Limit Synchronization in Markov Decision Processes*

Laurent Doyen[1], Thierry Massart[2], and Mahsa Shirmohammadi[1,2]

[1] LSV, ENS Cachan & CNRS, France
[2] Université Libre de Bruxelles, Belgium

Abstract. Markov decision processes (MDP) are finite-state systems with both strategic and probabilistic choices. After fixing a strategy, an MDP produces a sequence of probability distributions over states. The sequence is eventually synchronizing if the probability mass accumulates in a single state, possibly in the limit. Precisely, for $0 \leq p \leq 1$ the sequence is p-synchronizing if a probability distribution in the sequence assigns probability at least p to some state, and we distinguish three synchronization modes: (*i*) *sure* winning if there exists a strategy that produces a 1-synchronizing sequence; (*ii*) *almost-sure* winning if there exists a strategy that produces a sequence that is, for all $\epsilon > 0$, a (1-ϵ)-synchronizing sequence; (*iii*) *limit-sure* winning if for all $\epsilon > 0$, there exists a strategy that produces a (1-ϵ)-synchronizing sequence. We consider the problem of deciding whether an MDP is sure, almost-sure, or limit-sure winning, and we establish the decidability and optimal complexity for all modes, as well as the memory requirements for winning strategies. Our main contributions are as follows: (a) for each winning modes we present characterizations that give a PSPACE complexity for the decision problems, and we establish matching PSPACE lower bounds; (b) we show that for sure winning strategies, exponential memory is sufficient and may be necessary, and that in general infinite memory is necessary for almost-sure winning, and unbounded memory is necessary for limit-sure winning; (c) along with our results, we establish new complexity results for alternating finite automata over a one-letter alphabet.

1 Introduction

Markov decision processes (MDP) are finite-state stochastic models used in the design of systems that exhibit both controllable and stochastic behavior, such as in planning, randomized algorithms, and communication protocols [2, 13, 4]. The controllable choices along the execution are fixed by a strategy, and the stochastic choices describe the system response. When a strategy is fixed in an MDP, the *symbolic* outcome is a sequence of probability distributions over states of the MDP, which differs from the traditional semantics where a probability measure

* This work was partially supported by the Belgian Fonds National de la Recherche Scientifique (FNRS), and by the PICS project *Quaverif* funded by the French Centre National de la Recherche Scientifique (CNRS).

is considered over sets of sequences of states. This view is adequate in many applications, such as systems biology, sensor networks, robot planning, etc. [15, 5], where the system consists of several copies of the same process (molecules, sensors, robots, etc.), and the relevant information along the execution of the system is the number of processes in each state, or the relative frequency (i.e., the probability) of each state. In recent works, the verification of quantitative properties of the symbolic outcome was shown undecidable [18]. Decidability is obtained for special subclasses [6], or through approximations [1].

In this paper, we consider a general class of strategies that select actions depending on the full history of the system execution. In the context of several identical processes, the same strategy is used in every process, but the internal state of each process need not be the same along the execution, since probabilistic transitions may have different outcome in each process. Therefore, the execution of the system is best described by the sequence of probability distributions over states along the execution. Previously, the special case of word-strategies have been considered, that at each step select the same control action in all states, and thus only depend on the number of execution steps of the system. Several problems for MDPs with word-strategies (also known as probabilistic automata) are undecidable [3, 14, 18, 11]. In particular the limit-sure reachability problem, which is to decide whether a given state can be reached with probability arbitrarily close to one, is undecidable for probabilistic automata [14].

We establish the decidability and optimal complexity of deciding synchronizing properties for the symbolic outcome of MDPs under general strategies. Synchronizing properties require that the probability distributions tend to accumulate all the probability mass in a single state, or in a set of states. They generalize synchronizing properties of finite automata [20, 10]. Formally for $0 \leq p \leq 1$, a sequence $\bar{X} = X_0 X_1 \dots$ of probability distributions $X_i : Q \to [0, 1]$ over state space Q of an MDP is *eventually p-synchronizing* if for some $i \geq 0$, the distribution X_i assigns probability at least p to some state. Analogously, it is *always p-synchronizing* if in all distributions X_i, there is a state with probability at least p. For $p = 1$, these definitions are the qualitative analogous for sequences of distributions of the traditional reachability and safety conditions [9]. In particular, an eventually 1-synchronizing sequence witnesses that there is a length ℓ such that all paths of length ℓ in the MDP reach a single state, which is thus reached synchronously no matter the probabilistic choices.

Viewing MDPs as one-player stochastic games, we consider the following traditional winning modes (see also Table 1): (*i*) *sure* winning, if there is a strategy that generates an {eventually, always} 1-synchronizing sequence; (*ii*) *almost-sure* winning, if there exists a strategy that generates a sequence that is, for all $\epsilon > 0$, {eventually, always} $(1 - \epsilon)$-synchronizing; (*iii*) *limit-sure* winning, if for all $\epsilon > 0$, there is a strategy that generates an {eventually, always} $(1 - \epsilon)$-synchronizing sequence.

We show that the three winning modes form a strict hierarchy for eventually synchronizing: there are limit-sure winning MDPs that are not almost-sure win-

	Always	Eventually
Sure	$\exists\alpha \quad \forall n \ \ \mathcal{M}_n^\alpha(T) = 1$	$\exists\alpha \quad \exists n \ \ \mathcal{M}_n^\alpha(T) = 1$
Almost-sure	$\exists\alpha \ \ \inf_n \mathcal{M}_n^\alpha(T) = 1$	$\exists\alpha \ \ \sup_n \mathcal{M}_n^\alpha(T) = 1$
Limit-sure	$\sup_\alpha \inf_n \mathcal{M}_n^\alpha(T) = 1$	$\sup_\alpha \sup_n \mathcal{M}_n^\alpha(T) = 1$

Table 1. Winning modes and synchronizing objectives (where $\mathcal{M}_n^\alpha(T)$ denotes the probability that under strategy α, after n steps the MDP $\mathcal{M}$ is in a state of T).

ning, and there are almost-sure winning MDPs that are not sure winning. For always synchronizing, the three modes coincide.

For each winning mode, we consider the problem of deciding if a given initial distribution is winning. We establish the decidability and optimal complexity bounds for all winning modes. Under general strategies, the decision problems have much lower complexity than with word-strategies. We show that all decision problems are decidable, in polynomial time for always synchronizing, and PSPACE-complete for eventually synchronizing. This is also in contrast with almost-sure winning in the traditional semantics of MDPs, which is solvable in polynomial time for both safety and reachability objectives [8]. Our complexity results are shown in Table 2.

We complete the picture by providing optimal memory bounds for winning strategies. We show that for sure winning strategies, exponential memory is sufficient and may be necessary, and that in general infinite memory is necessary for almost-sure winning, and unbounded memory is necessary for limit-sure winning.

Some results in this paper rely on insights related to games and alternating automata that are of independent interest. First, the sure-winning problem for eventually synchronizing is equivalent to a two-player game with a synchronized reachability objective, where the goal for the first player is to ensure that a target state is reached after a number of steps that is independent of the strategy of the opponent (and thus this number can be fixed in advance by the first player). This condition is stronger than plain reachability, and while the winner in two-player reachability games can be decided in polynomial time, deciding the winner for synchronized reachability is PSPACE-complete. This result is obtained by turning the synchronized reachability game into a one-letter alternating automaton for which the emptiness problem (i.e., deciding if there exists a word accepted by the automaton) is PSPACE-complete [16, 17]. Second, our PSPACE lower bound for the limit-sure winning problem in eventually synchronizing uses a PSPACE-completeness result that we establish for the *universal finiteness problem*, which is to decide, given a one-letter alternating automata, whether from every state the accepted language is finite.

A full version of this paper with all proofs is available [12].

2 Markov Decision Processes and Synchronization

A *probability distribution* over a finite set S is a function $d : S \to [0,1]$ such that $\sum_{s \in S} d(s) = 1$. The *support* of d is the set $\mathsf{Supp}(d) = \{s \in S \mid d(s) > 0\}$. We denote by $\mathcal{D}(S)$ the set of all probability distributions over S. For $T \neq \emptyset$, the *uniform distribution* on T assigns probability $\frac{1}{|T|}$ to every state in T. Given $s \in S$, the *Dirac distribution* on s assigns probability 1 to s, and by a slight abuse of notation, we usually denote it simply by s.

2.1 Markov decision processes

A *Markov decision process* (MDP) $\mathcal{M} = \langle Q, \mathsf{A}, \delta \rangle$ consists of a finite set Q of states, a finite set A of actions, and a probabilistic transition function $\delta : Q \times \mathsf{A} \to \mathcal{D}(Q)$. A state q is *absorbing* if $\delta(q,a)$ is the Dirac distribution on q for all actions $a \in \mathsf{A}$.

We describe the behavior of an MDP as a one-player stochastic game played for infinitely many rounds. Given an initial distribution $\mu_0 \in \mathcal{D}(Q)$, the game starts in the first round in state q with probability $\mu_0(q)$. In each round, the player chooses an action $a \in \mathsf{A}$, and if the game is in state q, the next round starts in the successor state q' with probability $\delta(q,a)(q')$.

Given $q \in Q$ and $a \in \mathsf{A}$, denote by $\mathsf{post}(q,a)$ the set $\mathsf{Supp}(\delta(q,a))$, and given $T \subseteq Q$ let $\mathsf{Pre}(T) = \{q \in Q \mid \exists a \in \mathsf{A} : \mathsf{post}(q,a) \subseteq T\}$ be the set of states from which the player has an action to ensure that the successor state is in T. For $k > 0$, let $\mathsf{Pre}^k(T) = \mathsf{Pre}(\mathsf{Pre}^{k-1}(T))$ with $\mathsf{Pre}^0(T) = T$.

A *path* in $\mathcal{M}$ is an infinite sequence $\pi = q_0 a_0 q_1 a_1 \dots$ such that $q_{i+1} \in \mathsf{post}(q_i, a_i)$ for all $i \geq 0$. A finite prefix $\rho = q_0 a_0 q_1 a_1 \dots q_n$ of a path (or simply a finite path) has length $|\rho| = n$ and last state $\mathsf{Last}(\rho) = q_n$. We denote by $\mathsf{Play}(\mathcal{M})$ and $\mathsf{Pref}(\mathcal{M})$ the set of all paths and finite paths in $\mathcal{M}$ respectively.

For the decision problems considered in this paper, only the support of the probability distributions in the transition function is relevant (i.e., the exact value of the positive probabilities does not matter); therefore, we can encode an MDP as an A-labelled transition system (Q, R) with $R \subseteq Q \times \mathsf{A} \times Q$ such that $(q, a, q') \in R$ is a transition if $q' \in \mathsf{post}(q,a)$.

Strategies. A *randomized strategy* for $\mathcal{M}$ (or simply a strategy) is a function $\alpha : \mathsf{Pref}(\mathcal{M}) \to \mathcal{D}(\mathsf{A})$ that, given a finite path ρ, returns a probability distribution $\alpha(\rho)$ over the action set, used to select a successor state q' of ρ with probability $\sum_{a \in \mathsf{A}} \alpha(\rho)(a) \cdot \delta(q,a)(q')$ where $q = \mathsf{Last}(\rho)$.

A strategy α is *pure* if for all $\rho \in \mathsf{Pref}(\mathcal{M})$, there exists an action $a \in \mathsf{A}$ such that $\alpha(\rho)(a) = 1$; and *memoryless* if $\alpha(\rho) = \alpha(\rho')$ for all ρ, ρ' such that $\mathsf{Last}(\rho) = \mathsf{Last}(\rho')$. We view pure strategies as functions $\alpha : \mathsf{Pref}(\mathcal{M}) \to \mathsf{A}$, and memoryless strategies as functions $\alpha : Q \to \mathcal{D}(\mathsf{A})$, Finally, a strategy α uses *finite-memory* if it can be represented by a finite-state transducer $T = \langle \mathsf{Mem}, m_0, \alpha_u, \alpha_n \rangle$ where Mem is a finite set of modes (the memory of the strategy), $m_0 \in \mathsf{Mem}$ is the initial mode, $\alpha_u : \mathsf{Mem} \times \mathsf{A} \times Q \to \mathsf{Mem}$ is an update function, that given the current memory, last action and state updates the memory, and $\alpha_n : \mathsf{Mem} \times Q \to \mathcal{D}(\mathsf{A})$

is a next-move function that selects the probability distribution $\alpha_n(m, q)$ over actions when the current mode is m and the current state of $\mathcal{M}$ is q. For pure strategies, we assume that $\alpha_n : \mathsf{Mem} \times Q \to \mathsf{A}$. The *memory size* of the strategy is the number $|\mathsf{Mem}|$ of modes. For a finite-memory strategy α, let $\mathcal{M}(\alpha)$ be the Markov chain obtained as the product of $\mathcal{M}$ with the transducer defining α.

2.2 State semantics

In the traditional state semantics, given an initial distribution $\mu_0 \in \mathcal{D}(Q)$ and a strategy α in an MDP $\mathcal{M}$, a *path-outcome* is a path $\pi = q_0 a_0 q_1 a_1 \dots$ in $\mathcal{M}$ such that $q_0 \in \mathsf{Supp}(\mu_0)$ and $a_i \in \mathsf{Supp}(\alpha(q_0 a_0 \dots a_{i-1} q_i))$ for all $i \geq 0$. The probability of a finite prefix $\rho = q_0 a_0 q_1 a_1 \dots q_n$ of π is $\mu_0(q_0) \cdot \prod_{j=0}^{n-1} \alpha(q_0 a_0 \dots q_j)(a_j) \cdot \delta(q_j, a_j)(q_{j+1})$. We denote by $\mathit{Outcomes}(\mu_0, \alpha)$ the set of all path-outcomes from μ_0 under strategy α. An *event* $\Omega \subseteq \mathsf{Play}(\mathcal{M})$ is a measurable set of paths, and given an initial distribution μ_0 and a strategy α, the probabilities $Pr^\alpha(\Omega)$ of events Ω are uniquely defined [19]. In particular, given a set $T \subseteq Q$ of target states, and $k \in \mathbb{N}$, we denote by $\Box T = \{q_0 a_0 q_1 \dots \in \mathsf{Play}(\mathcal{M}) \mid \forall i : q_i \in T\}$ the safety event of always staying in T, by $\Diamond T = \{q_0 a_0 q_1 \dots \in \mathsf{Play}(\mathcal{M}) \mid \exists i : q_i \in T\}$ the event of reaching T, and by $\Diamond^k T = \{q_0 a_0 q_1 \dots \in \mathsf{Play}(\mathcal{M}) \mid q_k \in T\}$ the event of reaching T after exactly k steps. Hence, $\Pr^\alpha(\Diamond T)$ is the probability to reach T under strategy α.

We consider the following classical winning modes. Given an initial distribution μ_0 and an event Ω, we say that $\mathcal{M}$ is:

- *sure winning* if there exists a strategy α such that $\mathit{Outcomes}(\mu_0, \alpha) \subseteq \Omega$;
- *almost-sure winning* if there exists a strategy α such that $\Pr^\alpha(\Omega) = 1$;
- *limit-sure winning* if $\sup_\alpha \Pr^\alpha(\Omega) = 1$.

It is known for safety objectives $\Box T$ in MDPs that the three winning modes coincide, and for reachability objectives $\Diamond T$ that an MDP is almost-sure winning if and only if it is limit-sure winning. For both objectives, the set of initial distributions for which an MDP is sure (resp., almost-sure or limit-sure) winning can be computed in polynomial time [8].

2.3 Distribution semantics

In contrast to the state semantics, we consider the outcome of an MDP $\mathcal{M}$ under a fixed strategy as a sequence of probability distributions over states defined as follows [18]. Given an initial distribution $\mu_0 \in \mathcal{D}(Q)$ and a strategy α in $\mathcal{M}$, the *symbolic outcome* of $\mathcal{M}$ from μ_0 is the sequence $(\mathcal{M}_n^\alpha)_{n \in \mathbb{N}}$ of probability distributions defined by $\mathcal{M}_k^\alpha(q) = Pr^\alpha(\Diamond^k \{q\})$ for all $k \geq 0$ and $q \in Q$. Hence, $\mathcal{M}_k^\alpha$ is the probability distribution over states after k steps under strategy α. Note that $\mathcal{M}_0^\alpha = \mu_0$.

Informally, synchronizing objectives require that the probability of some state (or some group of states) tends to 1 in the sequence $(\mathcal{M}_n^\alpha)_{n \in \mathbb{N}}$. Given a set $T \subseteq Q$, consider the functions $\mathit{sum}_T : \mathcal{D}(Q) \to [0, 1]$ and $\mathit{max}_T : \mathcal{D}(Q) \to$

	Always		Eventually	
	Complexity	Memory requirement	Complexity	Memory requirement
Sure	PTIME-C	memoryless	PSPACE-C	exponential
Almost-sure			PSPACE-C	infinite
Limit-sure			PSPACE-C	unbounded

Table 2. Computational complexity of the membership problem, and memory requirement for the strategies (for always synchronizing, the three modes coincide).

$[0, 1]$ that compute $sum_T(X) = \sum_{q \in T} X(q)$ and $max_T(X) = \max_{q \in T} X(q)$. For $f \in \{sum_T, max_T\}$ and $p \in [0, 1]$, we say that a probability distribution X is p-synchronized according to f if $f(X) \geq p$, and that a sequence $\bar{X} = X_0 X_1 \dots$ of probability distributions is:

(a) *always p-synchronizing* if X_i is p-synchronized for all $i \geq 0$;
(b) *event* (or *eventually*) *p-synchronizing* if X_i is p-synchronized for some $i \geq 0$.

For $p = 1$, we view these definitions as the qualitative analogous for sequences of distributions of the traditional safety and reachability conditions for sequences of states [9]. Now, we define the following winning modes. Given an initial distribution μ_0 and a function $f \in \{sum_T, max_T\}$, we say that for the objective of {always, eventually} synchronizing from μ_0, $\mathcal{M}$ is:

- *sure winning* if there exists a strategy α such that the symbolic outcome of α from μ_0 is {always, eventually} 1-synchronizing according to f;
- *almost-sure winning* if there exists a strategy α such that for all $\epsilon > 0$ the symbolic outcome of α from μ_0 is {always, eventually} $(1 - \epsilon)$-synchronizing according to f;
- *limit-sure winning* if for all $\epsilon > 0$, there exists a strategy α such that the symbolic outcome of α from μ_0 is {always, eventually} $(1 - \epsilon)$-synchronizing according to f;

We often use $X(T)$ instead of $sum_T(X)$, as in Table 1 where the definitions of the various winning modes and synchronizing objectives for $f = sum_T$ are summarized. In Section 2.4, we present an example to illustrate the definitions.

2.4 Decision problems

For $f \in \{sum_T, max_T\}$ and $\lambda \in \{\text{always, event}\}$, the *winning region* $\langle\langle 1 \rangle\rangle^{\lambda}_{sure}(f)$ is the set of initial distributions such that $\mathcal{M}$ is sure winning for λ-synchronizing (we assume that $\mathcal{M}$ is clear from the context). We define analogously the winning regions $\langle\langle 1 \rangle\rangle^{\lambda}_{almost}(f)$ and $\langle\langle 1 \rangle\rangle^{\lambda}_{limit}(f)$. For a singleton $T = \{q\}$ we have $sum_T = max_T$, and we simply write $\langle\langle 1 \rangle\rangle^{\lambda}_{\mu}(q)$ (where $\mu \in \{\text{sure, almost, limit}\}$). We are interested in the algorithmic complexity of the *membership problem*, which is to decide, given a probability distribution μ_0, whether $\mu_0 \in \langle\langle 1 \rangle\rangle^{\lambda}_{\mu}(f)$. As we show

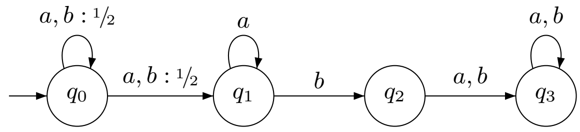

Fig. 1. An MDP $\mathcal{M}$ such that $\langle\!\langle 1 \rangle\!\rangle^{event}_{sure}(q_1) \neq \langle\!\langle 1 \rangle\!\rangle^{event}_{almost}(q_1)$ and $\langle\!\langle 1 \rangle\!\rangle^{event}_{almost}(q_2) \neq \langle\!\langle 1 \rangle\!\rangle^{event}_{limit}(q_2)$.

below, it is easy to establish the complexity of the membership problems for always synchronizing, while it is more tricky for eventually synchronizing. The complexity results are summarized in Table 2.

Always synchronizing. We first remark that for always synchronizing, the three winning modes coincide.

Lemma 1. *Let T be a set of states. For all functions $f \in \{max_T, sum_T\}$, we have $\langle\!\langle 1 \rangle\!\rangle^{always}_{sure}(f) = \langle\!\langle 1 \rangle\!\rangle^{always}_{almost}(f) = \langle\!\langle 1 \rangle\!\rangle^{always}_{limit}(f)$.*

It follows from the proof of Lemma 1 that the winning region for always synchronizing according to sum_T coincides with the set of winning initial distributions for the safety objective $\Box T$ in the traditional state semantics, which can be computed in polynomial time [7]. Moreover, always synchronizing according to max_T is equivalent to the existence of an infinite path staying in T in the transition system $\langle Q, R \rangle$ of the MDP restricted to transitions $(q, a, q') \in R$ such that $\delta(q, a)(q') = 1$, which can also be decided in polynomial time. In both cases, pure memoryless strategies are sufficient.

Theorem 1. *The membership problem for always synchronizing can be solved in polynomial time, and pure memoryless strategies are sufficient.*

Eventually synchronizing. For all functions $f \in \{max_T, sum_T\}$, the following inclusions hold: $\langle\!\langle 1 \rangle\!\rangle^{event}_{sure}(f) \subseteq \langle\!\langle 1 \rangle\!\rangle^{event}_{almost}(f) \subseteq \langle\!\langle 1 \rangle\!\rangle^{event}_{limit}(f)$ and we show that the inclusions are strict in general. Consider the MDP in Fig. 1 with initial state q_0 and target $T = \{q_1\}$. For all strategies, the probability in q_0 is always positive, implying that the MDP is not sure-winning in $\{q_1\}$. However, the MDP is almost-sure winning in $\{q_1\}$ using a strategy that always plays a. Now, consider target $T = \{q_2\}$. For all $\epsilon > 0$, we can have probability at least $1 - \epsilon$ in q_2 by playing a long enough, and then b. For a fixed strategy, this probability never tends to 1 since if the probability $p > 0$ in q_2 is positive at a certain step, then it remains bounded by $1 - p < 1$ for all next steps. Therefore, the MDP is not almost-sure winning in $\{q_2\}$, but it is limit-sure winning.

Lemma 2. *There exists an MDP $\mathcal{M}$ and states q_1, q_2 such that:*

(i) $\langle\!\langle 1 \rangle\!\rangle^{event}_{sure}(q_1) \subsetneq \langle\!\langle 1 \rangle\!\rangle^{event}_{almost}(q_1)$, *and*

(ii) $\langle\!\langle 1 \rangle\!\rangle^{event}_{almost}(q_2) \subsetneq \langle\!\langle 1 \rangle\!\rangle^{event}_{limit}(q_2)$.

The rest of this paper is devoted to the solution of the membership problem for eventually synchronizing. We make some preliminary remarks to show that it is sufficient to solve the membership problem according to $f = sum_T$ and for MDPs with a single initial state. Our results will also show that pure strategies are sufficient in all modes.

Remark. For eventually synchronizing and each winning mode, we show that the membership problem with function max_T is polynomial-time equivalent to the membership problem with function $sum_{T'}$ with a singleton T'. First, for $\mu \in \{\text{sure, almost, limit}\}$, we have $\langle\!\langle 1 \rangle\!\rangle_{\mu}^{event}(max_T) = \bigcup_{q \in T} \langle\!\langle 1 \rangle\!\rangle_{\mu}^{event}(q)$, showing that the membership problems for max are polynomial-time reducible to the corresponding membership problem for $sum_{T'}$ with singleton T'. The reverse reduction is as follows. Given an MDP $\mathcal{M}'$, a singleton $T' = \{q\}$ and an initial distribution μ'_0, we can construct an MDP $\mathcal{M}$ and initial distribution μ_0 such that $\mu'_0 \in \langle\!\langle 1 \rangle\!\rangle_{\mu}^{event}(q)$ iff $\mu_0 \in \langle\!\langle 1 \rangle\!\rangle_{\mu}^{event}(max_T)$ where $T = Q$ is the state space of $\mathcal{M}$. The idea is to construct $\mathcal{M}$ and μ_0 as a copy of $\mathcal{M}'$ and μ'_0 where all states except q are duplicated, and the initial and transition probabilities are evenly distributed between the copies. Therefore if the probability tends to 1 in some state, it has to be in q.

Remark. To solve the membership problems for eventually synchronizing with function sum_T, it is sufficient to provide an algorithm that decides membership of Dirac distributions (i.e., assuming MDPs have a single initial state), since to solve the problem for an MDP $\mathcal{M}$ with initial distribution μ_0, we can equivalently solve it for a copy of $\mathcal{M}$ with a new initial state q_0 from which the successor distribution on all actions is μ_0. Therefore, it is sufficient to consider initial Dirac distributions μ_0.

3 One-Letter Alternating Automata

In this section, we consider *one-letter alternating automata* (1L-AFA) as they have a structure of alternating graph analogous to MDP (i.e., when ignoring the probabilities). We review classical decision problems for 1L-AFA, and establish the complexity of a new problem, the *universal finiteness problem* which is to decide if from every initial state the language of a given 1L-AFA is finite. These results of independent interest are useful to establish the PSPACE lower bounds for eventually synchronizing in MDPs.

One-letter alternating automata. Let $\mathsf{B}^+(Q)$ be the set of positive Boolean formulas over Q, i.e. Boolean formulas built from elements in Q using $\wedge$ and $\vee$. A set $S \subseteq Q$ *satisfies* a formula $\varphi \in \mathsf{B}^+(Q)$ (denoted $S \models \varphi$) if φ is satisfied when replacing in φ the elements in S by true, and the elements in $Q \setminus S$ by false.

A *one-letter alternating finite automaton* is a tuple $\mathcal{A} = \langle Q, \delta_{\mathcal{A}}, \mathcal{F} \rangle$ where Q is a finite set of states, $\delta_{\mathcal{A}} : Q \to \mathsf{B}^+(Q)$ is the transition function, and $\mathcal{F} \subseteq Q$ is the set of accepting states. We assume that the formulas in transition function are in disjunctive normal form. Note that the alphabet of the automaton is omitted, as it has a single letter. In the language of a 1L-AFA, only the length of words is relevant. For all $n \geq 0$, define the set $Acc_{\mathcal{A}}(n, \mathcal{F}) \subseteq Q$ of states from which the word of length n is accepted by $\mathcal{A}$ as follows:

- $Acc_{\mathcal{A}}(0, \mathcal{F}) = \mathcal{F}$;
- $Acc_{\mathcal{A}}(n, \mathcal{F}) = \{q \in Q \mid Acc_{\mathcal{A}}(n-1, \mathcal{F}) \models \delta(q)\}$ for all $n > 0$.

The set $\mathcal{L}(\mathcal{A}_q) = \{n \in \mathbb{N} \mid q \in Acc_{\mathcal{A}}(n, \mathcal{F})\}$ is the *language* accepted by $\mathcal{A}$ from initial state q.

For fixed n, we view $Acc_{\mathcal{A}}(n, \cdot)$ as an operator on 2^Q that, given a set $\mathcal{F} \subseteq Q$ computes the set $Acc_{\mathcal{A}}(n, \mathcal{F})$. Note that $Acc_{\mathcal{A}}(n, \mathcal{F}) = Acc_{\mathcal{A}}(1, Acc_{\mathcal{A}}(n-1, \mathcal{F}))$ for all $n \geq 1$. Denote by $\mathsf{Pre}_{\mathcal{A}}(\cdot)$ the operator $Acc_{\mathcal{A}}(1, \cdot)$. Then for all $n \geq 0$ the operator $Acc_{\mathcal{A}}(n, \cdot)$ coincides with $\mathsf{Pre}^n_{\mathcal{A}}(\cdot)$, the n-th iterate of $\mathsf{Pre}_{\mathcal{A}}(\cdot)$.

Decision problems. We present the classical emptiness and finiteness problems for alternating automata, and we introduce a variant of the finiteness problem that will be useful for solving synchronizing problems for MDPs.

- The *emptiness problem* for 1L-AFA is to decide, given a 1L-AFA $\mathcal{A}$ and an initial state q, whether $\mathcal{L}(\mathcal{A}_q) = \emptyset$. The emptiness problem can be solved by checking whether $q \in \mathsf{Pre}^n_{\mathcal{A}}(\mathcal{F})$ for some $n \geq 0$. It is known that the emptiness problem is PSPACE-complete, even for transition functions in disjunctive normal form [16, 17].
- The *finiteness problem* is to decide, given a 1L-AFA $\mathcal{A}$ and an initial state q, whether $\mathcal{L}(\mathcal{A}_q)$ is finite. The sequence $\mathsf{Pre}^n_{\mathcal{A}}(\mathcal{F})$ is ultimately periodic, and for all $n \geq 0$, there exists $n_0 \leq 2^{|Q|}$ such that $\mathsf{Pre}^{n_0}_{\mathcal{A}}(\mathcal{F}) = \mathsf{Pre}^n_{\mathcal{A}}(\mathcal{F})$. Therefore, the finiteness problem can be solved in (N)PSPACE by guessing $n, k \leq 2^{|Q|}$ such that $\mathsf{Pre}^{n+k}_{\mathcal{A}}(\mathcal{F}) = \mathsf{Pre}^n_{\mathcal{A}}(\mathcal{F})$ and $q \in \mathsf{Pre}^n_{\mathcal{A}}(\mathcal{F})$. The finiteness problem is PSPACE-complete by a simple reduction from the emptiness problem: from an instance $(\mathcal{A}, q)$ of the emptiness problem, construct $(\mathcal{A}', q')$ where $q' = q$ and $\mathcal{A}' = \langle Q, \delta', \mathcal{F} \rangle$ is a copy of $\mathcal{A} = \langle Q, \delta, \mathcal{F} \rangle$ with a self-loop on q (formally, $\delta'(q) = q \vee \delta(q)$ and $\delta'(r) = \delta(r)$ for all $r \in Q \setminus \{q\}$). It is easy to see that $\mathcal{L}(\mathcal{A}_q) = \emptyset$ iff $\mathcal{L}(\mathcal{A}'_{q'})$ is finite.
- The *universal finiteness problem* is to decide, given a 1L-AFA $\mathcal{A}$, whether $\mathcal{L}(\mathcal{A}_q)$ is finite for all states q. This problem can be solved by checking whether $\mathsf{Pre}^n_{\mathcal{A}}(\mathcal{F}) = \emptyset$ for some $n \leq 2^{|Q|}$, and thus it is in PSPACE. Note that if $\mathsf{Pre}^n_{\mathcal{A}}(\mathcal{F}) = \emptyset$, then $\mathsf{Pre}^m_{\mathcal{A}}(\mathcal{F}) = \emptyset$ for all $m \geq n$.

Given the PSPACE-hardness proofs of the emptiness and finiteness problems, it is not easy to see that the universal finiteness problem is PSPACE-hard.

Lemma 3. *The universal finiteness problem for 1L-AFA is PSPACE-hard.*

Relation with MDPs. The underlying structure of a Markov decision process $\mathcal{M} = \langle Q, \mathsf{A}, \delta \rangle$ is an alternating graph, where the successor q' of a state q is obtained by an existential choice of an action a and a universal choice of a state $q' \in \mathsf{Supp}(\delta(q,a))$. Therefore, it is natural that some questions related to MDPs have a corresponding formulation in terms of alternating automata. We show that such connections exist between synchronizing problems for MDPs and language-theoretic questions for alternating automata, such as emptiness and universal finiteness. Given a 1L-AFA $\mathcal{A} = \langle Q, \delta_{\mathcal{A}}, \mathcal{F} \rangle$, assume without loss of generality that the transition function $\delta_{\mathcal{A}}$ is such that $\delta_{\mathcal{A}}(q) = c_1 \vee \cdots \vee c_m$ has the same number m of conjunctive clauses for all $q \in Q$. From $\mathcal{A}$, construct the MDP $\mathcal{M}_{\mathcal{A}} = \langle Q, \mathsf{A}, \delta_{\mathcal{M}} \rangle$ where $\mathsf{A} = \{a_1, \dots, a_m\}$ and $\delta_{\mathcal{M}}(q, a_k)$ is the uniform distribution over the states occurring in the k-th clause c_k in $\delta_{\mathcal{A}}(q)$, for all $q \in Q$ and $a_k \in \mathsf{A}$. Then, we have $Acc_{\mathcal{A}}(n, \mathcal{F}) = \mathsf{Pre}^n_{\mathcal{M}}(\mathcal{F})$ for all $n \geq 0$. Similarly, from an MDP $\mathcal{M}$ and a set T of states, we can construct a 1L-AFA $\mathcal{A} = \langle Q, \delta_{\mathcal{A}}, \mathcal{F} \rangle$ with $\mathcal{F} = T$ such that $Acc_{\mathcal{A}}(n, \mathcal{F}) = \mathsf{Pre}^n_{\mathcal{M}}(T)$ for all $n \geq 0$ (let $\delta_{\mathcal{A}}(q) = \bigvee_{a \in \mathsf{A}} \bigwedge_{q' \in \mathsf{post}(q,a)} q'$ for all $q \in Q$).

Several decision problems for 1L-AFA can be solved by computing the sequence $Acc_{\mathcal{A}}(n, \mathcal{F})$, and we show that some synchronizing problems for MDPs require the computation of the sequence $\mathsf{Pre}^n_{\mathcal{M}}(\mathcal{F})$. Therefore, the above relation between 1L-AFA and MDPs establishes bridges that we use in Section 4 to transfer complexity results from 1L-AFA to MDPs.

4 Eventually Synchronization

In this section, we show the PSPACE-completeness of the membership problem for eventually synchronizing objectives and the three winning modes. By the remarks at the end of Section 2, we consider the membership problem with function *sum* and Dirac initial distributions (i.e., single initial state).

4.1 Sure eventually synchronization

Given a target set T, the membership problem for sure-winning eventually synchronizing objective in T can be solved by computing the sequence $\mathsf{Pre}^n(T)$ of iterated predecessor. A state q_0 is sure-winning for eventually synchronizing in T if $q_0 \in \mathsf{Pre}^n(T)$ for some $n \geq 0$.

Lemma 4. *Let $\mathcal{M}$ be an MDP and T be a target set. For all states q_0, we have $q_0 \in \langle\!\langle 1 \rangle\!\rangle^{event}_{sure}(sum_T)$ if and only if there exists $n \geq 0$ such that $q_0 \in \mathsf{Pre}^n_{\mathcal{M}}(T)$.*

By Lemma 4, the membership problem for sure eventually synchronizing is equivalent to the emptiness problem of 1L-AFA, and thus PSPACE-complete (even when T is a singleton). Moreover if $q_0 \in \mathsf{Pre}^n_{\mathcal{M}}(T)$, a finite-memory strategy with n modes that at mode i in a state q plays an action a such that $\mathsf{post}(q,a) \subseteq \mathsf{Pre}^{i-1}(T)$ is sure winning for eventually synchronizing. There exists a family of MDPs $\mathcal{M}_n$ ($n \in \mathbb{N}$) that are sure winning for eventually synchronization, and where the sure winning strategies require exponential memory [12]. Essentially, the structure of $\mathcal{M}_n$ is an initial uniform probabilistic transition to n components $H_1, \dots, H_n$ where H_i is a cycle of length p_i the i-th prime number, and sure eventually synchronization requires memory size $p_n^{\#} = \prod_{i=1}^{n} p_i$. The following theorem summarizes the results for sure eventually synchronizing.

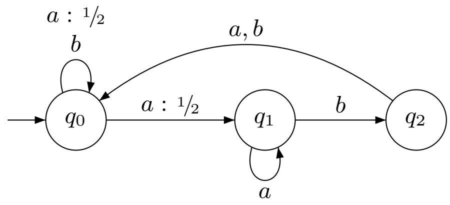

Fig. 2. An MDP where infinite memory is necessary for almost-sure eventually synchronizing strategies.

Theorem 2. *For sure eventually synchronizing in MDPs:*

1. *(Complexity). The membership problem is PSPACE-complete.*
2. *(Memory). Exponential memory is necessary and sufficient for both pure and randomized strategies, and pure strategies are sufficient.*

4.2 Almost-sure eventually synchronization

We show an example where infinite memory is necessary to win for almost-sure eventually synchronizing. Consider the MDP in Fig. 2 with initial state q_0. We construct a strategy that is almost-sure eventually synchronizing in $\{q_2\}$, showing that $q_0 \in \langle\!\langle 1 \rangle\!\rangle^{event}_{almost}(q_2)$. First, observe that for all $\epsilon > 0$ we can have probability at least $1 - \epsilon$ in q_2 after finitely many steps: playing n times a and then b leads to probability $1 - \frac{1}{2^n}$ in q_2. Thus the MDP is limit-sure eventually synchronizing in q_2. Moreover the remaining probability mass is in q_0. It turns out that from any (initial) distribution with support $\{q_0, q_2\}$, the MDP is again limit-sure eventually synchronizing in q_2 (and with support in $\{q_0, q_2\}$). Therefore we can take a smaller value of ϵ and play a strategy to have probability at least $1-\epsilon$ in q_2, and repeat this for $\epsilon \to 0$. This strategy ensures almost-sure eventually synchronizing in q_2. The next result shows that infinite memory is necessary for almost-sure winning in this example.

Lemma 5. *There exists an almost-sure eventually synchronizing MDP for which all almost-sure eventually synchronizing strategies require infinite memory.*

It turns out that in general, almost-sure eventually synchronizing strategies can be constructed from a family of limit-sure eventually synchronizing strategies if we can also ensure that the probability mass remains in the winning region (as in the MDP in Fig. 2). We present a characterization of the winning region for almost-sure winning based on an extension of the limit-sure eventually synchronizing objective *with exact support*. This objective requires to ensure probability arbitrarily close to 1 in the target set T, and moreover that after the same number of steps the support of the probability distribution is contained in a given set U. Formally, given an MDP $\mathcal{M}$, let $\langle\!\langle 1 \rangle\!\rangle^{event}_{limit}(sum_T, U)$ for $T \subseteq U$ be the set of all initial distributions such that for all $\epsilon > 0$ there exists a strategy α and

$n \in \mathbb{N}$ such that $\mathcal{M}_n^\alpha(T) \geq 1 - \epsilon$ and $\mathcal{M}_n^\alpha(U) = 1$. We say that α is limit-sure eventually synchronizing in T with support in U.

We will present an algorithmic solution to limit-sure eventually synchronizing objectives with exact support in Section 4.3. Our characterization of the winning region for almost-sure winning is as follows. There must exist a support U such that (i) the MDP is sure winning for eventually synchronizing in target U, and (ii) from distributions with support in U, it is possible to get probability arbitrarily close to 1 in T, and support back in U. In the example of Fig. 2 with $T = \{q_2\}$, we can take $U = \{q_0, q_2\}$.

Lemma 6. *Let $\mathcal{M}$ be an MDP and T be a target set. For all states q_0, we have $q_0 \in \langle\langle 1 \rangle\rangle^{event}_{almost}(sum_T)$ if and only if there exists a set U such that:*

- *$q_0 \in \langle\langle 1 \rangle\rangle^{event}_{sure}(sum_U)$, and*
- *$d_U \in \langle\langle 1 \rangle\rangle^{event}_{limit}(sum_T, U)$ where d_U is the uniform distribution over U.*

As we show in Section 4.3 that the membership problem for limit-sure eventually synchronizing with exact support can be solved in PSPACE, it follows from the characterization in Lemma 6 that the membership problem for almost-sure eventually synchronizing is in PSPACE, using the following (N)PSPACE algorithm: guess the set U, and check that $q_0 \in \langle\langle 1 \rangle\rangle^{event}_{sure}(sum_U)$, and that $d_U \in \langle\langle 1 \rangle\rangle^{event}_{limit}(sum_T, U)$ where d_U is the uniform distribution over U (both can be done in PSPACE by Theorem 2 and Theorem 4). We present a matching lower bound using a reduction from the membership problem for sure eventually synchronization [12], which is PSPACE-complete by Theorem 2.

Lemma 7. *The membership problem for $\langle\langle 1 \rangle\rangle^{event}_{almost}(sum_T)$ is PSPACE-hard even if T is a singleton.*

The results of this section are summarized as follows.

Theorem 3. *For almost-sure eventually synchronizing in MDPs:*

1. *(Complexity). The membership problem is PSPACE-complete.*
2. *(Memory). Infinite memory is necessary in general for both pure and randomized strategies, and pure strategies are sufficient.*

4.3 Limit-sure eventually synchronization

In this section, we present the algorithmic solution for limit-sure eventually synchronizing with exact support. Note that the limit-sure eventually synchronizing objective is a special case where the support is the state space of the MDP. Consider the MDP in Fig. 1 which is limit-sure eventually synchronizing in $\{q_2\}$, as shown in Lemma 2. For $i = 0, 1, \ldots$, the sequence $\mathsf{Pre}^i(T)$ of predecessors of $T = \{q_2\}$ is ultimately periodic: $\mathsf{Pre}^0(T) = \{q_2\}$, and $\mathsf{Pre}^i(T) = \{q_1\}$ for all $i \geq 1$. Given $\epsilon > 0$, a strategy to get probability $1 - \epsilon$ in q_2 first accumulates probability mass in the *periodic* subsequence of predecessors (here $\{q_1\}$), and when the probability mass is greater than $1-\epsilon$ in q_1, the strategy injects the probability mass in q_2 (through the aperiodic prefix of the sequence of predecessors). This is the typical shape of a limit-sure eventually synchronizing strategy. Note that in this scenario, the MDP is also limit-sure eventually synchronizing in every set $\mathsf{Pre}^i(T)$ of the sequence of predecessors. A special case is when it is possible to get probability 1 in the sequence of predecessors after finitely many steps. In this case, the probability mass injected in T is 1 and the MDP is even sure-winning. The algorithm for deciding limit-sure eventually synchronization relies on the above characterization, generalized in Lemma 8 to limit-sure eventually synchronizing with exact support, saying that limit-sure eventually synchronizing in T with support in U is equivalent to either limit-sure eventually synchronizing in $\mathsf{Pre}^k(T)$ with support in $\mathsf{Pre}^k(U)$ (for arbitrary k), or sure eventually synchronizing in T (and therefore also in U).

Lemma 8. *For all $T \subseteq U$ and all $k \geq 0$, we have $\langle\langle 1\rangle\rangle^{event}_{limit}(sum_T, U) = \langle\langle 1\rangle\rangle^{event}_{sure}(sum_T) \cup \langle\langle 1\rangle\rangle^{event}_{limit}(sum_R, Z)$ where $R = \mathsf{Pre}^k(T)$ and $Z = \mathsf{Pre}^k(U)$.*

Thanks to Lemma 8, since sure-winning is already solved in Section 4.1, it suffices to solve the limit-sure eventually synchronizing problem for target $R = \mathsf{Pre}^k(T)$ and support $Z = \mathsf{Pre}^k(U)$ with arbitrary k, instead of T and U. We can choose k such that both $\mathsf{Pre}^k(T)$ and $\mathsf{Pre}^k(U)$ lie in the periodic part of the sequence of pairs of predecessors $(\mathsf{Pre}^i(T), \mathsf{Pre}^i(U))$. We can assume that $k \leq 3^{|Q|}$ since $\mathsf{Pre}^i(T) \subseteq \mathsf{Pre}^i(U) \subseteq Q$ for all $i \geq 0$. For such value of k the limit-sure problem is conceptually simpler: once some probability is injected in $R = \mathsf{Pre}^k(T)$, it can loop through the sequence of predecessors and visit R infinitely often (every r steps, where $r \leq 3^{|Q|}$ is the period of the sequence of pairs of predecessors). It follows that if a strategy ensures with probability 1 that the set R can be reached by finite paths whose lengths are congruent modulo r, then the whole probability mass can indeed synchronously accumulate in R in the limit. Therefore, limit-sure eventually synchronizing in R reduces to standard limit-sure reachability with target set R and the additional requirement that the numbers of steps at which the target set is reached be congruent modulo r. In the case of limit-sure eventually synchronizing with support in Z, we also need to ensure that no mass of probability leaves the sequence $\mathsf{Pre}^i(Z)$. In a state $q \in \mathsf{Pre}^i(Z)$, we say that an action $a \in \mathsf{A}$ is *Z-safe* at position i if[3] $\mathsf{post}(q, a) \subseteq \mathsf{Pre}^{i-1}(Z)$. In states $q \notin \mathsf{Pre}^i(Z)$ there is no Z-safe action at position i.

To encode the above requirements, we construct an MDP $\mathcal{M}_Z \times [r]$ that allows only Z-safe actions to be played (and then mimics the original MDP), and tracks the position (modulo r) in the sequence of predecessors, thus simply decrementing the position on each transition since all successors of a state $q \in \mathsf{Pre}^i(Z)$ on a safe action are in $\mathsf{Pre}^{i-1}(Z)$. Formally, if $\mathcal{M} = \langle Q, \mathsf{A}, \delta\rangle$ then $\mathcal{M}_Z \times [r] = \langle Q', \mathsf{A}, \delta'\rangle$ where:

[3] Since $\mathsf{Pre}^r(Z) = Z$ and $\mathsf{Pre}^r(R) = R$, we assume a modular arithmetic for exponents of Pre, that is $\mathsf{Pre}^x(\cdot)$ is defined as $\mathsf{Pre}^{x \bmod r}(\cdot)$. For example $\mathsf{Pre}^{-1}(Z)$ is $\mathsf{Pre}^{r-1}(Z)$.

- $Q' = Q \times \{r-1, \ldots, 1, 0\} \cup \{\mathsf{sink}\}$; intuitively, we expect that $q \in \mathsf{Pre}^i(Z)$ in the reachable states $\langle q, i \rangle$ consisting of a state q of $\mathcal{M}$ and a *position* i in the predecessor sequence;
- δ' is defined as follows (assuming an arithmetic modulo r on positions) for all $\langle q, i \rangle \in Q'$ and $a \in \mathsf{A}$: if a is a Z-safe action in q at position i, then $\delta'(\langle q, i \rangle, a)(\langle q', i-1 \rangle) = \delta(q, a)(q')$, otherwise $\delta'(\langle q, i \rangle, a)(\mathsf{sink}) = 1$ (and sink is absorbing).

Note that the size of the MDP $\mathcal{M}_Z \times [r]$ is exponential in the size of $\mathcal{M}$ (since r is at most $3^{|Q|}$).

Lemma 9. *Let $\mathcal{M}$ be an MDP and $R \subseteq Z$ be two sets of states such that $\mathsf{Pre}^r(R) = R$ and $\mathsf{Pre}^r(Z) = Z$ where $r > 0$. Then a state q_0 is limit-sure eventually synchronizing in R with support in Z ($q_0 \in \langle\langle 1 \rangle\rangle^{event}_{limit}(sum_R, Z)$) if and only if there exists $0 \leq t < r$ such that $\langle q_0, t \rangle$ is limit-sure winning for the reachability objective $\Diamond(R \times \{0\})$ in the MDP $\mathcal{M}_Z \times [r]$.*

Since deciding limit-sure reachability is PTIME-complete, it follows from Lemma 9 that limit-sure synchronization (with exact support) can be decided in EXPTIME. We can show that the problem can be solved in PSPACE by exploiting the special structure of the exponential MDP in Lemma 9.

Lemma 10. *The membership problem for limit-sure eventually synchronization with exact support is in PSPACE.*

To establish the PSPACE-hardness for limit-sure eventually synchronizing in MDPs, we use a reduction from the universal finiteness problem for 1L-AFAs.

Lemma 11. *The membership problem for $\langle\langle 1 \rangle\rangle^{event}_{limit}(sum_T)$ is PSPACE-hard even if T is a singleton.*

The example in Fig. 2 can be used to show that the memory needed by a family of strategies to win limit-sure eventually synchronizing objective (in target $T = \{q_2\}$) is unbounded.

Theorem 4. *For limit-sure eventually synchronizing (with or without exact support) in MDPs:*

1. *(Complexity). The membership problem is PSPACE-complete.*
2. *(Memory). Unbounded memory is required for both pure and randomized strategies, and pure strategies are sufficient.*

Acknowledgment We are grateful to Winfried Just and German A. Enciso for helpful discussions on Boolean networks and for the gadget in the proof of Lemma 3.

References

1. M. Agrawal, S. Akshay, B. Genest, and P. S. Thiagarajan. Approximate verification of the symbolic dynamics of Markov chains. In *LICS*, pages 55–64. IEEE, 2012.
2. J. Aspnes and M. Herlihy. Fast randomized consensus using shared memory. *J. Algorithm*, 11(3):441–461, 1990.
3. C. Baier, N. Bertrand, and M. Größer. On decision problems for probabilistic Büchi automata. In *Proc. of FoSSaCS*, LNCS 4962, pages 287–301. Springer, 2008.
4. C. Baier, N. Bertrand, and P. Schnoebelen. On computing fixpoints in well-structured regular model checking, with applications to lossy channel systems. In *Proc. of LPAR*, volume 4246 of *LNCS*, pages 347–361. Springer, 2006.
5. R. Baldoni, F. Bonnet, A. Milani, and M. Raynal. On the solvability of anonymous partial grids exploration by mobile robots. In *Proc. of OPODIS*, LNCS 5401, pages 428–445. Springer, 2008.
6. R. Chadha, V. A. Korthikanti, M. Viswanathan, G. Agha, and Y. Kwon. Model checking MDPs with a unique compact invariant set of distributions. In *Proc. of QEST*, pages 121–130. IEEE Computer Society, 2011.
7. K. Chatterjee and T. A. Henzinger. A survey of stochastic ω-regular games. *J. Comput. Syst. Sci.*, 78(2):394–413, 2012.
8. L. de Alfaro. *Formal Verification of Probabilistic Systems.* PhD thesis, Stanford University, 1997.
9. L. de Alfaro, T. A. Henzinger, and O. Kupferman. Concurrent reachability games. *Theor. Comput. Sci.*, 386(3):188–217, 2007.
10. L. Doyen, T. Massart, and M. Shirmohammadi. Infinite synchronizing words for probabilistic automata. In *MFCS*, LNCS 6907, pages 278–289. Springer, 2011.
11. L. Doyen, T. Massart, and M. Shirmohammadi. Infinite synchronizing words for probabilistic automata (Erratum). *CoRR*, abs/1206.0995, 2012.
12. L. Doyen, T. Massart, and M. Shirmohammadi. Limit synchronization in Markov decision processes. *CoRR*, abs/1310.2935, 2013.
13. W. Fokkink and J. Pang. Variations on Itai-Rodeh leader election for anonymous rings and their analysis in PRISM. *Journal of Universal Computer Science*, 12(8):981–1006, 2006.
14. H. Gimbert and Y. Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *Proc. of ICALP (2)*, LNCS 6199, pages 527–538. Springer, 2010.
15. T. A. Henzinger, M. Mateescu, and V. Wolf. Sliding window abstraction for infinite Markov chains. In *Proc. of CAV*, LNCS 5643, pages 337–352. Springer, 2009.
16. M. Holzer. On emptiness and counting for alternating finite automata. In *Developments in Language Theory*, pages 88–97, 1995.
17. P. Jancar and Z. Sawa. A note on emptiness for alternating finite automata with a one-letter alphabet. *Inf. Process. Lett.*, 104(5):164–167, 2007.
18. V. A. Korthikanti, M. Viswanathan, G. Agha, and Y. Kwon. Reasoning about MDPs as transformers of probability distributions. In *Proc. of QEST*, pages 199–208. IEEE Computer Society, 2010.
19. M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of FOCS*, pages 327–338. IEEE Computer Society, 1985.
20. M. V. Volkov. Synchronizing automata and the Cerny conjecture. In *Proc. of LATA*, LNCS 5196, pages 11–27. Springer, 2008.