [\text{Lemma 27 and Proposition 22(a)}] \\
 &\subseteq \Lambda(X_i, Y_i^{j-1}) && [\text{by induction, } D_1 \subseteq Y_i^{j-1} \text{ and Proposition 22(b)}] \\
 &= Y_i^j.
 \end{aligned}$$

(b) Let $i \in \mathbb{N}$. We prove this proposition by induction on j . We distinguish the following two cases.

– If $j = 0$ then

$$\begin{aligned}
 Y_i^0 &= D_1 \\
 &= \Lambda(D_1, D_1) && [\text{Proposition 24}] \\
 &\subseteq \Lambda(X_i, D_1) && [\text{Lemma 27 and Proposition 22(a)}] \\
 &= \Lambda(X_i, Y_i^0) \\
 &= Y_i^1.
 \end{aligned}$$

– If $j > 0$ then

$$\begin{aligned}
 Y_i^j &= \Lambda(X_i, Y_i^{j-1}) \\
 &\subseteq \Lambda(X_i, Y_i^j) && [\text{by induction, } Y_i^{j-1} \subseteq Y_i^j \text{ and Proposition 22(b)}] \\
 &= Y_i^{j+1}.
 \end{aligned}$$

□

Next, we prove some other key properties of the sets X_i and Y_i^j .

Lemma 29.

(a) $X_m = \nu X. \mu Y. \Lambda(X, Y)$ for some $m \in \mathbb{N}$.

(b) $Y_m^n = \mu Y. \Lambda(X_m, Y)$ for some $n \in \mathbb{N}$.

(c) $X_m = Y_m^n$.

PROOF.

(a) Follows from Theorem 4(b) and Proposition 22(c).

(b) First, we observe that

$$\begin{aligned} D_1 &\subseteq X_m && [\text{Lemma 27}] \\ &= \mu Y. \Lambda(X_m, Y) && [X_m \text{ is a fixed point of } \lambda X. \mu Y. \Lambda(X, Y) \text{ by part (a)}] \end{aligned}$$

The desired result follows from the above and Theorem 4(c) and Proposition 22(b).

(c) We have that

$$\begin{aligned} Y_m^n &= \mu Y. \Lambda(X_m, Y) && [\text{part (b)}] \\ &= X_m && [X_m \text{ is a fixed point of } \lambda X. \mu Y. \Lambda(X, Y) \text{ by part (a)}] \end{aligned}$$

□

6.3. Max Policy A'

In this section, we will construct a max policy A' . The construction of A' relies on partitioning $X_m \setminus D_1$ into n disjoint subsets Z_0, Z_1, \dots, Z_{n-1} , as depicted in Figure 6. Note that m and n are the constants from Lemma 29.

Definition 15. For each $0 \leq i < n$, the set $Z_i \subseteq S \times S$ is defined by

$$Z_i = Y_m^{i+1} \setminus Y_m^i.$$

We collect some properties of the sets Z_i in the proposition below.

Proposition 30.

- (a) For all $0 \leq i < n$, $Z_i \subseteq S_7^2$.
- (b) For all $0 \leq i < j < n$, $Z_i \cap Z_j = \emptyset$.
- (c) $\bigcup_{0 \leq i < n} Z_i = X_m \setminus D_1$.
- (d) For all $0 \leq i \leq n$, $Y_m^i = D_1 \cup \bigcup_{0 \leq j < i} Z_j$.

PROOF.

- (a) Let $0 \leq i < n$. By Proposition 28(a), $S_1^2 \subseteq D_1 \subseteq Y_m^i$. Also, $Y_m^{i+1} = \Lambda(X_m, Y_m^i) \subseteq S_1^2 \cup S_7^2$. Hence, $Z_i = Y_m^{i+1} \setminus Y_m^i \subseteq S_7^2$.
- (b) Let $0 \leq i < j < n$. Because $i < j$, $Y_m^{i+1} \subseteq Y_m^j$ due to Proposition 28(b). Since also $Z_i = Y_m^{i+1} \setminus Y_m^i \subseteq Y_m^{i+1}$ and $Z_j = Y_m^{j+1} \setminus Y_m^j$, we can conclude that $Z_i \cap Z_j = \emptyset$.

(c) We first show that for all $1 \leq j \leq n$, $\bigcup_{0 \leq i < j} Z_i = Y_m^j \setminus D_1$ by induction on j . We distinguish the following two cases.

– If $j = 1$ then

$$Z_0 = Y_m^1 \setminus Y_m^0 = Y_m^1 \setminus D_1.$$

– If $j > 1$ then

$$\begin{aligned} \bigcup_{0 \leq i < j} Z_i &= Z_{j-1} \cup \bigcup_{0 \leq i < j-1} Z_i \\ &= Z_{j-1} \cup (Y_m^{j-1} \setminus D_1) \quad [\text{by induction}] \\ &= (Y_m^j \setminus Y_m^{j-1}) \cup (Y_m^{j-1} \setminus D_1) \\ &= Y_m^j \setminus D_1 \quad [Y_m^{j-1} \subseteq Y_m^j \text{ by Proposition 28(b)}] \end{aligned}$$

The result now follows from the observation that $X_m = Y_m^n$ (Lemma 29(c)).

(d) We prove that for all $Y_m^i = D_1 \cup \bigcup_{0 \leq j < i} Z_j$ by induction on i . We distinguish the following two cases.

– By definition, $Y_m^0 = D_1$.

– If $i > 0$ then

$$\begin{aligned} Y_m^i &= Y_m^{i-1} \cup (Y_m^i \setminus Y_m^{i-1}) \\ &= Y_m^{i-1} \cup Z_{i-1} \\ &= (D_1 \cup \bigcup_{0 \leq j < i-1} Z_j) \cup Z_{i-1} \quad [\text{by induction}] \\ &= D_1 \cup \bigcup_{0 \leq j < i} Z_j \end{aligned}$$

□

According to Proposition 30(b) and (c), the sets Z_0, Z_1, \dots, Z_{n-1} form a partition of $X_m \setminus D_1$. The next proposition is the key ingredient for the construction of the max policy A' .

Proposition 31. *For all $0 \leq i < n$ and $(s, t) \in Z_i$,*

$$\exists s \rightarrow \mu : \forall t \rightarrow \nu : \forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq X_m \wedge \text{support}(\omega) \cap Y_m^i \neq \emptyset \vee \quad (14)$$

$$\exists t \rightarrow \nu : \forall s \rightarrow \mu : \forall \omega \in V(\Omega(\nu, \mu)) : \text{support}(\omega) \subseteq X_m \wedge \text{support}(\omega) \cap Y_m^i \neq \emptyset \quad (15)$$

PROOF. Let $0 \leq i < n$ and $(s, t) \in Z_i$. Then

$$(s, t) \in Z_i$$

$$\text{iff } (s, t) \in Y_m^{i+1} \setminus Y_m^i$$

$$\text{implies } (s, t) \in Y_m^{i+1}$$

$$\text{iff } (s, t) \in \Lambda(X_m, Y_m^i)$$

$$\text{iff } \exists s \rightarrow \mu : \forall t \rightarrow \nu : \forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq X_m \wedge \text{support}(\omega) \cap Y_m^i \neq \emptyset \vee$$

$$\exists t \rightarrow \nu : \forall s \rightarrow \mu : \forall \omega \in V(\Omega(\nu, \mu)) : \text{support}(\omega) \subseteq X_m \wedge \text{support}(\omega) \cap Y_m^i \neq \emptyset$$

$$[(s, t) \in Z_i \subseteq S_\gamma^2 \text{ by Proposition 30(a)}]$$

□

Based on the above proposition, we construct a max policy A' .

Definition 16. The function $A' : S_7^2 \rightarrow (S \times \text{Distr}(S))$ is defined by

$$A'(s, t) = \begin{cases} (t, \mu) & \text{if } (s, t) \in Z_i \text{ and (14)} \\ (s, \nu) & \text{if } (s, t) \in Z_i \text{ and (15)} \\ A^*(s, t) & \text{if } (s, t) \in S_7^2 \setminus (X_m \setminus D_1). \end{cases}$$

Note that the max policy A' only differs from the optimal max policy A^* on $X_m \setminus D_1$. If the game starts in X_m and the max player uses the max policy A' and the min player plays according to the optimal min policy I^* , then the game stays in X_m .

Proposition 32. For all $(s, t) \in X_m \setminus S_1^2$, $\text{support}(I^*(A'(s, t))) \subseteq X_m$.

PROOF. Let $(s, t) \in X_m \setminus S_1^2$. We distinguish two cases.

- If $(s, t) \in D_1 \setminus S_1^2$, then

$$\begin{aligned} \text{support}(I^*(A'(s, t))) &= \text{support}(I^*(A^*(s, t))) \\ &[(s, t) \notin X_m \setminus D_1 \text{ since } (s, t) \in D_1] \\ &\subseteq D_1 \quad [\text{Proposition 25}] \\ &\subseteq X_m \quad [\text{Lemma 27}] \end{aligned}$$

- Otherwise, $(s, t) \in X_m \setminus D_1$ since $S_1^2 \subseteq D_1$. Hence, $(s, t) \in Z_i$ for some $0 \leq i < n$ according to Proposition 30(b) and (c). Without loss of generality, assume that $A'(s, t) = (t, \mu)$. Then $s \rightarrow \mu$ according to (14). Assume that $I^*(t, \mu) = \omega$. Then $t \rightarrow \nu$ and $\omega \in V(\Omega(\mu, \nu))$. From (14) we can conclude that $\text{support}(\omega) \subseteq X_m$. Therefore, $\text{support}(I^*(A'(s, t))) \subseteq X_m$. \square

If the max player uses the max policy A' and the min player plays according to the optimal min policy I^* , then the value of (s, t) is one if and only if s and t have probabilistic bisimilarity distance one.

Lemma 33. *For all $s, t \in S$, $(s, t) \in D_1$ if and only if $\mu(\Gamma_1^{A', I^*})(s, t) = 1$.*

PROOF. Let $s, t \in S$. We prove two implications. Assume that $(s, t) \in D_1$. We will show that for all $i \in \mathbb{N}$, we have that

$$(\Gamma_1^{A', I^*})^i(\vec{0})(s, t) = (\Gamma_1^{A^*, I^*})^i(\vec{0})(s, t). \quad (16)$$

From this fact we can conclude that

$$\begin{aligned} \mu(\Gamma_1^{A', I^*})(s, t) &= \mu(\Gamma_1^{A^*, I^*})(s, t) \quad [\text{Proposition 11 and Theorem 7}] \\ &= \mu(\Delta_1)(s, t) \quad [\text{Corollary 21}] \\ &= 1 \quad [(s, t) \in D_1] \end{aligned}$$

Next, we prove (16). The base case, $i = 0$, is vacuously true. Let $i > 0$. We distinguish two cases.

- If $(s, t) \in S_1^2$ then

$$\begin{aligned} (\Gamma_1^{A', I^*})^i(\vec{0})(s, t) &= \Gamma_1^{A', I^*}((\Gamma_1^{A', I^*})^{i-1}(\vec{0}))(s, t) \\ &= 1 \\ &= \Gamma_1^{A^*, I^*}((\Gamma_1^{A^*, I^*})^{i-1}(\vec{0}))(s, t) \\ &= (\Gamma_1^{A^*, I^*})^i(\vec{0})(s, t). \end{aligned}$$

- Otherwise, $(s, t) \in S_7^2$ since $(s, t) \notin S_1^2$ and $(s, t) \in D_1$. Then

$$\begin{aligned}
(\Gamma_1^{A', I^*})^i(\vec{0})(s, t) &= \Gamma_1^{A', I^*}((\Gamma_1^{A', I^*})^{i-1}(\vec{0}))(s, t) \\
&= \sum_{u, v \in S} I^*(A'(s, t))(u, v) (\Gamma_1^{A', I^*})^{i-1}(\vec{0})(u, v) \\
&= \sum_{(u, v) \in X_m} I^*(A'(s, t))(u, v) (\Gamma_1^{A', I^*})^{i-1}(\vec{0})(u, v) \\
&\quad [(s, t) \in D_1 \cap S_7^2 \subseteq X_m \setminus S_1^2, \text{ Proposition 32}] \\
&= \sum_{(u, v) \in X_m} I^*(A^*(s, t))(u, v) (\Gamma_1^{A', I^*})^{i-1}(\vec{0})(u, v) \\
&\quad [(s, t) \in D_1 \cap S_7^2 \subseteq S_7^2 \setminus (X_m \setminus D_1)] \\
&= \sum_{(u, v) \in D_1} I^*(A^*(s, t))(u, v) (\Gamma_1^{A', I^*})^{i-1}(\vec{0})(u, v) \\
&\quad [(s, t) \in D_1 \cap S_7^2 \subseteq D_1 \setminus S_1^2, \text{ Proposition 25}] \\
&= \sum_{(u, v) \in D_1} I^*(A^*(s, t))(u, v) (\Gamma_1^{A^*, I^*})^{i-1}(\vec{0})(u, v) \\
&\quad [\text{by induction}] \\
&= \sum_{u, v \in S} I^*(A^*(s, t))(u, v) (\Gamma_1^{A^*, I^*})^{i-1}(\vec{0})(u, v) \\
&\quad [(s, t) \in D_1 \cap S_7^2 \subseteq D_1 \setminus S_1^2, \text{ Proposition 25}] \\
&= \Gamma_1^{A^*, I^*}((\Gamma_1^{A^*, I^*})^{i-1}(\vec{0}))(s, t) \\
&= (\Gamma_1^{A^*, I^*})^i(\vec{0})(s, t).
\end{aligned}$$

To prove the other implication, assume that $\boldsymbol{\mu}(\Gamma_1^{A', I^*})(s, t) = 1$. Then

$$\begin{aligned}
1 &= \boldsymbol{\mu}(\Gamma_1^{A', I^*})(s, t) \\
&\leq \boldsymbol{\mu}(\Delta_1)(s, t) \quad [\text{Theorem 12}] \\
&\leq 1
\end{aligned}$$

Hence, $\boldsymbol{\mu}(\Delta_1)(s, t) = 1$, that is, $(s, t) \in D_1$. \square

Combining the above results, we arrive at the following.

Theorem 34. $X_m \setminus D_1 = \emptyset$

PROOF. Towards a contradiction, assume that $X_m \setminus D_1 \neq \emptyset$. Let

$$\begin{aligned} \min &= \min\{ \boldsymbol{\mu}\Gamma_1^{A', I^*}(s, t) \mid (s, t) \in X_m \setminus D_1 \} \\ M &= \{ (s, t) \in X_m \setminus D_1 \mid \boldsymbol{\mu}\Gamma_1^{A', I^*}(s, t) = \min \} \end{aligned}$$

Since $X_m \setminus D_1 \neq \emptyset$, we can conclude that \min exists and $M \neq \emptyset$.

First, we will argue that $\min < 1$. Towards a contradiction, assume that $\min = 1$. Then there exists $(s, t) \in X_m \setminus D_1$ such that $\boldsymbol{\mu}\Gamma_1^{A', I^*}(s, t) = 1$. By Lemma 33, $(s, t) \in D_1$, which contradicts $(s, t) \in X_m \setminus D_1$.

Next, we will prove that

$$\forall 1 \leq i < n : Z_i \cap M \neq \emptyset \Rightarrow \exists 0 \leq j < i : Z_j \cap M \neq \emptyset. \quad (17)$$

Let $0 \leq i < n$ and assume that $Z_i \cap M \neq \emptyset$. Let $(s, t) \in Z_i$ with $\boldsymbol{\mu}\Gamma_1^{A', I^*}(s, t) = \min$. Then

$$\begin{aligned} \min &= \boldsymbol{\mu}\Gamma_1^{A', I^*}(s, t) \\ &= \Gamma_1^{A', I^*}(\boldsymbol{\mu}\Gamma_1^{A', I^*})(s, t) \\ &= \sum_{u, v \in S} I^*(A'(s, t))(u, v) \boldsymbol{\mu}\Gamma_1^{A', I^*}(u, v) \\ &= \sum_{(u, v) \in X_m} I^*(A'(s, t))(u, v) \boldsymbol{\mu}\Gamma_1^{A', I^*}(u, v) \\ &\quad [(s, t) \in Z_i \subseteq X_m \setminus D_1 \subseteq X_m \setminus S_1^2, \text{ Proposition 32}] \end{aligned}$$

From Lemma 33 we can deduce that $\mu\Gamma_1^{A',I^*}(u,v) = 1$ for all $(u,v) \in D_1$ and, by the definition of \min , $\mu\Gamma_1^{A',I^*}(u,v) \geq \min$ for all $(u,v) \in X_m \setminus D_1$. Since $\min < 1$, from the above we can deduce that $\text{support}(I^*(A'(s,t))) \subseteq X_m \setminus D_1$ and $\mu\Gamma_1^{A',I^*}(u,v) = \min$ for all $(u,v) \in \text{support}(I^*(A'(s,t)))$.

Let $i \geq 1$. Since $(s,t) \in Z_i$, from Proposition 31 and Definition 16 we can conclude that $\text{support}(I^*(A'(s,t))) \cap Y_m^i \neq \emptyset$, that is, $\text{support}(I^*(A'(s,t))) \cap (D_1 \cup \bigcup_{0 \leq j < i} Z_j) \neq \emptyset$ by Proposition 30(d). Combining the above, we can conclude that $\text{support}(I^*(A'(s,t))) \cap \bigcup_{0 \leq j < i} Z_j \neq \emptyset$. Hence, there exists $0 \leq j < i$ and $(u,v) \in Z_j$ such that $\mu\Gamma_1^{A',I^*}(u,v) = \min$.

Recall from Proposition 30(c) that $X_m \setminus D_1 = \bigcup_{0 \leq i < n} Z_i$. Since $M \neq \emptyset$, we can deduce that $Z_i \cap M \neq \emptyset$ for some $0 \leq i < n$. From (17) we can conclude that $Z_0 \cap M \neq \emptyset$. Assume that $(s,t) \in Z_0 \cap M$. As we have seen above, $\text{support}(I^*(A'(s,t))) \subseteq X_m \setminus D_1$. From Proposition 31 and Definition 16 we can conclude that $\text{support}(I^*(A'(s,t))) \cap Y_m^0 \neq \emptyset$, that is, $\text{support}(I^*(A'(s,t))) \cap D_1 \neq \emptyset$, which contradicts $\text{support}(I^*(A'(s,t))) \subseteq X_m \setminus D_1$. \square

Corollary 35. $D_1 = \nu X. \mu Y. \Lambda(X, Y)$.

PROOF. Immediate consequence of Lemma 27, Lemma 29(a), and Theorem 34. \square

7. Experiments

In this section, we evaluate the performance of the algorithm of deciding distance one on several probabilistic automata. These probabilistic automata model probabilistic protocols that are part of the distribution of the probabilistic model checker PRISM [57].

We implemented the iterative algorithm of deciding distance one presented in Section 5 in Java.⁴ As $S_7^2 = (S \times S) \setminus (S_0^2 \cup S_1^2)$ is required when applying the Λ operator, we need to compute S_0^2 and S_1^2 . The set S_1^2 simply denotes the set of state pairs which have different labels and can be computed by comparing the labels of each state pair, whereas S_0^2 denotes the state pairs which have distance zero. As we mentioned earlier, distance zero coincides with probabilistic bisimilarity. We thus implemented the two-phased partitioning algorithm of deciding probabilistic bisimilarity for probabilistic automata by Baier et al. [58] in Java. This algorithm runs in time $O(mn(\log m + \log n))$ where m is the number of probabilistic transitions and n the number of states of the probabilistic automaton.

In our experiments, we keep track of $|S_0^2|$, $|S_1^2|$ and $|D_1 \setminus S_1^2|$. As a consequence, we can determine the number of distances that are non-trivial, that is, greater than zero and smaller than one: $|S|^2 - (|S_0^2| + |S_1^2| + |D_1 \setminus S_1^2|)$. The distances of the state pairs in S_7^2 can be computed using, for example, policy iteration [59]. This algorithm runs in exponential time in the worst case [24]. Note that the state pairs in $D_1 \setminus S_1^2$ can be computed in polynomial time. Hence, the number of state pairs that remains to be computed has been reduced by $|D_1 \setminus S_1^2|$.

We applied our implementation to probabilistic automata obtained from PRISM. We compute the above mentioned reduction of the number of non-trivial distances for the following models: the randomized consensus shared coin protocol due to Aspnes and Herlihy [60] and the IPv4 zeroconf protocol

⁴The source code is available at github.com/qiyitang71/distance-one-probabilistic-automata.

by Cheshire, Adoba and Guttman [61].

In the model of the randomized consensus shared coin protocol, there are two parameters: N denotes the number of processes and K a constant which together with N determines when the algorithm terminates. The results for the randomized consensus shared coin protocol are shown in the table below. We can only handle up to 528 states for this protocol. As we can see in Table 2, the algorithm takes more than 20 hours to terminate when the number of states is 528. For the system with $N = 2$ and $K = 2$, there remain only 30 non-trivial distances to compute. Without our algorithm to decide distance one, 14,427 non-trivial distances remain to be computed. An even more dramatic decrease in the number of non-trivial distances can be witnessed for the case $N = 2$ and $K = 4$.

N	K	$S_0^2 + D_1$	$ S $	$ S_0^2 $	$ S_1^2 $	$ D_1 \setminus S_1^2 $	non-trivial
2	2	32.6 min	272	422	22,279	14,397	30
2	4	20.4 hrs	528	790	82,479	56,357	30

Table 2: the results for the randomized consensus shared coin protocol.

The IPv4 zeroconf protocol is a dynamic configuration protocol for IPv4 addresses. The protocol modelled in PRISM has three parameters: N denotes the number of abstract hosts, K the number of probes each host sends and a boolean value *reset*. When *reset* is set to *true*, each host should delete the previously received messages before choosing a new IP address. For details of the model, we refer interested reader to [62]. The smallest size of this model has 451 states, when $K = 1$ and *reset* = *true*. This is actually the only size we can handle for this protocol. We consider two systems by considering

two different values for N : 20 and 1000. The results are shown in Table 3. Both systems have the same number of states and also the same number of state pairs of distance zero and one. The algorithm runs for about 17 hours and reduces the number of non-trivial distances from 90,656 to 71,236 for both systems.

N	$S_0^2 + D_1$	$ S $	$ S_0^2 $	$ S_1^2 $	$ D_1 \setminus S_1^2 $	non-trivial
20	16.7 hrs	451	10,820	450	19,420	71,236
1000	17.1 hrs	451	10,820	450	19,420	71,236

Table 3: the results for IPv4 zeroconf protocol.

8. Conclusion

In the introduction, we alluded to the connection between reinforcement learning and behavioural pseudometrics for labelled Markov chains (see Table 1). Here, we briefly discuss a connection between game theory and behavioural pseudometrics for probabilistic automata (see Table 4).

behavioural pseudometrics	game theory
probabilistic automaton	simple stochastic game
distances	values
policy iteration	

Table 4: correspondence between ingredients of behavioural pseudometrics and game theory.

Chen et al. [21] have provided an alternative characterization of the prob-

abilistic bisimilarity distances for a labelled Markov chain as the values of a Markov decision process. This characterization forms the basis for the algorithm to compute the probabilistic bisimilarity distances for labelled Markov chains by Bacci et al. [23]. Their algorithm is similar to Howard’s policy iteration algorithm [28]. In this paper we have presented an alternative characterization of the probabilistic bisimilarity distances for a probabilistic automaton as the values of a simple stochastic game. Bacci et al. [59] have recently used a similar characterization as the foundation for an algorithm to compute the probabilistic bisimilarity distances for probabilistic automata based on the policy iteration algorithm from game theory due to Hoffman and Karp [63].

As shown by Baier [25], probabilistic bisimilarity distance zero for probabilistic automata can be decided in polynomial time. In this paper we have shown that distance one can also be decided in polynomial time. As a consequence, we can determine in polynomial time how many, if any, distances are non-trivial, that is, greater than zero and smaller than one. As we have already shown in [26] in the context of labelled Markov chains, being able to decide distance zero and distance one in polynomial time has significant impact on computing probabilistic bisimilarity distances for labelled Markov chains. The algorithm by Bacci et al. [23], that does not decide distance one before computing the non-trivial distances using policy iteration, can compute distances for labelled Markov chains up to 150 states. For one such labelled Markov chain, their algorithm takes more than 49 hours. Our algorithm that we present in [26] decides distance zero and distance one before using policy iteration to compute the non-trivial distances. Our algorithm

takes 13 milliseconds instead of 49 hours. For more details, we refer the reader to [27, Chapter 9].

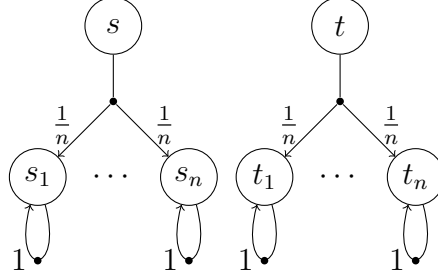


Figure 7: a probabilistic automaton with uniform distributions

Consider the probabilistic automaton in Figure 7. This probabilistic automaton induces the game graph depicted in Figure 8. If μ and ν are both the uniform distribution on n elements, then the vertices of $\Omega(\mu, \nu)$ can be viewed as permutations (see [64]). As a result, from the state pair (s, t) after one move by the max player and one move by the min player, $n!$ vertices can be reached. Hence, we may encounter an exponential blow-up when we transform a probabilistic automaton into a simple stochastic game. As a consequence, it is not immediately obvious which results from game theory can be transferred to behavioural pseudometrics. Let us briefly discuss one paper about games that contains seemingly related results that deserve further study. In [65] de Alfaro, Henzinger and Kupferman study reachability games. These games have two players and the strategies can be randomized. They present an algorithm to compute the almost-sure reachable states, that is, those states for which one of the players has a strategy to reach a particular set of states with probability one. Similarly, the function Λ induces a

strategy of the max player and is related to reachability of the set S_1^2 .

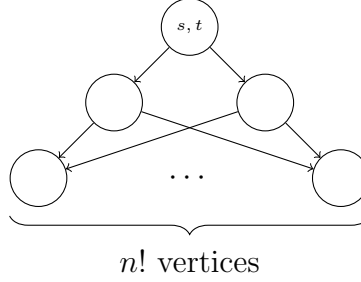


Figure 8: part of the game graph corresponding to the probabilistic automaton of Figure 7.

To prove Theorem 20, which provides the second part of the proof of the alternative characterization of the probabilistic bisimilarity distances, we rely on the discounted functions Δ_c and $\Gamma_c^{A^*, I}$ for $c \in (0, 1)$. In particular, in the proof of Lemma 19 we use the fact that $\Gamma_c^{A^*, I}$ has a unique fixed point. If we were able to prove that $\Gamma_1^{A^*, I}$ has a unique fixed point, then we would be able to give a proof of Theorem 20 that does not rely on discounted functions. We also leave that for future research.

Acknowledgements

The authors would like to thank the referees for their detailed and constructive feedback.

References

- [1] Q. Tang, F. v. Breugel, Deciding probabilistic bisimilarity distance one for probabilistic automata, in: S. Schewe, L. Zhang (Eds.), Proceedings of the 29th International Conference on Concurrency Theory, Vol. 118

of Leibniz International Proceedings in Informatics, Schloss Dagstuhl
– Leibniz-Zentrum für Informatik, Beijing, China, 2018, pp. 9:1–9:17.
`doi:10.4230/LIPIcs.CONCUR.2018.9`.

- [2] R. Milner, An algebraic definition of simulation between programs, in:
D. Cooper (Ed.), Proceedings of the 2nd International Joint Conference
on Artificial intelligence, Kaufmann, London, UK, 1971, pp. 481–489.
- [3] D. Park, Concurrency and automata on infinite sequences, in:
P. Deussen (Ed.), Proceedings of 5th GI-Conference on Theoretical
Computer Science, Vol. 104 of Lecture Notes in Computer Science,
Springer-Verlag, Karlsruhe, Germany, 1981, pp. 167–183. `doi:10.1007/
BFb0017309`.
- [4] A. Giacalone, C.-C. Jou, S. Smolka, Algebraic reasoning for probabilistic
concurrent systems, in: Proceedings of the IFIP WG 2.2/2.3 Working
Conference on Programming Concepts and Methods, North-Holland,
Sea of Gallilee, Israel, 1990, pp. 443–458.
- [5] R. Segala, Modeling and verification of randomized distributed real-
time systems, Ph.D. thesis, Massachusetts Institute of Technology, Cam-
bridge, MA, USA (Jun. 1995).
- [6] R. Segala, N. Lynch, Probabilistic simulations for probabilistic pro-
cesses, in: B. Jonsson, J. Parrow (Eds.), Proceedings of the 5th Inter-
national Conference on Concurrency Theory, Vol. 836 of Lecture Notes
in Computer Science, Springer-Verlag, Uppsala, Sweden, 1994, pp. 481–
496. `doi:10.1007/BFb0015027`.

- [7] K. Larsen, A. Skou, Bisimulation through probabilistic testing, in: Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages, ACM, Austin, TX, USA, 1989, pp. 344–352. doi:10.1145/75277.75307.
- [8] Y. Deng, T. Chothia, C. Palamidessi, J. Pang, Metrics for action-labelled quantitative transition systems, in: A. Cerone, H. Wiklicky (Eds.), Proceedings of the 3rd Workshop on Quantitative Aspects of Programming Languages, Vol. 153(2) of Electronic Notes in Theoretical Computer Science, Elsevier, Edinburgh, Scotland, 2005, pp. 79–96. doi:10.1016/j.entcs.2005.10.033.
- [9] J. Desharnais, V. Gupta, R. Jagadeesan, P. Panangaden, Metrics for labeled Markov systems, in: J. Baeten, S. Mauw (Eds.), Proceedings of the 10th International Conference on Concurrency Theory, Vol. 1664 of Lecture Notes in Computer Science, Springer-Verlag, Eindhoven, The Netherlands, 1999, pp. 258–273. doi:10.1007/3-540-48320-9_19.
- [10] F. Hausdorff, Grundzüge der Mengenlehre, Von Veit & Comp., Leipzig, 1914.
- [11] L. Kantorovich, On the translocation of masses (in Russian), Doklady Akademii Nauk 5 (1) (1942) 1–4, translated in Management Science 5 (1) (1958) 1–4. doi:10.1287/mnsc.5.1.1.
- [12] J. d. Bakker, J. Zucker, Denotational semantics of concurrency, in: Proceedings of the 14th Annual ACM Symposium on Theory of Com-

- puting, ACM, San Francisco, CA, USA, 1982, pp. 153–158. doi:10.1145/800070.802188.
- [13] F. v. Breugel, J. Worrell, Towards quantitative verification of probabilistic systems, in: F. Orejas, P. Spirakis, J. van Leeuwen (Eds.), Proceedings of 28th International Colloquium on Automata, Languages and Programming, Vol. 2076 of Lecture Notes in Computer Science, Springer-Verlag, Crete, 2001, pp. 421–432. doi:10.1007/3-540-48224-5_35.
 - [14] M. Tracol, J. Desharnais, A. Zhioua, Computing distances between probabilistic automata, in: M. Massink, G. Norman (Eds.), Proceedings 9th Workshop on Quantitative Aspects of Programming Languages, Vol. 57 of Electronic Proceedings in Theoretical Computer Science, Open Publishing Association, Saarbrücken, Germany, 2011, pp. 148–162. doi:10.4204/EPTCS.57.11.
 - [15] F. v. Breugel, B. Sharma, J. Worrell, Approximating a behavioural pseudometric without discount, Logical Methods in Computer Science 4 (2). doi:10.2168/lmcs-4(2:2)2008.
 - [16] A. Tarski, A decision method for elementary algebra and geometry, University of California Press, Berkeley, CA, USA, 1951.
 - [17] J. Canny, Some algebraic and geometric computations in PSPACE, in: J. Simon (Ed.), Proceedings of the 20th Annual ACM Symposium on Theory of Computing, ACM, Chicago, IL, USA, 1988, pp. 460–467. doi:10.1145/62212.62257.

- [18] J. Renegar, On the computational complexity and geometry of the first-order theory of the reals, *Journal of Symbolic Computation* 13 (3) (1992) 255–299. doi:10.1016/S0747-7171(10)80003-3.
- [19] W. Visser, K. Havelund, G. Brat, S. Park, F. Lerda, Model checking programs, *Automated Software Engineering* 10 (2) (2003) 203–232. doi:10.1023/A:1022920129859.
- [20] J. Desharnais, V. Gupta, R. Jagadeesan, P. Panangaden, The metric analogue of weak bisimulation for probabilistic processes, in: *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*, IEEE, Copenhagen, Denmark, 2002, pp. 413–422. doi:10.1109/LICS.2002.1029849.
- [21] D. Chen, F. v. Breugel, J. Worrell, On the complexity of computing probabilistic bisimilarity, in: L. Birkedal (Ed.), *Proceedings of the 15th International Conference on Foundations of Software Science and Computational Structures*, Vol. 7213 of *Lecture Notes in Computer Science*, Springer-Verlag, Tallinn, Estonia, 2012, pp. 437–451. doi:10.1007/978-3-642-28729-9_29.
- [22] R. Sutton, A. Barto, *Reinforcement learning: an introduction*, MIT Press, Cambridge, MA, USA, 2018.
- [23] G. Bacci, G. Bacci, K. Larsen, R. Mardare, On-the-fly exact computation of bisimilarity distances, in: N. Piterman, S. Smolka (Eds.), *Proceedings of the 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Vol. 7795 of *Lec-*

- ture Notes in Computer Science, Springer-Verlag, Rome, Italy, 2013, pp. 1–15. doi:10.1007/978-3-642-36742-7_1.
- [24] Q. Tang, F. v. Breugel, Computing probabilistic bisimilarity distances via policy iteration, in: J. Desharnais, R. Jagadeesan (Eds.), Proceedings of the 27th International Conference on Concurrency Theory, Vol. 59 of Leibniz International Proceedings in Informatics, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Quebec City, QC, Canada, 2016, pp. 22:1–22:15. doi:10.4230/LIPIcs.CONCUR.2016.22.
 - [25] C. Baier, Polynomial time algorithms for testing probabilistic bisimulation and simulation, in: R. Alur, T. Henzinger (Eds.), Proceedings of the 8th International Conference on Computer Aided Verification, Vol. 1102 of Lecture Notes in Computer Science, Springer-Verlag, New Brunswick, NJ, USA, 1996, pp. 50–61. doi:10.1007/3-540-61474-5_57.
 - [26] Q. Tang, F. v. Breugel, Deciding probabilistic bisimilarity distance one for labelled Markov chains, in: H. Chockler, G. Weissenbacher (Eds.), Proceedings of the 30th International Conference on Computer Aided Verification, Vol. 10981 of Lecture Notes in Computer Science, Springer-Verlag, Oxford, UK, 2018, pp. 681–699. doi:10.1007/978-3-319-96145-3_39.
 - [27] Q. Tang, Computing probabilistic bisimilarity distances, Ph.D. thesis, York University, Toronto, Canada (Aug. 2018).
 - [28] R. Howard, Dynamic programming and Markov processes, MIT Press, Cambridge, MA, USA, 1960.

- [29] H. Fu, Computing game metrics on Markov decision processes, in: A. Czumaj, K. Mehlhorn, A. Pitts, R. Wattenhofer (Eds.), Proceedings of the 39th International Colloquium on Automata, Languages, and Programming, Vol. 7392 of Lecture Notes in Computer Science, Springer-Verlag, Warwick, UK, 2012, pp. 227–238. doi:10.1007/978-3-642-31585-5_23.
- [30] H. Fu, Verifying probabilistic systems: New algorithms and complexity results, Ph.D. thesis, RWTH Aachen, Aachen, Germany (Nov. 2014).
- [31] F. v. Breugel, J. Worrell, The complexity of computing a bisimilarity pseudometric on probabilistic automata, in: F. van Breugel, E. Kashefi, C. Palamidessi, J. Rutten (Eds.), Horizons of the Mind – A Tribute to Prakash Panangaden, Vol. 8464 of Lecture Notes in Computer Science, Springer-Verlag, 2014, pp. 191–213. doi:10.1007/978-3-319-06880-0_10.
- [32] C. Papadimitriou, On the complexity of the parity argument and other inefficient proofs of existence, Journal of Computer and System Sciences 48 (3) (1994) 498–532. doi:10.1016/S0022-0000(05)80063-7.
- [33] C. Lemke, J. Howson, Equilibrium points of bimatrix games, Journal of the Society for Industrial and Applied Mathematics 12 (2) (1964) 413–423. doi:10.1137/0112033.
- [34] H. Scarf, The approximation of fixed points of a continuous mapping, Journal of the Society for Industrial and Applied Mathematics 15 (5) (1967) 1328–1343. doi:10.1137/0115116.

- [35] X. Chen, X. Deng, Settling the complexity of two-player Nash equilibrium, in: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE, Berkeley, CA, USA, 2006, pp. 261–272. doi:10.1109/FOCS.2006.69.
- [36] S. Kintali, L. Poplawski, R. Rajaraman, R. Sundaram, S.-H. Teng, Reducibility among fractional stability problems, in: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, IEEE, Atlanta, GA, USA, 2009, pp. 283–292. doi:10.1109/FOCS.2009.57.
- [37] K. Etessami, M. Yannakakis, On the complexity of Nash equilibria and other fixed points (extended abstract), in: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, IEEE, Providence, RI, USA, 2007, pp. 113–123. doi:10.1109/FOCS.2007.39.
- [38] T. Chen, T. Han, J. Lu, On behavioral metric for probabilistic systems: definition and approximation algorithm, in: J. Lei, J. Yu, S. Zhou (Eds.), Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, Haikou, China, 2007, pp. 21–25. doi:10.1109/FSKD.2007.426.
- [39] A. Condon, The complexity of stochastic games, *Information and Computation* 96 (2) (1992) 203–224. doi:10.1016/0890-5401(92)90048-k.
- [40] B. Davey, H. Priestley, Introduction to lattices and order, Cambridge University Press, Cambridge, UK, 2002. doi:10.1017/cbo9780511809088.

- [41] J. d. Bakker, E. d. Vink, Control flow semantics, MIT Press, Cambridge, MA, USA, 1996.
- [42] B. Knaster, Un théorème sur les fonctions d'ensembles, Annales de la Société Polonaise de Mathématique 6 (1928) 133–134.
- [43] A. Tarski, A lattice-theoretic fixed point theorem and its applications, Pacific Journal of Mathematics 5 (2) (1955) 285–309. doi:10.2140/pjm.1955.5.285.
- [44] E. Clarke, O. Grumberg, D. Kroening, D. Peled, H. Veith, Model checking, MIT Press, Cambridge, MA, USA, 2018.
- [45] S. Banach, Sur les opérations dans les ensembles abstraits et leurs applications aux equations intégrales, Fundamenta Mathematicae 3 (1922) 133–181. doi:10.4064/fm-3-1-133-181.
- [46] F. v. Breugel, On behavioural pseudometrics and closure ordinals, Information Processing Letters 112 (19) (2012) 715–718. doi:10.1016/j.ipl.2012.06.019.
- [47] W. Doeblin, Exposé de la théorie des chaînes simples constant es de Markov à un nombre fini d'états, Revue mathématique de l'union inter-balkanique 2 (1938) 77–105.
- [48] V. Klee, C. Witzgall, Facets and vertices of transportation polytopes, in: G. Dantzig, A. Veinott (Eds.), Proceedings of 5th Summer Seminar on the Mathematis of the Decision Sciences, Vol. 11 of Lectures in Applied Mathematics, AMS, Stanford, CA, USA, 1967, pp. 257–282.

- [49] B. Jonsson, K. Larsen, Specification and refinement of probabilistic processes, in: Proceedings of the 6th Annual Symposium on Logic in Computer Science, IEEE, Amsterdam, The Netherlands, 1991, pp. 266–277. doi:10.1109/lics.1991.151651.
- [50] F. v. Breugel, Probabilistic bisimilarity distances, SIGLOG News 4 (4) (2017) 33–51. doi:10.1145/3157831.3157837.
- [51] L. Shapley, Stochastic games, Proceedings of the Academy of Sciences 39 (10) (1953) 1095–1100. doi:10.1073/pnas.39.10.1095.
- [52] U. Zwick, M. Paterson, The complexity of mean payoff games on graphs, Theoretical Computer Science 158 (1/2) (1996) 343–359. doi:10.1016/0304-3975(95)00188-3.
- [53] C. Stirling, Bisimulation, modal logic and model checking games, Logic Journal of the IGPL 7 (1) (1999) 103–124. doi:10.1093/jigpal/7.1.103.
- [54] J. Filar, K. Vrieze, Competitive Markov decision processes, Springer-Verlag, New York, NY, USA, 1997. doi:10.1007/978-1-4612-4054-9.
- [55] J. Orlin, A polynomial time primal network simplex algorithm for minimum cost flows, Mathematical Programming 78 (2) (1997) 109–129. doi:10.1007/bf02614365.
- [56] G. Dantzig, Linear programming and extensions, Princeton University Press, Princeton, NJ, USA, 1963.

- [57] M. Kwiatkowska, G. Norman, D. Parker, Prism 4.0: Verification of probabilistic real-time systems, in: G. Gopalakrishnan, S. Qadeer (Eds.), Proceedings of the 23rd International Conference on Computer Aided Verification, Vol. 6806 of Lecture Notes in Computer Science, Springer-Verlag, Snowbird, UT, USA, 2011, pp. 585–591. doi:10.1007/978-3-642-22110-1_47.
- [58] C. Baier, B. Engelen, M. Majster-Cederbaum, Deciding bisimilarity and similarity for probabilistic processes, Journal of Computer and System Sciences 60 (1) (2000) 187–231. doi:10.1006/jcss.1999.1683.
- [59] G. Bacci, G. Bacci, K. Larsen, R. Mardare, Q. Tang, F. v. Breugel, Computing probabilistic bisimilarity distances for probabilistic automata, in: W. Fokkink, R. van Glabbeek (Eds.), Proceedings of the 30th International Conference on Concurrency Theory, Vol. 140 of Leibniz International Proceedings in Informatics, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Amsterdam, The Netherlands, 2019, pp. 9:1–9:17. doi:10.4230/LIPIcs.CONCUR.2019.9.
- [60] J. Aspnes, M. Herlihy, Fast randomized consensus using shared memory, Journal of Algorithms 11 (3) (1990) 441–461. doi:10.1016/0196-6774(90)90021-6.
- [61] S. Cheshire, B. Adoba, E. Gutterman, Dynamic configuration of IPv4 link local addresses, available from www.ietf.org/rfc/rfc3927.txt (May 2005).
- [62] M. Kwiatkowska, G. Norman, D. Parker, J. Sproston, Performance

- analysis of probabilistic timed automata using digital clocks, *Formal Methods in System Design* 29 (1) (2006) 33–78. doi:10.1007/s10703-006-0005-2.
- [63] A. Hoffman, R. Karp, On nonterminating stochastic games, *Management Science* 12 (5) (1966) 359–370. doi:10.1287/mnsc.12.5.359.
- [64] G. Birkhoff, Tres observaciones sobre el álgebra lineal, *Universidad Nacional de Tucumán Revista A* 5 (1946) 147–151.
- [65] L. de Alfaro, T. Henzinger, O. Kupferman, Concurrent reachability games, in: *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, IEEE, Palo Alto, CA, USA, 1998, pp. 564–575. doi:10.1109/SFCS.1998.743507.