2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Satisfiability Bounds for ω-Regular Properties in Bounded-Parameter Markov Decision Processes

Maximilian Weininger* Tobias Meggendorfer* Jan Křetínský*
Department of Informatics, Technical University of Munich

***Abstract*— We consider the problem of computing minimum and maximum probabilities of satisfying an ω-regular property in a bounded-parameter Markov decision process (BMDP). BMDP arise from Markov decision processes (MDP) by allowing for uncertainty on the transition probabilities in the form of intervals where the actual probabilities are unknown. ω-regular languages form a large class of properties, expressible as, e.g., Rabin or parity automata, encompassing rich specifications such as linear temporal logic. In a BMDP the probability to satisfy the property depends on the unknown transitions probabilities as well as on the policy. In this paper, we compute the extreme values. This solves the problem specifically suggested by Dutreix and Coogan in CDC 2018, extending their results on interval Markov chains with no adversary. The main idea is to reinterpret their work as analysis of interval MDP and accordingly the BMDP problem as analysis of an ω-regular stochastic game, where a solution is provided. This method extends smoothly further to bounded-parameter stochastic games.**

I. Introduction

Markov decision processes (MDP) are a classical formalism encompassing both probabilistic and nondeterministic features: in each state some actions are enabled and each action is assigned a distribution over the successor states. In other words, each action corresponds to a set of transitions, each of which is assigned a transition probability. *Bounded-parameter MDP (BMDP)* [16] are like MDP, but to each transition is instead assigned an interval of possible transition probabilities. Thus each BMDP specifies a set of MDP. There are two interpretations of these intervals. Firstly, in the *uncertain* interpretation, the BMDP specifies MDP with unknown but constant transition probabilities in the intervals. An MDP is thus derived from the BMDP by picking a value in each interval once for all. Secondly, in the *adversarial* interpretation, the BMDP specifies a decision process where the transition probabilities may be different numbers (in the intervals) every time we come to a state. Each interpretation found its use. The former can model, for instance, various degrees of uncertainty for each action or confidence intervals for the transition probabilities learnt from experience. In this case there is one true transition probability, however unknown. The latter can be used as a formalism for abstracting MDP: states with different outgoing transition probabilities can be abstracted into a single state with an interval covering all the values [16]. In this case, the interval can stand for any of the values whenever visiting the state. As such BMDP extend interval Markov chains (IMC) [21], [23] by an adversary (or an underspecified/nondeterministic controller). The uncertain interpretation of IMC then yields uncertain Markov chains (UMC), while the adversarial interpretation of IMC yields interval MDP (IMDP), as distinguished in [33].

ω-regular languages, e.g. [36], form a robust class of rich specifications, which can be represented in various ways, e.g., by formulae of monadic second-order logic or by automata over infinite words. In the setting of probabilistic systems, it is often advantageous to use deterministic Rabin automata (DRA) or their variations. In particular, this class encompasses properties expressible in linear temporal logic (LTL) [27] and there are efficient ways of translating LTL to DRA [24]. Control of MDP with LTL specifications is widely studied, e.g. [35], [22], [38], [30], [40], and typically uses the DRA representation.

In [13], Dutreix and Coogan argue for computing minimum and maximum probabilities of satisfying an ω-regular property in an IMC interpreted as IMDP. In future work, they wish to apply the technique to solve the problem for BMDP, the controllable counterpart of IMC. In this paper, we re-interpret their technique in a different light and using that perspective give a solution to BMDP, in both the uncertain and the adversarial understanding of the intervals. We consider both the upper bound (also called *design choice* of values in intervals [13]) and the lower bound (*antagonistic* in [13]). We present the results for controllers that try to maximize the probability to satisfy the ω-regular property; minimization is analogous as ω-regular languages are closed under complement.

The main idea of our approach is to not only view the IMC as an IMDP, but also as an MDP, since an IMDP is a special case of MDP where the adversary chooses the transition probabilities. We show the standard analysis on the respective MDP coincides with the tailored algorithm of [13] applied to the IMC. Our main contribution is taking this perspective on BMDP, yielding a stochastic game. Since we can solve the stochastic game with an ω-regular objective we can obtain also the solutions. Moreover, for the upper bound, the two players play cooperatively and we can solve the problem in polynomial time using adapted MDP techniques. Finally, we show how the game extension of IMC with two competing agents can be solved analogously to BMDP, this time without the need of introducing an additional agent.

*All authors contributed equally to this work. It was funded in part by the German Research Foundation (DFG) project KR 4890/2-1 "Statistical Unbounded Verification".

Further related work: The general model of MDP with imprecise parameters (MDPIP) was introduced in [32]. BMDP [16] are then a subclass where the parametrization is limited to independent intervals. BMDP have been investigated with respect to various objectives, such as stochastic shortest path (minimum expected reward) [4], expected total reward [26], [39], [18], discounted reward [26], [15], LTL [37], PCTL [28], [19], or mean payoff [34].

The special non-controlled case of IMC has also been investigated for various objectives, e.g. PCTL [33], [5], LTL [3] ([2] observes this algorithm may not converge to the optimum) or ω-regular properties [9].

Recent improvements include importance sampling techniques for IMC [20] or topological policy iteration for BMDP [31]. IMC and BMDP are used as abstractions of systems in [25].

Organization of the paper: After recalling the used formalisms in Section II, we state our problem in Section III. We provide the solution in Section IV and an illustrative case study (adjusted from [13]) in Section V. Section VI concludes and presents ideas for future work.

II. Preliminaries

In this section, we recall basics of probabilistic systems and set up the notation. As usual, $\mathbb{N}$ refers to the natural numbers (including 0). A *probability distribution* on a finite set X is a mapping $p : X \to [0,1]$, such that $\sum_{x \in X} p(x) = 1$. We use $\mathcal{D}(X)$ to denote the set of all probability distributions on the set X. Given some set S, we use $S^\star$ and S^ω to denote the set of all finite and infinite sequences comprising elements of S, respectively.

A. Markov Decision Processes

Definition 1 *A* Markov decision process (MDP) *is a tuple $\mathcal{M} = (S, s_0, A, \mathsf{Av}, \Delta, Acc)$, where S is a finite set of* states*, $s_0 \in S$ is the* initial *state, A is a finite set of* actions*, $\mathsf{Av} : S \to 2^A \setminus \{\emptyset\}$ assigns to every state a non-empty set of* available actions*, $\Delta : S \times A \to \mathcal{D}(S)$ is a* transition function *that for each state s and (available) action $a \in \mathsf{Av}(s)$ yields a probability distribution over successor states, and $Acc \in 2^S \times 2^S$ is the* Rabin acceptance*. Furthermore, for ease of notation we assume w.l.o.g. that actions are unique for each state, i.e. $\mathsf{Av}(s) \cap \mathsf{Av}(s') = \emptyset$ for $s \neq s'$.*[1] *An element $(F_i, I_i) \in Acc$ is called* Rabin pair*. We assume w.l.o.g. that $F_i \cap I_i = \emptyset$ for all pairs.*

In figures, we denote a Rabin pair (F, I) by  .

An MDP with $|\mathsf{Av}(s)| = 1$ for all $s \in S$ is called *Markov chain (MC)*. For ease of notation, we overload functions that map to distributions $f : Y \to \mathcal{D}(X)$ by $f : Y \times X \to [0,1]$, where $f(y,x) := f(y)(x)$. For example, instead of $\Delta(s,a)(s')$ we write $\Delta(s,a,s')$ for the probability of transitioning from state s to s' using action a.

An *infinite path* in an MDP is an infinite sequence $\rho = s_0 a_0 s_1 a_1 \cdots$, such that $a_i \in \mathsf{Av}(s_i)$ and $\Delta(s_i, a_i, s_{i+1}) > 0$ for every $i \in \mathbb{N}$. We use ρ_i to refer to the i-th state s_i in a given path. A *finite path* is a finite prefix of an infinite path. $\mathrm{Inf}(\rho) \subseteq S$ denotes the set of all states which are visited infinitely often in the path ρ.

A path ρ is *accepted*, denoted $\rho \models Acc$, if an only if there exists a Rabin pair $(F_i, I_i) \in Acc$ such that all states in F_i are visited *finitely often*, i.e. $F_i \cap \mathrm{Inf}(\rho) = \emptyset$, and at least one state of I_i is visited *infinitely often*, i.e. $I_i \cap \mathrm{Inf}(\rho) \neq \emptyset$. We call such a Rabin pair *accepting for* ρ.

Remark 1 *Often, system and specification are modelled separately, e.g., by a labelled MDP together with a description of an ω-regular property such as an LTL formula [27] or an automaton. The common approach then is to build the product of the system, the BMDP, and an automaton describing the specification; this results in a system as described in Definition 1. Since our work is largely independent of this construction's details we omit this step for simplicity. We highlight that indeed [13] only refers to the product throughout the main body of their work. Details can be found in Section VII and, e.g., [1].*

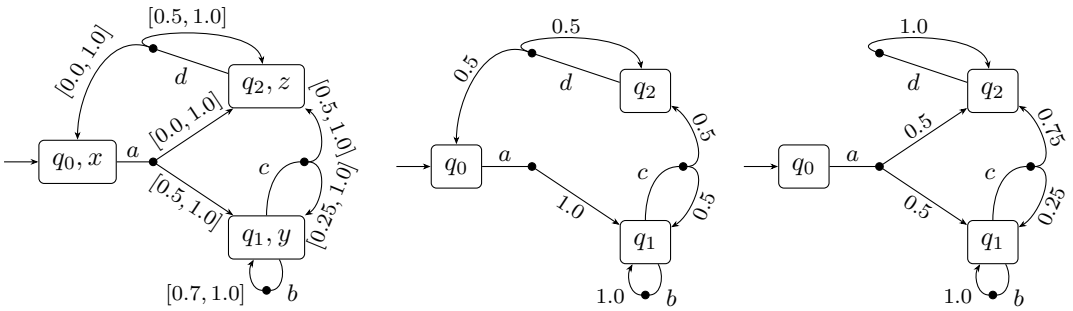
A *policy* (also called *controller*, *strategy*) on an MDP is a function $\pi : (S \times A)^* \times S \to \mathcal{D}(A)$ which given a finite path $\varrho = s_0 a_0 s_1 a_1 \dots s_n$ yields a probability distribution $\pi(\varrho) \in \mathcal{D}(\mathsf{Av}(s_n))$ on the actions to be taken next. We call a policy *memoryless randomized* (or *stationary*) if it is of the form $\pi : S \to \mathcal{D}(A)$, and *memoryless deterministic* (or *positional*) if it is of the form $\pi : S \to A$. Later in the paper, we prove that positional strategies are indeed sufficient for all considered problems. We denote the set of all policies of an MDP by Π. By fixing the policy π in an MDP $\mathcal{M}$, we naturally obtain a MC and thus a probability measure $\mathbb{P}^\pi_\mathcal{M}$ over potential runs [29]. Throughout this work, we are interested in finding policies which maximize the probability of accepting runs, i.e. $\sup_{\pi \in \Pi} \mathbb{P}^\pi_\mathcal{M}[\rho \models Acc]$.

An *end component* in an MDP is a pair (T, A) of a set of states T and a set of actions A such that the system *can* remain within the states T indefinitely, using only actions from A. Formally, a pair (T, A), where $\emptyset \neq T \subseteq S$ and $\emptyset \neq A \subseteq \bigcup_{s \in T} \mathsf{Av}(s)$, is an end component of an MDP $\mathcal{M}$ if (i) for all $s \in T, a \in A \cap \mathsf{Av}(s)$ we have $\{s' \mid \Delta(s,a,s') > 0\} \subseteq T$, and (ii) for all $s, s' \in T$ there is a finite path $\varrho = s a_0 \dots a_n s' \in (T \times A)^* \times T$, i.e. the path stays inside T and only uses actions in A. An end component (T, A) is a *maximal end component (MEC)* if there is no other end component (T', A') such that $T \subseteq T'$ and $A \subseteq A'$. The set of MECs of an MDP $\mathcal{M}$ is denoted by $\mathsf{MEC}(\mathcal{M})$ and can be obtained in polynomial time [12]. For further detail, see [1, Sec. 10.6.3].

B. Bounded-parameter Markov Decision Processes

Definition 2 *A* bounded-parameter Markov decision process (BMDP) *is a tuple $\mathfrak{M} = (S, s_0, A, \mathsf{Av}, \check{\Delta}, \hat{\Delta}, Acc)$, where S, s_0, A, Av and Acc are as in the definition of MDP, and $\check{\Delta}, \hat{\Delta} : S \times A \times S \to [0,1]$ are* lower and upper bounds *on the transition probability in each state. Again, we assume that actions are unique for each state.*

[1] The usual procedure to achieve this in general is to replace A by $S \times A$ and to adapt Av and Δ appropriately.



(a) Example BMDP (b) MDPs consistent with the BMDP

Fig. 1. Examples of an BMDP and its instantiations. The acceptance is $Acc = \{(q_2\ q_1)\}$.

For consistency, we require that $\sum_{s' \in S} \check{\Delta}(s,a,s') \leq 1 \leq \sum_{s' \in S} \hat{\Delta}(s,a,s')$ *for each state* s *and action* $a \in \mathsf{Av}(s)$.

Given a BMDP $\mathfrak{M} = (S, s_0, A, \mathsf{Av}, \check{\Delta}, \hat{\Delta}, Acc)$, we call a Markov decision process $\mathcal{M} = (S, s_0, A, \mathsf{Av}, \Delta, Acc)$ *consistent with* $\mathfrak{M}$, denoted $\mathcal{M} \in \mathfrak{M}$, if and only if $\mathcal{M}$'s transition probabilities satisfy $\mathfrak{M}$'s bounds, i.e. $\check{\Delta}(s,a,s') \leq \Delta(s,a,s') \leq \hat{\Delta}(s,a,s')$ for all states $s, s' \in S$ and actions $a \in \mathsf{Av}(s)$. Note that in general there are uncountably many MDPs consistent with a BMDP $\mathfrak{M}$. A BMDP with $|\mathsf{Av}(s)| = 1$ for all $s \in S$ is called *Interval Markov chain (IMC)* (e.g., [9]).

See Fig. 1a for an example BMDP and Fig. 1b for two MDPs consistent with this BMDP.

C. Stochastic Games

For our analysis, we additionally need the concept of *stochastic games*. These can be understood as an MDP where, instead of a single agent controlling the process, we have two antagonistic players. Intuitively, the first player's aim is to obtain an accepted path, while the second player aims to stop the first player from doing so. Each state in the game is "owned" by one of the two players and the owner of a particular state can decide which action to take in that state.

Definition 3 *A* stochastic game (SG) *is a tuple* $\mathcal{G} = (S, s_0, A, \mathsf{Av}, \Delta, Acc, \mathcal{O})$*, where* $(S, s_0, A, \mathsf{Av}, \Delta, Acc)$ *is an MDP and* $\mathcal{O} : S \to \{1, 2\}$ *is an* ownership function*, assigning each state to either player* 1 *or player* 2*. This naturally gives rise to the sets of states* S_1 *and* S_2*, which are controlled by the respective player.*

The definitions of (in)finite paths directly extend to stochastic games. Policies are slightly modified, since each player can only make decisions in a part of the game. Formally, we have two kinds of policies $\pi_1 : (S \times A)^* \times S_1 \to \mathcal{D}(A)$ and $\pi_2 : (S \times A)^* \times S_2 \to \mathcal{D}(A)$, one for each player.We denote the set of all policies of the respective players by Π_1 and Π_2. Fixing the policy of a single player yields an MDP, denoted $\mathcal{G}(\pi_i)$; fixing both players' policies π_1 and π_2 again yields an MC and measure over the set of runs, denoted $\mathbb{P}_{\mathcal{G}}^{\pi_1,\pi_2}$ [11].

III. Problem Statement

In this work, we are given a BMDP and want to control it such that the probability of an accepting run is maximized. This raises two orthogonal questions:

Firstly, the semantics of the interval constraints have to be fixed. We consider two different popular interpretations, called *uncertain* and *antagonistic*. In the *uncertain* (or *design-choice*) model, an external environment fixes the transition probabilities once and for all, i.e. for a BMDP $\mathfrak{M}$ a particular consistent MDP $\mathcal{M} \in \mathfrak{M}$ is chosen. In the *antagonistic* model, the external environment instead is allowed to change the transition probabilities at every step, taking into account the full path so far. These interpretations have been shown to be yield the same optima for reachability in interval Markov chains [10]. In the following, we show that this also is the case for BMDP with Rabin objectives; hence we do not distinguish the semantics in our formal problem statement.

Secondly, it is not specified whether the aforementioned environment is cooperative or antagonistic. We consider both of these two extreme cases. In particular, we are interested in finding the maximal probability of acceptance while assuming that all transition probabilities are chosen (i) to our liking and (ii) as bad as possible.

Formally, given a BMDP $\mathfrak{M}$ we want to compute

(i) $\hat{\mathbb{P}}(\mathfrak{M}) = \sup_{\pi \in \Pi} \sup_{\mathcal{M} \in \mathfrak{M}} \mathbb{P}_{\mathcal{M}}^{\pi}[\rho \models Acc]$, and

(ii) $\check{\mathbb{P}}(\mathfrak{M}) = \sup_{\pi \in \Pi} \inf_{\mathcal{M} \in \mathfrak{M}} \mathbb{P}_{\mathcal{M}}^{\pi}[\rho \models Acc]$.

Further, we are interested in the optimal policy and the best- / worst-case MDP consistent with the given BMDP, if it exists, i.e. the witnesses for the above values.

Case (i) can be understood as a "design challenge". We are interested in building our system, i.e. finding an optimal assignment for all transitions, such that we maximize the probability of acceptance. On the other hand, Case (ii) can be thought of as uncertainty about the real world or measurement imprecisions. Here, we rather are interested in optimizing the worst case and want to find a safe strategy, able to reasonably cope with any concrete instantiation of the intervals.

Consider the situation depicted in Fig. 1. We are interested in finding the upper and lower bounds for the BMDP given in Fig. 1a. The upper bound, 1.0, is exhibited by the left MDP of Fig. 1b. There, we end up in q_1 with probability 1 and, by always playing action b, get an accepting run with probability 1. The right MDP shows the lower bound of 0.5, since we get stuck in state q_2 with probability 0.5. Observe that if action d in state q_2 had a non-zero probability of moving to q_0,

the resulting run would be accepting with probability 1, since we would almost surely eventually reach q_1.

In the following, we re-interpret existing approaches, and present our unified approach for solving BMDP.

IV. Solution approach

In order to explain our approach, we first shed some light on the simpler case of interval Markov chains (IMC), handled in [13], and provide a different perspective on their approach.

In [13], the authors present a specialized algorithm for dealing with IMCs. In essence, they compute maximal sets of (non)accepting states and then obtain the final value by solving a reachability query for these states. We now provide a different viewpoint on their algorithm which will help us solve the more general case of BMDPs.

The key observation is the following. We can view the intervals in an IMC as a player (the external environment) picking the transition probabilities at every step. This can be understood as an MDP, where in every state the player has an action for each distribution satisfying the interval constraints. This MDP has uncountably many actions in general, as there are infinitely many possibilities to choose the probabilities. However, all possible distributions can be constructed as a convex combination of finitely many special cases which are *basic feasible solutions* (BFS) of a linear program [9], [17] (also known as *corner-point abstraction*). We can identify each of these special solutions and use them to construct a finite MDP. This MDP is often called *interval Markov decision process* (IMDP); not to be confused with BMDP. By model checking the IMDP, using established methods, we obtain the desired result for the IMC.

Indeed, the algorithm presented in [13, Sec. IV-C] actually can be interpreted as a symbolic adaption of the standard model checking procedure for Rabin objectives on the MDP, namely an adapted MEC decomposition together with a reachability query [1, Sec. 10.6.4]. Similar to the methods presented in [17], their specialized algorithm cleverly avoids explicit computation of the exponentially large IMDP, achieving polynomial runtime.

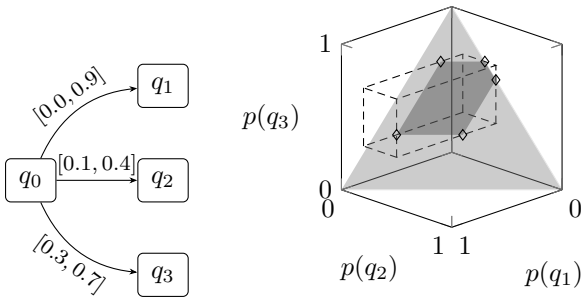
Before we can extend these ideas to BMDP, we explain some details of the basic feasible solutions, since they are essential to our idea.

A. Basic feasible solutions

Given an IMC $(S, s_0, \check{\Delta}, \hat{\Delta}, \lambda)$ and a state $s \in S$, we are interested in the set of all successor distributions $p \in \mathcal{D}(S)$ consistent with the constraints imposed by the IMC. In particular, the following constraints have to be satisfied:

(I) $\sum_{s' \in S} p(s') = 1$, and
(II) $\check{\Delta}(s, s') \leq p(s') \leq \hat{\Delta}(s, s')$ for all $s' \in S$.

Geometrically, Item I constrains the solution set to a plane, while Item II bounds it in a box, as shown in Fig. 2. Since each point in the solution corresponds to a valid transition distribution and vice versa, we identify solutions of the constraints with their corresponding distributions. Observe that the resulting solution set



(a) Example IMC (b) Visualized constraints.

Fig. 2. Visualization of the set of basic feasible solutions. The left picture shows a state together with its transition constraints in an IMC, the right picture depicts a geometric representation of the inequalities induced by the interval constraints. The light grey plane corresponds to the distribution constraints (Item I), while the dashed box represents the interval constraints (Item II). Finally, the dark grey area shows the set of consistent distributions and the diamonds mark the basic feasible solutions.

(and the corresponding set of distributions) is convex and any element of this set can be obtained by a convex combination of the corner-points, which are the basic feasible solutions.

There is one dimension per successor state and thus at most exponentially many basic feasible solutions. This set can be computed in exponential time using standard theory of linear programs or simple geometric computations. Due to lack of space we refer the reader to, e.g., [17, Sec. 4] for further detail.

B. Solving BMDP

When solving Rabin objectives on BMDPs, we observe one central problem: The set of end components depends on the choice of intervals. For example, consider the BMDP in Fig. 1a. This BMDP comprises only one MEC containing all three states. However, depending on the choice of probabilities, edges may be removed from the underlying graph, as for example in the right MDP of Fig. 1b. This MDP contains two MECs, namely $(\{q_1\}, \{b\})$ as well as $(\{q_2\}, \{d\})$.

In [13], a similar problem was encountered, as in IMC the set of *bottom strongly connected components* can be modified by the choice of intervals. The authors solved the problem implicitly by considering states where the Rabin objective is not satisfied as "leaky", i.e. not part of any strongly connected component.

In contrast to that, our general solution relies on making the possible choices of the probabilities explicit. We utilize the key observation that the non-determinism induced by intervals essentially corresponds to adding a player, who picks the probabilities for the intervals. Thus, we reduce the BMDP to a stochastic game that can be solved by known model checking methods. However, in the next section we introduce a more sophisticated approach inspired by the ideas of [13].

Intuitively, the reduction from BMDP to SG amounts to replacing every action in the BMDP with an additional state owned by the new player. In this state, the new player can choose from the corresponding basic feasible solutions, allowing him to construct any consistent

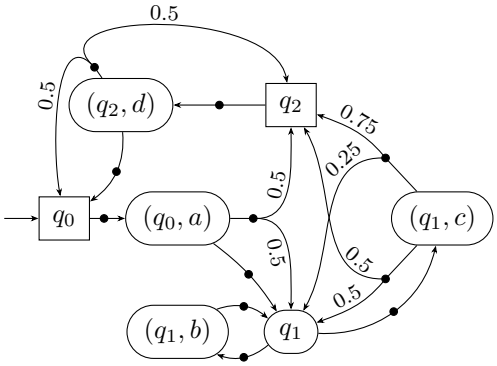


Fig. 3. The stochastic game obtained from the BMDP in Fig. 1a by the construction in the proof of Theorem 2. States owned by the system are depicted by rectangles, while environment states have rounded corners. For readability, we omit all action labels. Furthermore, we omit action nodes for probability 1 transitions.

distribution over the successors. An example of this construction is shown in Fig. 3, where the game constructed from the BMDP in Fig. 1a is depicted. This new player can play against the system (yielding $\check{\mathbb{P}}$) or cooperate (yielding $\widehat{\mathbb{P}}$).

Theorem 1 *For every BMDP $\mathfrak{M}$, there exists an SG $\mathcal{G}(\mathfrak{M})$ such that*

(i) $\widehat{\mathbb{P}}(\mathfrak{M}) = \sup_{\pi_1 \in \Pi_1} \sup_{\pi_2 \in \Pi_2} \mathbb{P}^{\pi_1,\pi_2}_{\mathcal{G}(\mathfrak{M})}[\rho \models Acc]$*, and*

(ii) $\check{\mathbb{P}}(\mathfrak{M}) = \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathbb{P}^{\pi_1,\pi_2}_{\mathcal{G}(\mathfrak{M})}[\rho \models Acc]$

Proof: Let $\mathfrak{M} = (S, s_0, A, \mathsf{Av}, \check{\Delta}, \widehat{\Delta}, \lambda)$ be a BMDP. In contrast to IMC, where we found the basic feasible solutions for a state, in BMDP we have to consider state-action pairs. Recall that we assumed that each action belongs to a single state. Hence, each $a \in A$ induces a set of basic feasible solutions, given by the constraints $\check{\Delta}(s,a)$ and $\widehat{\Delta}(s,a)$, where s is the unique state with $a \in \mathsf{Av}(s)$. We denote this set by $\mathsf{BFS}(a)$, and it can be computed as described in Section IV-A. Recall that any $p \in \mathsf{BFS}(a)$ corresponds to a distribution over states.

The SG $\mathcal{G}(\mathfrak{M}) = (S', s_0, A', \mathsf{Av}', \Delta, Acc, \mathcal{O})$ is constructed from $\mathfrak{M}$ as follows.

- $S' = S \cup \{(s,a) \mid s \in S \wedge a \in \mathsf{Av}(s)\}$,
- $A' = A \cup \{(a,p) \mid a \in A, p \in \mathsf{BFS}(a)\}$, and
- $\mathcal{O}(s) = 1$ if $s \in S$ and 2 otherwise, i.e. all newly introduced states (s,a) belong to the environment.
- For every old state $s \in S$ we set
  - $\mathsf{Av}'(s) = \mathsf{Av}(s)$ and
  - $\Delta(s, a, (s,a)) = 1$ (and 0 for all other states) for all actions $a \in \mathsf{Av}(s)$.
- For every new state $(s,a) \in S' \setminus S$ we set
  - $\mathsf{Av}'((s,a)) = \{(a,p) \mid p \in \mathsf{BFS}(a)\}$ and
  - $\Delta((s,a),(a,p),s') = p(s')$ for all $p \in \mathsf{BFS}(a)$.

For any consistent MDP $\mathcal{M} \in \mathfrak{M}$, there exists a policy for the other player π_2 inducing the transition function of $\mathcal{M}$, i.e. for all $\pi_1 \in \Pi_1$ we have $\mathbb{P}^{\pi_1}_{\mathcal{M}}[\rho \models Acc] = \mathbb{P}^{\pi_1,\pi_2}_{\mathcal{G}(\mathfrak{M})}[\rho \models Acc]$, and vice versa. This can be proven completely analogously to the proof for IMC, see, e.g., [9, Thm. 8]. Intuitively, randomizing over the BFS exactly yields the set of valid distributions. This immediately yields the desired equality. ∎

Corollary 1 *Positional policies suffice to achieve optimal solutions in BMDP. Consequently, the uncertain and antagonistic semantics are equivalent for the optima.*

Proof: Positional policies are sufficient for Rabin objectives in SG [8], and thus by the reduction of Theorem 1 also for BMDP. Hence the best policy for choosing the intervals is positional both for maximization and minimization, and there is no benefit in switching. ∎

Remark 2 *This result relies on the fact that the objective Acc is already part of the BMDP. See Appendix VII-C for further discussion.*

To solve ω-regular objectives for SGs, we use the strategy improvement algorithm presented in [7] and the reachability algorithm for MDPs from [1], which yields both the maximum and minimum probability, as well as the optimal controller. Given a BMDP $\mathfrak{M}$, our procedure works as follows:

1) Lower bound:
   a) Use [7, Alg. 2] to solve $\mathcal{G}(\mathfrak{M})$ with the given Rabin objective *Acc*. This yields the optimal player 1 policy π_1, which induces an MDP $\mathcal{M}_{\pi_1}$. Note that the actions of the system, the original non-determinism of the BMDP, are fixed in $\mathcal{M}_{\pi_1}$, and the remaining non-determinism belongs to the environment player choosing the intervals.
   b) Compute the minimum reachability probability with the algorithm from [1, Sec. 10.6.4], to obtain the the value $\check{\mathbb{P}}$ and the worst-case environment policy π_2 for $\mathcal{M}_{\pi_1}$, and thus the worst-case instantiation of the intervals.
2) Upper bound:
   a) Let $\mathcal{M}(\mathfrak{M})$ be the MDP obtained by assigning all nodes in $\mathcal{G}(\mathfrak{M})$ to player 1.
   b) Use standard model checking procedures to obtain the best policy π for $\mathcal{M}(\mathfrak{M})$ and the value $\widehat{\mathbb{P}}$; the policy handles both the non-determinism of the system and the instantiation of the intervals.

Theorem 2 *This procedure correctly computes*

1) $\check{\mathbb{P}}(\mathfrak{M})$, the optimal policy π_1 to achieve it and the worst-case MDP, and

2) $\widehat{\mathbb{P}}(\mathfrak{M})$, and π, describing the best-case system controller as well as the best choice of intervals.

It terminates in time exponential in the size of the $\mathcal{G}(\mathfrak{M})$, which in turn is exponential in the size of $\mathfrak{M}$.

Proof: The correctness follows directly from Theorem 1 and the correctness of the the used algorithms [7, Thm. 3], [1, Sec. 10.6.4]. The complexity is dominated by the computation of the game $\mathcal{G}(\mathfrak{M})$ and its solution process. The SG is exponential in the size of the BMDP (see Section IV-A) and the solution algorithm takes time exponential in the size the game [7, Thm. 3]. ∎

Remark 3 *We can immediately apply the presented methods to "bounded-parameter stochastic games", i.e. stochastic games with transition probability intervals. In*

particular, we do not need to introduce another player, as the states added by the construction in the proof of Theorem 1 can be assigned to one of the existing players – player 1 in the uncertain and player 2 in the antagonistic setting. Thus the system remains a stochastic game.

C. Improving computation of upper bounds

Note that for the computation of the upper bounds, we did not make use of the second player, but instead only introduced new states for the existing player, yielding an exponentially large MDP. By adapting observations from [13], we can improve on this algorithm by directly analysing the MC and avoiding the explicit construction of the SG, yielding a polynomial time algorithm for computing the upper bound.

Intuitively, the improved procedure symbolically identifies for each Rabin pair in $(F, I) \in Acc$ the winning end components. It does so by computing MECs while temporarily excluding states in F. After obtaining all states that are in a winning end-component for some pair, we compute the probability to reach any of these states. The structure of this improved procedure is similar to [13, Alg. 1], and relies on methods to compute the MEC decomposition on a BMDP from [17] and the reachability algorithm from [28].[2]

Given a BMDP $\mathfrak{M}$, our procedure works as follows:

1) For each Rabin pair $(F_i, I_i) \in Acc$:
   a) Construct a modified copy $\mathfrak{M}_i$ of $\mathfrak{M}$ where all (s, q) with $q \in F_i$ absorbing, i.e. all outgoing transitions are replaced with a self-loop.
   b) Compute the MEC decomposition of the resulting BMDP using [17, Alg. 3].
   c) If a MEC (T, A) has a non-empty intersection with I_i, then it is winning and all states T are added to the set of winning states W.
2) Compute the maximal probability of reaching W, using the methods presented in, e.g., [28].

Theorem 3 *This procedure correctly computes $\widehat{\mathbb{P}}(\mathfrak{M})$ and terminates in polynomial time.*

Proof: By [17, Prop. 6], we get that the MEC computation of a BMDP $\mathfrak{M}$ through [17, Alg. 3] is correct, i.e. the computed MECs correspond to the MECs of $\mathcal{G}(\mathfrak{M})$.The MECs identified as winning in Step 1b and 1c thus indeed are winning MECs in $\mathcal{M}(\mathfrak{M})$, where $\mathcal{M}(\mathfrak{M})$ is as in Step 2a of the procedure of Section IV-B. Consequently, we exactly identify the set of potentially winning states. Finally, the correctness of the algorithms used to compute the reachability in Step 2 yields the overall correctness.

For the complexity, observe that each step requires at most polynomial time ([17, Prop. 6] for Step 1b, [28, Thm. 4.1] for Step 2) and there are only linearly many Rabin pairs in Acc. ∎

Remark 4 *Note that in the setting of [13] a procedure for computing the upper bound is sufficient, as the lower bound can be computed as 1 minus the maximal probability to satisfy the negation of the specification. However, this idea is not applicable in the BMDP setting due to the alternation of* sup *and* inf *in the definition of $\check{\mathbb{P}}$.*

[2]In [17] and [28], the authors refer to BMDP as "IMDP".

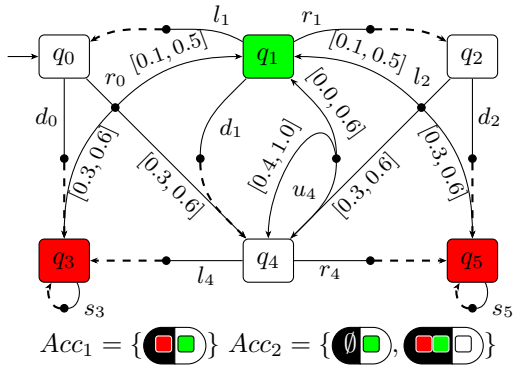


Fig. 4. Graphic representation of our extension of the case study from [13]. A robot navigates through a grid of six states according to the BMDP, using the actions **u**p, **d**own, **l**eft, and **r**ight, where enabled. For readability, all $[1.0, 1.0]$ constraints are omitted and the corresponding edges are drawn dashed. In our analysis, we consider two different acceptance conditions Acc_1 and Acc_2.

TABLE I
RESULTS FOR THE BMDP IN FIG. 4

	$\check{\mathbb{P}}$	$\widehat{\mathbb{P}}$
Acc_1	0.0	0.7
Acc_2	0.4	0.7

Remark 5 *Our algorithm can easily be extended to use* generalized Rabin transition acceptance*, allowing for efficient practical implementation. See [6] for details.*

V. CASE STUDY

We extend the case study from [13], where an agent moves on a coloured grid of six states. We added several actions in each state, modelling a robot which navigates the grid as depicted in Fig. 4. It can choose between going down, up, left or right; it cannot leave the grid, e.g., in q_0 only down and right are available. The robot is pulled towards the red states q_3 and q_5, e.g. when moving right from q_0 there is some chance to be pulled down onto q_4 or even q_3. The strength of the pulling force and hence also the probability distribution over the successor states is unknown and hence modelled by uncertainty intervals.

As a first example, we calculate the probability to reach state q_2 starting from q_0. From q_1 we can surely reach q_2 and from q_3 and q_5 there is no possibility of reaching q_2. From q_4, it depends on the instantiation of the BMDP. The probability to remain in q_4 may equal 1, preventing the robot from reaching q_2 when it is in q_4. On the other hand, we might have $\Delta(q_4, u_4, q_1) = 0.6$ and q_1 can almost surely be reached from q_4. Consequently, q_2 can be reached from q_4 with probability 1. Finally, the best strategy in q_0 is to go right, as otherwise the robot is immediately stuck in q_3. Together, this gives us a lower bound of 0.1 and an upper bound of 0.7.

Now we consider the properties from [13], adjusted to our setting as depicted in Fig. 4. The resulting probabilities are given in Table I and explained below.

Acc_1 corresponds to "The agent visits a green state infinitely often while visiting red states finitely often". In the antagonistic interpretation, the probability of satisfying this property is zero, since there is no MEC containing q_1. Intuitively, actions r_0, d_1 and l_2 all have a positive probability of moving to the bottom row, where we may be forced to remain forever. For the upper bound, observe that by setting $\Delta(q_4, u_4, q_1) = 0.6$ we obtain the winning MEC $(\{q_1, q_4\}, \{d_1, u_4\})$, which can be reached with probability 0.7.

Acc_2 corresponds to "The agent visits a red state infinitely often only if it visits a green state infinitely often". Observe that in this case both q_4 and q_1 are winning in any consistent MDP, as playing actions d_1 and u_4 always leads to a winning path. In contrast to Acc_1, remaining in q_4 is winning by the second pair of Acc_2, since only white states are visited. We thus get a lower and upper bound of 0.4 and 0.7, respectively, by computing the probability to reach q_1 or q_4.

VI. Conclusion

We have presented a solution to the open problem of bounding the probabilities to satisfy an ω-regular property on a bounded-parameter Markov systems. A different perspective on previous approaches enabled us to solve the problem by analysis of ω-regular stochastic games. Future work includes applications of our approach to more general settings such as MDPIP, as well as a practical implementation. For the latter, we plan to apply approaches based on learning and real-time dynamic programming, see e.g. [15].

References

[1] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

[2] Anicet Bart, Benoît Delahaye, Didier Lime, Eric Monfroy, and Charlotte Truchet. Reachability in parametric interval markov chains using constraints. In *QEST*, pages 173–189, 2017.

[3] Michael Benedikt, Rastislav Lenhardt, and James Worrell. LTL model checking of interval markov chains. In *TACAS*, pages 32–46, 2013.

[4] Olivier Buffet. Reachability analysis for uncertain ssps. In *ICTAI*, pages 515–522, 2005.

[5] Souymodip Chakraborty and Joost-Pieter Katoen. Model checking of open interval markov chains. In *ASMTA*, pages 30–42, 2015.

[6] Krishnendu Chatterjee, Andreas Gaiser, and Jan Kretínský. Automata with generalized rabin pairs for probabilistic model checking and LTL synthesis. In *CAV*, volume 8044 of *LNCS*, pages 559–575. Springer, 2013.

[7] Krishnendu Chatterjee and Thomas A. Henzinger. Strategy improvement for stochastic rabin and streett games. In *CONCUR*, volume 4137 of *LNCS*, pages 375–389. Springer, 2006.

[8] Krishnendu Chatterjee and Thomas A. Henzinger. A survey of stochastic ω-regular games. *J. Comput. Syst. Sci.*, 78(2):394–413, 2012.

[9] Krishnendu Chatterjee, Koushik Sen, and Thomas A. Henzinger. Model-checking omega-regular properties of interval markov chains. In *FOSSACS*, pages 302–317, 2008.

[10] Taolue Chen, Tingting Han, and Marta Z. Kwiatkowska. On the complexity of model checking interval-valued discrete time markov chains. *Inf. Process. Lett.*, 113(7):210–216, 2013.

[11] Anne Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992.

[12] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, July 1995.

[13] Maxence Dutreix and Samuel Coogan. Satisfiability bounds for co-regular properties in interval-valued markov chains. In *CDC*, pages 1047–1052, 2018.

[14] Javier Esparza, Jan Kretínský, and Salomon Sickert. One theorem to rule them all: A unified translation of LTL into ω-automata. In *LICS*, pages 384–393. ACM, 2018.

[15] Fernando L. Fussuma, Karina Valdivia Delgado, and Leliane Nunes de Barros. B^2rtdp: An efficient solution for bounded-parameter markov decision process. In *BRACIS*, pages 128–133, 2014.

[16] Robert Givan, Sonia M. Leach, and Thomas L. Dean. Bounded-parameter markov decision processes. *Artif. Intell.*, 122(1-2):71–109, 2000.

[17] Serge Haddad and Benjamin Monmege. Interval iteration algorithm for MDPs and IMDPs. *Theor. Comput. Sci.*, 735:111–131, 2018.

[18] Ernst Moritz Hahn, Vahid Hashemi, Holger Hermanns, Morteza Lahijanian, and Andrea Turrini. Multi-objective robust strategy synthesis for interval markov decision processes. In *QEST*, pages 207–223, 2017.

[19] Vahid Hashemi, Holger Hermanns, and Andrea Turrini. Compositional reasoning for interval markov decision processes. *CoRR*, abs/1607.08484, 2016.

[20] Cyrille Jégourel, Jingyi Wang, and Jun Sun. Importance sampling of interval markov chains. In *DSN*, pages 303–313, 2018.

[21] Bengt Jonsson and Kim Guldstrand Larsen. Specification and refinement of probabilistic processes. In *LICS*, pages 266–277, 1991.

[22] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Trans. Automat. Contr.*, 53(1):287–297, Feb 2008.

[23] Igor Kozine and Lev V. Utkin. Interval-valued finite markov chains. *Reliable Computing*, 8(2):97–113, 2002.

[24] Jan Kretínský, Tobias Meggendorfer, Salomon Sickert, and Christopher Ziegler. Rabinizer 4: From LTL to your favourite deterministic automaton. In *CAV (1)*, volume 10981 of *LNCS*, pages 567–577. Springer, 2018.

[25] Morteza Lahijanian, Sean B. Andersson, and Calin Belta. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Trans. Automat. Contr.*, 60(8):2031–2045, 2015.

[26] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[27] Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57. IEEE Computer Society, 1977.

[28] Alberto Puggelli, Wenchao Li, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In *CAV*, pages 527–542, 2013.

[29] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[30] Vasumathi Raman, Alexandre Donzé, Mehdi Maasoumy, Richard M. Murray, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. Model predictive control with signal temporal logic specifications. In *CDC*, pages 81–87. IEEE, 2014.

[31] Willy Arthur Silva Reis, Leliane Nunes de Barros, and Karina Valdivia Delgado. Robust topological policy iteration for infinite horizon bounded markov decision processes. *Int. J. Approx. Reasoning*, 105:287–304, 2019.

[32] Jay K. Satia and Roy E. Lave Jr. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740, 1973.

[33] Koushik Sen, Mahesh Viswanathan, and Gul Agha. Model-checking markov chains in the presence of uncertainties. In *TACAS*, pages 394–410, 2006.

[34] Ambuj Tewari and Peter L. Bartlett. Bounded parameter markov decision processes with average reward criterion. In *COLT*, pages 263–277, 2007.

[35] J. G. THISTLE and W. M. WONHAM. Control problems in a temporal logic framework. *International Journal of Control*, 44(4):943–976, 1986.

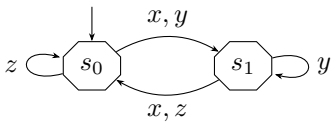


Fig. 5. Example DRA with acceptance $Acc = \{(\{s_0\}, \{s_1\}), (\{s_1\}, \{s_0\})\}$.

[36] Wolfgang Thomas. Languages, automata, and logic. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages, Volume 3: Beyond Words.*, pages 389–455. Springer, 1997.

[37] Eric M. Wolff, Ufuk Topcu, and Richard M. Murray. Robust control of uncertain markov decision processes with temporal logic specifications. In *CDC*, pages 3372–3379, 2012.

[38] Tichakorn Wongpiromsarn, Ufuk Topcu, and Richard M. Murray. Receding horizon control for temporal logic specifications. In *HSCC*, pages 101–110. ACM, 2010.

[39] Di Wu and Xenofon D. Koutsoukos. Reachability analysis of uncertain systems using bounded-parameter markov decision processes. *Artif. Intell.*, 172(8-9):945–954, 2008.

[40] B. Yordanov, J. Tumova, I. Cerna, J. Barnat, and C. Belta. Temporal logic control of discrete-time piecewise affine systems. *IEEE Transactions on Automatic Control*, 57(6):1491–1504, June 2012.

VII. Appendix – The product construction

In this section, we briefly explain how linear objectives usually are specified and how they are model checked using the product construction. First, we define the concept of *ω-regular languages* and *automata*. Fix a finite *alphabet* Σ. Elements of Σ^ω are called *words* and sets of words $\mathcal{L} \subseteq \Sigma^\omega$ are called *languages*. Such a language is ω-regular if it can be *recognized* by an automaton.

A. Automata & regular languages

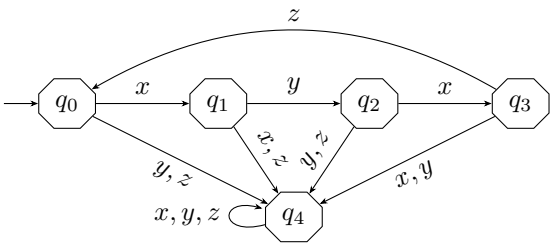
Definition 4 *A* deterministic Rabin automaton (DRA) *is a tuple $\mathcal{A} = (Q, T, q_0, Acc)$, where Q is a finite set of* states*, $T : Q \times \Sigma \to Q$ is a* transition function[3]*, $q_0 \in Q$ is an* initial state*, and $Acc \subseteq 2^Q \times 2^Q$ is the* Rabin condition*. An element $(F_i, I_i) \in Acc$ is called* Rabin pair*. We assume w.l.o.g. that $F_i \cap I_i = \emptyset$.*

Let $\mathcal{A}$ be a DRA and $w \in \Sigma^\omega$ a word. A word w induces a *run*, i.e. a sequence of states $\mathcal{A}(w) = q_0 q_1 q_2 \cdots \in Q^\omega$, where $q_{i+1} = T(q_i, w_i)$. As with MDP, let $\mathrm{Inf}(w)$ denote the set of states occurring *infinitely often* on the run $\mathcal{A}(w)$. A word is accepted by the automaton, denoted $w \models \mathcal{A}$, if there exists a Rabin pair $(F_i, I_i) \in Acc$ with $F_i \cap \mathrm{Inf}(w) = \emptyset$ and $I_i \cap \mathrm{Inf}(w) \neq \emptyset$. Such a Rabin pair is called *accepting for w*.

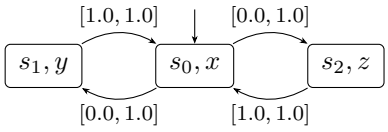
A language $\mathcal{L} \subseteq \Sigma^\omega$ is called *ω-regular* if and only if there exists a DRA $\mathcal{A}$ recognizing $\mathcal{L}$, i.e. some word $w \in \Sigma^\omega$ is accepted by the automaton if and only if it is in $\mathcal{L}$. See Fig. 5 for an example of a DRA recognizing the language "eventually only y or eventually only z".

Remark 6 *A wide variety of specifications are ω-regular. For example, reachability and liveness constraints can easily be translated to an automaton. Moreover, the whole of linear temporal logic is expressible through Rabin automata and efficient translations from LTL to Rabin automata exist [14].*

[3]Recall that the alphabet Σ is already fixed.



(a) Example DRA with acceptance $Acc = \{(\{q_2\}, \{q_0, q_1\})\}$



(b) Example IMC.

Fig. 6. Uncertain/adversarial interval interpretations are not equal.

B. Labelled MDPs & product

For the product construction, we modify the definition of MDPs by replacing the acceptance by a labelling function $\lambda : S \to \Sigma$, assigning to each state of the MDP a letter. We are given such a labelled MDP and a Rabin automaton. We construct the product by tracking both the evolution of the MDP and the automaton, where the automaton progresses based on the letter assigned to the current state.

Definition 5 *Let $\mathcal{M} = (S, s_0, A, \mathsf{Av}, \Delta, \lambda)$ be a labelled MDP and $\mathcal{A} = (Q, T, q_0, Acc)$ a Rabin automaton. The* product *$\mathcal{M} \otimes \mathcal{A} = (S \times Q, (s_0, q_0), A, \mathsf{Av}', \Delta', Acc')$ is an MDP where $\mathsf{Av}'((s, q)) := \mathsf{Av}(s)$, $\Delta((s, q), a, (s', q')) := \Delta(s, a, s')$ if $q' = T(q, \lambda(s))$ and 0 otherwise, and $Acc' = \{(F_i \times S, I_i \times S) \mid (F_i, I_i) \in Acc\}$.*

We analogously define this product construction for BMDP. Observe that the product is now of the form as we defined it in Definition 5. Applying our methods to this product yields a solution for the original system.

C. Caveat

Since this construction modifies the state space, it is not obvious how optimal policies on the product relate to policies on the original MDP. Indeed, while a memoryless policy may be optimal on the product, it might be the case that finite memory is needed to behave optimally in the given system. Moreover, it is the case that for ω-regular objectives, the optimal values for the uncertainty and the antagonistic interpretation are not equal already for IMCs.

Fix the alphabet $\Sigma = \{x, y, z\}$ and consider the language $\mathcal{L} = \{(xyxz)^\omega\}$, i.e. a language containing a single word w which repeats the string $xyxz$ indefinitely. This language is regular and is recognized by the automaton shown in Fig. 6a. Consider now the IMC in Fig. 6b. Clearly, the probability of satisfying the property is zero under the uncertainty interpretation – any MC consistent with the IMC eventually violates the structure of the property with probability 1. On the other hand, interpreted as an IMDP, the transitions can be chosen such that the required word is always produced.