Strategies for MDP Bisimilarity Equivalence and Inequivalence

Stefan Kiefer
Department of Computer Science, University of Oxford, UK

Qiyi Tang
Department of Computer Science, University of Liverpool, UK

Abstract

A labelled Markov decision process (MDP) is a labelled Markov chain with nondeterminism; i.e., together with a strategy a labelled MDP induces a labelled Markov chain. Motivated by applications to the verification of probabilistic noninterference in security, we study problems whether there exist strategies such that the labelled MDPs become bisimilarity equivalent/inequivalent. We show that the equivalence problem is decidable; in fact, it is EXPTIME-complete and becomes NP-complete if one of the MDPs is a Markov chain. Concerning the inequivalence problem, we show that (1) it is decidable in polynomial time; (2) if there are strategies for inequivalence then there are memoryless strategies for inequivalence; (3) such memoryless strategies can be computed in polynomial time.

2012 ACM Subject Classification Theory of computation → Program verification; Theory of computation → Models of computation; Mathematics of computing → Probability and statistics

Keywords and phrases Markov decision processes, Markov chains

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.32

Funding This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 956123 (FOCETA).

Acknowledgements We thank the anonymous reviewers of this paper for their constructive feedback.

1 Introduction

Given a model of computation (e.g., finite automata), and two instances of it, are they semantically equivalent (e.g., do the automata accept the same language)? Such *equivalence* problems can be viewed as a fundamental question for almost any model of computation. As such, they permeate computer science, in particular, theoretical computer science.

In *labelled Markov chains (LMCs)*, which are Markov chains whose states (or, equivalently, transitions) are labelled with an observable letter, there are two natural and very well-studied versions of equivalence, namely *trace (or language) equivalence* and *probabilistic bisimilarity*.

The *trace equivalence* problem has a long history, going back to Schützenberger [18] and Paz [15] who studied *weighted* and *probabilistic* automata, respectively. Those models generalize LMCs, but the respective equivalence problems are essentially the same. For LMCs, trace equivalence asks if the same label sequences have the same probabilities in the two LMCs. It can be extracted from [18] that equivalence is decidable in polynomial time, using a technique based on linear algebra; see also [21, 5].

Probabilistic bisimilarity is an equivalence that was introduced by Larsen and Skou [14]. It is finer than trace equivalence, i.e., probabilistic bisimilarity implies trace equivalence. A similar notion for Markov chains, called *lumpability*, can be traced back at least to the classical text by Kemeny and Snell [10]. Probabilistic bisimilarity can also be computed in polynomial time [1, 4, 22]. Indeed, in practice, computing the bisimilarity quotient is fast and has become a backbone for highly efficient tools for probabilistic verification such as Prism [13] and Storm [8].

© Stefan Kiefer and Qiyi Tang;
licensed under Creative Commons License CC-BY 4.0

In this paper, we study probabilistic bisimilarity in *(labelled) Markov decision processes (MDPs)*, which are LMCs plus nondeterminism; i.e., each state may have several *actions* (or "moves") one of which is chosen by a controller, potentially randomly. An MDP and a controller *strategy* together induce an LMC (potentially with infinite state space, depending on the complexity of the strategy). The nondeterminism in MDPs gives rise to a spectrum of equivalence queries: one may ask about the existence of strategies for two given MDPs such that the induced LMCs become trace/bisimilarity equivalent, or such that they become trace/bisimilarity *in*equivalent. Another potential dimension of this spectrum is whether to consider general strategies or more restricted ones, such as memoryless or even memoryless deterministic (MD) ones.

Much of this spectrum has been covered in previous work. It was shown in [6] that whether there exist (general) strategies such that two given MDPs become trace equivalent is undecidable. In fact, even whether there exists a strategy such that a given MDP becomes trace equivalent to a given LMC is undecidable [6, Theorem 3.1]. This points to a fundamental difficulty when dealing with a general strategy in an MDP: since the strategy may use unrestricted memory, the induced LMC can have an infinite (countable) state space, even when the MDP is finite. For this reason it is not a priori clear whether the *bisimilarity equivalence* problem, namely whether there exist general strategies such that two given MDPs become bisimilar, is even decidable.

The problem was "dodged" in [11], where trace and bisimilarity (in)equivalence problems were covered, but under the explicit assumption of *memoryless* strategies. There are good reasons to consider memoryless strategies, particularly their naturalness and simplicity in implementations, and their connection to *interval Markov chains* (see, e.g., [9, 3]) and *parametric MDPs* (see, e.g., [7, 23]). It was shown in [11, Theorem 19] that the bisimilarity equivalence problem is NP-complete for memoryless strategies.

It remained open in [11] whether the bisimilarity equivalence problem is decidable for general strategies, which would be in contrast to the undecidability of the corresponding trace equivalence problem [6]. There are also good reasons to consider general unrestricted strategies, primarily their naturalness (in their definition for MDPs) and their generality. The latter is important particularly for security applications, see below, where an attacker should be conservatively assumed to be powerful to employ an arbitrary strategy.

As one of our main results, we show that the bisimilarity problem for general strategies is decidable, in fact, EXPTIME-complete. This high computational complexity means that in order to induce two bisimilar LMCs in the two given MDPs it is generally necessary to employ complex strategies, inducing complex behaviours. We also show that the computational complexity reduces to NP if one of the MDPs is already an LMC.

The challenges of the corresponding bisimilarity *in*equivalence problems are somewhat analogous, but the results are opposite. It was shown in [11, Corollary 13] that whether there are *memoryless* strategies in two given MDPs that induce *non*bisimilar LMCs can be decided in polynomial time and that such memoryless strategies, if they exist, can be computed in polynomial time. As our second main result we show that this extends in an almost ideal way (although the proof is nontrivial): whenever there are general strategies for inequivalence, there are memoryless ones (and thus can be computed as in [11]). This means that, very much unlike for equivalence, memoryless strategies suffice for inequivalence, inducing relatively simple inequivalent LMCs.

Complementing the theoretical nature of these questions, let us mention an application from the field of security. *Noninterference* refers to an information-flow property of a program, stipulating that information about *high* data (i.e., data with high confidentiality) may not

leak to *low* (i.e., observable) data, or, quoting [17], "that a program is secure whenever varying the initial values of high variables cannot change the low-observable (observable by the attacker) behaviour of the program". It was proposed in [17] to reason about *probabilistic* noninterference in probabilistic multi-threaded programs by proving probabilistic bisimilarity; see also [20, 16]. More precisely, probabilistic noninterference is established if it can be shown that any two states that differ only in high data are probabilistic bisimilar, as then an attacker who only observes the low part of a state learns nothing about the high part. The observable behaviour of a multi-threaded program depends strongly on the *scheduler*, which raises the question whether bisimilarity holds under some or even under all schedulers [17]. A scheduler in this context amounts to a strategy in the corresponding MDP.

The rest of the paper is organized as follows. We give preliminaries in Section 2. In Sections 3 and 4 we prove our results on bisimilarity equivalence and inequivalence, respectively. We conclude in Section 5. Missing proofs can be found in an appendix.

2 Preliminaries

We write $\mathbb{N}$ for the set of nonnegative integers. Let S be a finite set. We denote by $\mathrm{Distr}(S)$ the set of probability distributions on S. For a distribution $\mu \in \mathrm{Distr}(S)$ we write $\mathrm{support}(\mu) = \{s \in S \mid \mu(s) > 0\}$ for its support.

A *labelled Markov chain* (LMC) is a quadruple $\langle S, L, \tau, \ell\rangle$ consisting of a nonempty set S of states[1], a nonempty finite set L of labels, a transition function $\tau : S \to \mathrm{Distr}(S)$, and a labelling function $\ell : S \to L$.

We denote by $\tau(s)(t)$ the transition probability from s to t. Similarly, we denote by $\tau(s)(E) = \sum_{t\in E} \tau(s)(t)$ the transition probability from s to $E \subseteq S$.

An equivalence relation $R \subseteq S \times S$ is a *probabilistic bisimulation* if for all $(s,t) \in R$, $\ell(s) = \ell(t)$ and $\tau(s)(E) = \tau(t)(E)$ for each R-equivalence class E. *Probabilistic bisimilarity*, denoted by $\sim$, is the largest probabilistic bisimulation.

A *(labelled) Markov decision process* (MDP) is a tuple $\langle S, Act, L, \varphi, \ell\rangle$ consisting of a finite set S of states, a finite set Act of actions, a finite set L of labels, a partial function $\varphi : S \times Act \nrightarrow \mathrm{Distr}(S)$ denoting the probabilistic transition, and a labelling function $\ell : S \to L$. The set of available actions in a state s is $Act(s) = \{\mathsf{m} \in Act \mid \varphi(s,\mathsf{m}) \text{ is defined}\}$.

A path is a sequence $\rho = s_0\mathsf{m}_1 s_1 \cdots \mathsf{m}_n s_n$ such that $\varphi(s_i, \mathsf{m}_{i+1})$ is defined and $\varphi(s_i, \mathsf{m}_{i+1})(s_{i+1}) > 0$ for all $0 \le i < n$. The last state of ρ is $\mathsf{last}(\rho) = s_n$. Let $\mathsf{Paths}(\mathcal{D})$ denote the set of paths in $\mathcal{D}$.

A (general) strategy for an MDP is a function $\alpha : \mathsf{Paths}(\mathcal{D}) \to \mathrm{Distr}(Act)$ that given a path ρ, returns a probability distribution on the available actions at the last state of ρ, $\mathsf{last}(\rho)$. A memoryless strategy depends only on $\mathsf{last}(\rho)$; so we can identify a memoryless strategy with a function $\alpha : S \to \mathrm{Distr}(Act)$ that given a state s, returns a probability distribution on the available actions at that state.

Given a general strategy α for $\mathcal{D}$, an LMC $\mathcal{D}(\alpha) = \langle \mathcal{P}, L, \tau, \ell'\rangle$ is induced, where $\mathcal{P} \subseteq \mathsf{Paths}(\mathcal{D})$. For $\rho \in \mathcal{P}$, we have $\tau(\rho)(\rho\mathsf{m}t) = \alpha(\rho)(\mathsf{m})\varphi(s,\mathsf{m})(t)$ and $\ell'(\rho) = \ell(s)$ where $s = \mathsf{last}(\rho)$ and $\mathsf{m} \in Act(s)$.

[1] We mainly consider LMCs with finitely many states unless otherwise stated.

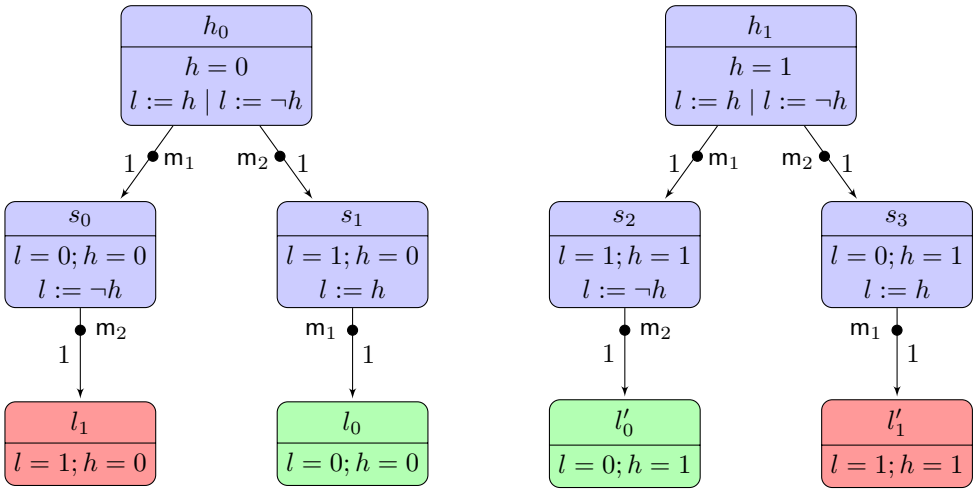

Figure 1 The program $l := h \mid l := \neg h$ as an MDP.

3 Bisimilarity Problems

In this section we consider the bisimilarity problem which, given an MDP and two (initial) states, asks whether there is a general strategy such that the two states are probabilistic bisimilar in the LMC induced by the general strategy.

▶ **Example 1.** Borrowing an example from [17, Section 4], consider the following simple program composed of two threads, involving a *high* boolean variable h (high confidentiality) and a *low* boolean variable l (observable):

$$l := h \quad | \quad l := \neg h$$

The vertical bar | separates two threads. The order in which the threads are executed is determined by a scheduler. We assume that the variable l becomes visible upon program termination. Its value will be h or $\neg h$, depending on whether the left or the right thread is executed *last*. The program can be viewed as an MDP as shown in Figure 1. The top part of each state indicates its name; the bottom part indicates the values of the variables, as well as the part of the program that is yet to be executed. Different colours indicate different state labels. The two top states h_0, h_1 differ in their value of h, but this difference is not observable; thus, h_0, h_1 have the same label. The actions $\mathsf{m}_1, \mathsf{m}_2$ available in h_0, h_1 correspond to the two scheduling options of the program: m_1 means that the thread $l := h$ is scheduled next and m_2 means that the thread $l := \neg h$ is scheduled next. The strategy that in h_0, h_1 picks one of the two actions uniformly at random induces an LMC in which h_0 and h_1 are probabilistic bisimilar. In fact, the bisimilarity equivalence classes under this strategy are $\{h_0, h_1\}$, $\{h_0\mathsf{m}_1 s_0, h_1\mathsf{m}_2 s_3\}$, $\{h_0\mathsf{m}_2 s_1, h_1\mathsf{m}_1 s_2\}$, $\{h_0\mathsf{m}_1 s_0\mathsf{m}_2 l_1, h_1\mathsf{m}_2 s_3\mathsf{m}_1 l'_1\}$, $\{h_0\mathsf{m}_2 s_1\mathsf{m}_1 l_0, h_1\mathsf{m}_1 s_2\mathsf{m}_2 l'_0\}$. Since h_0, h_1 are probabilistic bisimilar, this memoryless strategy prevents a leak of the value of h: an attacker who observes l in the end learns nothing about h.

In this section we show that the bisimilarity problem is EXPTIME-complete. We prove the upper and lower bound in Sections 3.1 and 3.2, respectively.

3.1 Membership in EXPTIME

To prove that the bisimilarity problem can be decided in EXPTIME, we define and analyse an auxiliary game, the *attacker-defender game*. It is a two-person (zero-sum, non-stochastic, turn-based) game, and is defined from an MDP $\mathcal{D} = \langle S, Act, L, \varphi, \ell \rangle$ and a set of states $E_1 \subseteq S$. The two players are called *Defender* and *Attacker*. The intuition (which we will prove in Proposition 3 below) is that Defender can win the game if and only if there is a general strategy α for $\mathcal{D}$ such that in the LMC induced by α all states in E_1 are probabilistic bisimilar. The attacker-defender game proceeds in rounds $1, 2, \ldots$. At the beginning of round 1 the game is in state E_1. Suppose at the beginning of round i the game is in state $E_i \subseteq S$. Then in round i Defender chooses (and announces publicly)

- $S' \subseteq 2^S$ such that for any $E \in S'$ and any $s, t \in E$ we have $\ell(s) = \ell(t)$ (intuitively, Defender claims for each $E' \in S'$ that there is a general strategy such that all states in E' are probabilistic bisimilar);
- a distribution $\upsilon \in \mathrm{Distr}(S')$;
- for each $s \in E_i$ a memoryless strategy (possibly randomised) $\alpha^s \in \mathrm{Distr}(Act(s))$; and
- a function $f : E_i \times Act \times S \to S'$ with $t \in f(s, \mathsf{m}, t)$ for all $(s, \mathsf{m}, t) \in E_i \times Act \times S$

such that for all $s \in E_i$ and all $E' \in S'$ we have

$$\upsilon(E') \;=\; \sum_{\mathsf{m} \in Act(s)} \;\; \sum_{t \in S \text{ s.t. } f(s,\mathsf{m},t) = E'} \alpha^s(\mathsf{m}) \varphi(s, \mathsf{m})(t)\,.$$

If objects with the properties required above do not exist, Defender loses and Attacker wins. Otherwise, to complete round i, Attacker chooses from S' a set E_{i+1}, which is the state of the game at the beginning of round $i{+}1$. If the game goes on forever, Defender wins and Attacker loses.

Memoryless winning strategies suffice for Defender:

▶ **Lemma 2.** *Given an MDP $\mathcal{D} = \langle S, Act, L, \varphi, \ell \rangle$ and a set $E_1 \subseteq S$, if Defender has a winning strategy for the attacker-defender game, Defender has a memoryless winning strategy, i.e., a winning strategy that depends only on the current state of the game.*

Proof. Recall that the states in the attacker-defender game are sets $E \subseteq S$. Let $S_w \subseteq 2^S$ denote the set of those $E \subseteq S$ such that starting from E Defender can win the attacker-defender game. We define a memoryless strategy, σ', for Defender such that starting from any $E \in S_w$, all possible successor states are also in S_w. Therefore, using σ', starting from any $E \in S_w$, the game remains in S_w indefinitely; i.e., σ' is a winning strategy for Defender.

Let $E \in S_w$. Then Defender has a (not necessarily memoryless) winning strategy σ starting from E. According to the rules of the game, in the first round σ chooses various objects, including $S' \subseteq 2^S$. Since σ is winning for Defender, Defender has a (not necessarily memoryless) winning strategy for all $E' \in S'$, i.e., $S' \subseteq S_w$. The memoryless strategy σ' is defined so that in E it makes the same choices that σ makes in E in the first round. All possible successor states are in S_w, as required. ◀

The following proposition establishes the connection between the bisimilarity problem and the attacker-defender game:

▶ **Proposition 3.** *Given an MDP $\mathcal{D} = \langle S, Act, L, \varphi, \ell \rangle$ and a set $E_1 \subseteq S$, Defender has a winning strategy for the attacker-defender game if and only if there exists a general strategy α for $\mathcal{D}$ such that in the LMC induced by α all states in E_1 are probabilistic bisimilar.*

Proof. ($\Longrightarrow$) Assume Defender can win the attacker-defender game which starts in E_1. By Lemma 2, Defender has a memoryless strategy. Using this memoryless strategy (the objects chosen by Defender), a general strategy α for the MDP $\mathcal{D}$ can be constructed.

Let $\mathcal{P}_i$ be a set of paths in $\mathcal{D}$ such that for any $\rho \in \mathcal{P}_i$ we have that ρ begins with a state $s \in E_1$ and the number of states in ρ is i. Let $\mathcal{P} = \bigcup_{i \geq 1} \mathcal{P}_i$. A path in $\mathcal{P}$ can be mapped to a possible state of the attacker-defender game. Let us define $\mathcal{P}$ and such a mapping W inductively on i as follows:

- Base case $i = 1$. We have $\mathcal{P}_1 := E_1$. For any state $s \in E_1$, it is mapped to the start state of the game E_1, that is, $W(s) := E_1$ for $s \in E_1$.
- Inductive case when $i > 1$. For a path $\rho' = \rho \mathsf{m} t$, it belongs to $\mathcal{P}_i$ if and only if $\rho \in \mathcal{P}_{i-1}$ and $\rho' \in \mathsf{Paths}(\mathcal{D})$. Next, we define $W(\rho \mathsf{m} t)$ for a path $\rho \mathsf{m} t \in \mathcal{P}_i$. Let $s = \mathsf{last}(\rho)$ and $E_{i-1} = W(\rho)$. If E_{i-1} is the state at the beginning of some round of the attacker-defender game, Defender chooses a set $S' \subseteq 2^S$ and a function $f : E_{i-1} \times \mathit{Act} \times S \to S'$ with $t \in f(s, \mathsf{m}, t)$ for all $(s, \mathsf{m}, t) \in E_{i-1} \times \mathit{Act} \times S$. We define $W(\rho \mathsf{m} t) := f(s, \mathsf{m}, t)$. Since $f(s, \mathsf{m}, t) \in S'$, $W(\rho \mathsf{m} t)$ may be chosen by Attacker as the new state.

Let $i \geq 1$. A path $\rho \in \mathcal{P}_i$ is mapped to $W(\rho)$, a possible state at the beginning of round i of the attacker-defender game. It can be shown by induction that Defender has a winning strategy for $W(\rho)$. We assume that Defender chooses memoryless strategies $\alpha^s \in \mathrm{Distr}(\mathit{Act}(s))$ for $s \in W(\rho)$. We define the general strategy $\alpha : \mathcal{P} \to \mathrm{Distr}(\mathit{Act})$ as $\alpha(\rho) := \alpha^s$ where $s = \mathsf{last}(\rho)$. We show in the appendix that all states in E_1 are probabilistic bisimilar in the LMC induced by α.

($\Longleftarrow$) We define the notion of an equalisable set. A set $E \subseteq S$ is equalisable if and only if there is a general strategy such that all states in E are probabilistic bisimilar in the induced LMC.

Let $E \subseteq S$ be an arbitrary equalisable set. We define a strategy for Defender when the attacker-defender game is in state E at the beginning of some round.

By definition, there is a general strategy, say α_E, such that all states in E are probabilistic bisimilar in the LMC induced by α_E. In the induced LMC, the successor states of any state of E can be partitioned into probabilistic bisimulation classes, say $B_1, \ldots, B_k$ where k is a positive integer. The transition probability distributions υ' over $B_1, \ldots, B_k$ from any state in E are the same, that is, $\upsilon'(B_i) = \sum_{s \mathsf{m} t \in B_i} \alpha_E(s)(\mathsf{m})\varphi(s, \mathsf{m})(t)$ for any $1 \leq i \leq k$ and $s \in E$. This probability distribution will be the one chosen by Defender. In the LMC induced by α_E, a state $\rho \in B_i$ where $1 \leq i \leq k$, a successor state of $s \in E$, is of the form $s \mathsf{m} t$ where $\mathsf{m} \in \mathrm{support}(\alpha_E(s))$ and $t \in \mathrm{support}(\varphi(s, \mathsf{m}))$. We define $E'_i = \{t \mid s \in E \wedge s \mathsf{m} t \in B_i\}$ for $1 \leq i \leq k$. We are ready to define the objects chosen by Defender when the game is in state E at the beginning of some round:

- $S' = \{E'_1, \ldots, E'_k\}$;
- A probability distribution υ over S' where $\upsilon(E'_i) = \upsilon'(B_i)$ for $1 \leq i \leq k$;
- for any $s \in E$ a memoryless strategy $\alpha^s = \alpha_E(s)$; and
- a function $f : E \times \mathit{Act} \times S \to S'$ such that $f(s, \mathsf{m}, t) = E'_i$ for any $s \in E$ and $s \mathsf{m} t \in B_i$.

We verify in the appendix that the objects chosen by Defender satisfy the required properties. If the game starts with an equalisable set and all the future game states are equalisable sets, Defender can always choose objects with required properties and hence win the game. We prove in the appendix that all sets in S' are equalisable.

As we assume there is a general strategy such that all states in E_1 are probabilistic bisimilar in the induced LMC, E_1 by definition is an equalisable set. This completes the proof. ◀

Now we can prove membership in EXPTIME.

▶ **Lemma 4.** *The bisimilarity problem is in EXPTIME.*

Proof. As EXPTIME equals APSPACE, it suffices to construct a PSPACE-bounded alternating Turing machine M that accepts the bisimilarity problem. By the definition of alternating Turing machines, the set of control states of M is partitioned into existential and universal states. There is an existential (respectively, universal) player who controls the existential (respectively, universal) states, and the player who controls the state of the current configuration chooses a successor configuration that is consistent with the transition relation of M. The goal of the existential player is to drive the computation into an accepting configuration (defined by a special control state), and the goal of the universal player is to prevent that from happening. If the existential player has a winning strategy, M is said to accept the input; otherwise M rejects the input.

In our case, the input of M is an MDP $\mathcal{D} = \langle S, Act, L, \varphi, \ell \rangle$ and two states $s, t \in S$. We need to construct M so that the existential player has a winning strategy if and only if there exists a general strategy α for $\mathcal{D}$ such that s and t are probabilistic bisimilar in the induced LMC $\mathcal{D}(\alpha)$. Using Proposition 3, it suffices to construct M so that the existential player has a winning strategy if and only if Defender has a winning strategy in the attacker-defender game defined by $\mathcal{D}$ and $E_1 := \{s, t\}$.

We construct M so that it implements the attacker-defender game: the existential (respectively, universal) player in M takes the role of Defender (respectively, Attacker) in the attacker-defender game. We have to ensure that M uses only polynomial space. The state of the attacker-defender game at the beginning of each round consists of a set $E_i \subseteq S$, which can clearly be stored in polynomial space. In each round, Defender (i.e., the existential player) needs to choose a set $S' \subseteq 2^S$, a distribution υ on S', memoryless strategies α^s for $s \in E_i$, and a function f. The equations that υ and α^s are required to satisfy imply that Defender can restrict herself to choosing the set S' as the image of the function f; then S' has at most polynomially many sets of states. Further, Defender can restrict herself to choosing υ and α^s such that the numbers are fractions of integers with polynomially many bits (in particular, the numbers are rational). Indeed, given S' and f, there is a linear program of polynomial size whose solutions are exactly those υ and α^s that satisfy, for all $s \in E_i$ and all $E' \in S'$,

$$\upsilon(E') \;=\; \sum_{\mathsf{m} \in Act(s)} \;\; \sum_{t \in S \text{ s.t. } f(s,\mathsf{m},t) = E'} \alpha^s(\mathsf{m})\varphi(s,\mathsf{m})(t)\,.$$

So if there exist any υ and α^s satisfying these equations, then there also exist rational ones with polynomially many bits.

Finally, we have to ensure that M enters an accepting configuration when Defender can win the attacker-defender game (recall that Defender wins the attacker-defender game if and only if it goes on forever). Using a counter on the tape, we make M enter an accepting configuration once $2^{|S|}$ rounds of the attacker-defender game have been played without Attacker having won. This is justified as follows. If Attacker has a winning strategy for the attacker-defender game, Attacker also has a winning strategy that guarantees that every set $E_i \subseteq S$ appears at most once as the state of the game at the beginning of a round. It follows that if Attacker can win, Attacker can also win in at most $2^{|S|}$ rounds. ◀

3.2 EXPTIME-Hardness

Recall that in the previous section an intermediate technical notion (attacker-defender games) was useful to derive the EXPTIME upper bound in the previous section. In this section we show that the bisimilarity problem is EXPTIME-hard. For this lower bound we leverage another non-stochastic intermediate tool, namely intersection emptiness of deterministic tree automata. Let us introduce the required definitions.

A *ranked* alphabet Σ is a finite set of symbols such that each symbol $a \in \Sigma$ is associated with a rank, $rank(a) \in \mathbb{N} \setminus \{0\}$. A *tree* over Σ is an ordered tree in which each node is labelled with a symbol $a \in \Sigma$ (we call such a node an a-node) and every a-node has $rank(a)$ children. Since we exclude symbols of rank 0, a tree over Σ is necessarily infinite. A *deterministic top-down tree automaton (DTTA)* is a quadruple $\mathcal{A} = (Q, \Sigma, \delta, q_0)$ where Q is a finite set of states, Σ is a ranked alphabet, $\delta : Q \times \Sigma \nrightarrow Q^*$ is a *partial* transition function with $|\delta(q, a)| = rank(a)$ for all q, a for which $\delta(q, a)$ is defined, and $q_0 \in Q$ is the initial state. A *run* of DTTA $\mathcal{A}$ on tree t is a labelling of the nodes of t such that the root is labelled with q_0 and for every a-node ($a \in \Sigma$), if it is labelled with $q \in Q$, then its children, read from left to right, are labelled with $\delta(q, a)$. Note that a DTTA has at most one run on any tree. Write $L(\mathcal{A})$ for the set of trees on which $\mathcal{A}$ has a run. Given DTTAs $\mathcal{A}_1, \ldots, \mathcal{A}_k$ over the same ranked alphabet, the *intersection nonemptiness* problem asks whether $\bigcap_{i=1}^{k} L(\mathcal{A}_i) \neq \emptyset$, i.e., whether there is a tree on which every DTTA $\mathcal{A}_i$ has a run. A version of this problem, for finite trees, was proved EXPTIME-hard by Seidl [19]. The version of this problem for DFAs and finite words is a well-known PSPACE-complete problem [12]. By adapting these proofs we show:

▶ **Lemma 5.** *The intersection nonemptiness problem is EXPTIME-hard.*

We use this result to prove the following lemma.

▶ **Lemma 6.** *The bisimilarity problem is EXPTIME-hard.*

Proof. The reduction is from the intersection nonemptiness problem of DTTAs, which by Lemma 5 is EXPTIME-hard.

Given DTTAs $\mathcal{A}_1, \ldots, \mathcal{A}_k$ over the same alphabet, we construct k MDPs $\mathcal{D}_1, \ldots, \mathcal{D}_k$ and show that $\bigcap_{i=1}^{k} L(\mathcal{A}_i) \neq \emptyset \iff$ there is a general strategy for each MDP such that the k initial states in the induced LMCs are probabilistic bisimilar. Later we show how to replace the k MDPs by two MDPs, proving the statement of the lemma.

Let r be the maximum rank of symbols in the DTTAs. Let $L_{order} = \{1, \ldots, r\}$ be a set of integer labels which is disjoint from the set Σ. Let m be an action different from any symbol in Σ. For each DTTA $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, q_0^i)$ where $1 \leq i \leq k$, we construct an MDP $\mathcal{D}_i = \langle S_i, Act, L_i, \varphi_i, \ell_i \rangle$ as follows:

- $S_i := \{qj \mid q \in Q_i \text{ and } j \in \mathbb{N} \text{ and } 1 \leq j \leq r\} \cup \{qa \mid q \in Q_i \text{ and } a \in \Sigma \text{ and } \delta_i(q, a) \text{ is defined}\} \cup \{d_i\}$ where d_i is a special sink state only available in $\mathcal{D}_i$;
- $Act := \Sigma \cup \{\mathsf{m}\}$;
- $L_i := \Sigma \cup L_{order} \cup \{deadlock_i\}$ where $deadlock_i$ is a special symbol only available in $\mathcal{D}_i$;
- $\varphi_i(qj, a) := \{qa \mapsto 1\}$ for all $q \in Q_i, a \in \Sigma$ such that $\delta_i(q, a)$ is defined and all $1 \leq j \leq r$; $\varphi_i(qa, \mathsf{m}) = \{q'j \mapsto \frac{1}{rank(a)} \mid q' \text{ is the } j\text{th symbol in } \delta(q, a)\}$ for all $q \in Q_i, a \in \Sigma$ such that $\delta_i(q, a)$ is defined; $\varphi_i(qj, \mathsf{m}) := \{d_i \mapsto 1\}$ for all $1 \leq j \leq r$ and all $q \in Q_i$ such that $\delta_i(q, a)$ is not defined for any $a \in \Sigma$; since d_i is a sink state, we also have $\varphi_i(d_i, \mathsf{m}) := \{d_i \mapsto 1\}$;
- $\ell_i(qj) = j$ for all $q \in Q_i$ and $1 \leq j \leq r$; $\ell_i(qa) = a$ for all $q \in Q_i$ and $a \in \Sigma$ such that $\delta_i(q, a)$ is defined; $\ell_i(d_i) = deadlock_i$.

The initial state of $\mathcal{D}_i$ is $q_0^i 1$. Each state q of Q_i corresponds to a set of states qj in $\mathcal{D}_i$ where the number j represents that q is the jth child. Such a state qj is assigned the label j. For each $q \in Q_i$ and $a \in \Sigma$ such that $\delta_i(q, a)$ is defined, we also have a state qa in $\mathcal{D}_i$. Such a state qa is assigned the label a. There is a special sink state d_i for each MDP $\mathcal{D}_i$. The set of actions is the same for all MDPs while each MDP $\mathcal{D}_i$ has a special label $deadlock_i$ which is used to label the sink state d_i. Since the only states in the MDPs that may have multiple actions are those qj states where q is a state in the automata and j is a number, we only need to specify the general strategy upon reaching those states.

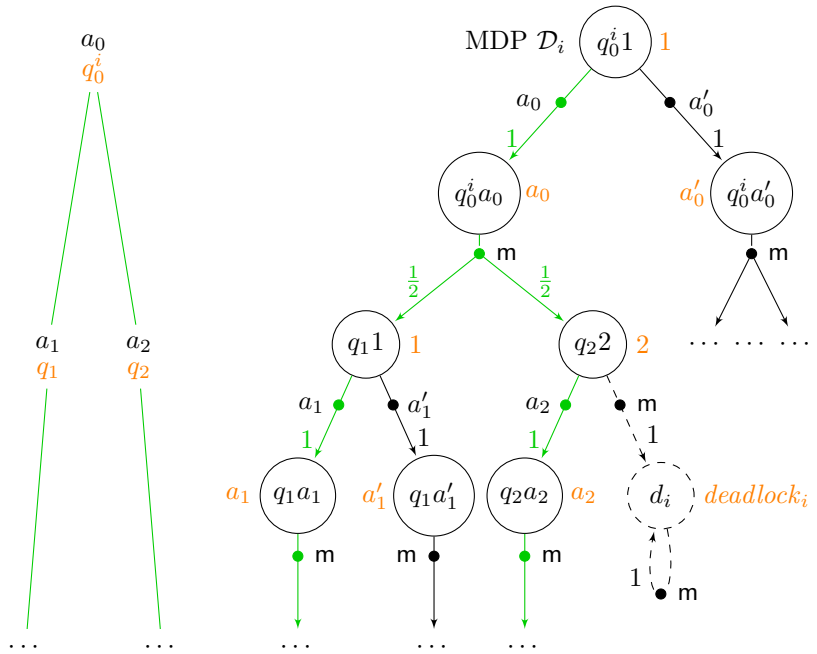

Figure 2 Consider a DTTA $\mathcal{A}_i$ with $rank(a_0) = 2$, $\delta(q_0^i, a_0) = q_1 q_2$ and $\{(q_0^i, a_0'), (q_1, a_1')\} \subseteq$ support(δ). On the right, there is the MDP $\mathcal{D}_i$ corresponding to the DTTA $\mathcal{A}_i$. If $\delta(q_2, a_2)$ is defined in the DTTA $\mathcal{A}_i$, the state $q_2 2$ of $\mathcal{D}_i$ has a single action a_2 taking it to the state $q_2 a_2$. Otherwise, if $\delta(q_2, a_2)$ is undefined in the DTTA $\mathcal{A}_i$, the state $q_2 2$ of $\mathcal{D}_i$ has a single action m taking it to the deadlock state d_i (see the dashed transitions). The labels of the states of $\mathcal{D}_i$ are written next to the states in orange. The left shows a part of a run of the DTTA $\mathcal{A}_i$ on an ordered tree over Σ, that is, a labelling (in orange) of $q \in Q_i$ on the nodes of the tree. This run corresponds to a deterministic general strategy (highlighted in green) of the MDP $\mathcal{D}_i$ on the right.

We show in the appendix that $\bigcap_{i=1}^{k} L(\mathcal{A}_i) \neq \emptyset$ if and only if there is a general strategy α_i for each MDP $\mathcal{D}_i$ such that the k initial states in the induced LMCs are probabilistic bisimilar. See Figure 2 for an illustration.

It remains to replace the k MDPs $\mathcal{D}_1, \ldots, \mathcal{D}_k$ by two MDPs. Specifically, we construct $\mathcal{D}_1' = \langle S_1', Act, L_1, \varphi_1', \ell_1' \rangle$ and $\mathcal{D}_2' = \langle S_2', Act, L_2', \varphi_2', \ell_2' \rangle$ so that the following property holds: there is a general strategy $\mathcal{D}_1, \ldots, D_k$ respectively such that the k initial states in the induced LMCs are probabilistic bisimilar $\iff$ there is a general strategy for $\mathcal{D}_1'$ and $\mathcal{D}_2'$ respectively such that the two initial states in the induced LMCs are probabilistic bisimilar.

We distinguish the two cases: $k \geq 2$ and $k = 1$. When $k = 1$, for any general strategy, the initial state in the induced LMC is trivially probabilistic bisimilar with itself.

If $k \geq 2$, as shown in Figure 3, the initial state s_1 of the MDP $\mathcal{D}_1'$ has a single action m taking it to the initial state $q_0^1 1$ of $\mathcal{D}_1$ with probability one, while the initial state s_2 of the MDP $\mathcal{D}_2'$ has a single action m taking it to the initial states of $\mathcal{D}_2, \ldots, \mathcal{D}_k$ with equal probabilities, that is,

- $S_1' = S_1 \dot\cup \{s_1\}$, $\varphi_1'(s,a) = \begin{cases} \varphi_1(s,a) & \text{if } (s,a) \in \text{support}(\varphi_1) \\ \{q_0^1 1 \mapsto 1\} & \text{if } s = s_1 \text{ and } a = \mathsf{m} \end{cases}$
  and $\ell_1'(s) = \begin{cases} \ell_1(s) & \text{if } s \in S_1 \\ 1 & \text{if } s = s_1; \end{cases}$
- $S_2' = \dot\bigcup_{i \in \{2,\ldots,k\}} S_i \dot\cup \{s_2\}$, $L_2' = \dot\bigcup_{i \in \{2,\ldots,k\}} L_i$,

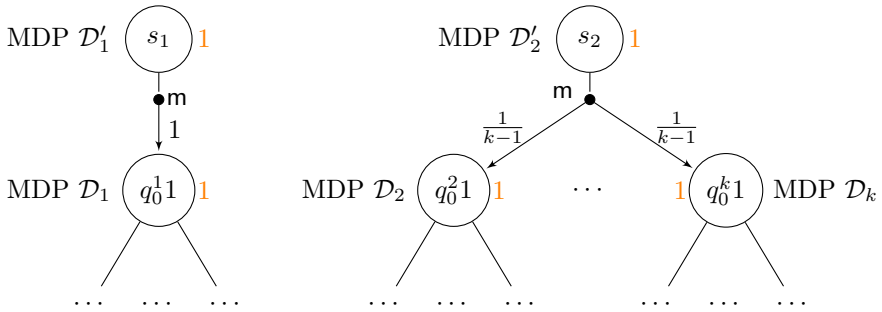

Figure 3 Case $k \geq 2$. Two MDPs $\mathcal{D}'_1 = \langle S'_1, Act, L_1, \varphi'_1, \ell'_1 \rangle$ and $\mathcal{D}'_2 = \langle S'_2, Act, L'_2, \varphi'_2, \ell'_2 \rangle$ are constructed using the k MDPs $\mathcal{D}_1, \ldots, \mathcal{D}_k$.

$$\varphi'_2(s,a) = \begin{cases} \varphi_i(s,a) & \text{if } (s,a) \in \text{support}(\varphi_i) \text{ where } i = 2,\ldots,k \\ \{q_0^i 1 \mapsto \frac{1}{k-1} \mid i = 2,\ldots,k\} & \text{if } s = s_2 \text{ and } a = \mathsf{m} \end{cases}$$

$$\text{and } \ell'_2(s) = \begin{cases} \ell_i(s) & \text{if } s \in S_i \text{ where } i \in \{2,\ldots,k\} \\ 1 & \text{if } s = s_2. \end{cases}$$

Consider the two MDPs $\mathcal{D}'_1$ and $\mathcal{D}'_2$. Assume that there is a general strategy α_i for $\mathcal{D}'_i$ where $i \in \{1,2\}$ such that s_1 and s_2 are probabilistic bisimilar in the induced LMCs. We define a general strategy for each MDP $\mathcal{D}_i$ as follows: the general strategy for D_1 maps each path ρ in $\mathcal{D}_1$ to $\alpha_1(s_1\mathsf{m}\rho)$ and the general strategy for D_i where $i > 1$ maps each path ρ in $\mathcal{D}_i$ to $\alpha_2(s_2\mathsf{m}\rho)$. We have that the k initial states in the induced LMCs are probabilistic bisimilar. For the other direction, assume there is a general strategy α_i for $\mathcal{D}_i$ where $i \in \{1,\ldots,k\}$ such that the k initial states in the induced LMCs are probabilistic bisimilar. Since both s_1 and s_2 only have a single available action m, s_1 and s_2 can be made probabilistic bisimilar by the following general strategies: the strategy for $\mathcal{D}'_1$ maps a path $s_1\mathsf{m}\rho$, where ρ is in $\mathcal{D}_1$, to $\alpha_1(\rho)$ and the strategy for $\mathcal{D}'_2$ maps a path $s_2\mathsf{m}\rho$, where $i > 1$ and ρ is in $\mathcal{D}_i$, to $\alpha_i(\rho)$. ◀

Lemmas 4 and 6 imply the main result of this section:

▶ **Theorem 7.** *The bisimilarity problem is EXPTIME-complete.*

3.3 The Bisimilarity Problem for an LMC and an MDP

We consider the subproblem when one MDP is restricted to be an LMC, that is, given an LMC and an MDP, and two states from the LMC and the MDP respectively, whether there exists a general strategy for the MDP to make these two states probabilistic bisimilar.

In general, memoryless strategies do not suffice for the problem. Consider the example in Figure 4. For the MDP on the left, the general strategy which in state s alternates between the two actions m_1 and m_2 witnesses that s and t are probabilistic bisimilar in the induced LMC. However, no memoryless strategy can make s and t probabilistic bisimilar.

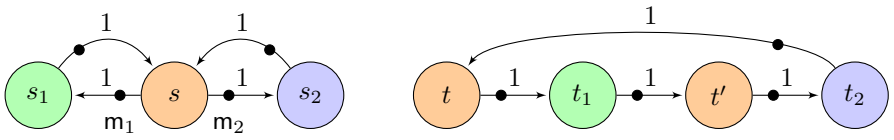

Figure 4 In this MDP no memoryless strategy witnesses $s \sim t$.

We show that this problem is NP-complete.

▶ **Lemma 8.** *The bimilarity problem for an LMC and an MDP is in NP.*

Proof. Given an LMC $\mathcal{M} = \langle S_1, L, \tau, \ell_1 \rangle$, an MDP $\mathcal{D} = \langle S_2, Act, L, \varphi, \ell_2 \rangle$, and two states $s_1 \in S_1$ and $s_2 \in S_2$, we decide whether there is a general strategy α for $\mathcal{D}$ such that s_1 and s_2 are probabilistic bisimilar in the LMC $\mathcal{M} \oplus \mathcal{D}_\alpha$.

Consider the attacker-defender game defined in terms of the MDP $\mathcal{M} \oplus \mathcal{D}$ and the set $\{s_1, s_2\}$. According to Lemma 2 and Proposition 3, it suffices to check whether Defender has a memoryless winning strategy for the attacker-defender game.

Without loss of generality, assume that the LMC $\mathcal{M}$ is a quotient LMC, that is, no two states in S_1 are probabilistic bisimilar, and all states in S_1 are reachable from s_1. Each state $s \in S_1$ corresponds to a game state and we have $|S_1|$ game states. We guess the following components of a winning strategy for Defender:

- For each state $s \in S_1$, guess a set of states $E_s \subseteq S_2$. Intuitively, Defender claims that there is a general strategy such that all states in E_s are probabilistic bisimilar with s. Let S_w be the set of all the E_s sets. Let $E'_s = \{s\} \cup E_s$ be the game state which corresponds to the state s. The state s_2 is in E_{s_1}, and is also in E'_{s_1}.
- For each $E \in S_w$, guess a function $f_E : E \times Act \times S_2 \to S_w$ with $v \in f_E(u, \mathsf{m}, v)$ for all $(u, \mathsf{m}, v) \in E \times Act \times S_2$.

In the game state E'_s where $s \in S_1$, the probability distribution υ over the successor game states is determined by the probability transition function of the LMC $\mathcal{M}$, that is, $\upsilon(E_t) = \tau(s)(t)$ for all $t \in S_1$ where $E_t \in S_w$ is the set of states which Defender claims can be made probabilistic bisimilar with t.

For each $E_s \in S_w$ and each $u \in E_s$, a memoryless strategy $\alpha^u_s \in \mathrm{Distr}(Act(u))$ can be characterised by numbers $x_{s,u,\mathsf{m}}$ where $\mathsf{m} \in Act(u)$ such that $x_{s,u,\mathsf{m}} = \alpha^u_s(\mathsf{m})$. We write $\bar{x}$ for the collection $(x_{s,u,\mathsf{m}})_{s \in S_1, u \in E_s, \mathsf{m} \in Act(u)}$. Checking whether there is a memoryless winning strategy for Defender amounts to a feasibility test of the following linear program: $\exists \bar{x}$ such that

- $x_{s,u,\mathsf{m}} \geq 0$ for all $s \in S_1, u \in E_s, \mathsf{m} \in Act(u)$;
- $\sum_{\mathsf{m} \in Act(u)} x_{s,u,\mathsf{m}} = 1$ for all $s \in S_1, u \in E_s$;
- $\tau(s)(t) = \sum_{\mathsf{m} \in Act(u)} \sum_{v \in S_2 \text{ s.t. } f_{E_s}(u,\mathsf{m},v) = E_t} x_{s,u,\mathsf{m}} \varphi(u, \mathsf{m})(v)$ for all $s, t \in S_1$ and $u \in E_s$.

Hence, this can be decided in polynomial time. ◀

NP-hardness follows from a reduction from the Subset Sum problem. The reduction is similar to [11, Theorem 19]. Given a set $P = \{p_1, \ldots, p_n\}$ where $P \subseteq \mathbb{N}$ and $N \in \mathbb{N}$, Subset Sum asks whether there exists a set $Q \subseteq P$ such that $\sum_{p_i \in Q} p_i = N$. Subset Sum is known to be NP-complete [2].

▶ **Lemma 9.** *The bimilarity problem for an LMC and an MDP is NP-hard.*

By combining Lemma 8 and Lemma 9 we get:

▶ **Theorem 10.** *The bimilarity problem for an LMC and an MDP is NP-complete.*

4 Bisimilarity Inequivalence Problem

In this section, we consider the bisimilarity inequivalence problem which, given an MDP and two initial states, asks whether there is a general strategy such that in the induced LMC the two states are not probabilistic bisimilar.

▶ **Example 11.** Consider again the 2-threaded program and MDP from Example 1 and Figure 1. This time we take the view of an eavesdropper. The memoryless strategy that, in h_0 and h_1, chooses action m_1 with probability 0.4 and action m_2 with probability 0.6 induces an LMC in which h_0 and h_1 are not probabilistic bisimilar. Thus, if the eavesdropper has control over the scheduling and chooses a suitable strategy, they can glean information about the value of h. In this sense, the bisimilarity inequivalence problem can be viewed as the problem whether a program is safe from such information leaks, even if an adversary controls the scheduler.

▶ **Example 12.** In general, a single memoryless strategy does not suffice for the bisimilarity inequivalence problem. Consider the MDP in Figure 5. The general strategy which in state

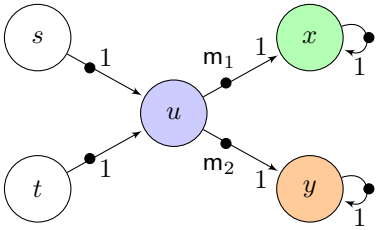

Figure 5 In this MDP no memoryless strategy witnesses $s \not\sim t$.

u selects the action m_1 after seeing the path su and selects the action m_2 after seeing the path tu witnesses that s and t are not probabilistic bisimilar in the induced LMC. However, given any memoryless strategy, s and t are probabilistic bisimilar in the induced LMC.

We write $\rho[\alpha]$ for the state ρ in the LMC $\mathcal{D}_\alpha$ induced by a strategy α. In the following, we show that if there is a general strategy such that s and t are not probabilistic bisimilar in the induced LMC, there are memoryless strategies σ and τ such that $s[\sigma]$ and $t[\tau]$[2] are not probabilistic bisimilar. These two memoryless strategies are not necessarily the same and they can be combined to form a general witnessing strategy for the bisimilarity inequivalence problem. Take the MDP in Figure 5 as an example. Although no single memoryless strategy witnesses the probabilistic bisimilarity inequivalence of s and t, with the memoryless strategies σ and τ where $\sigma(u) = \mathsf{m}_1$ and $\tau(u) = \mathsf{m}_2$, we have that $s[\sigma]$ and $t[\tau]$ are not probabilistic bisimilar.

A partition of the states S is a set X consisting of pairwise disjoint subsets E of S with $\bigcup_{E\in X} = S$. Recall that $\varphi(s,\mathsf{m})(s')$ is the transition probability from s to s' when choosing action m. Similarly, $\varphi(s,\mathsf{m})(E) = \sum_{s'\in E} \varphi(s,\mathsf{m})(s')$ is the transition probability from $s \in S$ to $E \subseteq S$ when choosing action m. We write $\varphi(s,\mathsf{m})(X)$ to denote the probability distribution $(\varphi(s,\mathsf{m})(E))_{E\in X}$. We define $\varphi(s)(X) = \{\varphi(s,\mathsf{m})(X) : \mathsf{m} \in Act(s)\}$, which is a set of probabilistic distributions over the partition X when choosing all available actions of s.

Abusing the notation slightly, for a memoryless strategy α we write $\alpha(s)(s')$ for the transition probability from s to s' in the LMC induced by α, that is, $\alpha(s)(s') = \sum_{\mathsf{m}\in Act(s)} \alpha(s)(\mathsf{m})\varphi(s,\mathsf{m})(s')$. Similarly, the transition probability from s to $E \subseteq S$ in the LMC induced by a memoryless strategy is $\alpha(s)(E) = \sum_{s'\in E} \alpha(s)(s')$ and the probability distribution on a partition X is $\alpha(s)(X) = \big(\alpha(s)(E)\big)_{E\in X}$.

The assumption that for two states s,t we have $s[\sigma] \sim t[\tau]$ for all general strategies σ and τ has consequences on s, on t, and on their successors, as detailed in the following lemma.

[2] Here s and t are paths of length 1.

▶ **Lemma 13.** *Let $s, t \in S$ with $s[\sigma] \sim t[\tau]$ for all general strategies σ and τ.*

1. *For all general strategies σ and τ, we have $s[\sigma] \sim s[\tau]$ and $t[\sigma] \sim t[\tau]$;*
2. *For all successors u of s and all general strategies σ and τ, we have $u[\sigma] \sim u[\tau]$.*

Given an MDP $\mathcal{D} = \langle S, Act, L, \varphi, \ell \rangle$, define a *superbisimulation* relation to be any equivalence relation $R \subseteq S \times S$ such that $(s,t) \in R$ if and only if $\ell(s) = \ell(t)$ and $\alpha(s)(X) = \alpha(t)(X)$ for all memoryless strategies α where $X = S/_R$. The union of superbisimulations is a superbisimulation. Let superbisimilarity $\approx_{\mathcal{D}}$ be the largest superbisimulation, i.e., the union of all superbisimulations. The subscript $\mathcal{D}$ can be omitted if it is clear from the context. We write $s \approx t$ if $(s,t) \in \approx$.

Let $\bar{S} := S \times \{0,1\}$. We define an MDP $\bar{\mathcal{D}} = (\bar{S}, Act, L, \bar{\varphi}, \bar{\ell})$ where $\bar{\varphi}\big((s,i)\big)(\mathsf{m})\big((t,i)\big) = \varphi(s,\mathsf{m})(t)$ for all $s, t \in S$, $i \in \{0,1\}$ and $\mathsf{m} \in Act$ and $\bar{\ell}\big((s,i)\big) = \ell(s)$ for all $(s,i) \in \bar{S}$. The MDP $\bar{\mathcal{D}}$ is basically made up of two disjoint copies of the original MDP $\mathcal{D}$.

The following lemma is a counterpart to Lemma 13. It spells out consequences of the assumption that $(s,0)$ and $(t,1)$ are superbisimilar in $\bar{\mathcal{D}}$.

▶ **Lemma 14.** *Let $s, t \in S$ with $(s,0) \approx_{\bar{\mathcal{D}}} (t,1)$.*

1. *Let R be any superbisimulation that contains $\big((s,0),(t,1)\big)$. For any successor $(u,0)$ of $(s,0)$, there exists a successor $(v,1)$ of $(t,1)$ such that $\big((u,0),(v,1)\big) \in R$. Similarly, for any successor $(v,1)$ of $(t,1)$, there exists a successor $(u,0)$ of $(s,0)$ such that $\big((u,0),(v,1)\big) \in R$. In other words, any successor of $(s,0)$ is superbisimilar with some successor of $(t,1)$ and vice versa.*
2. *We have $(s,1) \approx_{\bar{\mathcal{D}}} (t,0)$, $(s,0) \approx_{\bar{\mathcal{D}}} (s,1)$ and $(t,0) \approx_{\bar{\mathcal{D}}} (t,1)$.*

The following theorem, whose proof is based on Lemmas 13 and 14, is the main technical result of this section. It provides a superbimilarity-based characterisation of s and t being bisimilar under all general strategies.

▶ **Theorem 15.** *For all $s, t \in S$, we have $(s,0) \approx (t,1) \iff \forall$ general strategies $\sigma, \tau : s[\sigma] \sim t[\tau]$.*

Proof. ($\Longleftarrow$) Let $S' = \{s \in S \mid \forall \text{ general strategies } \sigma, \tau : s[\sigma] \sim s[\tau]\}$ and $\bar{S}' = \{(s,i) \in \bar{S} \mid s \in S'\}$.

Let $R := \{\big((s,i),(t,j)\big) \in \bar{S} \times \bar{S}\big) \mid \forall \text{ general strategies } \sigma, \tau : s[\sigma] \sim t[\tau]\}$.

Firstly, R is an equivalence relation on $\bar{S}'$:

- $R \subseteq \bar{S}' \times \bar{S}'$: Assume for all general strategies $\sigma, \tau : s[\sigma] \sim t[\tau]$. By Lemma 13, we have that $s[\sigma] \sim s[\tau]$ and $t[\sigma] \sim t[\tau]$ for all general strategies σ and τ. Both s and t are in S', hence, $(s,i),(t,i) \in \bar{S}'$ for all $i \in \{0,1\}$.
- R is reflexive. (trivial)
- R is symmetric. (trivial)
- R is transitive. (trivial)

By Lemma 13, all successors of a state $s \in S'$ in the MDP $\mathcal{D}$ are in S', hence, all successors of a state $(s,i) \in \bar{S}'$ are in $\bar{S}'$. Let $\bar{\mathcal{D}}'$ be the sub-MDP of $\bar{\mathcal{D}}$ that contains $\bar{S}'$ and all the transitions between $\bar{S}'$.

To show that $(s,i) \approx (t,j)$ for any $\big((s,i),(t,j)\big) \in R$, we show that R is a superbisimulation of $\bar{\mathcal{D}}'$. The details can be found in the appendix.

($\Longrightarrow$) Let $S' = \{s \mid \exists (t,1) \text{ such that } (s,0) \approx (t,1)\}$. Define a relation $R = \{(s,t) \mid (s,0) \approx (t,1)\}$ on S'. By Lemma 14, R is an equivalence relation. Let $X = S'/_R$.

By Item 1 of Lemma 14, all successors of a state in S' are in S'. Let $\mathcal{D}'$ be the sub-MDP that contains S' and all the transitions between S'. Let σ, τ be two arbitrary general strategies of $\mathcal{D}'$.

Let $R' = \{(\rho_1[\mu_1], \rho_2[\mu_2]) \mid \rho_1, \rho_2$ are paths in $\mathcal{D}', \big(\mathsf{last}(\rho_1), \mathsf{last}(\rho_2)\big) \in R$ and $\mu_1, \mu_2 \in \{\sigma, \tau\}\}$ be a relation on the states of the LMC $\mathcal{D}'_\sigma \oplus \mathcal{D}'_\tau$, the disjoint union of the induced LMCs $\mathcal{D}'_\sigma$ and $\mathcal{D}'_\tau$.

Since R is an equivalence relation, it is not hard to see that R' is also an equivalence relation. We show in the appendix that R' is a probabilistic bisimulation on the states of the LMC $\mathcal{D}'_\sigma \oplus \mathcal{D}'_\tau$.

For any $(s, 0) \approx (t, 1)$, we have $(s[\sigma], t[\tau]) \in R'$, and hence, $s[\sigma]$ and $t[\tau]$ are probabilistic bisimilar in the LMC $\mathcal{D}'_\sigma \oplus \mathcal{D}'_\tau$. Since σ and τ are arbitrary general strategies for $\mathcal{D}'$ and the sub-MDP $\mathcal{D}'$ has all the available actions and successors of S' from $\mathcal{D}$, we have $s[\sigma] \sim t[\tau]$ for all general strategies σ and τ for $\mathcal{D}$. ◀

By Theorem 15, to decide whether there exist general strategies σ and τ such that $s[\sigma] \not\sim t[\tau]$, it suffices to decide whether $(s, 0) \not\approx (t, 1)$, which can be done by running [11, Algorithm 2] on the MDP $\bar{\mathcal{D}}$. This partition refinement algorithm is polynomial-time and the relation computed is superbisimilarity. By [11, Theorem 12, Corollary 13], if two states s and t are not superbisimilar, we can compute in polynomial time a memoryless strategy that witnesses $s \not\sim t$. Since the two states $(s, 0)$ and $(t, 1)$ are from two disjoint MDPs, if $(s, 0) \not\approx (t, 1)$, we can also compute in polynomial time two memoryless strategies σ and τ that witness $(s, 0)[\sigma] \not\sim (t, 1)[\tau]$, equivalently $(s)[\sigma] \not\sim (t)[\tau]$. Hence we have proved the following theorem.

▶ **Theorem 16.** *The bisimilarity inequivalence problem is in* P*. For any positive instance of the problem, there are memoryless strategies σ and τ such that $s[\sigma] \not\sim t[\tau]$. Further, for any positive instance of the problem, we can compute in polynomial time memoryless strategies σ and τ that witness $s[\sigma] \not\sim t[\tau]$.*

5 Conclusion

In this paper we have settled the decidability and complexity of the bisimilarity equivalence and inequivalence problems of MDPs under general strategies. Let us review the key technical steps.

We have proved that bisimilarity equivalence is decidable, albeit with a high, EXPTIME, computational complexity. For the EXPTIME upper bound we have provided a reduction to a non-stochastic two-player game, the attacker-defender game, which can be decided in EXPTIME. For the EXPTIME lower bound we have provided a reduction from the intersection emptiness problem for deterministic tree automata, which we have shown to be EXPTIME-hard. Further, we have obtained NP-completeness for the case that one of the MDPs is a Markov chain.

We have also shown that the bisimilarity inequivalence problem has much lower computational complexity, as it can be decided in polynomial time. This extends an earlier result that the corresponding inequivalence problem for memoryless strategies is in P. The key novel technique we have developed here is the notion of superbisimilarity, whose definition is similar to bisimilarity but with a different quantification over strategies.

References

1 Christel Baier. Polynomial time algorithms for testing probabilistic bisimulation and simulation. In Rajeev Alur and Thomas A. Henzinger, editors, *Computer Aided Verification*, pages 50–61, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

2 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

3 Benoît Delahaye. Consistency for parametric interval Markov chains. In Étienne André and Goran Frehse, editors, *2nd International Workshop on Synthesis of Complex Parameters, SynCoP 2015, April 11, 2015, London, United Kingdom*, volume 44 of *OASICS*, pages 17–32. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015.

4 Salem Derisavi, Holger Hermanns, and William H. Sanders. Optimal state-space lumping in Markov chains. *Inf. Process. Lett.*, 87(6):309–315, 2003.

5 L. Doyen, T.A. Henzinger, and J.-F. Raskin. Equivalence of labeled Markov chains. *International Journal on Foundations of Computer Science*, 19(3):549–563, 2008.

6 Nathanaël Fijalkow, Stefan Kiefer, and Mahsa Shirmohammadi. Trace refinement in labelled Markov decision processes. *Logical Methods in Computer Science*, 16(2), 2020.

7 Ernst Moritz Hahn, Holger Hermanns, and Lijun Zhang. Probabilistic reachability for parametric Markov models. *Int. J. Softw. Tools Technol. Transf.*, 13(1):3–19, 2011.

8 Christian Hensel, Sebastian Junges, Joost-Pieter Katoen, Tim Quatmann, and Matthias Volk. The probabilistic model checker Storm, 2020. `arXiv:arXiv:2002.07080`.

9 Bengt Jonsson and Kim Guldstrand Larsen. Specification and refinement of probabilistic processes. In *Proceedings of the Sixth Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 15-18, 1991*, pages 266–277. IEEE Computer Society, 1991.

10 John G. Kemeny and J. Laurie Snell. *Finite Markov Chains*. Van Nostrand, 1960.

11 Stefan Kiefer and Qiyi Tang. Comparing labelled Markov decision processes. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, volume 182 of *LIPIcs*, pages 49:1–49:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.FSTTCS.2020.49`.

12 Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 254–266. IEEE Computer Society, 1977. `doi:10.1109/SFCS.1977.16`.

13 Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

14 Kim Guldstrand Larsen and Arne Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991.

15 Azaria Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.

16 Andrei Popescu, Johannes Hölzl, and Tobias Nipkow. Formalizing probabilistic noninterference. In Georges Gonthier and Michael Norrish, editors, *Certified Programs and Proofs – Third International Conference*, volume 8307 of *Lecture Notes in Computer Science*, pages 259–275. Springer, 2013. `doi:10.1007/978-3-319-03545-1_17`.

17 Andrei Sabelfeld and David Sands. Probabilistic noninterference for multi-threaded programs. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop*, pages 200–214. IEEE Computer Society, 2000. `doi:10.1109/CSFW.2000.856937`.

18 Marcel-Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4:245–270, 1961.

19 Helmut Seidl. Haskell overloading is DEXPTIME-complete. *Inf. Process. Lett.*, 52(2):57–60, 1994. `doi:10.1016/0020-0190(94)00130-8`.

20 Geoffrey Smith. Probabilistic noninterference through weak probabilistic bisimulation. In *16th IEEE Computer Security Foundations Workshop (CSFW-16 2003)*, pages 3–13. IEEE Computer Society, 2003. `doi:10.1109/CSFW.2003.1212701`.

21 Wen-Guey Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM Journal on Computing*, 21(2):216–227, 1992.

22 Antti Valmari and Giuliana Franceschinis. Simple $O(m \log n)$ time Markov chain lumping. In Javier Esparza and Rupak Majumdar, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6015 of *Lecture Notes in Computer Science*, pages 38–52. Springer, 2010.

23 Tobias Winkler, Sebastian Junges, Guillermo A. Pérez, and Joost-Pieter Katoen. On the complexity of reachability in parametric markov decision processes. In Wan J. Fokkink and Rob van Glabbeek, editors, *30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands*, volume 140 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CONCUR.2019.14.

A Proofs of Section 3

▶ **Proposition 3.** *Given an MDP $\mathcal{D} = \langle S, Act, L, \varphi, \ell\rangle$ and a set $E_1 \subseteq S$, Defender has a winning strategy for the attacker-defender game if and only if there exists a general strategy α for $\mathcal{D}$ such that in the LMC induced by α all states in E_1 are probabilistic bisimilar.*

Proof. ($\Longrightarrow$) We show in the following that all states in E_1 are probabilistic bisimilar in the LMC induced by α defined in the main text.

Given the MDP $\mathcal{D}$ and the general strategy α, an LMC $\mathcal{D}(\alpha) = \langle \mathcal{P}, L, \tau_\alpha, \ell_\alpha\rangle$ is induced. To show that all states in E_1 are probabilistic bisimilar in the LMC $\mathcal{D}(\alpha)$, we show that for any two states ρ_1 and ρ_2 of $\mathcal{D}(\alpha)$, if $\rho_1, \rho_2 \in \mathcal{P}$ and $W(\rho_1) = W(\rho_2)$, we have $\rho_1 \sim \rho_2$. It suffices to show the relation $\sim_W \subseteq \mathcal{P} \times \mathcal{P}$ defined by $\rho_1 \sim_W \rho_2$ if and only if $W(\rho_1) = W(\rho_2)$ is a probabilistic bisimulation.

Let $\rho_1, \rho_2 \in \mathcal{P}$ be two states in the LMC $\mathcal{D}(\alpha)$ and $E = W(\rho_1) = W(\rho_2)$ be a state at the beginning of some round of the attacker-defender game. Let $s_1 = \mathsf{last}(\rho_1)$ and $s_2 = \mathsf{last}(\rho_2)$. We have $\ell_\alpha(\rho_1) = \ell(s_1) = \ell(s_2) = \ell_\alpha(\rho_2)$ since $s_1, s_2 \in E$ and all states in E have the same label. Defender has a winning strategy for E and we assume that she chooses a set $S' \subseteq 2^S$, a distribution υ on S', and a function f. Consider a set $E' \in \mathrm{support}(\upsilon)$. By definition of $\sim_W$, for all $\rho_1\mathsf{m}_1 t_1, \rho_2\mathsf{m}_2 t_2 \in \mathcal{P}$ such that $W(\rho_1\mathsf{m}_1 t_1) = W(\rho_2\mathsf{m}_2 t_2) = E'$ we have $\rho_1\mathsf{m}_1 t_1 \sim_W \rho_2\mathsf{m}_2 t_2$. We also have

$$
\begin{aligned}
& \upsilon(E') \\
= & \sum_{\mathsf{m}_1 \in Act(s_1)} \sum_{t_1 \in S \text{ s.t. } f(s_1,\mathsf{m}_1,t_1)=E'} \alpha^{s_1}(\mathsf{m}_1)\varphi(s_1,\mathsf{m}_1)(t_1) \\
& \qquad \text{[property satisfied by } \upsilon\text{]} \\
= & \sum_{\mathsf{m}_1 \in Act(s_1)} \sum_{t_1 \in E'} \alpha(\rho_1)(\mathsf{m}_1)\varphi(s_1,\mathsf{m}_1)(t_1) \\
= & \sum_{\mathsf{m}_1 \in Act(s_1)} \sum_{t_1 \in E'} \tau_\alpha(\rho_1)(\rho_1\mathsf{m}_1 t_1). \qquad \text{[definition of } \tau_\alpha\text{]} \\
= & \sum_{\rho_1\mathsf{m}_1 t_1 \in \mathcal{P}_{i+1} \text{ and } t_1 \in E'} \tau_\alpha(\rho_1)(\rho_1\mathsf{m}_1 t_1).
\end{aligned}
$$

Similarly, we have $\upsilon(E') = \sum_{\rho_2\mathsf{m}_2 t_2 \in \mathcal{P}_{i+1} \text{ and } t_2 \in E'} \tau_\alpha(\rho_2)(\rho_2\mathsf{m}_2 t_2)$.

Thus, the relation $\sim_W$ is a probabilistic bisimulation. Consider any state $s \in \mathcal{P}_1$ of $\mathcal{D}(\alpha)$. We have $W(s) = E_1$, it concludes that all states in E_1 are probabilistic bisimilar.

($\Longleftarrow$) We verify that the objects chosen by Defender satisfies the required properties. For all $s \in E$ and all $E'_i \in S'$ where $1 \leq i \leq k$ we have

$$\begin{aligned}
& \upsilon(E'_i) \\
= \; & \upsilon'(B_i) \\
= \; & \sum_{s\mathsf{m}t \in B_i} \alpha_E(s)(\mathsf{m})\varphi(s,\mathsf{m})(t) \\
= \; & \sum_{s\mathsf{m}t \in B_i} \alpha^s(\mathsf{m})\varphi(s,\mathsf{m})(t) \\
= \; & \sum_{\mathsf{m} \in Act(s)} \; \sum_{t \in S \text{ s.t. } f(s,\mathsf{m},t)=E'_i} \alpha^s(\mathsf{m})\varphi(s,\mathsf{m})(t)\,.
\end{aligned}$$

If the game starts with an equalisable set and all the future game states are equalisable sets, Defender can always choose objects with required properties and hence win the game. It remains to show that all sets in S' are equalisable. Consider the probabilistic bisimulation class B_i which is used to construct $E'_i \in S'$. We define a general strategy α_i for any path that starts with a state $t \in E'_i$. Let ρ be such a path. We define $\alpha_i(\rho) := \alpha_E(s\mathsf{m}\rho)$ where $s \in E$ and $s\mathsf{m}t \in B_i$. Basically, after going along the path ρ, the general strategy α_i plays as $\alpha_E(s\mathsf{m}\rho)$. The LMC induced by the general strategy α_i can be seen as part of the LMC induced by α_E where a state ρ in the former LMC corresponds to the state $s\mathsf{m}\rho$ in the latter. That is, a state $t \in E'_i$ in the LMC induced by α_i corresponds to a state $s\mathsf{m}t \in B_i$ in the LMC induced by α_E. Since all states in B_i are probabilistic bisimilar in the LMC induced by α_E, all states in E'_i in the LMC induced by α_i are also probabilistic bisimilar. ◀

▶ **Lemma 5.** *The intersection nonemptiness problem is EXPTIME-hard.*

Proof. We give a polynomial-time reduction from the problem of acceptance of a word by a PSPACE-bounded alternating Turing machine. Let $M = (P_\exists, P_\forall, \Gamma, \Delta, p_0, p_{acc}, p_{rej})$ be a PSPACE-bounded alternating Turing machine, where $P = P_\exists \cup P_\forall$ is the finite set of (control) states partitioned into existential states $P_\exists$ and universal states $P_\forall$, and Γ is the tape alphabet, and $\Delta \subseteq P \times \Gamma \times P \times \Gamma \times \{-1, +1\}$ is the transition relation, and $p_0, p_{acc}, p_{rej} \in P$ are the initial, accepting, rejecting state, respectively. A transition $(p, a, p', a', d) \in \Delta$ means that if M is in state p and its read-write head reads letter a, then $\mathcal{M}$ rewrites the contents of the current cell with the letter a', moves the head in direction d (either left if $d = -1$, or right if $d = +1$), and changes its state to p'. Such a transition is called *outgoing* from (p, a). We assume that for all $(p, a) \in P \times \Gamma$ there is at least one outgoing transition, and for all $(p, a) \in P_\forall \times \Gamma$ there are exactly two outgoing transitions. A *configuration* of M is given by the current state $p \in P$, the tape content (from Γ^*), and the position of the head. If the current state p is existential, i.e., $p \in P_\exists$, and the head reads $a \in \Gamma$, then the *existential* player picks a transition that is outgoing from (p, a). Similarly for the universal states and the *universal* player. Starting from an input word $w \in \Gamma^*$ on the tape, strategies of the two players define a *computation*, i.e., a sequence of configurations. It is the goal of the existential player to form a computation that reaches p_{acc}; the goal of the universal player is to avoid this. We can assume that all computations reach either p_{acc} or p_{rej} and no configuration is repeated before that (this is achieved, e.g., using a counter on the tape), and after reaching p_{acc} or p_{rej} the control state no longer changes. If the existential player has a strategy to reach p_{acc} no matter what strategy the universal player uses, then we say that M *accepts* w. Since APSPACE = EXPTIME, there is a (fixed) PSPACE-bounded alternating Turing machine, say $M = (P_\exists, P_\forall, \Gamma, \Delta, p_0, p_{acc}, p_{rej})$, such that it is EXPTIME-complete to decide if it accepts a given input word.

Let $w \in \Gamma^n$ be the input word. The Turing machine M uses only space polynomial in n, say N. We construct, in polynomial time, DTTAs $\mathcal{A}_0, \ldots, \mathcal{A}_N$ such that $\bigcap_{i=0}^{N} L(\mathcal{A}_i) \neq \emptyset$ if and only if M accepts w. To do so, we encode a strategy of the existential player by a tree whose branches encode the computations that are consistent with the existential player's strategy; the strategy of the universal player effectively chooses one branch in the tree, which encodes the computation defined by both players' strategies. We want to construct DTTAs $\mathcal{A}_0, \ldots, \mathcal{A}_N$ so that $\bigcap_{i=0}^{N} L(\mathcal{A}_i)$ contains exactly those trees that encode a winning strategy of the existential player. Each $\mathcal{A}_i$ ensures some aspect of correctness of such strategy trees; when $\mathcal{A}_i$ encounters a problem with a tree, it uses the partiality of its transition function δ so that it does not have a run on that tree.

The trees consist of blocks of the form $a_1 \cdots a_{m-1} \# a_m \cdots a_N p$, which encode a configuration. Here, $a_1 \cdots a_N \in \Gamma^N$ is the tape content, the position of the symbol # indicates the position of the head (reading a_m), and $p \in P$ is the current control state. As ranked alphabet we take $\Sigma = \Gamma \cup \{\#\} \cup P$ (we can assume this is a union of disjoint sets), where all symbols have rank 1, except those in $P_\forall$, which have rank 2. As a result, a p-node at the end of a block, where $p \in P_\exists$, has one child, which starts another block encoding a successor configuration. Similarly, if $p \in P_\forall$, then the node has two children, both of which start another block encoding successor configurations.

We construct, in polynomial time, DTTA $\mathcal{A}_0$ so that it ensures that the input tree starts with $\# w_1 \cdots w_n \sqcup \cdots \sqcup p_0$, encoding the initial configuration of M; here $w = w_1 \cdots w_n$ is the input word, which is followed by $N - n$ blank symbols '$\sqcup$'. DTTA $\mathcal{A}_0$ also ensures that p_{rej} occurs *nowhere* in the tree. It also ensures that the blocks, which encode configurations as described above, are well-formed in that each of them consists of N symbols from Γ, a single occurrence of # in front of one of the symbols from Γ, and a $p \in P$ at the end.

Let $i \in \{1, \ldots, N\}$. We construct, in polynomial time, DTTA $\mathcal{A}_i$ so that it ensures the following properties for all blocks in the input tree.

- If the symbol # precedes the ith symbol from Γ in the block, say a, and the block ends with $p \in P_\exists$ and, in the directly following block, the symbol # precedes the $(i+d)$th symbol ($i \in \mathbb{N}$) from Γ in the block and the ith symbol from Γ in the block is a' and the block ends with $p' \in \Gamma$, then $(p, a, p', a', d) \in \Delta$.
- If the symbol # precedes the ith symbol from Γ in the block, say a, and the block ends with $p \in P_\forall$ and $(p, a, p_1, a_1, d_1), (p, a, p_2, a_2, d_2) \in \Delta$ are the two outgoing transitions from (p, a) (we assume that these two transitions are ordered in some way), then in the left (respectively, right) successor block the symbol # precedes the $(i+d_1)$th (respectively, $(i+d_2)$th) symbol from Γ and the ith symbol from Γ in the block is a_1 (respectively, a_2) and the block ends with p_1 (respectively, p_2).
- If the symbol # does not precede the ith symbol from Γ in the block, say a, then the ith symbol from Γ in the (either one or two) directly following block(s) is also a.

In this way, $\bigcap_{i=0}^{N} L(\mathcal{A}_i)$ contains exactly those trees that encode a winning strategy of the existential player. ◀

▶ **Lemma 6.** *The bisimilarity problem is EXPTIME-hard.*

Proof. We show in the following that $\bigcap_{i=1}^{k} L(\mathcal{A}_i) \neq \emptyset$ if and only if there is a general strategy α_i for each MDP $\mathcal{D}_i$ such that the k initial states in the induced LMCs are probabilistic bisimilar.

($\implies$) Assume $\bigcap_{i=1}^{k} L(\mathcal{A}_i) \neq \emptyset$. Let t be an ordered tree over Σ in the intersection $\bigcap_{i=1}^{k} L(\mathcal{A}_i)$. There is a deterministic general strategy α_i for each MDP $\mathcal{D}_i$ corresponding to t. An example of a deterministic general strategy of an MDP corresponding to a run tree is given in Figure 2.

We define a function f_i which maps a path starting from the root of the run tree of $\mathcal{A}_i$ (t labelled with states of $\mathcal{A}_i$) to a path in the MDP $\mathcal{D}_i$:

- $f_i(a) = q_0^i 1$ where the root of the tree t is an a-node (the root of t is labelled with $q_0^i 1$);
- $f_i(xa) = f_i(x)\mathsf{last}(x)(q\mathsf{last}(x))\mathsf{m}(q'j)$ where $x \in \Sigma^+, a \in \Sigma$, $q \in Q_i$, l is a number and $ql = \mathsf{last}(f_i(x))$, a is the jth child of its parent and is labelled with $q' \in Q_i$.

For a path $x \in \Sigma^+$ of t, $f_i(x)$ is a path in $\mathcal{D}_i$ which ends with a state of the form qj where $q \in Q_i$ and j is a number. We now define the deterministic general strategy α_i for the MDP $\mathcal{D}_i$. For a path ρ in the MDP $\mathcal{D}_i$, we have

$$\alpha_i(\rho) = \begin{cases} a & \text{if } f_i(x) = \rho \text{ for a path } x \in \Sigma^+ \text{ of the tree } t \text{ and } a = \mathsf{last}(x) \\ \mathsf{m} & \text{if } \mathsf{last}(\rho) \text{ is of the form } q_i a \text{ where } q_i \in Q_i \text{ and } a \in \Sigma \\ \mathsf{m}' & \text{otherwise, } \mathsf{m}' \text{ is an arbitrary available action of } \mathsf{last}(\rho) \end{cases}$$

Every state in the induced LMC $\mathcal{D}_i(\alpha_i)$ corresponds to a tree path. The states ρ in the induced LMC $\mathcal{D}_i(\alpha_i)$, where $\mathsf{last}(\rho)$ is of the form $q_i j$ such that $q_i \in Q_i$ and j is a number, correspond to the tree path x such that $f_i(x) = \rho$. The states $\rho a(q_i a)$ where $q_i \in Q_i$ and $a \in \Sigma$ in the induced LMC $\mathcal{D}_i(\alpha_i)$ correspond to the tree path x such that $f_i(x) = \rho$. Let P be the union of the states of the induced LMCs. Let $R \subseteq P \times P$ be a relation in which $(\rho, \rho') \in R$ if and only if either (1) there exist a tree path $x \in \Sigma^+$ and $i, j \in \{1, \ldots, k\}$ such that $f_i(x) = \rho$ and $f_j(x) = \rho'$ or (2) both states correspond to the same tree path and $\mathsf{last}(\rho)$ (respectively, $\mathsf{last}(\rho')$) is of the form qa where $q \in Q_i$ for some i and $a \in \Sigma$. In case (1), we have the pairs of states ρ and ρ' in the induced LMCs where $\mathsf{last}(\rho)$ (respectively, $\mathsf{last}(\rho')$) is of the form qj where $q \in Q_i$ for some i and j is a number. We can show that the relation R is a probabilistic bisimulation. Hence, the initial states $q_0^i 1$ of the LMCs $\mathcal{D}_i(\alpha_i)$ are probabilistic bisimilar.

($\Longleftarrow$) Assume that there is a general strategy α_i for each MDP $\mathcal{D}_i$ such that the k initial states in the induced LMCs are probabilistic bisimilar. Since the general strategies are possibly randomised, there might be multiple trees embedded in each of the induced LMC. We extract one common tree from these LMCs. We will then show that this tree witnesses the intersection nonemptiness of the k DTTAs as there is a run on this common tree for every DTTA $\mathcal{A}_i$.

Let us prioritise the symbols in Σ such that each symbol has a different priority. In the following, we show how to obtain an ordered tree t_i for the MDP $\mathcal{D}_i$ level by level. The process is similar to the construction of the LMC induced by the general strategy α_i. We first add the initial state $q_0^i 1$ (it is also a path with only one state) to the tree and make it the root of the tree. We call this node $q_0^i 1$. For every node ρ in the tree without children, we assign a symbol from Σ to it and add the children as follows. Since the LMCs induced by the general strategies are probabilistic bisimilar, there is no state in the LMCs labelled with $deadlock_i$. For a node ρ in the tree, we have that $\mathsf{last}(\rho)$ is of the form qj where q is a state in the DTTA $\mathcal{A}_i$ and j is a number. The strategy α_i over the path ρ is a distribution over the set $\mathrm{support}(\alpha_i(\rho)) \subseteq Act_i(qj)$, a subset of the available actions at qj. Let $a \in \Sigma$ be the one with the highest priority in $\mathrm{support}(\alpha_i(\rho))$ and now make the node ρ an a-node. From now on, we only consider paths in the MDP $\mathcal{D}_i$ with prefix $\rho a(qa)$ and discard all the other paths with prefix ρ. The strategy α_i on the path $\rho a(qa)$ is a Dirac distribution on the only available action m. Each of the $rank(a)$ successor of the path $\rho a(qa)$ is of the form $\rho a(qa)\mathsf{m}(q'j')$, which is then added as the j'th child of the node ρ. The tree is infinite and is over Σ.

For all the MDPs, the ordered tree constructed in the above way is the same. The nodes have the same symbols from Σ. We label each node ρ in t_i with $q \in \mathcal{A}_i$, where $qj = \mathsf{last}(\rho)$. It is not hard to verify that this labelling of the tree t_i is a run of the DTTA $\mathcal{A}_i$ on t_i. Thus, the tree is in the intersection of the languages of the DTTAs. ◀

▶ **Lemma 9.** *The bimilarity problem for an LMC and an MDP is NP-hard.*

Proof. Given an instance of Subset Sum $\langle P, N\rangle$ where $P = \{p_1, \ldots, p_n\}$ and $N \in \mathbb{N}$, we construct an MDP $\mathcal{D}$; see Figure 6. Let $T = \sum_{p_i \in P} p_i$. In the MDP, state s transitions to state s_i with probability p_i/T for all $1 \leq i \leq n$. Each state s_i has two available actions, each transitions to s_a and s_b by taking the action m_i and m'_i, respectively. State t transitions to t_1 and t_2 with probability N/T and $1 - N/T$, respectively. All the remaining states have only one available action transitioning to the successor state with probability one. States s_b and t_b have label b and all other states have label a.

It suffices to consider memoryless strategies in the MDP constructed since the only states that we need to specify a strategy are the s_i states and there is only one path from s to any of the s_i state.

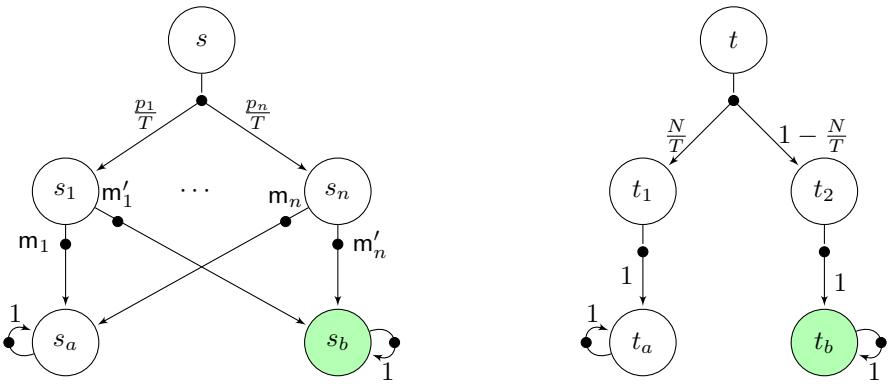

Figure 6 The MDP $\mathcal{D}$ in the reduction for NP-hardness of the bisimilarity problem. All states have the same label a except s_b and t_b which have label b.

Next, we show that $\langle P, N\rangle \in$ Subset Sum if and only if there exists a general strategy α such that s and t are probabilistic bisimilar in the induced LMC $\mathcal{D}_\alpha$.

Intuitively, making s_i probabilistic bisimilar with t_1 simulates the membership of p_i in Q. Conversely, making s_i probabilistic bisimilar with t_2 simulates the membership of p_i in $P \setminus Q$.

($\Longrightarrow$) Let $Q \subseteq P$ be the set such that $\sum_{p_i \in Q} p_i = N$. Let α be an memoryless deterministic strategy such that if $p_i \in Q$ then $\alpha(s_i) = \mathsf{m}_i$ and $\alpha(s_i) = \mathsf{m}'_i$ otherwise. It is clear that in the induced LMC, all states s_i where $p_i \in Q$ are probabilistic bisimilar with t_1 and all the other states are probabilistic bisimilar with t_2. Since $\sum_{p_i \in Q} p_i = N$, s and t are probabilistic bisimilar in the induced LMC.

($\Longleftarrow$) Assume there is a memoryless strategy α such that s and t are probabilistic bisimilar in the induced LMC $\mathcal{D}_\alpha$. We have $t_1 \not\sim t_2$. Let S_1 be the set of successor states of s that are probabilistic bisimilar to t_1. Then, $\sum_{s_i \in S_1} \frac{p_i}{T} = \tau(s)(S_1) = \tau(t)(t_1) = \frac{N}{T}$. Let $Q = \{p_i \in P \mid s_i \in S_1\}$. We have $Q \subseteq P$ be the set such that $\sum_{p_i \in Q} p_i = N$. ◀

B Proofs of Section 4

▶ **Lemma 13.** *Let $s, t \in S$ with $s[\sigma] \sim t[\tau]$ for all general strategies σ and τ.*

1. *For all general strategies σ and τ, we have $s[\sigma] \sim s[\tau]$ and $t[\sigma] \sim t[\tau]$;*
2. *For all successors u of s and all general strategies σ and τ, we have $u[\sigma] \sim u[\tau]$.*

Proof.

1. Assume for all general strategies $\sigma, \tau : s[\sigma] \sim t[\tau]$. Since bisimulation is an equivalence relation, we have that for all general strategies $\sigma, \tau : s[\sigma] \sim s[\tau]$. Similarly, we have $t[\sigma] \sim t[\tau]$ for all general strategies σ and τ.
2. Assume $s[\sigma] \sim t[\tau]$ for all general strategies σ and τ. By Item 1, we have $s[\sigma] \sim s[\tau]$ for all general strategies σ and τ. Assume there exist a successor u of s and general strategies σ, τ such that $u[\sigma] \not\sim u[\tau]$.
   Let σ' be a general strategy that in s takes an action m to reach u with positive chance in the first step and plays the strategy σ once u is reached, that is, $\varphi(s, \mathsf{m})(u) > 0$, $\sigma'(s)(\mathsf{m}) = 1$ and $\sigma'(s\mathsf{m}\rho_u) = \sigma(\rho_u)$ where ρ_u is any path starting with u.
   Let τ' be the same general strategy as σ' except that it plays the strategy τ once u is reached, that is, $\tau'(\rho) = \sigma'(\rho)$ for any path $\rho \neq s\mathsf{m}\rho_u$ and $\tau'(s\mathsf{m}\rho_u) = \tau(\rho_u)$ where ρ_u is any path starting with u.
   In the LMCs $\mathcal{D}_{\sigma'}$ and $\mathcal{D}_{\tau'}$, for all the other successor states v of s in $\mathcal{D}$ where $v \neq u$, we have $s\mathsf{m}v[\sigma'] \sim s\mathsf{m}v[\tau']$. However, since $s\mathsf{m}u[\sigma'] \not\sim s\mathsf{m}u[\tau']$, we have $s[\sigma'] \not\sim s[\tau']$, which contradicts that $s[\sigma] \sim s[\tau]$ holds for all general strategies σ and τ. ◀

▶ **Lemma 14.** *Let $s, t \in S$ with $(s,0) \approx_{\bar{\mathcal{D}}} (t,1)$.*

1. *Let R be any superbisimulation that contains $\big((s,0),(t,1)\big)$. For any successor $(u,0)$ of $(s,0)$, there exists a successor $(v,1)$ of $(t,1)$ such that $\big((u,0),(v,1)\big) \in R$. Similarly, for any successor $(v,1)$ of $(t,1)$, there exists a successor $(u,0)$ of $(s,0)$ such that $\big((u,0),(v,1)\big) \in R$. In other words, any successor of $(s,0)$ is superbisimilar with some successor of $(t,1)$ and vice versa.*
2. *We have $(s,1) \approx_{\bar{\mathcal{D}}} (t,0)$, $(s,0) \approx_{\bar{\mathcal{D}}} (s,1)$ and $(t,0) \approx_{\bar{\mathcal{D}}} (t,1)$.*

Proof. Assume $(s,0) \approx (t,1)$.

1. Let R be a superbisimulation such that $\big((s,0),(t,1)\big) \in R$. Let $X = \bar{S}/_R$. For a contradiction, assume for a successor of $(s,0)$, say $(u,0)$, there exists no successor $(v,1)$ of $(t,1)$ such that $\big((u,0),(v,1)\big) \in R$. Then, $(u,0)$ is in an equivalence class $E \in X$ in which there are no successors of $(t,1)$. Let α be a memoryless strategy such that $\alpha\big((s,0)\big)\big((u,0)\big) > 0$. We have $\alpha\big((s,0)\big)(E) > 0 = \alpha\big((t,1)\big)(E)$, which contradicts that $\big((s,0),(t,1)\big) \in R$.
2. If $(s,0) \approx (t,1)$ then, by symmetry, $(s,1) \approx (t,0)$.
   Let $R = \{\big((s,i),(s,i)\big) \mid s \in S \wedge i \in \{0,1\}\} \cup \{\big((s,0),(s,1)\big), \big((s,1),(s,0)\big) \mid \exists t \in S$ such that $(s,0) \approx (t,1)\}$. To show $(s,0) \approx (s,1)$ and $(t,0) \approx (t,1)$, it suffices to show that R is a superbisimulation.
   Firstly, note that R is an equivalence relation.
   For any $\big((s,i),(s,j)\big) \in R$, clearly we have $\bar{\ell}\big((s,i)\big) = \bar{\ell}\big((s,j)\big)$. Let $X = \bar{S}/_R$. It remains to show that for all $\big((s,i),(s,j)\big) \in R$ it holds that $\alpha\big((s,i)\big)(X) = \alpha\big((s,j)\big)(X)$ for all memoryless strategies α. Assume $\big((s,i),(s,j)\big) \in R$. If $i = j$, it is trivially true. Otherwise, $j = 1 - i$. There must exist a state $t \in S$ such that $(t,j) \approx (s,i)$. For any successor (u,i) of (s,i), by Item 1, there is some successor (v,j) of (t,j) such that $\big((u,i),(v,j)\big) \in R$. Thus, for all successors (u,i) of (s,i), $\big((u,i),(u,j)\big)$ is in R and $(u,i),(u,j)$ are in the same equivalence class in X. Thus, we have $\alpha\big((s,i)\big)(X) = \alpha\big((s,j)\big)(X)$ for all memoryless strategies α.
   Hence, R is a superbisimulation. ◀

▶ **Theorem 15.** *For all $s, t \in S$, we have $(s,0) \approx (t,1) \iff \forall$ general strategies $\sigma, \tau : s[\sigma] \sim t[\tau]$.*

Proof. ($\Longleftarrow$) To show that $(s,i) \approx (t,j)$ for any $\big((s,i),(t,j)\big) \in R$, it suffices to show that R is a superbisimulation of $\bar{\mathcal{D}}'$.

Let α be an arbitrary memoryless strategy for $\bar{\mathcal{D}}'$. We define a relation $R_\alpha = \{\big((s,i),(t,j)\big) \in \bar{S}' \times \bar{S}' \mid (s,i)[\alpha] \sim (t,j)[\alpha]\}$. Then, R_α is a probabilistic bisimulation on $\bar{S}'$ and $X_\alpha = \bar{S}'/_{R_\alpha}$ is the set of probabilistic bisimulation classes.

Next, we show $R = R_\alpha$. It is obvious that $R \subseteq R_\alpha$. To show $R_\alpha \subseteq R$, we assume $\big((s,i),(t,j)\big) \in \bar{S}' \times \bar{S}'$ and $(s,i)[\alpha] \sim (t,j)[\alpha]$. Since $(s,i) \in \bar{S}'$, we have $(s,i)[\alpha] \sim (s,i)[\sigma]$ for all general strategies σ. Similarly, we have $(t,j)[\alpha] \sim (t,j)[\tau]$ for all general strategies τ. Since probabilistic bisimulation is transitive, we have $(s,i)[\sigma] \sim (s,i)[\alpha] \sim (t,j)[\alpha] \sim (t,j)[\tau]$ for all general strategies σ and τ.

Let $X = \bar{S}'/_R$. We have $X_\alpha = \bar{S}'/_{R_\alpha} = \bar{S}'/_R = X$. Let $\big((s,i),(t,j)\big) \in R$. We have $\big((s,i),(t,j)\big) \in R_\alpha$ and $\alpha\big((s,i)\big)(X) = \alpha\big((s,i)\big)(X_\alpha) = \alpha\big((t,j)\big)(X_\alpha) = \alpha\big((t,j)\big)(X)$. Since α was arbitrary, R is a superbisimulation of $\bar{\mathcal{D}}'$.

($\Longrightarrow$) We show that the relation R' defined in the proof in the main text is a probabilistic bisimulation on the states of the LMC $\mathcal{D}'_\sigma \oplus \mathcal{D}'_\tau$.

Let X' be the partition of the states of the LMC $\mathcal{D}'_\sigma \oplus \mathcal{D}'_\tau$ with respect to R'. Let $(\rho_1[\mu_1], \rho_2[\mu_2]) \in R'$ with $\mathsf{last}(\rho_1) = s$ and $\mathsf{last}(\rho_2) = t$. To show R' is a probabilistic bisimulation on the states of the LMC $\mathcal{D}'_\sigma \oplus \mathcal{D}'_\tau$, it suffices to show that (1) $\rho_1[\mu_1]$ and $\rho_2[\mu_2]$ have the same label; (2) the probability distributions over X' from $\rho_1[\mu_1]$ and from $\rho_2[\mu_2]$ are the same.

Since $(\rho_1[\mu_1], \rho_2[\mu_2]) \in R'$, we have $(s,0) \approx (t,1)$. It follows that $\rho_1[\mu_1]$ and $\rho_2[\mu_2]$ have the same label. Furthermore, we have $\alpha_1(s)(X) = \alpha_2(t)(X)$ for all memoryless strategies α_1 and α_2 for $\mathcal{D}$. Let $\alpha_1(s) = \mu_1(\rho_1)$, $\alpha_2(t) = \mu_2(\rho_2)$. The successors of $(\rho_1[\mu_1]$ $((\rho_2[\mu_2])$ can be partitioned with respect to R' and each class can be identified by a set $E \in X$. We define $E_1 = \{\rho_1 \mathsf{m} u[\mu_1] \mid \mathsf{m} \in Act(s) \wedge u \in E\}$, which is the set of successors of $\rho_1[\mu_1]$ corresponding to E. Similarly, define $E_2 = \{\rho_2 \mathsf{m} u[\mu_2] \mid \mathsf{m} \in Act(t) \wedge u \in E\}$, which is the set of successors of $\rho_2[\mu_2]$ corresponding to E. We have that the transition probability from $\rho_1[\mu_1]$ to E_1 is $\sum_{\mathsf{m} \in Act(s)} \mu_1(\rho_1)(\mathsf{m})\varphi(s,\mathsf{m})(E_1)$, which is equal to $\alpha_1(s)(E)$. Similarly, the transition probability from $\rho_2[\mu_2]$ to E_2 is $\sum_{\mathsf{m} \in Act(t)} \mu_2(\rho_2)(\mathsf{m})\varphi(t,\mathsf{m})(E_2)$, which is equal to $\alpha_2(t)(E)$. Thus, the probability distribution over X' from $\rho_1[\mu_1]$ is equal to that from $\rho_2[\mu_2]$, and we conclude that R' is a probabilistic bisimulation on the states of the LMC $\mathcal{D}'_\sigma \oplus \mathcal{D}'_\tau$. ◀