

# Mixing Probabilistic and non-Probabilistic Objectives in Markov Decision Processes

Raphaël Berthon  
Université libre de Bruxelles  
University of Antwerp

Shibashis Guha  
Université libre de Bruxelles

Jean-François Raskin  
Université libre de Bruxelles

## Abstract

In this paper, we consider algorithms to decide the existence of strategies in MDPs for Boolean combinations of objectives. These objectives are omega-regular properties that need to be enforced either *surely*, *almost surely*, *existentially*, or with *non-zero probability*. In this setting, relevant strategies are *randomized infinite memory strategies*: both infinite memory and randomization may be needed to play optimally. We provide algorithms to solve the general case of Boolean combinations and we also investigate relevant subcases. We further report on complexity bounds for these problems.

**CCS Concepts:** • Mathematics of computing → Markov processes; • Theory of computation → Logic.

**Keywords:** Markov Decision Processes, synthesis, omega-regular

## ACM Reference Format:

Raphaël Berthon, Shibashis Guha, and Jean-François Raskin. 2020. Mixing Probabilistic and non-Probabilistic Objectives in Markov Decision Processes. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20)*, July 8–11, 2020, Saarbrücken, Germany. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3373718.3394805>

## 1 Introduction

Recently, there have been several works on how to mix the semantics of games and Markov decision processes [1, 5, 11, 12, 16]. This setting provides means to model the interaction between a system and its environment that is uncontrollable but obeys stochastic dynamics. The setting is then used to reason on strategies of the system that ensure for example some properties with *certainty* and others with *high probability*.

Here, we extend this line of work by studying a general setting where objectives for the system are Boolean combinations of atoms. These atoms are omega-regular properties, expressed as parity conditions, that need to be ensured either *surely* (A), *almost surely* (AS), *existentially* (E), or with *non-zero probability* (NZ). Sure (A) and existential (E) atoms are *non-probabilistic* while almost sure (AS) and non-zero (NZ) atoms are *probabilistic*. The coexistence of atoms of both types that need to be satisfied by a unique strategy makes this problem out of reach of classical techniques used to solve MDPs with CTL objectives or with PCTL objectives for example.

**Infinite memory and randomization.** In some previous works on models that mix games and MDPs [1, 5, 11, 12], infinite memory strategies are necessary to play optimally; randomization is not necessary. In [17], combination of parity conditions are studied. In that paper, randomization is necessary, but not infinite memory. In the setting that is considered in the current paper, relevant strategies for the systems are *randomized infinite memory strategies*: both infinite memory and randomization may be needed to play optimally. This implies that the techniques used here are more complex than for the previous works. Note that randomization is already necessary when considering conjunctions of two NZ atoms. The example we give above in Figure 1 is encompassed by the formalism of [17], and shows why we need to add randomization when compared to the work in [5]. In the MDP of Figure 1, there does not exist a deterministic choice from state  $s_0$  between action  $a$  and action  $b$  that ensures  $NZ(p_1) \wedge NZ(p_2)$ , while a randomized strategy can enforce this objective by taking  $a$  with probability  $\alpha$  ( $0 < \alpha < 1$ ) and taking  $b$  with probability  $1 - \alpha$ .

**Main contributions.** Our main contributions are summarized in Table 1. We provide a  $\Sigma_2^P$  algorithm to decide the existence of a strategy to enforce a Boolean combination of atomic objectives. We also show that this problem is both NP and coNP hard. Then we provide additional results for relevant subclasses of Boolean combinations. For the conjunctive case, we prove the existence of a polynomial algorithm that uses an NP oracle while the problem is shown to be coNP hard. For conjunctions that contain only one sure atom (1A) and a number of other atoms, the complexity goes down to  $NP \cap coNP$  and it is at least as hard as solving parity games. The complexity of this algorithm is dominated by

---

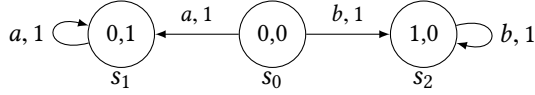
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

LICS '20, July 8–11, 2020, Saarbrücken, Germany

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7104-9/20/07...\$15.00

<https://doi.org/10.1145/3373718.3394805>



**Figure 1.** The MDP depicted here has 3 states and two parity functions  $p_1$  and  $p_2$ . The numbers assigned by the parity functions to the states are depicted by the integers inside the states. A parity condition is enforced if the maximum value of states that appear infinitely often is even. By taking  $a$  from  $s_0$ , state  $s_1$  is reached with probability one, and by taking  $b$ ,  $s_2$  is reached with probability one. Clearly on this example, a randomized strategy is needed to win for  $NZ(p_1) \wedge NZ(p_2)$  from state  $s_0$ .

	Hardness	Membership
$\wedge(AS, NZ, E)$	P	P (Thm. 7.2)
$\wedge(1A, AS, NZ, E)$	parity	$NP \cap coNP$ (Thm. 7.4)
$\wedge(A, AS, NZ, E)$	coNP	$P^{NP} (= \Delta_2^P)$ (Thm. 7.1)
$\mathbb{B}(A, AS, NZ, E)$	NP and coNP (Thm. 7.6)	$NP^{NP} (= \Sigma_2^P)$ (Thm. 7.5)

**Table 1.** Table of the main complexity results.

the complexity of solving parity games. A polynomial time solution to parity games would lead to a polynomial time solution for our problem. Also the recent quasi-polynomial time solutions for parity games, see e.g. [13], can be used to obtain a quasi-polynomial time solution to our problem. Finally, for conjunctions that do not contain sure atoms, the problem can be solved in polynomial time.

**Related works.** Logical formalisms to express properties of transition systems and Markov decision processes were plentifully studied in the literature. But most of the results in the literature only consider either logics based on *non-probabilistic* atoms, e.g. CTL, or logics based on *probabilistic* atoms only, e.g. PCTL. The logic PCTL is used to express constraints on the probability of events that are temporal properties of paths. In [10], the strategy synthesis problem for MDPs with PCTL objectives is studied. The full logic, i.e. with arbitrary probabilistic thresholds, is undecidable but the qualitative fragment of the logic (thresholds 0 and 1, corresponding to NZ and AS in our setting) is decidable in EXPTIME. This high complexity is due to the succinctness of PCTL. As PCTL cannot express our non-probabilistic atoms, the two formalisms have incomparable expressive power. Settings that mix both non-probabilistic properties, such as A or E, with probabilistic ones such as AS or NZ, are more recent. We make now a more detailed review of the recent relevant works in that direction.

In [11, 12], MDPs with mean-payoff and shortest path objectives are considered. This work was, to the best of our knowledge, the first work to consider the synthesis of strategies that optimize an *expectation* (a probabilistic property) while satisfying a long-run *worst-case objective* (a non-probability objective). Similarly, the authors of [1] consider the synthesis of strategies that ensure a parity condition *surely* and at the same time an  $\epsilon$ -optimal *expected* mean-payoff. Those works introduce refinements of the notion of *end-components* that we need to further refine here.

The authors of [19] study an extension of MSO, called  $MSO+\nabla$  which uses a probabilistic second order quantifier. The logic  $MSO+\nabla$  is expressive enough to encode the problem we study here, but this logic has been proved to be undecidable [3, 8]. In [6, 7], a fragment of  $MSO+\nabla$ , called Thin MSO has been introduced. The logic Thin MSO is expressive enough to encode the model-checking problem of the qualitative fragment  $CTL^* + PCTL^*$  (union of  $CTL^*$  and  $PCTL^*$ ) over Markov chains. Their algorithm has non-elementary complexity. The algorithm was recently improved in [18] where a model-checking algorithm with  $3NEXPTIME \cap co-3NEXPTIME$  complexity is proposed. The works in [6, 18] do not consider the richer model of Markov decision processes as we do here.

In [14], the authors study qualitative tree automata, that is automata with a probabilistic acceptance condition. The non-emptiness problem of nondeterministic tree automata with such acceptance condition has been proved decidable, but the problem has been proved undecidable for universal tree automata with such acceptance condition [3]. There is a deep connection between tree automata and Markov Decision Processes, as the existence of a strategy on an MDP corresponds to deciding the non-emptiness of a qualitative tree automaton with unary alphabet.

In [7, 20], the authors study *subzero* automata: a class of tree automata with an acceptance condition that mixes the classical Rabin acceptance condition with probabilistic constraints. The problem of determining if a subzero automaton accepts some regular tree is decidable. This class of automata can in turn be used to solve synthesis problem for *finite-memory* strategies (that are equivalent to regular trees) that enforce a first parity condition  $p_1$  surely (A) and a second parity condition  $p_2$  almost-surely (AS). Our work consider more general properties (both E and NZ in addition to A and AS, and their Boolean combinations) and more general strategies: randomized infinite memory strategies, and not only finite memory deterministic strategies (regular trees).

In this paper, we provide non-trivial extensions of results in [5] where only the case of one sure parity objective (1A) and one almost-sure parity objective (1AS) is considered. An  $NP \cap coNP$  algorithm is provided in [5] for this special case. In the current paper, in addition to a  $\Sigma_2^P$  algorithm for the general case  $\mathbb{B}(A, AS, NZ, E)$ , we also provide an algorithm that solves conjunctions of one sure parity objective (1A) and

any number of almost-sure (AS), existential (E), and non-zero probability (NP) parity objectives with the same worst-case complexity as that of [5]. Algorithms in [5] heavily rely on notions of very good end-components (VGEC) and ultra good end-components (UGEC). Here, we need generalization of VGEC and UGEC, and additional technical results to build algorithms for our more general setting.

Finally, the authors of [16] consider the synthesis of *finite-memory strategies* for MDPs with a sure parity (S) and an almost-sure parity (AS) objectives. The restriction to finite memory strategies leads to simpler algorithms but the complexity is similar, i.e.  $\text{NP} \cap \text{coNP}$ . The authors of [16] also consider the case of  $2\frac{1}{2}$ -player games. In that setting the problem is  $\text{coNP}$ -complete.

**Structure of the paper.** In Section 2, we introduce necessary preliminaries about MDPs, and we formally define the class of properties that we consider, i.e. Boolean combinations of A, AS, E, and NZ atoms. In Sections 3, 4 and 5, we study notions of end-components that are the main technical ingredients of our algorithms. Section 6 introduces additional techniques needed to handle E and NZ atoms. In Section 7, we study the complexity of algorithms for the general case, and several relevant fragments.

Full proofs are provided in [4].

## 2 Preliminaries

For  $k \in \mathbb{N}$ , we denote by  $[k]_0$  and  $[k]$  the set of natural numbers  $\{0, \dots, k\}$  and  $\{1, \dots, k\}$  respectively. Given a finite set  $A$ , a (rational) *probability distribution* over  $A$  is a function  $\text{Pr}: A \rightarrow [0, 1] \cap \mathbb{Q}$  such that  $\sum_{a \in A} \text{Pr}(a) = 1$ . We denote the set of probability distributions on  $A$  by  $\mathcal{D}(A)$ . The *support* of the probability distribution  $\text{Pr}$  on  $A$  is  $\text{Supp}(\text{Pr}) = \{a \in A \mid \text{Pr}(a) > 0\}$ .

**Markov chain.** We denote by  $\mathbb{N}$  the set  $\{1, 2, \dots\}$ , and by  $\mathbb{N}_0$  the set  $\mathbb{N} \cup \{0\}$ . A *Markov chain* (MC, for short) is a tuple  $\mathcal{M} = (S, E, \text{Pr})$ , where  $S$  is a set of states,  $E \subseteq S \times S$  is a set of edges (we assume in this paper that the set  $E(s)$  of outgoing edges from  $s$  is nonempty and finite for all  $s \in S$ ), and  $\text{Pr}: S \rightarrow \mathcal{D}(E)$  assigns a probability distribution – on the set  $E(s)$  of outgoing edges from  $s$  – to all states  $s \in S$ . In the following,  $\text{Pr}(s, (s, s'))$  is denoted  $\text{Pr}(s, s')$ , for all  $s \in S$ . The Markov chain  $\mathcal{M}$  is *finite* if  $S$  is finite.

For  $s \in S$ , the set of infinite paths in  $\mathcal{M}$  starting from  $s$  is  $\text{Paths}^{\mathcal{M}}(s) = \{\pi = s_0 s_1 \dots \in S^\omega \mid s_0 = s, \forall n \in \mathbb{N}_0, \text{Pr}(s_n, s_{n+1}) > 0\}$ . The set of all infinite paths in  $\mathcal{M}$  is  $\text{Paths}^{\mathcal{M}} = \bigcup_{s \in S} \text{Paths}^{\mathcal{M}}(s)$ . For  $\pi = s_0 s_1 \dots \in \text{Paths}^{\mathcal{M}}$ , we denote by  $\pi(i, l)$  the sequence of  $l - i + 1$  states (or  $l + 1$  edges)  $s_i \dots s_{i+l}$ , and for simplicity, we denote  $\pi(i, 0)$  by  $\pi(i)$ . The infinite suffix of  $\pi$  starting in  $s_n$  is denoted by  $\pi(n, \infty) \in \text{Paths}^{\mathcal{M}}$ . The set of finite paths starting from a state  $s \in S$  is defined as  $\text{Fpaths}^{\mathcal{M}}(s) = \{\pi = s \dots s' \in S^+ \mid \exists \bar{\pi} \in \text{Paths}^{\mathcal{M}}, \pi \bar{\pi} \in \text{Paths}^{\mathcal{M}}(s)\}$  and  $\text{Fpaths}^{\mathcal{M}} = \bigcup_{s \in S} \text{Fpaths}^{\mathcal{M}}(s)$ . For  $\pi = s \dots s'$ , we denote by  $\text{Last}(\pi)$ ,

the last state  $s'$  in  $\pi$ . As in [21], we extend the probability distribution to the space of infinite paths by considering cylinders defined by finite prefixes and using Carathéodory's extension theorem. We denote this probability distribution over the set of infinite paths beginning from some initial state  $s$  by  $\text{Pr}_{\mathcal{M}}^s$ . When  $s$  is clear from the context, we omit it and only denote this distribution by  $\text{Pr}_{\mathcal{M}}$ .

**Markov decision process.** A finite *Markov decision process* (MDP, for short) is a tuple  $\Gamma = (S, E, \text{Act}, \text{Pr})$ , where  $S$  is a finite set of states,  $\text{Act}$  is a finite set of actions, and  $E \subseteq S \times \text{Act} \times S$  is a set of edges, and  $\text{Pr}: S \times \text{Act} \rightarrow \mathcal{D}(E)$  is a partial function that assigns a probability distribution – on the set  $E(s, a)$  of outgoing edges from  $s$  – to all states  $s \in S$  if action  $a \in \text{Act}$  is taken from  $s$ . For all  $s \in S$  there exists at least one  $a \in \text{Act}$  such that  $E(s, a)$  is defined. Given  $s \in S$  and  $a \in \text{Act}$ , we define  $\text{Post}(s, a) = \{s' \in S \mid \text{Pr}(s, a, s') > 0\}$ . Then, for all state  $s \in S$ , we denote by  $\text{Act}(s)$  the set of actions  $\{a \in \text{Act} \mid \text{Post}(s, a) \neq \emptyset\}$ . We assume that, for all  $s \in S$ , we have  $\text{Act}(s) \neq \emptyset$ . Given an MDP  $\Gamma = (S, E, \text{Act}, \text{Pr})$ , and a set of states  $C \subseteq S$ , we define the restriction of  $\Gamma$  to  $C$ , denoted  $\Gamma \upharpoonright C$ , as the MDP  $(C, E', \text{Act}, \text{Pr}')$  where  $E' = \{(s, a, s') \mid s, s' \in C, a \in \text{Act}, \text{Post}(s, a) \subseteq C, \text{ and } (s, a, s') \in E\}$ , and  $\text{Pr}'$  is a partial function defined as  $\text{Pr}'(s, a) = \text{Pr}(s, a)$  if  $(s, a, s') \in E'$  for  $a \in \text{Act}$ , and  $s, s' \in C$ , and is undefined otherwise.

A *strategy* in  $\Gamma$  is a function  $\sigma: S^+ \rightarrow \mathcal{D}(\text{Act})$  such that for all  $s_0 \dots s_n \in S^+$ , we have  $\text{Supp}(\sigma(s_0 \dots s_n)) \subseteq \text{Act}(s_n)$ . A strategy  $\sigma$  can be encoded by a transition system  $\mathsf{T} = (Q, S, \text{act}, \delta, \iota)$  where  $Q$  is a (possibly infinite) set of states, called modes,  $\text{act}: Q \times S \rightarrow \mathcal{D}(\text{Act})$  selects a distribution on actions such that, for all  $q \in Q$  and  $s \in S$ , we have,  $\text{act}(q, s) \in \mathcal{D}(\text{Act}(s))$ . The function  $\delta: Q \times S \rightarrow Q$  is a mode update function and  $\iota: S \rightarrow Q$  selects an initial mode for each state  $s \in S$ . If the current state is  $s \in S$ , and the current mode is  $q \in Q$ , then the strategy chooses the distribution  $\text{act}(q, s)$ , and the next state  $s'$  is chosen according to the distribution  $\text{act}(q, s)$ . Formally,  $(Q, S, \text{act}, \delta, \iota)$  defines the strategy  $\sigma$  such that  $\sigma(\rho \cdot s) = \text{act}(\delta^*(\iota(\rho(0)), \rho), s)$  for all  $\rho \in S^*$ , and  $s \in S$ , where  $\delta^*$  extends  $\delta$  to sequence of states starting from  $\iota$  as expected, i.e.,  $\delta^*(\iota(\rho(0)), \rho \cdot s) = \delta(\delta^*(\iota(\rho(0)), \rho), s)$ , and  $\delta^*(\iota(\rho(0)), \varepsilon) = \iota(\rho(0))$ . We denote by  $\mathsf{T}_\sigma$  a transition system with minimal number of modes that corresponds to a strategy  $\sigma$ . A strategy is said to be *memoryless* if there exists a transition system encoding the strategy with  $|Q| = 1$ , that is, the choice of action only depends on the current state. A memoryless strategy can be seen as a function  $\sigma: S \rightarrow \mathcal{D}(\text{Act})$ . Formally, a strategy  $\sigma$  is memoryless if for all finite sequences of states  $\rho_1$  and  $\rho_2$  in  $S^+$  such that  $\text{Last}(\rho_1) = \text{Last}(\rho_2)$ , we have  $\sigma(\rho_1) = \sigma(\rho_2)$ . A strategy is called a *finite memory* strategy if there exists a transition system encoding the strategy in which  $Q$  is finite. A strategy is *deterministic* if  $\sigma: S^+ \rightarrow \text{Act}$ . For deterministic strategies, we have  $\text{act}: Q \times S \rightarrow \text{Act}$  such that for all  $q \in Q$  and  $s \in S$ ,

we have  $act(q, s) \in Act(s)$ . Note that the state space of  $\Gamma^{[\sigma]}$  is  $Q \times S$ . For a sequence  $\pi$  of states in  $\Gamma^{[\sigma]}$ , we denote by  $\text{proj}_S(\pi)$  the corresponding sequence of states in the MDP  $\Gamma$ . Once we fix a strategy  $\sigma$  encoded by the transition system  $(Q, S, act, \delta, \iota)$  in an MDP  $\Gamma = (S, E, Act, Pr)$ , we obtain an MC  $\Gamma^{[\sigma]} = (S', E', Pr')$ , where  $S' = Q \times S$  is the set of states,  $E' = \{(q \times s) \times (q' \times s') \mid q, q' \in Q, s, s' \in S, \delta(q, s) = q', \exists a \in Act, a \in \text{Supp}(act(q, s)) \text{ and } (s, a, s') \in E\}$  is the set of edges, and for  $q, q' \in Q, s, s' \in S$  we have the probability distribution  $Pr'(q, s)(q', s') = \sum_{a \in \text{Supp}(act(q, s))} act(q, s)(a) \cdot Pr(s, a, s')$  if  $q' = \delta(q, s)$  and is not defined otherwise. In the sequel, by abuse of notation, we write the projection onto the second component, that is  $s$  instead of  $(q, s)$ , for a state of this MC, unless specifically stated.

**One and two-player games.** For a given objective, an MDP  $\Gamma = (S, E, Act, Pr)$  can also be considered to have the semantics of a zero-sum two-player turn-based game where the game is played for infinitely many rounds and the exact probabilities are not important (this is the case when we will consider A and E atoms). The first round starts from a designated initial state  $s_{\text{init}} \in S$ . In each round, Player 1 chooses an action  $a \in Act(s)$  from a state  $s$  while Player 2 that is adversarial resolves the nondeterminism by choosing a state  $s'$  such that  $Pr(s, a, s') > 0$ . We denote by  $G_\Gamma = (S, E, Act)$  the two-player game that is obtained from an MDP  $\Gamma = (S, E, Act, Pr)$ . When the players resolve the nondeterminism co-operatively, we have a one-player game, which is the same as a nondeterministic automaton. Equivalently, in a one-player game, Player 1 chooses both action  $a$  as well as the state  $s'$ .

Given a target set  $T$ , we define the attractor of  $T$ , denoted  $\text{Attr}_1(T)$  as the set of states from which there exists a strategy for Player 1 to reach  $T$  with certainty. This corresponds to reachability in a classical “and-or” graph. For a two-player game, given  $T$ , an algorithm to obtain its attractor computes a sequence of sets of states  $(\text{Attr}_1^n(T))_{n \geq 0}$  defined as follows: (i)  $\text{Attr}_1^0(T) = T$ ; and (ii) for all  $n \geq 0$ :  $\text{Attr}_1^{n+1}(T) = \text{Attr}_1^n(T) \cup \{s \in S \mid \exists a \in Act, \text{Post}(s, a) \subseteq \text{Attr}_1^n(T)\}$ . Clearly  $\text{Attr}_1^{n+1}(T) \supseteq \text{Attr}_1^n(T)$ . If  $S$  is finite, then there exists an  $m \in \mathbb{N}_0$  such that  $\text{Attr}_1^n(T) = \text{Attr}_1^m(T)$  for all  $n \geq m$ . The algorithm for the case of one-player game only changes in the induction step where we have for all  $n \geq 0$ :  $\text{Attr}_1^{n+1}(T) = \text{Attr}_1^n(T) \cup \{s \in S \mid \exists a \in Act, \text{Post}(s, a) \cap \text{Attr}_1^n(T) \neq \emptyset\}$ . The algorithm for the one-player case corresponds to classical graph reachability.

We denote the size of an MC  $\mathcal{M}$ , MDP  $\Gamma$  and two-player game  $G$  by  $|\mathcal{M}|$ ,  $|\Gamma|$  and  $|G|$  respectively. For each case, the size is the sum of the number of states, the number of edges, and the size of the representation of the transition matrix, that is,  $|S| + |E| + |Pr|$ .

**Parity conditions and qualitative parity logic.** Given an MDP  $\Gamma$ , a parity condition is a function  $p: S \rightarrow \mathbb{N}_0$ . Given a path  $\pi \in S^\omega$ , the set  $\text{inf}(\pi) = \{s \in S \mid \forall i \geq 0, \exists j \geq$

$i, \text{ such that } \pi(j) = s\}$  is the set of states visited infinitely often on this path. A path satisfies a parity condition  $p$  if  $\max\{p(s) \mid s \in \text{inf}(\pi)\}$  is even. Given a parity condition  $p$ , its dual is the condition  $\bar{p}: s \mapsto 1 + p(s)$ . We denote by  $\text{parity}$  the set of parity conditions. A path satisfies  $\bar{p}$  iff it does not satisfy  $p$ . We now define qualitative parity logic (QPL) which is defined by the following grammar.

$$\begin{aligned} atom &= A(p) \mid E(p) \mid AS(p) \mid NZ(p) \quad (p \in \text{parity}) \\ \varphi &= atom \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \end{aligned}$$

Given an MDP  $\Gamma = (S, E, Act, Pr)$ , a state  $s \in S$ , and a parity condition  $p$ , for the atomic formulas, we say that  $s$  under strategy  $\sigma$

- *surely* satisfies  $p$ , denoted  $s, \sigma \models_\Gamma A(p)$ , iff  $\forall \pi \in \text{Paths}^{\Gamma^{[\sigma]}}(s)$ , we have that  $\pi$  satisfies  $p$ .
- *almost-surely* satisfies  $p$ , denoted  $s, \sigma \models_\Gamma AS(p)$ , iff  $\Pr_{\Gamma^{[\sigma]}}(\{\pi \in \text{Paths}^{\Gamma^{[\sigma]}}(s) : \pi \text{ satisfies } p\}) = 1$ .
- *satisfies  $p$  with non-zero probability*, denoted  $s, \sigma \models_\Gamma NZ(p)$ , iff  $\Pr_{\Gamma^{[\sigma]}}(\{\pi \in \text{Paths}^{\Gamma^{[\sigma]}}(s) : \pi \text{ satisfies } p\}) > 0$ .
- *existentially* satisfies  $p$ , denoted  $s, \sigma \models_\Gamma E(p)$ , iff  $\exists \pi \in \text{Paths}^{\Gamma^{[\sigma]}}(s)$ , such that  $\pi$  satisfies  $p$ .

Given two QPL formulas  $\varphi$  and  $\psi$ , and a strategy  $\sigma$  we define the semantics of Boolean connectives as follows:

- $s, \sigma \models_\Gamma \varphi \wedge \psi$  iff  $s, \sigma \models_\Gamma \varphi$  and  $s, \sigma \models_\Gamma \psi$
- $s, \sigma \models_\Gamma \varphi \vee \psi$  iff  $s, \sigma \models_\Gamma \varphi$  or  $s, \sigma \models_\Gamma \psi$
- $s, \sigma \models_\Gamma \neg A(p)$  iff  $s, \sigma \models_\Gamma E(\bar{p})$
- $s, \sigma \models_\Gamma \neg E(p)$  iff  $s, \sigma \models_\Gamma AS(\bar{p})$
- $s, \sigma \models_\Gamma \neg AS(p)$  iff  $s, \sigma \models_\Gamma NZ(\bar{p})$
- $s, \sigma \models_\Gamma \neg NZ(p)$  iff  $s, \sigma \models_\Gamma AS(\bar{p})$
- $s, \sigma \models_\Gamma \neg(\varphi \wedge \psi)$  iff  $s, \sigma \models_\Gamma \neg\varphi \vee \neg\psi$
- $s, \sigma \models_\Gamma \neg(\varphi \vee \psi)$  iff  $s, \sigma \models_\Gamma \neg\varphi \wedge \neg\psi$

Given a formula  $\varphi$ , we will use  $s \models_\Gamma \varphi$  to denote  $\exists \sigma : s, \sigma \models_\Gamma \varphi$ . Given a formula  $\varphi$ , let  $\llbracket \varphi \rrbracket \stackrel{\text{def}}{=} \{s \in S \mid s \models_\Gamma \varphi\}$ . We note that satisfying surely a parity condition is the same as winning the parity objective in the two-player game corresponding to the MDP  $\Gamma$ . Satisfying existentially is the same as finding a satisfying path in the nondeterministic automaton associated to this MDP.

Given an MDP  $\Gamma = (S, E, Act, Pr)$ , a state  $s \in S$ , and a QPL formula  $\varphi$ , the QPL-*synthesis problem* is to find a strategy  $\sigma$  such that  $s, \sigma \models_\Gamma \varphi$ . The QPL-*realizability problem* is to decide whether  $s \models_\Gamma \varphi$ . In what follows, we focus on the QPL-realizability problem, but the algorithms we provide give all the elements necessary to build a winning strategy when such a strategy exists, and so they can be easily extended to solve QPL-synthesis.

**Remark 2.1.** We define the negation of the formulas using classical De Morgan’s laws. We note that the logic QPL is closed under negation. It is also important to note that in this semantics,  $s \models_\Gamma \neg\varphi$  is not equivalent to  $s \not\models_\Gamma \varphi$ . Indeed,  $s \models_\Gamma \neg\varphi$  implies that there exists a strategy  $\sigma$  such that  $s, \sigma \models_\Gamma \neg\varphi$

whereas  $s \not\models_{\Gamma} \varphi$  implies that for all strategies  $\sigma$ , we have  $s, \sigma \models_{\Gamma} \neg\varphi$ .

Similarly, even though  $s \models_{\Gamma} \varphi \vee \psi$  is equivalent to  $s \models_{\Gamma} \varphi$  or  $s \models_{\Gamma} \psi$ , we note that  $s \models_{\Gamma} \varphi \wedge \psi$  is not the same as  $s \models_{\Gamma} \varphi$  and  $s \models_{\Gamma} \psi$ . Also using De Morgan's laws, the negation can be applied only to the parity objectives to get their duals, for example,  $\neg(A(p_1) \wedge A(p_2))$  is the same as  $E(\bar{p}_1) \vee E(\bar{p}_2)$ . We can indeed define a negation free normal form that can be obtained by taking the DNF and pushing negations down to the atoms. In the rest of the paper, we thus restrict our attention to the subclass of the logic that is free of negation and disjunction.

**Additional objectives.** We define the following additional objectives, introduced for technical reasons, even though they are not part of QPL<sup>1</sup>. A parity condition is called a *Büchi condition* if it is defined as  $p: S \rightarrow \{1, 2\}$ . A path  $\pi \in S^{\omega}$  satisfies a *conjunction of parity conditions*  $\bigwedge_{x \in X} p_x$  if for all  $x \in X$  we have  $\max\{p_x(s) \mid s \in \text{inf}(\pi)\}$  is even. It is not hard to see that conjunctions of parity conditions can be expressed as *Streitt conditions*. A path  $\pi$  satisfies a *reachability condition* towards a set  $R \subseteq S$ , denoted  $\diamond R$ , if there exists  $i \in \mathbb{N}_0$  such that  $\pi(i) \in R$ .

Given an MDP, we can define, in the same way as previously the sure, almost-sure, non-zero, and existential objectives for these conditions, as well as conjunctions and disjunctions of these objectives.

**End-components.** An *end-component* (EC, for short)  $M = (C, A)$  such that  $C \subseteq S$ , and  $A: C \rightarrow 2^{Act}$  is a *sub-MDP* of  $\Gamma$  (for all  $s \in C$ , we have  $A(s) \subseteq Act(s)$ , and for all  $a \in A(s)$ , we have  $\text{Post}(s, a) \subseteq C$ ) that is strongly connected. We denote by  $EC(\Gamma)$  the set of end-components of MDP  $\Gamma$ . By abuse of notation, in the sequel, we often refer to a set  $C \subseteq S$  to be an end-component when there exists a function  $A: C \rightarrow 2^{Act}$  such that  $(C, A)$  is an end-component. A *maximal EC* (MEC, for short) is an EC that is not included in any other EC. For every strategy in an MDP, the set of states seen infinitely often during a path form an end-component with probability 1. Formally:

**Proposition 2.2** ([2]). *Given an MDP  $\Gamma$ , for all strategies  $\sigma$ , for all states  $s$ , we have  $\Pr_{\Gamma|\sigma}(\{\pi \in \text{Paths}^{\Gamma|\sigma}(s) \mid \text{inf}(\pi) \in EC(\Gamma)\}) = 1$ .*

### 3 Type I end-components

In this section, we define Type I ECs that are a generalization of super-good end components as defined in [1].

Lemma 3.5 is the main result of the section, where we state that we can compute the set of maximal Type I ECs. This will be used later, to compute the set of maximal ECs of other kinds, namely Type II and Type III that are used in Sections 4 and 5 to solve satisfiability of formulas of the

<sup>1</sup>Given an MDP  $\Gamma$ , we could have expressed these conditions using QPL, but this would involve constructing a larger and more complex MDP from the given MDP  $\Gamma$ .

form  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$  and  $\bigwedge(\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$  respectively. Lemmas 3.2, 3.3 and 3.4 are technical lemmas that are required in the proof of Lemma 3.5. The proof of Lemma 3.2 uses the notion of Street-Büchi games.

Given two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$  and  $\{p_{as} \mid as \in \mathcal{AS}\}$ , an end-component  $C$  of  $\Gamma$  is Type I( $\mathcal{A}, \mathcal{AS}$ ) if the following property holds:

- $(I_1) \forall s \in C, s \models_{\Gamma|C} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(\diamond C_{\text{even}}^{\max}(p_{as}))$ ,

where

$$C_{\text{even}}^{\max}(p_{as}) = \{s \in C \mid (p_{as}(s) \text{ is even}) \wedge (\forall s' \in C,$$

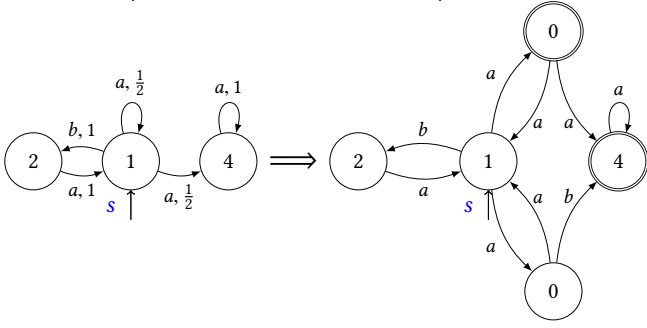
$$p_{as}(s') \text{ is odd} \implies p_{as}(s') < p_{as}(s))\}$$

contains the states with even priorities that are larger than any odd priority in  $C$  (this set can be empty for arbitrary ECs but needs to be non-empty for Type I( $\mathcal{A}, \mathcal{AS}$ ) ECs);

We write Type I( $\mathcal{A}, \mathcal{AS}$ ) EC as Type I EC when the parity sets are clear from the context. We introduce the following notations:  $\mathcal{EC}_1(\Gamma, \mathcal{A}, \mathcal{AS})$  is the set of all Type I( $\mathcal{A}, \mathcal{AS}$ ) ECs, and  $\mathcal{T}_{I, \Gamma, \mathcal{A}, \mathcal{AS}} = \bigcup_{U \in \mathcal{EC}_1(\Gamma, \mathcal{A}, \mathcal{AS})} U$  is the set of states belonging to some Type I EC. Given an EC  $C$ , we say a state  $s \in C$  is of Type I for  $C$  if  $C$  is Type I. In this paper, we only consider Type I( $\mathcal{A}, \mathcal{AS}$ ) ECs where  $\mathcal{AS}$  is either  $\mathcal{A}$  or  $\{a\}$ .

Intuitively, within a Type I EC, there is a strategy to visit all  $C_{\text{even}}^{\max}(p_{as})$  for all  $as \in \mathcal{AS}$  with probability 1 while guaranteeing  $A(p_a)$  for all  $a \in \mathcal{A}$ . We note that this property must hold while staying inside the end-component  $C$ . This notion strengthens the notion of *super-good end-component* (SGEC in [1]), that are defined for some parity condition  $p_a$ , and are Type I( $\{a\}, \{a\}$ ) ECs. In the case of SGEC, it has been shown in [1] that the existence of a strategy to enforce condition  $I_1$  in  $\Gamma$  can be reduced to checking the existence of a winning strategy in a game, constructed in polynomial time from  $\Gamma$ , with a conjunction of one parity objective and one Büchi objective. The existence of a winning strategy in such a game is in  $\text{NP} \cap \text{coNP}$ . The structure of this game is different from the one of the original MDP, as its size is polynomially increased to transform the qualitative reachability condition into a sure Büchi. In the sequel, we generalize this result to multiple parity conditions. We illustrate the reduction by the following example.

**Example 3.1.** Consider the example in Figure 2 where an MDP (on the left side of the figure) that is a Type I( $\{a\}, \{a\}$ ) EC for a parity condition  $p_a$  is transformed into a game (on the right side of the figure) that satisfies  $A(p_a) \wedge A(\square \diamond R)$ . In order to convert the  $AS(\diamond C_{\text{even}}^{\max}(p_a))$  condition of  $(I_1)$  into the  $A(\square \diamond R)$  condition, we add two states to the game: The top-most state and the bottom-most state. In the MDP on the left of Figure 1, a strategy that alternates between playing action  $a$  and playing action  $b$  at state  $s$  indeed satisfies the condition  $(I_1)$ .



**Figure 2.** An example of a Type I EC at left, and a game associated to it at right.

Now consider the game on the right side of Figure 2. The top-most state and the right-most state shown in double circles form the set  $R$ . A strategy to satisfy  $A(p_a) \wedge A(\Box\Diamond R)$  is as follows: When in state  $s$ , alternate between playing action  $a$  and playing action  $b$ . When in the bottom-most state, play action  $b$ . The  $A(p_a)$  atom is clearly satisfied. The sure Büchi  $A(\Box\Diamond R)$  holds for the following reason. When action  $a$  is chosen in state  $s$ , if player 2 chooses to go to the bottom-most state, then from this state player 1 plays action  $b$ , and reaches the right-most state that is absorbing and in  $R$ . If player 2 chooses to go to the top-most state always, as this state is in  $R$ , the Büchi condition is again satisfied.

We now state the first step of the reduction.

**Lemma 3.2.** *Given an MDP  $\Gamma = (S, E, Act, Pr)$ , a state  $s_0 \in S$ , a set of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$ , and a target set  $R \subseteq S$ , it can be decided if  $s_0 \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(\neg(\Diamond R) \rightarrow p_a) \wedge AS(\Diamond R)$ . If the answer is YES, then there exists a finite-memory witness strategy. This decision problem is coNP complete.*

This lemma relies on a reduction to a two-player game  $G_{R, \{p_a \mid a \in \mathcal{A}\}}^{\Gamma}$  with a conjunction of one Büchi and multiple parity conditions, that we call a Streett-Büchi game. A formal definition of this game is given in [4]. The approach is the same as in Lemma 3 of [1], that studies the case where  $\mathcal{A}$  is a singleton.

**Lemma 3.3.** *Given an MDP  $\Gamma = (S, E, Act, Pr)$ , a state  $s_0 \in S$ , a set of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$ , and a target set  $R \subseteq S$ , if for all  $s \in S$  it holds that  $s \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a)$  then we have that  $s_0 \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(\neg(\Diamond R) \rightarrow p_a) \wedge AS(\Diamond R)$  if and only if  $s_0 \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge AS(\Diamond R)$ .*

The proof of this lemma can be found in [4]. This lemma relates the  $\bigwedge_{a \in \mathcal{A}} A(\neg(\Diamond R) \rightarrow p_a) \wedge AS(\Diamond R)$  objective of Lemma 3.2 and the  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge AS(\Diamond R)$  objective of Type I  $(\mathcal{A}, \{a\})$  ECs under the condition that for all  $s \in S$  it holds that  $s \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a)$ . As  $I_1$  implies  $s \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a)$ , pruning states that do not satisfy  $\bigwedge_{a \in \mathcal{A}} A(p_a)$  before using Lemma 3.2 and Lemma 3.3

is always possible. Lemma 3.2 and Lemma 3.3 can only be used to compute Type I  $(\mathcal{A}, \{a\})$  ECs. We have the following lemma to relate Type I  $(\mathcal{A}, \{a\})$  ECs and Type I  $(\mathcal{A}, \mathcal{A})$  ECs.

**Lemma 3.4.** *In an EC  $C$ , for all  $s \in C$  we have that  $s \models_{\Gamma \downarrow C} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{a \in \mathcal{A}} AS(\Diamond_{\text{even}}^{\max}(p_a))$ , iff for all  $a_i \in \mathcal{A}$ , and for all  $s \in C$ , we have that  $s \models_{\Gamma \downarrow C} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge AS(\Diamond_{\text{even}}^{\max}(p_{a_i}))$ .*

Proof of Lemma 3.4 appears in [4]. We now state that we can compute the maximal Type I  $(\mathcal{A}, \mathcal{A})$  ECs.

**Lemma 3.5.** *Given an MDP  $\Gamma$ , it is possible to compute the set of maximal Type I  $(\mathcal{A}, \mathcal{A})$  ECs. This can be done by solving iteratively a number of Streett games that are polynomial in  $|\mathcal{A}|$  and  $|S|$ .*

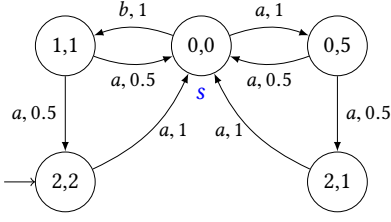
The proof of Lemma 3.5 and a detailed algorithmic procedure for computing the set of maximal Type I  $(\mathcal{A}, \mathcal{A})$  ECs can be found in [4]. In that procedure we iteratively compute the maximal Type I  $(\mathcal{A}, \{a_i\})$  ECs for all  $a_i \in \mathcal{A}$ . The combination of Lemma 3.2 and Lemma 3.3 is used for the computation of the set maximal Type I  $(\mathcal{A}, \{a_i\})$  ECs. Every time we do this computation, we prune all the states that do not belong to at least one of these ECs and solve Streett games again. We note that computing the maximal Type I  $(\mathcal{A}, \{a_i\})$  ECs follows an approach similar to the the procedure in [1] that computes the set of maximal SGENs. The difference is that we add an additional step in our algorithm, and use Lemma 3.4 to be able to combine the different  $\{p_{a_i}\}$ . We note that a naive generalization of the algorithm in [1] to compute the set of maximal Type I  $(\mathcal{A}, \mathcal{A})$  ECs results in an EXPTIME complexity, while we end up with a P<sup>NP</sup> complexity as we show later in Section 7.

## 4 Type II end-components

In this section, we define Type II ECs that are a generalization of UGEC defined in [5]. In the setting of the current paper, they are a generalization of Type I ECs, that have an additional condition. The main result of the section, Lemma 4.2, shows an equivalence between solving the realizability problem for formulas of the form  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{A}S} AS(p_{as})$  and solving the realizability problem for formulas involving sure parity conditions and almost-sure reachability of the Type II end-components. To do so, the two directions of the proof are done separately. For the right to left direction, first Lemma 4.4 shows that inside a Type II EC, there always exists a strategy  $\sigma$  for  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{A}S} AS(p_{as})$ . In Lemma 4.6, we use the strategy to reach a Type II EC and thereafter play  $\sigma$ , completing the proof of this direction.

For the other direction, we first show that for a strategy satisfying formula  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{A}S} AS(p_{as})$ , all the states that are visited under this strategy satisfy this formula. We then introduce the notion of density in Definition 4.8 to





**Figure 3.** An example of a Type II EC

relate in Lemma 4.9 the states satisfying formula  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$  to the Street-Büchi game of Section 3. Since Street-Büchi games are related to Type I ECs, and Type II ECs are extensions of Type I ECs, it then remains to prove that there exists at least one Type II EC in the MDP. This is done in Lemma 4.10.

Given two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$  and  $\{p_{as} \mid as \in \mathcal{AS}\}$ , an end-component  $C$  of  $\Gamma$  is Type II  $(\mathcal{A}, \mathcal{AS})$  if the following two properties hold:

- **(II<sub>1</sub>)**  $\forall s \in C, s \models_{\Gamma \downarrow C} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{a \in \mathcal{A}} AS(\diamond C_{\text{even}}^{\max}(p_a))$
- **(II<sub>2</sub>)**  $\forall s \in C, s \models_{\Gamma \downarrow C} \bigwedge_{a \in \mathcal{A}} AS(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$

We note that condition **(II<sub>1</sub>)** is exactly the one defining a Type I  $(\mathcal{A}, \mathcal{A})$  EC. We write Type II  $(\mathcal{A}, \mathcal{AS})$  EC as Type II EC when the parity sets are clear from the context. We introduce the following notations:  $\mathcal{EC}_{\text{II}}(\Gamma, \mathcal{A}, \mathcal{AS})$  is the set of all Type II  $(\mathcal{A}, \mathcal{AS})$  ECs, and  $\mathcal{T}_{\text{II}, \Gamma, \mathcal{A}, \mathcal{AS}} = \bigcup_{U \in \mathcal{EC}_{\text{II}}(\Gamma, \mathcal{A}, \mathcal{AS})} U$  is the set of states belonging to some Type II EC.

Intuitively, a Type II  $(\mathcal{A}, \mathcal{AS})$  EC, is a Type I  $(\mathcal{A}, \mathcal{A})$  EC where there also exists an additional strategy staying within the EC and almost-surely satisfying all parity conditions  $p_a$  and  $p_{as}$ . This notion generalizes the notion of a *ultra-good end-component* (UGEC in [5]) which is a Type II EC where both  $\mathcal{A}$  and  $\mathcal{AS}$  are singletons. In the sequel, we use the following notation:  $\bigwedge(\mathcal{A}, \mathcal{AS}) = \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$ .

Finding solutions for **(II<sub>2</sub>)** is done in [17], and it is shown in [5] how to use simple techniques from [2]. Winning strategies for **(II<sub>2</sub>)** may require either randomization or deterministic finite memory. We showed how to compute ECs satisfying **(II<sub>1</sub>)** (Type I ECs) in Lemma 3.5. In the sequel we relate Type II ECs to the formula  $\bigwedge(\mathcal{A}, \mathcal{AS})$ . In particular, from every state belonging to a Type II ECs there exists a strategy satisfying the formula  $\bigwedge(\mathcal{A}, \mathcal{AS})$ . We illustrate this by the following example.

**Example 4.1.** Consider the example in Figure 3. We show a strategy satisfying  $A(p_1) \wedge AS(p_2)$  in an MDP that is also a Type II EC. Indeed every state satisfies condition **(II<sub>1</sub>)** when action  $a$  is chosen from state  $s$  and every state satisfies condition **(II<sub>2</sub>)** when action  $b$  is chosen from state  $s$ .

Now the strategy from  $s$  to satisfy  $A(p_1) \wedge AS(p_2)$  is the following. In *round 1*, action  $b$  is chosen some  $i_0$  times. If the state with parity (2, 2) is not visited when action  $b$  is chosen  $i_0$  times, then action  $a$  is chosen until the state with parity (2, 1)

is visited. Once state (2, 1) is visited, we proceed to round 2 in which action  $b$  is chosen  $i_1 = i_0 + 1$  times. From every round, we proceed to the next round if either the state with parity (2, 2) is visited in the current round, or otherwise if the state with parity (2, 1) is visited and so on, resulting in action  $b$  being chosen  $i_j = i_0 + j$  times at round  $j$ . Now we compute the probability of not choosing action  $a$  from  $s$  during  $n$  rounds. The probability of not choosing  $a$  from  $s$  after the first round is  $1 - 2^{-i_0}$ . The probability of not choosing  $a$  from  $s$  after the first and the second round is  $(1 - 2^{-i_0}) \cdot (1 - 2^{-(i_0+1)})$ , and thus the probability of never choosing  $a$  when  $n$  rounds already happened is  $p(n) = \prod_{j=1}^n (1 - 2^{-(i_0+j)})$ . In [5], it has been shown that  $\lim_{n \rightarrow \infty} p(n) = 1$ , implying that with probability 1 action  $a$  will eventually stop being played.

The strategies for conditions **(II<sub>1</sub>)** and **(II<sub>2</sub>)** in the example above are deterministic memoryless strategies. However, in general, the strategy for **(II<sub>1</sub>)** may require finite memory and the strategy for **(II<sub>2</sub>)** may require memory or randomization. We illustrate this in the full version [4].

We now state the main result of this section.

**Lemma 4.2.** *Given an MDP  $\Gamma$ , and two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$  and  $\{p_{as} \mid as \in \mathcal{AS}\}$ , for all states  $s_0$ , we have  $s_0 \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$  iff  $s_0 \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge AS(\diamond \mathcal{T}_{\text{II}, \Gamma, \mathcal{A}, \mathcal{AS}})$ .*

*A strategy that enforces such conditions may require infinite memory.*

We show each direction separately. We fix some  $s_0 \in S$  and use it throughout this section. The necessity of using infinite memory is proved in Theorem 18 of [5], in the subcase where  $\mathcal{A}$  and  $\mathcal{AS}$  are singletons.

We start with the right-to-left implication. Since  $s_0 \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge AS(\diamond \mathcal{T}_{\text{II}, \Gamma, \mathcal{A}, \mathcal{AS}})$ , there exists a witness strategy  $\sigma_T$ . By definition, paths in  $\text{Paths}^{\Gamma[\sigma_T]}(s_0)$  eventually reaches some Type II EC  $C \in \mathcal{EC}_{\text{II}}(\Gamma, \mathcal{A}, \mathcal{AS})$  with probability one. Since  $C$  is a Type II EC, there exist two strategies,  $\sigma_1$  and  $\sigma_2$  respectively ensuring condition **(II<sub>1</sub>)** and **(II<sub>2</sub>)**. In what follows we define a strategy  $\sigma_C$  from  $\sigma_1$  and  $\sigma_2$  such that for all  $s \in C$  we have  $s, \sigma_C \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$ . Finally, we compute a strategy  $\sigma$  such that from  $s_0$ , we play  $\sigma_T$  until reaching a Type II EC  $C$ , where we play  $\sigma_C$ . This strategy  $\sigma$  satisfies  $\bigwedge(\mathcal{A}, \mathcal{AS})$  from  $s$ .

We first show how to construct the infinite memory strategy  $\sigma_C$  from  $\sigma_1$  and  $\sigma_2$ .

**Definition 4.3.** Let  $C \in \mathcal{EC}_{\text{II}}(\Gamma)$ . Let  $(n_i)_{i \in \mathbb{N}}$  be a sequence of naturals  $n_i$  such that for all  $i \in \mathbb{N}$  we have:

$$\Pr_{\Gamma \downarrow C^{[\sigma_2]}}[\{\rho \cdot \rho' \in \text{Paths}^{\Gamma \downarrow C^{[\sigma_2]}} \mid \rho \in \text{Fpaths}^{\Gamma \downarrow C^{[\sigma_2]}}\},$$

$$\forall a \in \mathcal{A}, \exists k_a < n_i, \rho(k_a) \in C_{\text{even}}^{\max}(p_a)\} \geq 1 - 2^{-i}$$

The strategy  $\sigma_C$  is defined as follows.

1. Play  $\sigma_2$  for  $n_i$  steps. Then  $i = i + 1$  and go to 2.

2. If for all  $a \in \mathcal{A} : C_{\text{even}}^{\max}(p_a)$  was visited in phase 1, then go to 1.  
Else, play  $\sigma_1$  until all  $C_{\text{even}}^{\max}(p_a)$  are reached, and then go to 1.

The following lemma states that for all  $s \in C$ , the strategy  $\sigma_C$  indeed satisfies  $\bigwedge(\mathcal{A}, \mathcal{AS})$ :

**Lemma 4.4.** *Let  $C \in \mathcal{EC}_{\text{II}}(\Gamma, \mathcal{A}, \mathcal{AS})$ . For all  $s \in C$ , it holds that  $s, \sigma_C \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$ .*

*Proof.* Let us first look at the  $\bigwedge_{a \in \mathcal{A}} A(p_a)$  condition. Each path  $\pi$  has to follow one of these three cases:

- Strategy  $\sigma_1$  is only played a finite number of times, and for a finite duration: This means that eventually for some  $i_0$ , in each round  $i > i_0$ , in episodes of  $n_i$  steps,  $C_{\text{even}}^{\max}(p_a)$  was visited for all  $a \in \mathcal{A}$ . This also means that eventually only  $\sigma_2$  is played and  $\pi$  stays in  $C$ , hence all  $p_a$  are satisfied on  $\pi$ .
- Strategy  $\sigma_1$  is eventually played for an infinite duration without coming back to **1**: By definition of  $\sigma_1$ , path  $\pi$  satisfies all  $p_a$ .
- Strategy  $\sigma_1$  and  $\sigma_2$  are both played infinitely often: The only way to stop strategy  $\sigma_1$  is to have visited all  $C_{\text{even}}^{\max}(p_a)$ . As  $\sigma_2$  and  $\sigma_1$  were both played infinitely often,  $\sigma_1$  was stopped infinitely often, and so  $C_{\text{even}}^{\max}(p_a)$  was visited infinitely often for all  $a \in \mathcal{A}$ . As  $\pi$  has to stay in  $C$ , it implies that  $\pi$  satisfies all  $p_a$ .

For the  $\bigwedge_{as \in \mathcal{AS}} A(p_{as})$  conditions, we can prove that with probability 1, eventually only  $\sigma_2$  is played. As  $\sigma_2$  has itself probability 1 of ensuring all  $p_{as}$ , we get that  $\sigma_C$  satisfies  $\bigwedge_{as \in \mathcal{AS}} A(p_{as})$ .  $\square$

Now we construct a strategy  $\sigma$  from  $\sigma_T$  and  $\sigma_C$ .

**Definition 4.5.** Based on strategies  $\sigma_T$  and  $\sigma_C$  for all  $C \in \mathcal{EC}_{\text{II}}(\Gamma)$ , we build the global strategy  $\sigma$  as follows.

1. Play  $\sigma_T$  until a Type II EC  $C$  is reached, then go to 2.
2. Play  $\sigma_C$  forever.

The following lemma concludes this direction of the proof of Lemma 4.2:

**Lemma 4.6.** *It holds that  $s_0, \sigma \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$ .*

Now we sketch the proof of the left-to-right implication of Lemma 4.2. We make use of the following lemma.

**Lemma 4.7.** *Given an MDP  $\Gamma$ , and two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$  and  $\{p_{as} \mid as \in \mathcal{AS}\}$ , for all states  $s$  and  $s'$ , and for all strategies  $\sigma$  the following holds: if  $s, \sigma \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$  and  $s' \notin \llbracket \bigwedge(\mathcal{A}, \mathcal{AS}) \rrbracket$ , then  $s' \notin \text{Paths}^{\Gamma[\sigma]}(s)$ .*

The proof follows from the fact that for a strategy  $\sigma$  in  $\Gamma$  that satisfies  $A(p)$  (resp.  $AS(p)$ ), for all finite paths  $\pi$  from  $s$  in  $\Gamma^{[\sigma]}$ , if  $\pi$  leads to a state  $\tilde{s}$ , then it holds that for the set of

paths originating from  $\tilde{s}$ , we have that  $A(p)$  (resp.  $AS(p)$ ) is satisfied. A formal proof can be found in [4]. We now introduce the following definition.

**Definition 4.8** (Density). Let  $\Gamma = (S, E, Act, Pr)$  be an MDP,  $s \in S$  an initial state,  $\sigma$  a strategy, and  $R \subseteq S$ . We say that  $R$  is *dense* in  $\sigma$  from  $s$  if and only if for all  $\rho \in \text{Fpaths}^{\Gamma[\sigma]}(s)$ , there exists  $\rho'$  such that  $\rho \cdot \rho' \in \text{Fpaths}^{\Gamma[\sigma]}(s)$  and  $\text{Last}(\rho') \in R$ . That is, after all prefixes in the tree  $\text{Paths}^{\Gamma[\sigma]}(s)$ , there is a continuation that visits  $R$ .

Now we state the following lemma that uses the above definition.

**Lemma 4.9.** *Given an MDP  $\Gamma = (S, E, Act, Pr)$ , a state  $s \in S$ , a set of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$ , a set  $R \subseteq S$ , if there exists a strategy  $\sigma$  such that  $s, \sigma \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a)$ , and  $R$  is dense in  $\sigma$  from  $s$ , then  $s \models_{G_{R, \{p_a \mid a \in \mathcal{A}\}}^{\Gamma}} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge A(\square \diamond B)$ , with  $G_{R, \{p_a \mid a \in \mathcal{A}\}}^{\Gamma}$  the Streett-Büchi game defined in Section 3 where the Büchi condition is  $B$ .*

Lemma 4.9 is proved in [4]. Lemma 4.7 implies that for all initial state  $s'$ , in  $\Gamma|_C^{[\sigma]}$ , after all finite path  $\rho$  beginning from some state  $(q', s')$ , and ending in  $(q'', s'')$  it holds that  $(q'', s''), \sigma \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$ . Thus after every finite path, there exists a continuation that visits  $\mathcal{T}_{\text{II}, \Gamma, \mathcal{A}, \mathcal{AS}}$ , hence  $\mathcal{T}_{\text{II}, \Gamma, \mathcal{A}, \mathcal{AS}}$  is dense from  $s'$  in  $\sigma$ , and so by Lemma 4.9, Lemma 3.2 and Lemma 3.3, we have that  $s_0 \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge AS(\diamond \mathcal{T}_{\text{II}, \Gamma, \mathcal{A}, \mathcal{AS}})$ .

Recall that  $s_0$  is the initial state in Lemma 4.2. We detail below why  $\mathcal{T}_{\text{II}, \Gamma, \mathcal{A}, \mathcal{AS}}$  is non-empty.

**Lemma 4.10.** *If  $s_0 \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$  then  $\mathcal{EC}_{\text{II}}(\Gamma) \neq \emptyset$ .*

The proof can be sketched as follows. Given an MDP  $\Gamma = (S, E, Act, Pr)$ , an initial state  $s$ , a strategy  $\sigma$  and a set of paths  $\Pi \subseteq \text{Paths}^{\Gamma[\sigma]}(s)$ , we define  $\text{States}(\Pi) = \{s \in S \mid \exists \pi \in \Pi, \exists n \in \mathbb{N}_0, \pi(n) = s\}$ . To prove this lemma, we first study the following set  $\mathcal{S}$  of subsets of  $S$ :

$$\mathcal{S} = \{R \subseteq S \mid \exists s \in S, \exists \sigma \text{ a strategy,}$$

$$(s, \sigma \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})) \wedge (R = \text{States}(\text{Paths}^{\Gamma[\sigma]}(s)))\}.$$

Intuitively, this set contains every subset of  $S$  that contains all states reachable by some witness strategy  $\sigma$  for  $\bigwedge(\mathcal{A}, \mathcal{AS})$ , from some state  $s \in S$ . First note that  $s_0 \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$  implies that  $\mathcal{S}$  is non-empty, as for a witness strategy  $\sigma$ , we have  $R = \text{States}(\text{Paths}^{\Gamma[\sigma]}(s_0)) \in \mathcal{S}$ , by definition.

Second, we show that all minimal elements of  $\mathcal{S}$  under set inclusion  $\subseteq$  are Type II ECs, i.e., for all  $R \in \min_{\subseteq}(\mathcal{S})$ , it holds that  $R \in \mathcal{EC}_{\text{II}}(\Gamma)$ . The details can be found in [4].

Finally, we state the following lemma. It refines Lemma 4.10 by showing that some Type II EC of  $\mathcal{EC}_{\text{II}}(\Gamma)$  can be reached almost-surely while satisfying  $\bigwedge_{a \in \mathcal{A}} (A(p_a))$ . We define

$$\mathcal{T}_{\text{II}, \Gamma, \mathcal{A}, \mathcal{AS}}^{\min} = \bigcup_{R \in \min_{\subseteq}(\mathcal{S})} R, \text{ that is the set of all states that belong to a minimal set } R \text{ of } \mathcal{S}.$$



**Lemma 4.11.** *If  $s_0 \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$  then  $s_0 \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} (A(p_a) \wedge AS(\diamond \mathcal{T}_{II, \Gamma, \mathcal{A}, \mathcal{AS}}^{\min}))$ .*

The details of the proof can be found in [4].

We end this section with the following observation. Lemma 4.7 implies that winning strategies only visit states belonging to  $\llbracket \bigwedge(\mathcal{A}, \mathcal{AS}) \rrbracket$ . As a consequence, pruning the states that do not satisfy  $\bigwedge(\mathcal{A}, \mathcal{AS})$  does not affect correctness. We use this pruned MDP in the rest of the paper. We state this formally:

**Assumption 4.12.** *For all states  $s$  of  $\Gamma$ , we have:  $s \in \llbracket \bigwedge(\mathcal{A}, \mathcal{AS}) \rrbracket$ .*

We detail how to do this pruning in Section 7, we use Lemma 4.2 to find the states that do not satisfy the objective.

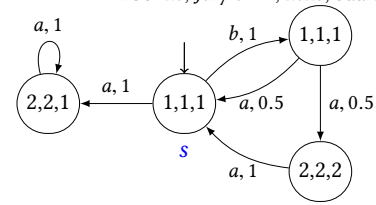
## 5 Type III end-components

In this section, we define Type III ECs. These end-components are used to characterize the winning strategies for formulas of the form  $\bigwedge(\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$ . To do so, we show in Lemma 5.3 that all Type III ECs satisfy such a formula. In Lemma 5.5, we use the previous lemma and the technical Lemma 5.4 to relate the formula  $\bigwedge(\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$  to the almost-sure reachability of the set of Type III ECs under the constraint  $\bigwedge(\mathcal{A}, \mathcal{AS})$ . We explain in Section 6 how to compute reachability of a set of states under the constraint  $\bigwedge(\mathcal{A}, \mathcal{AS})$ , and in Section 7 we explain how to compute the set of Type III ECs.

Given two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$ ,  $\{p_{as} \mid as \in \mathcal{AS}\}$  and another parity condition  $p_{nz}$ , an end-component  $C$  of  $\Gamma$  is Type III  $(\mathcal{A}, \mathcal{AS}, p_{nz})$  if the following two properties hold:

- **(III<sub>1</sub>)**  $\forall s \in C, s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$ ;
- **(III<sub>2</sub>)**  $\forall s \in C, s \models_{\Gamma|C} \bigwedge_{a \in \mathcal{A}} AS(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge AS(p_{nz})$

We write Type III  $(\mathcal{A}, \mathcal{AS}, p_{nz})$  EC as Type III ECs when the parity sets are clear from the context. We note that condition **(III<sub>1</sub>)** may require an infinite memory strategy (see Lemma 4.2), and can always be satisfied in the pruned MDP, due to Assumption 4.12. Condition **(III<sub>1</sub>)** can be checked using [17]. Note that condition **(III<sub>2</sub>)** is only about the parity conditions indexed by  $\mathcal{A}$  and  $\mathcal{AS}$ , and must hold while staying inside the end-component  $C$ , but the witness strategy for **(III<sub>1</sub>)** may leave  $C$ . This notion strengthens in a non-trivial way the notion of *very-good end-component* (VGEC in [5]) which are Type III ECs where  $\mathcal{A}$  is a singleton,  $\mathcal{AS} = \emptyset$ , and the  $p_{nz}$  stays as it is. From condition **(III<sub>1</sub>)** and Lemma 4.2 we know that if there exists a Type III EC in the MDP  $\Gamma$  then there also exists a Type II EC. We introduce the following notations:  $\mathcal{EC}_{III}(\Gamma, \mathcal{A}, \mathcal{AS}, p_{nz})$  is the set of all Type III  $(\mathcal{A}, \mathcal{AS}, p_{nz})$  ECs of  $\Gamma$ , and  $\mathcal{T}_{III, \mathcal{A}, \mathcal{AS}, p_{nz}} = \bigcup_{C \in \mathcal{EC}_{III}(\Gamma, \mathcal{A}, \mathcal{AS}, p_{nz})} C$  is the set of states belonging to some Type III EC in  $\Gamma$ .



**Figure 4.** An example of a Type III EC

In the sequel, we relate Type III ECs to the formula  $\bigwedge(\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$ . In particular, from all states belonging to a Type III ECs there exists a strategy satisfying the formula  $\bigwedge(\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$ .

We illustrate this by the following example.

**Example 5.1.** Consider the example in Figure 4. We show a strategy satisfying  $A(p_1) \wedge AS(p_2) \wedge NZ(p_3)$  in an MDP that is also a Type III EC: All the states satisfy condition **(III<sub>1</sub>)** when action  $a$  is chosen from state  $s$ , and all the states satisfy condition **(III<sub>2</sub>)** when action  $b$  is chosen from state  $s$ .

Now the strategy from  $s$  to satisfy  $A(p_1) \wedge AS(p_2) \wedge NZ(p_3)$  is the following. In *round 1*, action  $b$  is chosen some  $i_0$  times. If the state with parity value  $(2, 2, 2)$  is visited in round 1, then proceed to round 2. Otherwise action  $a$  is chosen from state  $s$  at the end of round 1. In round 2, action  $b$  is chosen  $i_1 = i_0 + 1$  times and so on, resulting in action  $b$  being chosen  $i_j = i_0 + j$  times in round  $j$ . From every round, we proceed to the next round if the state with parity value  $(2, 2, 2)$  is visited in the current round, or otherwise we switch to playing action  $a$  from state  $s$ . Now we compute the probability of never switching to action  $a$  in state  $s$ . The probability of not choosing action  $a$  from  $s$  after the first round is  $(1 - 2^{-i_0})$ . The probability of not choosing  $a$  from  $s$  after the first round and as well as after the second round is  $(1 - 2^{-i_0}) \cdot (1 - 2^{-(i_0+1)})$ , and thus the probability of not choosing  $a$  from  $s$  after each of the first  $n$  rounds is  $p(n) = \prod_{j=0}^{n-1} (1 - 2^{-(i_0+j)})$ . In [5], it has been shown that  $\lim_{n \rightarrow \infty} p(n) > 0$ , implying that with non-zero probability, action  $a$  will never be played in  $s$ . Hence with non-zero probability  $p_3$  holds.

**Proposition 5.2** (Optimal reachability [2]). *Given an MDP  $\Gamma = (S, E, Act, Pr)$ , and a target set  $T \subseteq S$ , we can compute for each state  $s \in S$  the maximal probability  $v_s^*$  to reach  $T$ , in polynomial time. There is an optimal deterministic memoryless strategy  $\sigma^*$  that enforces  $v_s^*$  from all  $s \in S$ . Now, fix  $s \in S$  and  $c \in \mathbb{Q}$  such that  $c < v_s^*$ . Then there exists  $k \in \mathbb{N}$  such that by playing  $\sigma^*$  from  $s$  for  $k$  steps, we reach  $T$  with probability larger than  $c$ .*

**Lemma 5.3.** *Given an MDP  $\Gamma$ , and a Type III EC  $C$  in  $\Gamma$ , for all states  $s \in C$ , we have  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$ .*

*Proof.* Consider some  $\varepsilon > 0$ , and  $f: \mathbb{N}_0 \rightarrow \mathbb{Q} \cap (0, 1]$  a series of probabilities such that the infinite product  $\prod_{i \in \mathbb{N}_0} f(i) > 1 - \varepsilon$ . Let  $\sigma_1$  be a strategy satisfying **(III<sub>1</sub>)**, and  $\sigma_2$  be a strategy satisfying **(III<sub>2</sub>)**. Using Proposition 5.2, we associate with  $\sigma_2$  a sequence of numbers  $g: \mathbb{N}_0 \rightarrow \mathbb{N}_0$  such that if  $\sigma_2$  is played for

$g(i)$  steps then for all  $a \in \mathcal{A}$  we have that  $C_{\text{even}}^{\max}(p_a)$  is visited with probability at least  $f(i)$ . We consider the following infinite-memory strategy  $\sigma$  by combining strategies  $\sigma_1$  and  $\sigma_2$  as follows:

- a) Let  $i = 0$
- b) Play  $\sigma_2$  for  $g(i)$  steps. Let  $i = i + 1$ .
- c) if for all  $a \in \mathcal{A}$  we have  $C_{\text{even}}^{\max}(p_a)$  was visited, then go to b), else play  $\sigma_1$  forever.

When following  $\sigma$ , in each round  $i$ , we have probability at least  $f(i)$  of continuing to play  $\sigma_2$ . The probability of playing  $\sigma_2$  forever is thus the same as the probability of visiting all  $C_{\text{even}}^{\max}(p_a)$  in each round, that is, at least  $1 - \varepsilon$ , and thus satisfying the parity conditions  $p_{as}$  for  $as \in \mathcal{AS}$  and  $p_{nz}$  with probability  $1 - \varepsilon$ . In all the paths where  $\sigma_2$  keeps being played, for all  $a \in \mathcal{A}$  we have that  $p_a$  is satisfied since  $C_{\text{even}}^{\max}(p_a)$  is visited infinitely often. For the plays switching to  $\sigma_1$  at some point (we have probability  $\varepsilon$  to switch to one of these plays at some point), we have  $\bigwedge(\mathcal{A}, \mathcal{AS})$ . This implies that considering all possibilities, we have that all conditions  $p_{as}$  are satisfied with probability  $1 - \varepsilon + \varepsilon = 1$ , and that all possible plays satisfy all  $p_a$ . Finally,  $p_{nz}$  is satisfied with probability  $1 - \varepsilon$ , and hence with probability greater than 0. Thus for  $s \in C$ , we have  $s, \sigma \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(p_{nz})$ .  $\square$

We now relate the existence of a Type III end component with the  $\bigwedge_{a \in \mathcal{A}} A(p_a)$  objective. We begin with the following observation.

**Lemma 5.4.** *Given an end-component  $C$  in an MDP  $\Gamma$ , a set of parity conditions  $\{p_x \mid x \in X\}$ , for all states  $s \in C$ , we have  $s \models_{\Gamma|C} \text{AS}(\bigwedge_{x \in X} p_x)$  iff  $s \models_{\Gamma|C} \text{NZ}(\bigwedge_{x \in X} p_x)$ .*

The difficult part of the proof is the right to left implication. It uses the fact that Proposition 2.2 implies that the set of states visited infinitely often forms an end-component with probability 1. From there, we find a sub-EC of  $C$  such that in this sub-EC for all  $x \in X$ , condition  $p_x$  has even maximum priority. As there exists a strategy to visit all the states of this sub-EC almost-surely, we have  $s \models_{\Gamma|C} \text{AS}(\bigwedge_{x \in X} p_x)$ .

The detailed proof of the lemma can be found in [4].

Now we state the main result of this section that relates the existence of a Type III end component to the objective  $\bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(p_{nz})$ .

**Lemma 5.5.** *Given an MDP  $\Gamma$ , two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$ ,  $\{p_{as} \mid as \in \mathcal{AS}\}$ , and another parity condition  $p_{nz}$ , for all states  $s$ , we have  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(p_{nz})$  iff  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(\diamond \mathcal{T}_{\text{III}, \mathcal{A}, \mathcal{AS}, p_{nz}})$ .*

*Proof.* We begin with the right to left implication. Consider a strategy  $\sigma_s$  such that  $s, \sigma_s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(\diamond \mathcal{T}_{\text{III}, \mathcal{A}, \mathcal{AS}, p_{nz}})$ . We show below that  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(p_{nz})$ . First note that for all states  $q \in \mathcal{T}_{\text{III}, \mathcal{A}, \mathcal{AS}, p_{nz}}$ , we consider a strategy  $\sigma_q$  such that  $q, \sigma_q \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(p_{nz})$  (we know that such a strategy always exists in a Type III EC, thanks

to Lemma 5.3). Now we construct a strategy  $\sigma$ , such that  $s, \sigma \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(p_{nz})$ . Strategy  $\sigma$  is defined from  $\sigma_s$  and  $\sigma_q$  as follows: We play  $\sigma_s$  from  $s$ . If we reach a state  $q$  belonging to a Type III EC, we play  $\sigma_q$  from  $q$  forever. Since such a state  $q$  can be reached with non-zero probability, and strategy  $\sigma_q$  satisfies  $p_{nz}$  with non-zero probability, we have that  $\sigma$  satisfies  $p_{nz}$  with non-zero probability. As both  $\sigma_s$  and  $\sigma_q$  satisfy  $\bigwedge(\mathcal{A}, \mathcal{AS})$ , and thus, while following  $\sigma$ , since every path in the corresponding MC ends up either following  $\sigma_s$  forever or following  $\sigma_q$  forever, we have that  $\sigma$  also satisfies  $\bigwedge(\mathcal{A}, \mathcal{AS})$ .

Now for the left to right implication, let  $\sigma$  be a strategy such that  $s, \sigma \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(p_{nz})$ . It is easy to see that  $s, \sigma \models_{\Gamma} \text{NZ}(\bigwedge_{a \in \mathcal{A}} (p_a) \wedge \bigwedge_{as \in \mathcal{AS}} (p_{as}) \wedge (p_{nz}))$ . From Proposition 2.2, there is probability 1 that an infinite path ends up in an end-component. Hence in the Markov chain  $\Gamma^{[\sigma]}$  there is a non-zero probability that an infinite path will reach an end-component  $C$  such that for all states  $s' \in C$ , we have  $s', \sigma \models_{\Gamma|C} \text{NZ}(\bigwedge_{a \in \mathcal{A}} (p_a) \wedge \bigwedge_{as \in \mathcal{AS}} (p_{as}) \wedge (p_{nz}))$ . From Lemma 5.4, we thus have that for all  $s' \in C$ , there exists  $\sigma'$  such that  $s', \sigma' \models_{\Gamma|C} \text{AS}(\bigwedge_{a \in \mathcal{A}} (p_a) \wedge \bigwedge_{as \in \mathcal{AS}} (p_{as}) \wedge (p_{nz}))$ . Thus condition (III<sub>2</sub>) is satisfied. As we consider a pruned MDP thanks to Assumption 4.12, for all  $s' \in C$  we have that  $s' \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$ .

Thus  $C$  is a Type III EC that can be reached from  $s$  with non-zero probability, and thus we have:  
 $s, \sigma \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(\diamond \mathcal{T}_{\text{III}, \mathcal{A}, \mathcal{AS}, p_{nz}})$ .  $\square$

## 6 Formulas with multiple Non-Zero and multiple Exists

In this section, we discuss how to compute strategies for formulas that consist of several sure parity objectives, several almost-sure parity objectives, several non-zero parity objectives, and several existential parity objectives. We show in Lemma 6.1 that such a formula can be split into sub-formulas having a single non-zero or a single existential parity objective. Further, we show in Lemma 6.3 that a single non-zero parity objective can be transformed into an existential parity objective. We finally show in Lemma 6.5 how to check the satisfiability of a formula that consists of several sure parity objectives, several almost-sure parity objectives, and one existential parity objective.

**Lemma 6.1.** *Given an MDP  $\Gamma$ , a state  $s$ , four sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$ ,  $\{p_{as} \mid as \in \mathcal{AS}\}$ ,  $\{p_{nz} \mid nz \in \mathcal{NZ}\}$ , and  $\{p_e \mid e \in \mathcal{E}\}$ , the following holds:  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \bigwedge_{nz \in \mathcal{NZ}} \text{NZ}(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} \text{E}(p_e)$  iff for all  $nz \in \mathcal{NZ}$  we have  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{NZ}(p_{nz})$ , and for all  $e \in \mathcal{E}$ , we have  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge \text{E}(p_e)$*

*Proof.* As the left to right implication is obvious, we prove here the other direction. For  $i \in \mathcal{NZ} \cup \mathcal{E}$ , we consider a strategy  $\sigma_i$  such that  $s, \sigma_i \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge Q_i(p_i)$  for the

appropriate  $Q_i \in \{NZ, E\}$ . Now we construct a randomized strategy  $\sigma$  given all  $\sigma_i$  in which each  $\sigma_i$  is chosen uniformly, that is with equal probability. Clearly,  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge \bigwedge_{nz \in NZ} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$ , and hence the result.  $\square$

We now show that a non-zero objective can be replaced with an existential parity objective. Towards this, we first observe the following.

**Proposition 6.2.** *Every reachability condition can be translated to a parity condition.*

*Proof.* Given an MDP  $\Gamma = (S, E, Act, Pr)$ , we construct an MDP  $\Gamma' = (S', E', Act, Pr')$  such that  $S' = S \times \{1, 2\}$  where intuitively  $\Gamma'$  consists of two copies of  $\Gamma$  with state space  $S \times \{1\}$  and  $S \times \{2\}$  respectively, and the reachability condition being satisfied corresponds to moving from the first copy to the second copy, and staying there forever. Formally, we consider the parity condition  $p$  such that for  $s' \in S'$ , with  $s' = (s, i)$  for  $s \in S$  and  $i \in \{1, 2\}$ , we have  $p(s') = i$ .  $\square$

We use this  $\Gamma'$  in the following two lemmas.

**Lemma 6.3.** *Given an MDP  $\Gamma$ , two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$  and  $\{p_{as} \mid as \in \mathcal{AS}\}$ , and a reachability set  $R$ , there exists a state  $s'$  in  $\Gamma'$ , and a parity condition denoted  $p_{\diamond R}$  such that for all states  $s$ , we have  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge NZ(\diamond R)$  iff  $s' \models_{\Gamma'} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge E(p_{\diamond R})$ .*

The result comes from noticing that a non-zero reachability objective and an existential reachability objective are equivalent. It is then enough to use Proposition 6.2 to get Lemma 6.3. The detailed proof of the lemma appears in [4].

We get from Lemma 5.5 and Lemma 6.3 the following:

**Lemma 6.4.** *Given an MDP  $\Gamma$ , and two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$  and  $\{p_{as} \mid as \in \mathcal{AS}\}$ , and a parity condition  $p_{nz}$ , for all states  $s$ , we have  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$  iff there exists  $s'$  in  $\Gamma'$  such that  $s' \models_{\Gamma'} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge E(p_{\diamond \mathcal{T}_{III, nz}})$ , where  $\mathcal{T}_{III, nz}$  is defined w.r.t.  $\Gamma'$ .*

Now, since by Assumption 4.12 we have removed all the states that do not satisfy  $\bigwedge (\mathcal{A}, \mathcal{AS})$ , we have the following:

**Lemma 6.5.** *Given an MDP  $\Gamma$ , a state  $s$ , two sets of parity conditions  $\{p_a \mid a \in \mathcal{A}\}$  and  $\{p_{as} \mid as \in \mathcal{AS}\}$ , and another parity condition  $p$ , we have that  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge E(p)$  iff  $s \models_{\Gamma} E(\bigwedge_{a \in \mathcal{A}} p_a \wedge p)$ .*

*Proof.* The left to right result is obvious as  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge E(p)$  implies  $E(\bigwedge_{a \in \mathcal{A}} p_a \wedge p)$ .

For the other direction, consider a strategy  $\widehat{\sigma}$  such that  $s, \widehat{\sigma} \models_{\Gamma} E(\bigwedge_{a \in \mathcal{A}} p_a \wedge p)$ . Now, a conjunction of parity conditions is a Streett condition [15], and non-emptiness problem of Streett automata is decidable.

We noted in Section 2 that satisfying existentially is the same as finding a satisfying path in the nondeterministic

automaton associated to  $\Gamma$ . This means that if the Streett automaton is non-empty, then there exists a finite-memory strategy  $\sigma$  (linear in the indices of  $p_a$  and  $p$ ) in  $\Gamma$  such that there exists a path  $\pi$  in the MC  $\Gamma^{\sigma}$  satisfying both  $p$  and all  $p_a$  for  $a \in \mathcal{A}$ .

Now by assumption, there exists a strategy  $\sigma_{\wedge}$  such that  $s, \sigma_{\wedge} \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS})$ . A strategy  $\sigma$  such that  $s, \sigma \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge E(p)$  is obtained below by combining  $\widehat{\sigma}$  and  $\sigma_{\wedge}$  as follows. At each step a coin is tossed. If it gives head, then we play  $\sigma_{\wedge}$  forever. Otherwise, we play this step as specified by the strategy  $\widehat{\sigma}$ . If this results in deviating from the path  $\pi$ , then we play  $\sigma_{\wedge}$  forever, else repeat the same process.

Strategy  $\sigma$  has a path ensuring  $p$ : the one where we always follow  $\widehat{\sigma}$ , that happens when all the coin tosses give tails, and the state randomly taken in the MDP is always in  $\pi$ . The probability of switching to  $\sigma_{\wedge}$  some time is 1, thus satisfying

$\bigwedge_{as \in \mathcal{AS}} AS(p_{as})$ . If we follow the path  $\pi$ , then for all  $a \in \mathcal{A}$  we have that  $p_a$  is satisfied, otherwise we switch to  $\sigma_{\wedge}$  at some point, and for all  $a \in \mathcal{A}$  we again have that  $p_a$  is satisfied, thus ensuring  $\bigwedge_{a \in \mathcal{A}} A(p_a)$ .  $\square$

This concludes the decidability proof of the realizability of the negation and disjunction-free fragment of QPL. Indeed, given a formula  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge \bigwedge_{nz \in NZ} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$ , we use Lemma 6.1 to split it into formulas of the form  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$ , and of the form  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge E(p_e)$ . We then apply Lemma 5.5 on formulas of the form  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$ , and then use Lemma 6.4 to transform the non-zero objective into an existential objective. For formulas of the form  $s \models_{\Gamma} \bigwedge (\mathcal{A}, \mathcal{AS}) \wedge E(p_e)$ , we use Lemma 6.5.

By Remark 2.1 it shows the decidability of the QPL - realizability problem. We state the complexity of this realizability problem in the next Section.

## 7 Complexity results

In this section, we analyze the complexity of deciding the existence of strategies to satisfy QPL formulas. Recall from the results of Sections 5 and 6, that in order to find strategies for formulas of the form  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge$

$\bigwedge_{nz \in NZ} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$ , we need to compute the set of max-

imal Type III ECs. In particular, we need these ECs for subformulas of the form  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge \bigwedge_{nz \in NZ} NZ(p_{nz})$ .

This in turn requires solving several Streett games in general (Lemma 3.5). The procedure is described in Algorithm 1. We show that the algorithm runs in time  $\Sigma_2^P = P^{NP}$  (Theorem 7.1). We also show that we have a polynomial algorithm for the special case where the set  $\mathcal{A}$  is empty (Theorem 7.2), and that randomization and finite memory are required (Theorem 7.3). The problem is in  $NP \cap CoNP$  when  $\mathcal{A}$  is singleton (Theorem 7.4). Finally, we show that finding a strategy for

**Algorithm 1**

**Input :** An MDP  $\Gamma$ ,  $s$  a state of  $\Gamma$ , parity conditions  $\{p_a \mid a \in \mathcal{A}\}$ ,  $\{p_{as} \mid as \in \mathcal{AS}\}$ ,  $\{p_{nz} \mid nz \in \mathcal{NZ}\}$ , and  $\{p_e \mid e \in \mathcal{E}\}$ .

**Output :** true if

$s \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge \bigwedge_{nz \in \mathcal{NZ}} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$   
else false.

- 1: Compute the set of maximal Type I  $(\mathcal{A}, \mathcal{A})$  ECs  $C$  such that  $\forall s' \in C$ ,  $s' \models_{\Gamma|C} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(\diamond_{\text{even}}^{C_{\text{max}}} p_a)$ . //

By Lemma 3.5.

- 2: Compute the set of maximal Type II  $(\mathcal{A}, \mathcal{AS})$  ECs  $C$ :  $C$  is a maximal Type I  $(\mathcal{A}, \mathcal{AS})$  EC and  $\forall s' \in C$ ,  $s' \models_{\Gamma|C} \bigwedge_{a \in \mathcal{A}} AS(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$ .

- 3: Compute the set  $\mathcal{S}_1$  of states  $s'$  such that  $s' \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge AS(\diamond_{\mathcal{T}_{II, \Gamma, \mathcal{A}, \mathcal{AS}}} p_a)$ . // Correct by Lemma 4.2.  
If  $s \notin \mathcal{S}_1$  then return false.

- 4: Compute  $\Gamma|_{\mathcal{S}_1}$  where all the states that do not satisfy  $\bigwedge(\mathcal{A}, \mathcal{AS})$  have been pruned. // Correct by Lemma 4.7.

5: **for all**  $nz \in \mathcal{NZ}$  **do**

- 6: Compute the set of maximal Type III  $(\mathcal{A}, \mathcal{AS}, p_{nz})$  ECs  $C$  of  $\Gamma|_{\mathcal{S}_1}$ :  $\forall s \in C$ , we have that  $s \models_{(\Gamma|_{\mathcal{S}_1})|C} \bigwedge_{a \in \mathcal{A}} AS(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge AS(p_{nz})$ . // By Lemma 5.5.

- 7: Compute the parity condition  $p_{\diamond_{\mathcal{T}_{III, \mathcal{A}, \mathcal{AS}, p_{nz}}}}$  and the MDP  $\Gamma'$  with set  $\mathcal{S}'_1$  of states //  $\Gamma'$  and  $\mathcal{S}'_1$  are defined in Proposition 6.2.

- 8: Check if  $(s, 1) \models_{\mathcal{S}'_1} E(\bigwedge_{a \in \mathcal{A}} p_a \wedge p_{\diamond_{\mathcal{T}_{III, \mathcal{A}, \mathcal{AS}, p_{nz}}}})$ . // By lemmas 6.4 and 6.5;  $(s, 1)$  is defined in Proposition 6.2.

9: **for all**  $e \in \mathcal{E}$  **do**

- 10: Check if  $s \models_{\mathcal{S}_1} E(\bigwedge_{a \in \mathcal{A}} p_a \wedge p_e)$ . // By Lemma 6.5.

- 11: If any of the checks in Steps 8 and 10 fails then return false, else return true.

an arbitrary QPL formula that may have combination of conjunctions and disjunctions is in  $\Sigma_2^P (= \text{NP}^{\text{NP}})$  (Theorem 7.5), and it is both NP-hard and coNP-hard (Theorem 7.6).

**Theorem 7.1.** *Given an MDP  $\Gamma$ , and a state  $s$ , Algorithm 1 decides if  $s \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge \bigwedge_{nz \in \mathcal{NZ}} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$  in  $\text{P}^{\text{NP}}$  time.*

*Proof.* For the correctness of Algorithm 1, consider a formula  $\varphi = \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge \bigwedge_{nz \in \mathcal{NZ}} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$ . Given  $\varphi$ , we explain below how to check its satisfiability.

According to Lemma 6.1, to check if  $s \models_{\Gamma} \varphi$ , we need to check for all  $nz \in \mathcal{NZ}$ , if  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$  and that for all  $e \in \mathcal{E}$  if  $s \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS}) \wedge E(p_e)$ .

For the non-zero (NZ) part, recall from Section 6, that for each  $p_{nz}$ , we need to compute the set of all Type III ECs for

the objective  $\bigwedge(\mathcal{A}, \mathcal{AS}) \wedge NZ(p_{nz})$ . Recall that each Type III EC  $C$  is such that two properties hold. First  $\forall s' \in C$ , we have  $s' \models_{\Gamma} \bigwedge(\mathcal{A}, \mathcal{AS})$ ; we do this in Step 3 of the algorithm. Then we also check that  $\forall s' \in C$ ,  $s' \models_{\Gamma|C} \bigwedge_{a \in \mathcal{A}} AS(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge AS(p_{nz})$ . This is done in Step 6.

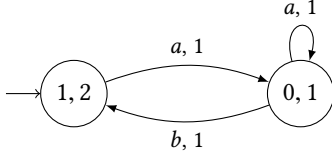
Now for an existential parity objective  $E(p_e)$ , by Lemma 6.5, we only need to check if  $s \models_{\Gamma|_{\mathcal{S}_{[\bigwedge(\mathcal{A}, \mathcal{AS})]}}} E(\bigwedge_{a \in \mathcal{A}} p_a \wedge p_e)$ . It remains to explain how the sub-MDP in which all states satisfy the formula  $\bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$  is computed. This is done in Steps 1 to 4.

To do so we first find the set of states  $s'$  such that  $s' \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$ . We find the set of maximal Type II ECs. By definition of Type I and Type II ECs, a maximal Type I EC that satisfies  $(\text{II}_2)$  is a maximal Type II EC. We now prove that if  $C$  is a maximal Type II EC, it is included in  $C_1$ , a maximal Type I EC. Since for all  $s' \in C$ , we have  $s' \models_{\Gamma|C} \bigwedge_{a \in \mathcal{A}} AS(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$ , we also have that for all  $s' \in C_1$ ,  $s' \models_{\Gamma|C_1} \bigwedge_{a \in \mathcal{A}} AS(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$  (it suffices to take a strategy that has probability 1 of reaching some state  $s'' \in C$ , and then play the strategy ensuring  $s'' \models_{\Gamma|C} \bigwedge_{a \in \mathcal{A}} AS(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as})$ ). Hence  $C_1$  is a Type II EC. By maximality of  $C$ , we have  $C = C_1$ . This means that to find the maximal Type II EC, it is sufficient to compute the maximal Type I EC and remove those maximal Type I ECs that do not ensure condition  $(\text{II}_2)$ . Finding the maximal Type I EC is done in Step 1 thanks to Lemma 3.5. We then check condition  $(\text{II}_2)$  in Step 2.

It then remains to find the states  $s'$  such that  $s' \models_{\Gamma} \bigwedge_{a \in \mathcal{A}} A(p_a) \wedge AS(\diamond_{\mathcal{T}_{II, \Gamma, \mathcal{A}, \mathcal{A}}})$  (Step 3). It can be done by transforming the MDP into a Streett-Büchi game that in turn can be transformed into a Streett game since a Büchi condition is a special case of a parity condition, and a conjunction of parity conditions is a Streett condition.

For the complexity, Steps 2, 4, 6, 7, 8 and 10 are polynomial. Step 3 is parity-complete if there is only one parity condition, polynomial if there is none, and is co-NP complete in general (we have to solve a Streett game). Step 1 is parity-complete if there is only one parity condition, polynomial if there is none. In the general case, Step 1 requires to iteratively solve a polynomial number of Streett games (which is in coNP), and use the result of this computation to remove some of the states, resulting in a  $\text{P}^{\text{NP}}$  complexity. Further details of the procedure are given in the full version in [4]. This leads us to a  $\text{P}^{\text{NP}}$  complexity for Algorithm 1.  $\square$

**Theorem 7.2.** *Given an MDP  $\Gamma$ , and a state  $s$ , we can decide if  $s \models_{\Gamma} \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge \bigwedge_{nz \in \mathcal{NZ}} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$  in polynomial time.*



**Figure 5.** An MDP requiring randomized finite-memory strategy for the objective  $AS(p_1) \wedge E(p_2)$ .

*Proof.* The proof is similar to that of Theorem 7.1, but as  $\mathcal{A} = \emptyset$  we compute Type I  $(\mathcal{A}, \mathcal{A})$  that is Type I  $(\emptyset, \emptyset)$  ECs, that is any EC, and so Step 1 returns the maximal ECs. In the same way, Step 3 polynomially computes one almost-sure reachability problem. As we consider Type I  $(\emptyset, \emptyset)$ , the checks in Steps 2,3,6,8 and 10 can be done with a randomized finite-memory strategy. Further, a strategy for each of these steps can be computed in polynomial time [17].  $\square$

In the above proof, we show that randomized finite-memory strategies are sufficient for formulas of the form  $s \models_{\Gamma} \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge \bigwedge_{nz \in \mathcal{NZ}} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$ . Below, we show that such strategies are indeed necessary.

**Theorem 7.3.** *Given an MDP  $\Gamma$ , and a state  $s$ , to decide if  $s \models_{\Gamma} \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge \bigwedge_{nz \in \mathcal{NZ}} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$ , a randomized finite-memory winning strategy may be necessary.*

*Proof.* We show that the lemma already holds for formulas of the form  $AS(p_1) \wedge E(p_2)$ . Consider the MDP in Figure 5. We want a strategy  $\sigma$  such that in  $\Gamma^{[\sigma]}$ , we have a subset of  $(ab)^*a^\omega$  with measure 1 to satisfy  $AS(p_1)$ , and we want at least one path in  $(a^+b)^\omega$  to satisfy  $E(p_2)$ . We can argue that no randomized memoryless strategy, as well as no deterministic memoryful strategy suffices here. However, we can use the following randomized strategy with one bit of memory: Toss a coin every time the token is in state (0, 1). If the coin gives heads, play  $ba$ , otherwise if the coin gives tails once, play  $a$  forever from the state (0, 1) (one bit of memory is used to store if the coin has given tails).  $\square$

**Theorem 7.4.** *Given an MDP  $\Gamma$ , and a state  $s$ , we can decide in  $NP \cap coNP$  if  $s \models_{\Gamma} A(p_a) \wedge \bigwedge_{as \in \mathcal{AS}} AS(p_{as}) \wedge \bigwedge_{nz \in \mathcal{NZ}} NZ(p_{nz}) \wedge \bigwedge_{e \in \mathcal{E}} E(p_e)$ . This requires solving a polynomial number of parity games.*

*Proof.* The proof is similar to that of Theorem 7.1, but Step 1 computes maximal Type I  $(\{a\}, \{a\})$  ECs which can be done in  $P^{NP \cap coNP}$  [1]. Also the set of states that ensure the objective in Step 3 with only one sure parity condition can be computed by solving a polynomial number of parity games [1, 5], and by pruning states after solving each parity game. The result follows since  $P^{NP \cap coNP} = NP \cap coNP$  [9].  $\square$

We now consider QPL without any restriction.

**Theorem 7.5.** *QPL realizability is in  $NP^{NP}$ .*

*Proof.* Assume that we are given an MDP  $\Gamma$ , a state  $s$ , and a QPL formula  $\varphi$ . We assume that  $\varphi$  is in negation-free normal form. A negation-free equivalent formula can be found in polynomial time, it suffices to take the negation normal form (by using De Morgan's law to push negations inwards), and to take the dual of the negated atomic parity conditions.

We note that there exists a DNF formula equivalent to  $\varphi$  with the same set of atomic objectives. If  $\varphi$  is satisfiable, then there exists a conjunctive clause  $\psi$  that is a conjunction of these atomic objectives, and  $\psi$  can be guessed by an NP machine such that  $s \models_{\Gamma} \psi$ . Given this witness  $\psi$ , we can use an NP oracle to check that  $\psi$  implies  $\varphi$  where the atomic objectives are regarded as classical propositional atoms. We can then use the  $P^{NP}$  algorithm of Theorem 7.1 to get a proof that indeed  $s \models_{\Gamma} \psi$  implying that  $\varphi$  is also satisfied by  $s$ . The result follows since the class  $NP^{NP}$  is the same as  $NP^{NP}$ .  $\square$

**Theorem 7.6.** *QPL realizability is NP-hard and coNP-hard.*

*Proof.* The coNP-hardness follows from the fragment of QPL made of conjunctions of  $A(p)$  atoms that is powerful enough to encode Streett games, as proved in [15].

We prove NP-hardness for the fragment of QPL made of  $\{\wedge, \vee, A(p)\}$ . Given a SAT formula  $\varphi$  in negation normal form, we define a QPL formula  $\varphi_{obj}$ , and an MDP  $\Gamma$  both polynomial in  $\varphi$  such that there exists a state  $s$  of  $\Gamma$  and there exists a winning strategy for  $\varphi_{obj}$  from  $s$  on  $\Gamma$  if and only if  $\varphi$  is satisfiable. Let  $Var = \{a_1, \dots, a_n\}$  be the set of propositional variables of  $\varphi$ . Let  $S = \{a_i, \bar{a}_i \mid i \in [n]\}$  be both the set of states and the set of actions. We define the MDP  $\Gamma = \{S, \{(a, b, b) \mid a, b \in S\}, S, Pr\}$  where for all  $a, b \in S$  we have  $Pr(a, b, b) = 1$ . Note that the underlying graph is a complete graph. For each  $i \in [n]$ , we define two parity conditions  $p_{a_i}$  and  $p_{\bar{a}_i}$  such that  $p_{a_i}(a_i) = p_{\bar{a}_i}(\bar{a}_i) = 2$ , such that  $p_{a_i}(\bar{a}_i) = p_{\bar{a}_i}(a_i) = 3$  and  $p_{a_i}(b) = p_{\bar{a}_i}(b) = 1$  if  $b \notin \{a_i, \bar{a}_i\}$ . We define  $\psi = \bigwedge_{i \in [1, n]} A(p_{a_i}) \vee A(p_{\bar{a}_i})$ . Given the SAT formula  $\varphi$  in

NNF, we define a QPL formula  $\varphi'$  by transforming  $\varphi$  in the following way: each  $\neg a_i$  is replaced by  $A(p_{\bar{a}_i})$  and each (non-negated)  $a_i$  is replaced by  $A(p_{a_i})$ . We define  $\varphi_{obj} = \psi \wedge \varphi'$ . In the MDP  $\Gamma$ , we consider an arbitrary state  $s$ , a strategy  $\sigma$ , and the paths in  $Paths^{\Gamma^{[\sigma]}}(s)$ . We show in [4] that  $s \models_{\Gamma} \varphi_{obj}$  iff  $\varphi$  is satisfiable.  $\square$

## Acknowledgments

Work partially supported by the PDR project *Subgame perfection in graph games* (F.R.S.-FNRS), the ARC project *Non-Zero Sum Game Graphs: Applications to Reactive Synthesis and Beyond* (Fédération Wallonie-Bruxelles), the EOS project *Verifying Learning Artificial Intelligence Systems* (F.R.S.-FNRS & FWO), and the COST Action 16228 *GAMENET* (European Cooperation in Science and Technology).

## References

- [1] Shaull Almagor, Orna Kupferman, and Yaron Velner. 2016. Minimizing Expected Cost Under Hard Boolean Constraints, with Applications to Quantitative Synthesis. In *27th International Conference on Concurrency Theory, CONCUR 2016*. 9:1–9:15.
- [2] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking*. MIT Press.
- [3] Raphaël Berthon, Nathanaël Fijalkow, Emmanuel Filiot, Shibashis Guha, Bastien Maubert, Aniello Murano, Laureline Pinault, Sophie Pinchinat, Sasha Rubin, and Olivier Serre. 2020. Alternating Tree Automata with Qualitative Semantics. *CoRR abs/2002.03664* (2020).
- [4] Raphaël Berthon, Shibashis Guha, and Jean-François Raskin. 2020. Mixing Probabilistic and non-Probabilistic Objectives in Markov Decision Processes. (2020). *arXiv:cs.LO/2004.13789*
- [5] Raphaël Berthon, Mickael Randour, and Jean-François Raskin. 2017. Threshold Constraints with Guarantees for Parity Objectives in Markov Decision Processes. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*. 121:1–121:15.
- [6] Mikołaj Bojańczyk. 2016. Thin MSO with a probabilistic path quantifier. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [7] Mikołaj Bojańczyk, Hugo Gimbert, and Edon Kelmendi. 2017. Emptiness of zero automata is decidable. *arXiv preprint arXiv:1702.06858* (2017).
- [8] Mikołaj Bojańczyk, Edon Kelmendi, and Michał Skrzypczak. 2019. MSO+ $\nabla$  is undecidable. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019*. IEEE, 1–13.
- [9] Gilles Brassard. 1979. A note on the complexity of cryptography (corresp.). *IEEE Transactions on information Theory* 25, 2 (1979), 232–233.
- [10] Tomáš Brázdil, Vojtěch Forejt, and Antonín Kučera. 2008. Controller synthesis and verification for Markov decision processes with qualitative branching time objectives. In *International Colloquium on Automata, Languages, and Programming, ICALP 2008*. Springer, 148–159.
- [11] Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-François Raskin. 2014. Meet Your Expectations With Guarantees: Beyond Worst-Case Synthesis in Quantitative Games. In *31st International Symposium on Theoretical Aspects of Computer Science, STACS 2014 (LIPIcs)*, Vol. 25. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 199–213.
- [12] Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-François Raskin. 2017. Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games. *Information and Computation* 254 (2017), 259–295. <https://doi.org/10.1016/j.ic.2016.10.011>
- [13] Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. 2017. Deciding parity games in quasipolynomial time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*. ACM, 252–263. <https://doi.org/10.1145/3055399>
- [14] Arnaud Carayol, Axel Haddad, and Olivier Serre. 2014. Randomization in automata on infinite trees. *ACM Transactions on Computational Logic* 15, 3 (2014), 1–33.
- [15] Krishnendu Chatterjee, Thomas A Henzinger, and Nir Piterman. 2007. Generalized parity games. In *International Conference on Foundations of Software Science and Computational Structures, FoSSaCS 2007*. Springer, 153–167.
- [16] Krishnendu Chatterjee and Nir Piterman. 2019. Combinations of Qualitative Winning for Stochastic Parity Games. In *30th International Conference on Concurrency Theory, CONCUR 2019 (LIPIcs)*, Vol. 140. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 6:1–6:17. <http://www.dagstuhl.de/dagpub/978-3-95977-121-4>
- [17] Kousha Etessami, Marta Kwiatkowska, Moshe Y Vardi, and Mihalis Yannakakis. 2007. Multi-objective model checking of Markov decision processes. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems TACAS 2007*. Springer, 50–65.
- [18] Paulin Fournier and Hugo Gimbert. 2018. Alternating Nonzero Automata. In *29th International Conference on Concurrency Theory, CONCUR 2018 (Leibniz International Proceedings in Informatics (LIPIcs))*, Sven Schewe and Lijun Zhang (Eds.), Vol. 118. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 13:1–13:16. <https://doi.org/10.4230/LIPIcs.CONCUR.2018.13>
- [19] Henryk Michalewski and Matteo Mio. 2016. Measure quantifier in monadic second order logic. In *International Symposium on Logical Foundations of Computer Science, LFCS 2016*. Springer, 267–282.
- [20] Henryk Michalewski, Matteo Mio, and Mikołaj Bojańczyk. 2016. On the regular emptiness problem of subzero automata. *arXiv preprint arXiv:1608.03319* (2016).
- [21] Moshe Y. Vardi. 1985. Automatic Verification of Probabilistic Concurrent Finite-State Programs. In *26th Annual Symposium on Foundations of Computer Science, FOCS 1985*. 327–338.