Solving Qualitative Multi-Objective Stochastic Games

Moritz Graf
RPTU University
Kaiserslautern-Landau & Max Planck
Institute for Software Systems
Kaiserslautern, Germany
moritz.graf@cs.rptu.de

Anthony Lin
RPTU University
Kaiserslautern-Landau & Max Planck
Institute for Software Systems
Kaiserslautern, Germany
awlin@mpi-sws.org

Rupak Majumdar
Max Planck Institute for Software
Systems
Kaiserslautern, Germany
rupak@mpi-sws.org

ABSTRACT

Many problems in compositional synthesis and verification of multi-agent systems—such as rational verification and assume-guarantee verification in probabilistic systems—reduce to reasoning about two-player multi-objective stochastic games. This motivates us to study the problem of characterizing the complexity and memory requirements for two-player stochastic games with Boolean combinations of qualitative reachability and safety objectives. Reachability objectives require that a given set of states is reached; safety requires that a given set is invariant. A qualitative winning condition asks that an objective is satisfied almost surely (AS) or (in negated form) with non-zero (NZ) probability.

We study the determinacy and complexity landscape of the problem. We show that games with conjunctions of AS and NZ reachability and safety objectives are determined, and determining the winner is PSPACE-complete. The same holds for positive boolean combinations of AS reachability and safety, as well as for negations thereof. On the other hand, games with full Boolean combinations of qualitative objectives are not determined, and are NEXPTIME-hard. Our hardness results show a connection between stochastic games and logics with partially-ordered quantification. Our results shed light on the relationship between determinacy and complexity, and extend the complexity landscape for stochastic games in the multi-objective setting.

KEYWORDS

Stochastic games; Temporal logic specifications; Reachability; Qualitative objectives; Multi-objective games; Complexity

ACM Reference Format:
Moritz Graf, Anthony Lin, and Rupak Majumdar. 2026. Solving Qualitative Multi-Objective Stochastic Games. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/PAAJ1168

1 INTRODUCTION

Stochastic games are a class of games consisting of multiple players, in which the environment exhibits stochastic behavior. The 2-player version of the game (often called $2\frac{1}{2}$-player games) has especially been extremely useful in modeling many problems in verification, including rational verification in multi-agent systems [16, 25], wherein Player 1 represents a deviating player and Player 2 represents a coalition of players who aim to punish the deviating player. However, in many applications of 2-player stochastic games, it is known that *multiple* objectives (i.e. a Boolean combination of them) are necessary (cf. [8, 16, 17, 26]).

ϕ	queries	objectives	determined?	complexity
$\bigwedge$	AS, NZ	$\Diamond, \Box$	yes	PSPACE-complete
$\bigvee$	AS, NZ	$\Diamond, \Box$	yes	PSPACE-complete
$\mathcal{B}^+$	AS	$\Diamond, \Box$	yes	PSPACE-complete
$\mathcal{B}^+$	NZ	$\Diamond, \Box$	yes	PSPACE-complete
$\mathcal{B}^+$	AS, NZ	$\Diamond, \Box$	no	NEXPTIME-hard
$\mathcal{B}^+$	$AS\Diamond,\ AS\Box,\ NZ\Diamond$		no	NEXPTIME-complete

Table 1: Main results. Queries that use only conjunctions ($\bigwedge$) or only disjunctions ($\bigvee$) are PSPACE-complete, as are positive Boolean combinations ($\mathcal{B}^+$) of only almost sure (AS) or only nonzero (NZ) queries. Queries that use conjunctions, disjunctions, and both AS and NZ are NEXPTIME-hard.

Many basic problems on solving 2-player stochastic games with multiple objectives are still open. Chen et al. [8] studied this problem, focusing primarily on *quantitative* objectives, i.e., a Boolean combination of expected total reward objectives. They proved that such games are not determined. *Determinacy* is the property that for every game and every winning objective, either player 1 wins the winning objective or player 2 wins the negation of the objective; it is a generalization of the minimax theorem for two player one-shot games. They also show that if players are restricted to deterministic strategies, deciding if a player has a winning strategy becomes undecidable. For general strategies, PSPACE-hard was shown, but decidability remains open.

Stan et al. [25] and Winkler and Weininger [26] studied multi-objective stochastic games with *qualitative* reachability objectives (requiring that a set of states is reached) and safety objectives (requiring that a given set of states is never left). Further, a qualitative condition requires that the underlying reachability or safety objective is satisfied almost surely (with probability one; also written AS) or with non-zero probability (also written NZ). Thus, winning conditions are general Boolean formulas over the propositions

$$(\{AS, NZ\} \times \{\Diamond, \Box\})F,$$

where F ranges over sets of states, $\Diamond$ and $\Box$ denote reachability and safety, respectively, and AS and NZ denote almost surely or non-zero, respectively.

This qualitative setting is, in fact, sufficient for many applications. For example, this is the case for *liveness* verification for probabilistic

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/PAAJ1168

distributed protocols [14, 18–20], e.g., whether a philosopher will eventually eat with probability 1 in a dining philosopher protocol. Such qualitative objectives were also considered in rational verification problems in various settings [16, 25]. For this qualitative setting, [26] show that disjunctions of almost sure reachability can be solved in polynomial time, and are between PSPACE and EXPTIME for deterministic strategies. In [25] it is shown that the case with conjunctions of almost sure and nonzero objectives is decidable, and is in EXPTIME. Decidability for the general case with an arbitrary Boolean combination remains an open problem.

Contributions. In this paper, we carry out a systematic investigation of the determinacy and complexity of two-player stochastic games with multiple qualitative objectives.

Our starting point is an observation that different applications give rise to different *classes* of Boolean formulas. For example, in the reduction [16, 25] from the problem of rational verification for probabilistic systems to two-player stochastic games, one obtains only a *conjunctive* Boolean formula. Similarly, in assume-guarantee reasoning, it has been noted in [8, 21] that the required Boolean formulas are of the form

$$\bigwedge_i \neg\varphi_i \vee \psi_i,$$

where φ_i (resp. ψ_i) encodes the required assumption (resp. guaranteed post-condition). For these reasons, it makes sense to investigate the problem by *varying the allowed Boolean operators* ($\wedge, \vee, \neg$) in the objectives. Moreover, owing to the duality of $AS(\Diamond F)$ and $NZ(\Box F)$ (and similarly the duality of $AS(\Box F)$ and $NZ(\Diamond F)$), we may allow only $\wedge, \vee$ (i.e., dispense with negation) and instead vary the individual propositions that are allowed.

For the general case (without restricting the Boolean formulas), we show that such games are not determined. This improves the result in [8] that two-player stochastic games with multiple *quantitative* objectives are not determined. As for the problem of deciding if player 1 has a winning strategy, we do not know if this is decidable, but we show a new NEXPTIME lower bound, improving the PSPACE-hardness from [8]. This exploits a connection between dependency quantified Boolean formulas [2] and stochastic games.

Table 1 summarizes our results for different subclasses of formulas. For the special case of Boolean combinations of non-zero reachability, we can show decidability and a matching NEXPTIME upper bound. Our proof uses a characterization of optimal policies: we show that if player 1 has a winning strategy, then player 1 has a winning strategy that uses exponential memory.

We explore the determinacy boundary. We show that games with positive Boolean combinations of *only* AS or *only* NZ objectives are determined, as are games where the formula is a pure disjunction or a pure conjunction. We complement this with a PSPACE upper bound to determine the winner, which matches a PSPACE-hardness inherited from multiple (nonstochastic) reachability games [12] or from multi-objective reachability in MDPs [23].

Related Work. Very few results were known for multi-objective stochastic games before our work. This is in contrast to *single-objective* stochastic games—the winning condition is exactly one temporal objective—for which there is a well-developed algorithmic theory [6, 9, 10]. It is also in contrast to multi-objective problems on Markov decision processes (single-player stochastic games), where algorithmic solutions are known for Boolean combinations of quantitative LTL objectives [11], for percentile queries [23], or for combinations probabilistic and non-probabilistic objectives [3].

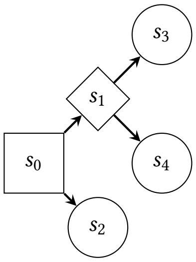

Figure 1: Example stochastic game.

In addition to the related work on stochastic games that we have discussed earlier, we also mention the work of [1, 4]. While in general it is only known that the Pareto set for conjunctions of reachability objectives can be approximated [1], it is shown in [4] that total reward is decidable for stopping games with two objectives. Notably, [4] also give an exponential time algorithm to compute the Pareto set exactly, assuming determinacy.

Organization. We define stochastic games in Section 2. We provide the proofs of our results for determined (resp. nondetermined) queries in Section 3 (resp. 4). Missing proofs are in the full version [15].

2 PRELIMINARIES

We introduce the setting of *turn-based* stochastic games [5, 13, 24].

Definition 1 (Stochastic Games). *A stochastic game is a tuple $\mathcal{G} = \langle S, s_0, A, P\rangle$, where S is a finite set of states, partitioned into disjoint subsets S_1, S_2, and S_c controlled by player 1, player 2, and the chance player, respectively. Thus, $S = S_1 \cup S_2 \cup S_c$. $s_0 \in S$ is an initial state. The map $A : S_1 \cup S_2 \to 2^S \setminus \{\emptyset\}$ maps player 1 and 2 states to possible successors in S. The map $P : S_c \to \Delta(S)$ maps states in S_c to a probability distribution over S.*

A game starts in the initial state s_0 and proceeds in rounds. In each round, if the current state is in S_i, for $i \in \{1, 2\}$ then player i picks a successor s' from $A(s)$ and the new state is s'. If the current state is in S_c, the game moves to a new state $s' \in S$ with probability $P(s)(s')$. A state s with $A(s) = \{s\}$ is called a terminal state. Note that these games are turn based: in each round, exactly one player decides the next state. For $s \in S$, we use $\mathcal{G}_s$ to denote the game that is identical to $\mathcal{G}$ but starts at s instead of s_0. We draw a stochastic game $\mathcal{G}$ as a directed graph, with nodes $s \in S$, denoted by a square if $s \in S_1$, a diamond if $s \in S_2$ or a circle if $s \in S_c$, and edges (s, s') if $s' \in A(s)$ or $P(s)(s') \neq 0$. We will usually omit the self loops on terminal states. Figure 1 is a stochastic game with $S_1 = \{s_0\}$, $S_2 = \{s_1\}$, $S_c = \{s_2, s_3, s_4\}$.

Definition 2 (Strategies). *A* strategy *for player 1 in $\mathcal{G}$ is a function $\sigma : S^*S_1 \to \Delta(S)$ with $supp(\sigma(\pi s)) \subseteq A(s)$, for all $\pi s \in S^*S_1$, where $\Delta(S)$ is the set of distributions over S and $supp(d)$ is the support of a distribution $d \in \Delta(S)$. Strategies for player 2 are defined symmetrically as functions $\tau : S^*S_2 \to \Delta(S)$.*

When both players fix their strategies, the game induces a Markov chain, where the successors at any state are picked according to the strategies (or according to P for chance states). Formally, specifying a pair of strategies σ, τ induces an infinite Markov Chain $\mathcal{M}_{\mathcal{G}}^{\sigma,\tau}$, whose nodes are paths in $\mathcal{G}$ starting from s_0 following the strategies σ and τ (branching is caused by probabilistic actions). See [7] for more details. The runs of this Markov chain are called *plays* of $\mathcal{G}$ according to σ, τ. We write $\mathcal{P}_{\mathcal{G}}^{\sigma,\tau}$ for the probability measure over these plays, defined by the usual cylinder sets [7].

An *objective* is a set of plays. We will focus on *reachability* and *safety* objectives. Let $T \subseteq S$. The *reachability objective* with target T, written $\Diamond T$, is the set of plays in S^ω that reach T. Its dual, the *safety objective*, written $\Box T$, is the set of plays that always remain in T:

$$\Diamond T = \{\pi \in S^\omega | \exists i : \pi_i \in T\} \quad \text{and} \quad \Box T = \{\pi \in S^\omega \mid \forall i : \pi_i \in T\}$$

Note that $\pi \in \Diamond T \iff \pi \notin \Box S \setminus T$.

Fix a game $\mathcal{G}$ and strategies σ and π. An objective $X \subseteq S^\omega$ is realized *almost surely*, written $\sigma, \tau \models_{\mathcal{G}} AS\, X$, iff X holds with probability one: $\mathbb{P}_{\mathcal{G}}^{\sigma,\tau}(X) = 1$. An objective X is realized with *nonzero probability*, written $\sigma, \tau \models_{\mathcal{G}} NZ\, X$, iff $\mathbb{P}_{\mathcal{G}}^{\sigma,\tau}(X) > 0$.

We extend AS and NZ queries to Boolean combinations, with the obvious semantics. For example, $\sigma, \tau \models_{\mathcal{G}} AS\, X \vee AS\, Y$ iff $\sigma, \tau \models_{\mathcal{G}} AS\, X$ or $\sigma, \tau \models_{\mathcal{G}} AS\, Y$. We refer to such a Boolean combination as a *query*.

In the following we only consider queries where the base objectives are either reachability or safety. Note that for any objective X, we have $\sigma, \tau \models_{\mathcal{G}} AS\, X$ iff $\sigma, \tau \not\models_{\mathcal{G}} NZ(S^\omega \setminus X)$. Together with the duality between reachability and safety, we see that queries are closed under complementation.

We refer to nontrivial Boolean combinations as *multi-objective queries* and an atomic AS or NZ query as a *single-objective query*.

Given a game $\mathcal{G}$ and query ϕ, a strategy σ of player 1 is a *winning* strategy if and only if for every strategy τ of player 2, ϕ is satisfied, that is, $\sigma, \tau \models_{\mathcal{G}} \phi$ holds. Player 1 is *winning* if they have a winning strategy: $\exists\sigma\forall\tau : \sigma, \tau \models_{\mathcal{G}} \phi$.

Player 2 is *winning* if they have a winning strategy for the negated objective $\exists\tau\forall\sigma : \sigma, \tau \models_{\mathcal{G}} \neg\phi$.

For a given query ϕ, the set $[[\phi]]_1 = \{s \in S | \exists\sigma\forall\tau\sigma, \tau \models_{\mathcal{G}_s} \phi\}$ of all states s such that player 1 has a winning strategy if the game starts at s is called the *winning region* of ϕ.

A query ϕ is *determined* if in every game $\mathcal{G}$ with query ϕ, either player 1 is winning or player 2 is winning. (We assume the queries and games share the same set of states.) Determinacy is a non-trivial property of stochastic games. This is because the logical negation of "player 1 is winning" is that $\forall\sigma\exists\tau : \sigma, \tau \models_{\mathcal{G}} \neg\phi$, which only ensures that player 2 has a *spoiling strategy* for any player 1 strategy. This does not mean that player 2 has a single winning strategy for every player 1 strategy. The following results hold.

Proposition 1 (Determinacy).
(1) *[9, 10] Games with $AS\, X$ and $NZ\, X$ objectives for safety and reachability objectives X are determined.*
(2) *[Determinacy Argument] A query ϕ is determined if there exists a determined query ϕ' such that for every game $\mathcal{G}$:*
  - *If player 1 wins ϕ' then they win ϕ*
  - *If player 2 wins ϕ' then they also win ϕ.*

The first result follows from a general determinacy theorem for stochastic games for all Borel objectives [22]. The second result follows because either player 1 or player 2 wins the game with query ϕ', and this determines who wins the game with query ϕ.

In our proofs we will sometimes restrict the game $\mathcal{G}$ to only a subset of S, replacing all other states with a new terminal state $s_\perp$.

Definition 3 (restricted game). *Let $\mathcal{G}$ be a stochastic game, and $U \subseteq S$ be a subset of states. We define game $\mathcal{G}|U$ restricted to U as: $\langle U \cup \{s_\perp\}, s_0', A', P'\rangle$ where $s_\perp$ is a new terminal state, with:*

$$s_0' = s_0 \text{ if } s_0 \in U,\ s_0' = s_\perp \text{ otherwise}$$

$$A'(s) = A(s) \text{ if } A(s) \subseteq U,\ A'(s) = (A(s) \cap U) \cup s_\perp \text{ otherwise}$$

$$P'(s)(s') = P(s)(s') \text{ if } s' \in U,\ P'(s)(s_\perp) = \sum_{s' \notin U:} P(s)(s')$$

If X is an objective with target set T in $\mathcal{G}$ then we consider it in $\mathcal{G}|U$ with the target set $T' = T \cap U$.

3 DETERMINED QUERIES

In this section, we prove the following main theorem.

Theorem 1. *Consider the following class of queries:*
(1) *Conjunctions of AS and NZ queries for both reachability and safety objectives;*
(2) *Positive Boolean combinations of AS queries for reachability and safety objectives.*

Both classes of queries are determined. Further, deciding if player 1 wins a game is PSPACE-complete.

This result solves the open problem from [16] of the precise complexity of rational verification for reachability/safety objectives.

By negating the classes and using determinacy, we conclude the following classes are also determined, and have the same PSPACE-completeness complexity to determine the winner:
(1) Disjunctions of AS and NZ queries for both reachability and safety objectives;
(2) Positive Boolean combinations of NZ queries for reachability and safety objectives.

We prove Theorem 1 using a sequence of lemmas, building up to the main result.

3.1 Conjunctions

We start with the first class of conjunctive queries.

Lemma 2 (Conjunction of nonzero). *Let $\phi = \bigwedge_{i \in I} NZ(X_i)$, where $X_i \subset S^\omega$ is any objective. ϕ is determined and player 1 has a winning strategy for ϕ if any only if player 1 has a winning strategy for each $\phi_i = NZ(X_i)$.*

Proof. Since each ϕ_i is determined, either player 1 has a winning strategy σ_i for each ϕ_i, or there is some ϕ_i for which player 2 has a winning strategy τ_i.

If player 1 has a winning strategy σ_i for each i, then the strategy σ that randomizes uniformly between each σ_i at the start of the game is a winning strategy for ϕ, since for all strategies τ of player 2 we have that $\mathbb{P}^{\sigma,\tau}(X_i) = \frac{1}{|I|}\mathbb{P}^{\sigma_i,\tau}(X_i) > 0$.

If player 2 has a winning strategy τ_i for some ϕ_i, then this strategy is also a winning strategy for ϕ, since for any strategy σ of player 1 we have that $\sigma, \tau_i \not\models \phi_i$ and therefore $\sigma, \tau_i \not\models \phi$.

It follows that ϕ is determined, and player 1 has a winning strategy if and only if they have a winning strategy for each ϕ_i. ◻

The following lemma shows that we can reduce a conjunction of multiple almost sure reachability queries to a single almost sure reachability query.

Lemma 3 (Conjunction of *AS*). *Let $\phi = \bigwedge_{i=1}^{n} AS(\Diamond T_i)$ be a conjunction of almost sure reachability queries. ϕ is determined and player 1 has a winning strategy for ϕ if and only if player 1 has a winning strategy for $\phi' = AS(\Diamond T')$ where*

$$T' = \bigcup_{i=1}^{n} T_i \cap [[\bigwedge_{j \neq i} AS(\Diamond T_j)]]_1$$

Proof. The set T' is a subset of the union of all target states $T = \bigcup_{i=1}^{n} T_i$ such that if $s \in T' \cap T_i$ is reached, player 1 can guarantee that all other target sets T_j $(j \neq i)$ are reached almost surely from each s. We show by induction over n that the lemma holds for any conjunction with up to n objectives. For the base case, we have since $T' = T$ and $\phi' = \phi = AS(\Diamond T_1)$, so the lemma holds. Now assume the lemma holds for any conjunctions of up to k objectives and consider the case of $n = k + 1$.

Suppose player 1 has a winning strategy σ for $AS(\Diamond T')$. Consider the following strategy: player 1 starts playing σ until a state $s \in T'$ is reached. Suppose $s \in T_i \cap [[\bigwedge_{j \neq i} AS(\Diamond T_j)]]_1$ for some i (pick any i in case multiple apply). From s player 1 switches to playing the strategy from $[[\bigwedge_{j \neq i} AS(\Diamond T_j)]]_1$.

By the induction hypothesis, each subquery $\bigwedge_{j \neq i} AS(\Diamond T_j)$ is determined. Consider the following strategy: player 2 plays a winning strategy for $\Diamond T'$. If the play reaches some $s \in T_i \setminus T'$ for some i, player 2 switches to a winning strategy for $\bigwedge_{j \neq i} AS(\Diamond T_j)$ from s.

It follows that ϕ is determined and player 1 has a winning strategy if an only if they have a winning strategy for $AS(\Diamond T')$. Inductively, this holds for conjunctions over any number of objectives. The complete proof is in the full version [15]. ◻

Next, we extend the results to conjunctions of both *AS* and *NZ* reachability. We start with the special case where there is exactly one *NZ* reachability query.

Lemma 4 (Conjunction of AS and one NZ reachability). *Let $\phi = NZ(\Diamond T_0) \wedge \bigwedge_{i=1}^{n} AS(\Diamond T_i)$ be a conjunction of one nonzero and multiple almost sure reachability queries. ϕ is determined, and deciding if player 1 has a winning strategy is PSPACE-complete.*

Proof. We first construct a nonstochastic reachability game $\langle \mathcal{G}^*, F \rangle$ and show that for each player, winning the query ϕ in $\mathcal{G}$ is equivalent to winning the reachability game $\langle \mathcal{G}^*, F \rangle$. Since nonstochastic reachability games are determined, it follows that ϕ is determined.

We construct the goal unfolding of $\mathcal{G}$ (see also [26], [12], [8]). This is a game constructed from $\mathcal{G}$ with state space $S \times \{0,1\}^{n+1}$, such that in a state (s, b), the vector b tracks which target sets T_i, $i \in \{0, \ldots, n\}$ have already been visited during a play. For each state $s \in S$, let $I_s = \{i \in \{0, \ldots n\} | s \in T_i\}$ be the index set of targets that include s. We define the goal unfolding formally as $\mathcal{G}' = \langle S', s'_0, A', P' \rangle$, where:

$$\begin{aligned} S' &= S \times \{0,1\}^{n+1} \\ s'_0 &= (s_0, (0, \ldots, 0)) \\ A'((s,b)) &= \{(s', b') |\ s' \in A(s), \\ &\qquad \forall i \in I_s : b'_i = 1, \forall i \notin I_s : b'_i = b_i\} \end{aligned}$$

$$P'((s,b))((s',b')) = \begin{cases} P(s)(s') & \text{if } \forall i \in I_s : b'_i = 1, \forall i \notin I_s : b'_i = b_i \\ 0 & \text{otherwise} \end{cases}$$

Note that we can identify paths in $\mathcal{G}'$ with paths in $\mathcal{G}$ via projection or by augmenting each state with the vector of previously visited target sets. Similarly we can identify strategies in $\mathcal{G}'$ with strategies in $\mathcal{G}$.

Consider in $\mathcal{G}'$ the query $\phi' = NZ(\Diamond T'_0) \wedge \bigwedge_{i=1}^{n} AS(\Diamond T'_i)$, where $\forall i \in \{0, \ldots, n\} : T'_i = \{(s, b) | s \in S, b_i = 1\}$. This query is equivalent to the query ϕ in $\mathcal{G}$, since for any pair of strategies σ, τ it holds that $\sigma, \tau \models_{\mathcal{G}'} \phi'$ if and only if $\sigma, \tau \models_{\mathcal{G}} \phi$. Since all target sets T'_i are absorbing, i.e. any play that reaches T'_i will stay in T'_i, it follows that $\sigma, \tau \models \bigwedge_{i=1}^{n} AS(\Diamond T'_i)$ if and only if $\sigma, \tau \models AS(\Diamond T')$ where $T' = \{(s, b) | s \in S, \forall i \neq 0 : b_i = 1\}$.

Let $M = [[AS(\Diamond T')]]_1$ be the winning region of player 1 for $AS(\Diamond T')$ in $\mathcal{G}'$, and let $\mathcal{G}' | [[AS(\Diamond T')]]_1$ be the game restricted to that set. Finally, let $\langle \mathcal{G}^*, \Diamond T'_0 \rangle$ be the nonstochastic reachability game obtained from the restricted game $\mathcal{G}' | [[AS(\Diamond T')]]_1$ by giving control of the stochastic states to player 1.

We now show that player 1 has a winning strategy for ϕ' in $\mathcal{G}'$ if and only if they win $\langle \mathcal{G}^*, \Diamond T'_0 \rangle$. Since $\langle \mathcal{G}^*, \Diamond T'_0 \rangle$ is a nonstochastic reachability game it is determined and winning strategies are deterministic and memoryless (in $\mathcal{G}^*$).

Assume player 2 has a winning strategy τ^* for $\langle \mathcal{G}^*, \Diamond T'_0 \rangle$, and let τ be the deterministic strategy for player 2 in $\mathcal{G}'$ that plays according to τ^* until some state $s \notin M$ is reached, and then switches to a winning strategy τ_s for player 2 for $AS(\Diamond T')$. Let σ be any strategy for player 1 in $\mathcal{G}'$ and assume that $\sigma, \tau \models_{\mathcal{G}'} \phi'$. It follows that there is a play π with $\mathbb{P}^{\sigma,\tau}(\pi) > 0, \pi \in \bigcap_{i=0}^{n} \Diamond T'_i$. If π would reach a state $s \notin M$, then player 2 would play τ_s and ensure $\sigma, \tau \not\models_{\mathcal{G}'} \bigwedge_{i=1}^{n} AS(\Diamond T'_i)$. Therefore π is a play in M. Let π^* be the corresponding play in $\mathcal{G}^*$ and let σ^* be the strategy that on each prefix of π^* plays the next state on π^* (player 1 plays at all states (s, b) with $s \in S'_1 \cup S'_c$). Then it follows that $\pi^* \in \Diamond T'_0$. Since playing σ^* against τ^* in $\mathcal{G}^*$ results in the path π^*, player 1 wins $\langle \mathcal{G}^*, T'_0 \rangle$, which is a contradiction to τ^* being a winning strategy for $\langle \mathcal{G}^*, \Diamond T'_0 \rangle$ for player 2. Therefore it follows that $\sigma, \tau \not\models_{\mathcal{G}} \phi'$, and τ is a winning strategy for player 2 for ϕ' in $\mathcal{G}$.

Assume now instead that player 1 has a winning strategy σ^* for $\langle \mathcal{G}^*, \Diamond T'_0 \rangle$ and let τ^* be any strategy of player 2. Without loss of generality, both σ^* and τ^* are memoryless strategies. Then the play according to σ^*, τ^* is a finite simple path π^* that ends in $s \in T'_0$ and therefore does not contain $s_\perp$. The union of all these simple paths form a tree Π that branches universally on states of player 2. Let σ be the strategy for player 1 in $\mathcal{G}$ that plays according to σ^* on Π and switches to playing σ_s after leaving Π at s or after reaching a leaf s of Π, where σ_s is the winning strategy for $\bigwedge_{i=1}^{n} AS(\Diamond T'_i)$ from s. Note that a play according to σ only leaves Π at stochastic states $s \in S'_c \cap M$. Let τ be any strategy of player 2. Because Π branches universally at states of player 2 and any play reaching a stochastic state $s \in \Pi$ has a nonzero chance to stay in Π, there is a root-to-leaf path in Π that has positive probability according to σ, τ. It follows that $\sigma, \tau \models_{\mathcal{G}'} NZ(\Diamond T_0')$. Since any play with positive probability eventually either reaches a leaf of $\Pi \subset M$ or leaves Π at some stochastic state $s \in S_c' \cap M$, player 1 eventually plays a strategy σ_s that is a winning strategy for $\bigwedge_{i=1}^{n} AS(\Diamond T_i')$. It follows that $\sigma, \tau \models_{\mathcal{G}'} \bigwedge_{i=1}^{n} AS(\Diamond T_i')$. Therefore $\sigma, \tau \models_{\mathcal{G}'} \phi'$ and σ is a winning strategy for ϕ' in $\mathcal{G}'$.

Therefore ϕ' in $\mathcal{G}'$ is determined, and player 1 has a winning strategy if and only if they win the reachability game $\langle \mathcal{G}^*, \Diamond T_0' \rangle$.

For complexity, note that the reachability game $\langle \mathcal{G}^*, \Diamond T_0' \rangle$ is constructed from the goal unfolding of $\mathcal{G}'$ and therefore the state space is exponential in terms of the number of almost sure target sets n. However, because along any path π, the value of b_i can only change from 0 to 1 once, the length of any simple path in $\mathcal{G}^*$ is at most of length $(n+2)|S|$. Therefore the depth of the tree Π spanned by any winning strategy in $\mathcal{G}^*$ is only linear in the initial state space S, and we can use a polynomial space algorithm to verify if a winning strategy exists by checking if there is a tree in $\mathcal{G}^*$ that branches universally at states of player 2, where each leaf is in T_0'. The algorithm can be found in the full version [15]. PSPACE-hardness follows from the multiple almost sure reachability problem in MDPs [23]. □

We now show the general case.

Lemma 5 (Conjunction of AS and NZ reachability). *Let $\phi = \bigwedge_{i \in I} AS(\Diamond T_i) \wedge \bigwedge_{j \in J} NZ(\Diamond T_i)$ be a conjunction of almost sure and nonzero reachability queries. ϕ is determined, and deciding if player 1 has a winning strategy is PSPACE-complete.*

Proof. We show that ϕ is determined and that player 1 has a winning strategy for ϕ if and only if they have a winning strategy for each $\phi_j = NZ(\Diamond T_j) \wedge \bigwedge_{i \in I} AS(\Diamond T_i)$, $j \in J$. The rest follows from Lemma 4.

Assume that for each j, player 1 has a winning strategy $\sigma_j$for ϕ_j. Then the strategy σ that plays each σ_j with probability $\frac{1}{|J|}$ is a winning strategy for ϕ: Let τ be any strategy of player 2, then $\sigma_j, \tau \models \bigwedge_{i \in I} AS(\Diamond T_i)$ for each $j \in J$, therefore $\sigma, \tau \models \bigwedge_{i \in I} AS(\Diamond T_i)$, and $\sigma_j, \tau \models NZ(\Diamond T_j)$ therefore $\sigma, \tau \models NZ(\Diamond T_j)$ for all $j \in J$.

Assume now instead that for some j, player 2 has a winning strategy τ_j for ϕ_j. For any strategy σ of player 1, it follows that $\sigma, \tau_j \not\models \phi_j$ and therefore $\sigma, \tau_j \not\models \phi$. Therefore τ_j is a winning strategy for player 2 for ϕ.

It follows that ϕ is determined and player 1 has a winning strategy if and only if they have a winning strategy for each ϕ_j. □

Note that the approach from Lemma 5 that generalizes from a single nonzero reachability objective to conjunctions involving multiple nonzero reachability objectives also works for nonzero safety objectives.

Finally, we add safety conditions. For almost sure safety, first note that for any pair of strategies σ, τ, it holds that $\sigma, \tau \models \bigwedge AS(\Box T_i)$ iff $\sigma, \tau \models AS(\bigcap \Box T_i)$ and $\bigcap \Box T_i = \Box \bigcap T_i$. Therefore conjunctions of multiple almost sure safety queries can be replaced with a single almost sure safety query.

Lemma 6 (Almost sure safety). *Let $\phi = \psi \wedge AS(\Box T)$, where ψ is positive Boolean formula over almost sure and nonzero reachability queries. Let $\mathcal{G}' = \mathcal{G}|[\![AS(\Box T)]\!]_1$ be the game restricted to the winning region of $AS(\Box T)$. Then ϕ is determined, and player 1 has a winning strategy if and only if player 1 has a winning strategy for ψ in $\mathcal{G}'$.*

We give a proof in the full version [15]. The proof only shows that Lemma 6 holds for conjunctions with reachability targets. However, since almost sure safety in stochastic games is equivalent to safety in the non-stochastic game where player 2 is given control of the stochastic states, Lemma 6 actually holds for all ψ.

For nonzero safety, we use a result from [25], which shows that in conjunctions, we can replace a nonzero safety objective with a nonzero reachability objective. Note that in general this result only applies to nonzero safety objectives where a play that leaves the target set T can not enter T again later. The construction of the goal unfolding ensures that this holds.

Lemma 7 (Conjunction of AS and one NZ safety). *Let $\phi = NZ(\Box T_0) \wedge \bigwedge_{i=1}^{n} AS(\Diamond T_i)$ be a conjunction of one nonzero safety query and multiple almost sure reachability queries. ϕ is determined, and deciding if player 1 has a winning strategy is PSPACE-complete.*

Proof. Similar to the proof of Lemma 4, we can construct the goal unfolding $\mathcal{G}'$, and consider the equivalent query

$$\phi' = NZ(\Box T_0') \wedge \bigwedge_{i=1}^{n} AS(\Diamond T_i')$$

with $T_i' = \{(s,b) | s \in S, b_i = 1\}$ for $i \in \{1, \dots, n\}$ as in Lemma 4 and $T_0' = \{(s,b) | s \in S, b_0 = 0\}$. Here we can apply a result from [25] that states that player 1 wins ϕ' if and only if they win $\phi'' = NZ(\Diamond T_0'') \bigwedge_{i=1}^{n} AS(\Diamond T_i')$ where

$$T_0'' = [\![NZ(\Box T_0')]\!]_1 \cap \bigcap_{i=1}^{n} T_i'$$

A winning strategy σ for ϕ'' reaches a state $(s,b) \in [\![NZ(\Box T_0')]\!]_1 \cap \bigcap_{i=1}^{n} T_i$ with positive probability. Since $b_0 = 0$, it follows that for each state (s', b') in the play so far $b_0' = 0$ and therefore the current play is in $\Box T_0'$. Since the almost sure reachability queries are already satisfied, player 1 can then switch to playing the winning strategy from $[\![NZ(\Box T_0')]\!]_1$ to satisfy $NZ(\Box T_0')$.

The rest of the lemma follows from Lemma 4, since now ϕ'' is a conjunction of one nonzero reachability and multiple almost sure reachability objectives. □

Finally, the proof for Theorem 1 for a conjunction ϕ of almost sure and nonzero reachability and safety queries is as follows:

1. For each nonzero query $NZ(X_i)$ in ϕ, consider the conjunction of $NZ(X_i)$ and all almost sure queries in the goal unfolding.
2. Use Lemma 6 to remove almost sure safety queries by moving to a restricted game.
3. If $X_i = \Box T_i$, use Lemma 7 to replace any nonzero safety queries with nonzero reachability queries.
4. Use Lemma 4 to check if player 1 has a winning strategy for the resulting conjunction of one nonzero and multiple almost sure reachability queries.

(5) With Lemma 3, player 1 has a winning strategy for ϕ if and only if they have a winning strategy for each of these conjunctions.

3.2 Positive Boolean combinations

We now move on to positive Boolean combinations. First note that we can negate Lemma 2 to get the following corollary for disjunctions over multiple almost sure queries.

Corollary 8 (Disjunction Almost-sure). *Let ϕ be a disjunction of almost sure queries. Then ϕ is determined and player 1 has a winning strategy for $\phi = \bigvee_{i=1}^{n} AS(X_i)$ if and only if they have a winning strategy for some $\phi_i = AS(X_i)$.*

Determinacy follows because the negation of a determined query is still determined. A separate proof for this can also be found in [26]. Together with Lemma 3, it follows that positive Boolean combinations of almost sure reachability and safety queries are determined and PSPACE-complete.

Lemma 9 (Positive Boolean AS-reachability and AS-safety). *Let ϕ be a positive Boolean formula over almost-sure reachability and safety queries. ϕ is determined, and deciding which player has a winning strategy is PSPACE-complete.*

Proof. For determinancy convert ϕ to DNF and note that for any conjunction of almost sure objectives it holds that $\sigma, \tau \models AS(X_1) \wedge AS(X_2)$ if and only if $\sigma, \tau \models AS(X_1 \wedge X_2)$. Therefore ϕ is equivalent to a disjunction of almost sure queries. Determinacy follows from Corollary 8.

For membership in PSPACE, consider the following algorithm. First, nondeterministically guess a satisfying assignment to ϕ and Let $\phi' = \bigwedge_{i \in I} AS(\Diamond T_i) \wedge \bigwedge_{j \in J} AS(\Box T_j)$ be a conjunction over the positive variables in that assignment. From Lemma 6, we know that player 1 has a winning strategy if and only if they have a winning strategy for $\phi'' = \bigwedge_{i \in I} AS(\Diamond T_i)$ in the game $\mathcal{G}|[[AS(\Box \bigcap_{j \in J} T_j)]]_1$. Since ϕ'' is a conjunction of almost sure reachability queries, we can decide if player 1 has a winning strategy in PSPACE using Lemma 3. If player 1 has no winning strategy for any satisfying assignment of ϕ, then it follows from Corollary 8 that player 1 has no winning strategy for ϕ. □

4 NONDETERMINED QUERIES

In this section, we show that the subclasses shown to be determined in Theorem 1, and their negations, form a maximal class. That is, games with queries outside these classes are not determined: neither player may have a winning strategy. Moreover, we show that the decision problem of determining if player 1 can win is at least NEXPTIME-hard.

We start with an example of a nondetermined query. Consider the with the game from Figure 1. We show that the query $\phi = AS(\Diamond\{s_3\}) \vee (NZ(\Diamond\{s_2\}) \wedge NZ(\Diamond\{s_4\}))$ is not determined. The available strategies for both players in this game can be characterized as follows: player 1 either plays a strategy σ_1 that plays s_1 with probability 1, or a strategy σ_2 that plays s_2 with probability greater than 0. Symmetrically player 2 can either play a strategy τ_1 than plays s_3 with probability 1, or a strategy τ_2 that plays s_4 with positive probability. It follows that $\sigma_1, \tau_1 \models \phi$ and $\sigma_2, \tau_2 \models \phi$, but $\sigma_1, \tau_2 \not\models \phi$ and $\sigma_2, \tau_1 \not\models \phi$. Therefore there are no winning strategies for either player, and ϕ is not determined.

This nondeterminancy example extends to the following classes of queries:

(1) positive Boolean combinations of AS and NZ reachability (example);
(2) Boolean combination of AS reachability with

$$\phi' = AS(\Diamond\{s_3\}) \vee (\neg AS(\Diamond\{s_3, s_4\}) \wedge NZ(\Diamond\{s_4\}))$$

(3) Boolean combinations of NZ reachability

$$\phi'' = \neg NZ(\Diamond\{s_2, s_4\}) \vee (NZ(\Diamond\{s_2\}) \wedge NZ(\Diamond\{s_4\}))$$

as well as for the qualitative multiple safety queries that arise from their respective negations.

4.1 Hardness

While the determined queries in the previous sections were PSPACE-complete, we show the extended classes of queries are NEXPTIME-hard. To show NEXPTIME-hardness, we use a reduction from Boolean formulas with Henkin quantifiers, known as *Dependency Quantified Boolean Formulas (DQBF)* [2]. In comparison to a regular QBF, where a quantified variable always depends on exactly the previously quantified variables, in DQBF this dependency can be specified explicitly. We are concerned with S-form DQBFs:

$$\Phi = \forall x_1, \dots, \forall x_n \exists y_{1,S_1}, \dots, \exists y_{m,S_m} \phi$$

where $X = \{x_1, \dots, x_n\}$ are the universally quantified variables, $Y = \{y_1, \dots, y_m\}$ the existentially quantified variables, and for each $j \in \{1, \dots, m\}$, the set $S_j \subset X$ is the set of variables that y_j depends on. This formula is satisfied if and only if there are Skolem functions $Y_j : S_j \to \{0, 1\}$, such that the Skolemization $\forall x_1, \dots x_n : \phi[y_j \to Y_j(X)]$ is satisfied, where $\phi[y_j \to Y_j(X)]$ denotes ϕ with each y_j replaced by $Y_j(X)$. Satisfiability of a DQBF in S-Form is NEXPTIME-complete [2].

Theorem 10 (reduction from DQBF). *Let ϕ be a positive Boolean combination of AS and NZ reachability queries. Deciding if player 1 has a winning strategy is NEXPTIME-hard.*

We describe the construction informally here, the complete proof can be found in the full version [15]. For a DQBF Φ in S-Form:

$$\Phi = \forall x_1, \dots, \forall x_n \exists y_{1,S_1}, \dots, \exists y_{m,S_m} \phi$$

We construct a stochastic game where the initial state s_0 randomizes between m branches.

Each branch $j \in \{1, \dots, m\}$ consists of $n + m$ modules, one for each variable. In each module for a variable $v \in X \cup Y$ we have a state controlled by player 1 if $v \in Y$ and by player 2 if $v \in X$, where the player chooses between a state that corresponds to setting the variable v to true, and a state corresponding to setting the variable v to false.

The modules in each branch are ordered according to the dependency induced by S_j: Each branch j begins with the modules for $x \in S_j$, followed by the module for y_j, then the remaining modules $x \notin S_j$ and finally the modules y_i, $i \neq j$. The winning condition for the game ψ is constructed by augmenting the formula ϕ with restrictions ψ_1 and ψ_2 that ensure that each player plays deterministically and identically on each branch:

$$\psi = (\phi' \wedge \psi_1) \vee \psi_2$$

where ϕ' is obtained from ϕ by replacing each positive variable v with the almost sure reachability query for the states that set v to true, and $\neg v$ with the almost sure reachability query for the states that set v to false.

Player 1 wins if player 2 does not follow the restriction (making ψ_2 true) or if they follow the restriction (making ψ_1 true) and $\sigma, \tau \models \phi'$. To show that player 1 has a winning strategy for ψ in $\mathcal{G}_\Phi$, we use that a strategy of player 2 that follows these restrictions can be translated to an assignment to the variables X, and a strategy of player 1 strategy that satisfies these restrictions can be translated to Skolem functions $Y_j : S_j \longrightarrow \{0, 1\}$.

Essentially the same proof works for any of the other query types previously shown to be nondetermined (e.g. Boolean combinations of AS reachability, Boolean combinations of NZ reachability, etc.).

4.2 Membership

While in general a strategy is a function depending on the entire play of the game so far, often strategies only need to remember limited information about the history of the play. A strategy can be realized by a *strategy automaton* [26] with a state space M called memory. In each round, the strategy automaton updates its memory state $m \in M$ (potentially in a probabilistic way) based on the new state s and, if it is the respective player's turn, outputs a distribution $\Delta(S)$ based on its memory state m and the game state s. For a strategy σ, the smallest $k \in \mathbb{N} \cup \{\infty\}$ such that there is a strategy automaton realizing σ with $|M| = k$ is the memory size of σ.

To put an upper bound on the complexity, we aim at bounding the memory requirement: Given a bound on the memory of a winning strategy, we can nondeterministically guess a strategy of that bound and check if the strategy is winning in the induced MDP using the algorithms of [11].

We will show that for positive Boolean combinations that include $AS(\Diamond)$, $AS(\Box)$ and $NZ(\Diamond)$ queries only require exponential memory, and therefore deciding if player 1 has a winning strategy is in NEXPTIME. For queries that also include $NZ(\Box)$ the memory requirement, and hence decidability/complexity remains open.

In generalized reachability games [12], it is known that winning strategies only need to remember the set of target sets previously visited. We show that this is not sufficient for our class of queries.

Consider in the game from Figure (2) the objective

$$(NZ(\Diamond A) \wedge NZ(\Diamond B)) \vee (AS(\Diamond\{A, B, D\}) \wedge NZ(\Diamond D)) \\ \vee (AS(\Diamond\{A, B, C\}) \wedge NZ(\Diamond C))$$

Since all target sets are terminal states, if memory of the target sets visited was sufficient, then player 1 should have a winning strategy if and only if they have a memoryless winning strategy. If player 2 plays both s_1 and s_2 with positive probability, then $(NZ(\Diamond A) \wedge NZ(\Diamond B))$ and therefore ϕ is satisfied, regardless of player 1s actions. Therefore we only have to consider spoiling strategies τ_A andτ_B that play s_1 and s_2 with probability 1 respectively. For player 1, any strategy that randomizes between C and D will fail to satisfy ϕ against both τ_B and τ_A. Therefore there are two possible memoryless strategies for player 1: σ_C and σ_D which play C and D with probability 1 respectively, and neither is a winning strategy: $\sigma_C, \tau_B \not\models \phi$ and $\sigma_D, \tau_A \not\models \phi$. Player 1 does however have a winning strategy that uses memory: If σ plays C with probability 1 if s_1 has been played, and plays D with probability 1 if s_2 has been played. This strategy wins against τ_A by playing C, wins against τ_B by playing D and trivially wins against any randomization.

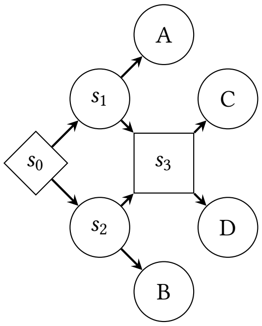

Figure 2: Example game where a winning strategy requires memory other than visited target sets.

For queries that do not include $NZ(\Box)$, we show that remembering the subset of all visited states (not just target sets) is sufficient for a winning strategy. To show this, we assume there is a winning strategy σ (with arbitrary memory) and construct a strategy $\bar{\sigma}$ from σ that only uses the set of previously visited states as memory, and show that $\bar{\sigma}$ is still a winning strategy.

Let σ be a strategy. For a play π, let $set(\pi) = \{s \in S \mid \exists i : \pi_i = s\}$ be the set of states visited by π. For a state $s \in S$ and set of states $M \subset S$, define the set $resp_\sigma(s, M)$ as:

$$\{s' \in A(s) \mid \exists \tau', \pi' : \mathbb{P}^{\sigma,\tau'}(\pi' s) > 0 \; set(\pi') = M \sigma(\pi' s)(s') > 0\}$$

i.e., the actions that are played with positive probability by σ after some play that reaches s and visits exactly the states in M. Let $\bar{\sigma}$ be the strategy derived from σ where:

$$\bar{\sigma}(\pi s)(s') = \begin{cases} \frac{1}{|resp_\sigma(s, set(\pi))|} & \text{if } s' \in resp_\sigma(s, set(\pi)) \\ 0 & \text{otherwise} \end{cases}$$

Lemma 11. *The strategy $\bar{\sigma}$ derived from σ is well defined, and only uses exponential memory.*

Proof. $\bar{\sigma}(\pi s)$ only depends on $resp_\sigma(s, set(\pi))$ and so only on s and $set(\pi)$. Thus, the only memory $\bar{\sigma}$ requires is the set of previously visited states, which is exponential in the number of states.

Let τ be any strategy for player 2 and πs be a path such that $\mathbb{P}^{\bar{\sigma},\tau}(\pi s) > 0$. We show that the set $resp_\sigma(s, set(\pi))$ is non-empty and thus $\bar{\sigma}$ is well defined. For this we prove:

$$\exists \tau', \pi' : Pr^{\sigma,\tau'}(\pi' s) > 0, \; set(\pi) = set(\pi') \qquad (*)$$

by induction over finite paths πs.

$(*)$ trivially holds for $\pi s = \langle s_0 \rangle$ with $\tau' = \tau$ and $\pi' = \pi$.

Assume $(*)$ holds for some πs with strategy τ' and alternative path π', and consider the path $\pi s s'$ for $s' \in A(s)$:

If $s' \in S_c$, then by assumption $\mathbb{P}^{\sigma,\tau'}(\pi' s) > 0$ and thus $\mathbb{P}^{\sigma,\tau'}(\pi' s s') = \mathbb{P}^{\sigma,\tau'}(\pi' s) P(s)(s') > 0$. Since $set(\pi) = set(\pi')$ it also follows that $set(\pi s) = set(\pi' s')$ and so $(*)$ holds for $\pi s s'$.

If $s' \in S_2$ and $\mathbb{P}^{\bar{\sigma},\tau}(\pi s s') > 0$, then consider the strategy τ'' that plays like τ' on the path π' and then plays like τ everywhere else. It follows that $\mathbb{P}^{\sigma,\tau''}(\pi' s s') = \mathbb{P}^{\sigma,\tau'}(\pi' s)\tau(\pi s)(s') > 0$. Since $set(\pi) = set(\pi')$ by assumption, it also follows that $set(\pi s) = set(\pi' s)$ and therefore $(*)$ holds for $\pi s s'$.

If $s' \in S_1$ and $\mathbb{P}^{\bar{\sigma},\tau}(\pi s s') > 0$, then it follows that $s' \in resp_{\sigma}(s, set(\pi))$. By construction there exist τ', π' such that $\mathbb{P}^{\sigma,\tau'}(\pi' s) > 0$, $set(\pi) = set(\pi')$, and $\sigma(\pi' s)(s) > 0$. It follows that $\mathbb{P}^{\sigma,\tau'}(\pi s s') > 0$ and $set(\pi s) = set(\pi' s)$, so $(*)$ holds for $\pi s s'$.

It follows that for any path πs with $\mathbb{P}^{\bar{\sigma},\tau}(\pi s) > 0$, the set $resp_{\sigma}(s, set(\pi))$ is non-empty and therefore $\bar{\sigma}$ is well defined. □

Lemma 12 (exponential memory). *Let ϕ be a positive Boolean combination of AS reachability, AS safety and NZ reachability queries. If σ is a winning strategy for ϕ, then $\bar{\sigma}$ is a winning strategy for ϕ.*

Proof. We prove that $\bar{\sigma}$ is a winning strategy by assuming that there is a strategy τ such that $\bar{\sigma}, \tau \not\models \phi$ and showing a contradiction by constructing τ^* with $\sigma, \tau^* \not\models \phi$.

This proof uses two important properties of $\bar{\sigma}$. First, from the previous proposition we know if there is a play πs according to $\bar{\sigma}, \tau$, then there is a play $\pi' s$ according to σ, τ' that visits the same set of states. Additionally from the construction of $\bar{\sigma}$ it is clear that any finite play that has positive probability according to σ, τ also has positive probability according to $\bar{\sigma}, \tau$.

From the second property it follows that for any target set T we have $\sigma, \tau \models NZ(\Diamond T) \implies \bar{\sigma}, \tau \models NZ(\Diamond T)$. Therefore there must be some almost sure safety or reachability queries ϕ_i, $i \in I^*$ in ϕ such that $\bar{\sigma}, \tau \not\models \phi_i$. Using the first property, we then construct a strategy τ_i^* for each of these queries such that $\sigma, tau_i^* \not\models \phi_i$, and let τ^* be the strategy that chooses to play any τ_i^* at random.

We then show that $\sigma, \tau^* \models NZ(\Diamond T) \implies \bar{\sigma}, \tau \models NZ(\Diamond T)$. It follows that $\sigma, \tau^* \not\models \phi$ which is a contradiction to σ being a winning strategy. Thus the assumption of $\bar{\sigma}$ not winning against τ is false, and $\bar{\sigma}$ is a winning strategy. Details are in the full version [15]. □

This construction does however not work with nonzero safety targets, as $\sigma, \tau \models NZ(\Box T) \implies \bar{\sigma}, \tau \models NZ(\Box T)$ does not hold. This can be seen in a simple example game where player 1 can choose to either stay in the initial state s_0 or choose to move to a terminal state s_1. A strategy σ_k, $k \in \mathbb{N}$ that randomizes between staying in s_0 and moving to s_1 for the first k iterations, and then chooses to play s_0 forever is a winning strategy for $NZ(\Box s_0)$, but $\bar{\sigma}$ constructed from σ would randomize uniformly between s_0 and s_1 at every step, eventually reaching s_1 with probability 1 and not satisfying $NZ(\Box s_0)$.

Since exponential memory is sufficient for positive Boolean combinations that do not include $NZ(\Box T)$ queries, we can decide if player 1 has a winning strategy in NEXPTIME.

Theorem 13. *Let ϕ be a positive Boolean combination of almost sure reachability, almost sure safety and nonzero reachability queries. Deciding if player 1 has a winning strategy is NEXPTIME-complete.*

Proof. From the previous proposition we know that if a winning strategy σ exists, then there exists a winning strategy $\bar{\sigma}$ that only uses exponential memory, and randomizes uniformly at every state. We guess such an exponentially sized strategy $\bar{\sigma}$, and verify in exponential time that player 2 does not have a winning strategy for $\neg\phi$ in the MDP $\mathcal{G}_{\bar{\sigma}}$, which can be done in exponential time using the algorithm from [11]. □

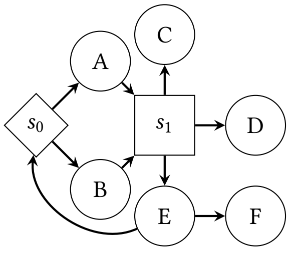

Figure 3: Example game where winning strategies require memory other than the set of visited states.

It remains open whether positive Boolean combinations including $NZ\Box$ can be decided in NEXPTIME, but we motivate why it is not trivial to extend our result. Consider again the previous example where $\bar{\sigma}$ fails for a simple $NZ(\Box T)$ in a game with 2 states. The two strategies $\sigma_1(s_0^k)(s_1) = \frac{1}{k}$ and $\sigma_1(s_0^k)(s_1) = \frac{1}{k^2}$ both play s_1 with positive probability at every step, but with arbitrarily small probability overall. However, the probability of reaching s_1 with σ_1 is $\lim \frac{k-1}{k} = 1$, while the probability of reaching s_1 with σ_2 is $\lim \frac{k-1}{2k} = \frac{1}{2} < 1$. A construction of $\bar{\sigma}$ that works for queries that include $NZ\Box$ would need to be able to distinguish between these two strategies.

Additionally, we show that simply remembering all visited states is not sufficient if we include $NZ(\Box)$ queries. Consider in the game from Figure (3) the objective $\phi = \phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4$ where:

$$\phi_1 = AS(\Diamond A) \wedge NZ(\Diamond B) \wedge NZ(\Diamond C) \wedge AS(\Box \overline{D})$$

$$\phi_1 = NZ(\Diamond A) \wedge AS(\Diamond B) \wedge AS(\Box \overline{C}) \wedge NZ(\Diamond D)$$

$$\phi_3 = NZ(\Box \overline{A}) \wedge NZ(\Box \overline{B})$$

$$\phi_4 = AS(\Diamond F) \wedge (AS(\Box \overline{A}) \vee AS(\Box \overline{B}))$$

ϕ_3 ensures that at the start of the game, player 2 plays either A or B with probability 1, while ϕ_4 ensures that if player 1 never plays C or D (and therefore the game almost surely reaches F), player 2 has to eventually both A and B. The strategy σ that plays E with probability 1 until both A and B have been reached and then plays C if A was reached before B and D otherwise is a winning strategy. However, a strategy that only uses the set of visited states as memory can not distinguish between these two cases, and there is no winning strategy with that memory structure. We do not know if a different exponential-sized memory structure is sufficient to solve queries that include $NZ(\Box)$ and therefore if these queries are also in NEXPTIME.

ACKNOWLEDGMENTS

We thank Daniel Stan and anonymous reviewers for their valuable feedback. This work is in part supported by the EU (ERC, LASD, grant number 101089343, https://doi.org/10.3030/101089343) and the DFG project 389792660 TRR 248–CPEC. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] Pranav Ashok, Krishnendu Chatterjee, Jan Křetínskỳ, Maximilian Weininger, and Tobias Winkler. 2020. Approximating values of generalized-reachability stochastic games. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. 102–115.

[2] Valeriy Balabanov, Hui-Ju Katherine Chiang, and Jie-Hong R. Jiang. 2014. Henkin quantifiers and Boolean formulae: A certification perspective o f DQBF. *Theoretical Computer Science* 523 (2014), 86–100. https://doi.org/10.1016/j.tcs.2013.12.020

[3] Raphaël Berthon, Shibashis Guha, and Jean-François Raskin. 2020. Mixing Probabilistic and non-Probabilistic Objectives in Markov Decision Processes. (2020), 195–208. https://doi.org/10.1145/3373718.3394805

[4] Romain Brenguier and Vojtech Forejt. 2016. Decidability Results for Multi-objective Stochastic Games. 9938 (2016), 227–243. https://doi.org/10.1007/978-3-319-46520-3_15

[5] Krishnendu Chatterjee and Thomas A. Henzinger. 2012. A survey of stochastic ω-regular games. *J. Comput. Syst. Sci.* 78, 2 (2012), 394–413. https://doi.org/10.1016/J.JCSS.2011.05.002

[6] Krishnendu Chatterjee, Marcin Jurdzinski, and Thomas A. Henzinger. 2004. Quantitative stochastic parity games. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, J. Ian Munro (Ed.). SIAM, 121–130. http://dl.acm.org/citation.cfm?id=982792.982808

[7] Taolue Chen, Vojtěch Forejt, Marta Kwiatkowska, Aistis Simaitis, Ashutosh Trivedi, and Michael Ummels. 2012. Playing stochastic games precisely. In *International Conference on Concurrency Theory*. Springer, 348–363.

[8] Taolue Chen, Vojtech Forejt, Marta Z. Kwiatkowska, Aistis Simaitis, and Clemens Wiltsche. 2013. On Stochastic Games with Multiple Objectives. In *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 8087)*, Krishnendu Chatterjee and Jirí Sgall (Eds.). Springer, 266–277. https://doi.org/10.1007/978-3-642-40313-2_25

[9] Luca de Alfaro. 1999. Computing Minimum and Maximum Reachability Times in Probabilistic Systems. In *CONCUR '99: Concurrency Theory, 10th International Conference, Eindhoven, The Netherlands, August 24-27, 1999, Proceedings (Lecture Notes in Computer Science, Vol. 1664)*, Jos C. M. Baeten and Sjouke Mauw (Eds.). Springer, 66–81. https://doi.org/10.1007/3-540-48320-9_7

[10] Luca de Alfaro and Thomas A. Henzinger. 2000. Concurrent Omega-Regular Games. In *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*. IEEE Computer Society, 141–154. https://doi.org/10.1109/LICS.2000.855763

[11] Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi, and Mihalis Yannakakis. 2008. Multi-Objective Model Checking of Markov Decision Processes. *Log. Methods Comput. Sci.* 4, 4 (2008). https://doi.org/10.2168/LMCS-4(4:8)2008

[12] Nathanaël Fijalkow and Florian Horn. 2010. The surprizing complexity of generalized reachability games. (Oct. 2010). https://hal.science/hal-00525762 working paper or preprint.

[13] Jerzy Filar and Koos Vrieze. 1997. *Competitive Markov Decision Processes - Theory, Algorithms and Applications*. Springer.

[14] Wan Fokkink. 2013. *Distributed Algorithms*. MIT Press.

[15] Moritz Graf, Anthony Lin, and Rupak Majumdar. 2026. Solving Qualitative Multi-Objective Stochastic Games. arXiv:2602.12927 [cs.GT] https://arxiv.org/abs/2602.12927

[16] Julian Gutierrez, Lewis Hammond, Anthony W. Lin, Muhammad Najib, and Michael J. Wooldridge. 2021. Rational Verification for Probabilistic Systems. In *KR*. 312–322.

[17] Marta Kwiatkowska, David Parker, and Clemens Wiltsche. 2016. PRISM-Games 2.0: A Tool for Multi-objective Strategy Synthesis for Stochastic Games. In *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings (Lecture Notes in Computer Science, Vol. 9636)*, Marsha Chechik and Jean-François Raskin (Eds.). Springer, 560–566. https://doi.org/10.1007/978-3-662-49674-9_35

[18] Ondrej Lengál, Anthony Widjaja Lin, Rupak Majumdar, and Philipp Rümmer. 2017. Fair Termination for Parameterized Probabilistic Concurrent Systems. In *TACAS (1) (Lecture Notes in Computer Science, Vol. 10205)*. 499–517.

[19] Anthony W. Lin and Philipp Rümmer. 2016. Liveness of Randomised Parameterised Systems under Arbitrary Schedulers. In *CAV (2) (Lecture Notes in Computer Science, Vol. 9780)*. Springer, 112–133.

[20] Nancy Lynch. 1996. *Distributed Algorithms*. Morgan Kaufmann.

[21] Rupak Majumdar, Kaushik Mallik, Anne-Kathrin Schmuck, and Damien Zufferey. 2020. Assume-Guarantee Distributed Synthesis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 39, 11 (2020), 3215–3226. https://doi.org/10.1109/TCAD.2020.3012641

[22] Donald A. Martin. 1998. The Determinacy of Blackwell Games. *J. Symb. Log.* 63, 4 (1998), 1565–1581. https://doi.org/10.2307/2586667

[23] Mickael Randour, Jean-François Raskin, and Ocan Sankur. 2017. Percentile queries in multi-dimensional Markov decision processes. *Formal Methods Syst. Des.* 50, 2-3 (2017), 207–248. https://doi.org/10.1007/S10703-016-0262-7

[24] L. S. Shapley. 1953. Stochastic Games. *Proceedings of the National Academy of Sciences* 39, 10 (1953), 1095–1100. https://doi.org/10.1073/pnas.39.10.1095 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.39.10.1095

[25] Daniel Stan, Muhammad Najib, Anthony Widjaja Lin, and Parosh Aziz Abdulla. 2024. Concurrent Stochastic Lossy Channel Games. , 46:1–46:19 pages. https://doi.org/10.4230/LIPICS.CSL.2024.46

[26] Tobias Winkler and Maximilian Weininger. 2021. Stochastic Games with Disjunctions of Multiple Objectives. 346 (2021), 83–100. https://doi.org/10.4204/EPTCS.346.6