Games with lexicographically ordered $\omega$-regular objectives

Véronique Bruyère, Quentin Hautem, and Jean-François Raskin

1 Computer Science Department, Université de Mons (UMONS), Belgium
2 Computer Science Department, Université libre de Bruxelles (ULB), Belgium

Abstract. In recent years, two-player zero-sum games with multidimensional objectives have received a lot of interest as a model for the synthesis of complex reactive systems. In this framework, player 1 wins if he can ensure that all objectives are satisfied against any behavior of player 2. It is however often natural to provide more significance to one objective over another, a situation that can be modeled with lexicographically ordered objectives. Inspired by recent work on concurrent games with $\omega$-regular objectives by Bouyer et al., we investigate in detail turned-based lexicographic $\omega$-regular games. We study the threshold problem which asks whether player 1 can ensure a payoff greater than or equal to a given threshold w.r.t. the lexicographic order. We provide precise results that refine and complete the ones by Bouyer et al. for turn-based games, including exact complexity classes, deterministic algorithms for computing the values, and tight memory requirements for those strategies. Whereas the threshold problem is $\text{PSPACE}$-complete for several $\omega$-regular objectives, our new algorithms show that it is fixed parameter tractable in those cases.

1 Introduction

Two-player zero-sum games played on directed graphs constitute an adequate framework for the synthesis of reactive systems facing an uncontrollable environment. To model properties to be enforced by the reactive system in its environment, games with Boolean objectives and games with quantitative objectives have been studied, for example games with $\omega$-regular objectives and mean-payoff games.

Recently, games with multidimensional objectives have received a lot of attention since in practice, a system must satisfy several properties. In this framework, the system wins if it can ensure that all objectives are satisfied against the environment. For instance, generalized parity games are studied in multi-mean-payoff games in, and multidimensional games with heterogeneous $\omega$-regular objectives in.

When modeling a system that must ensure multiple properties, an alternative is to assign more importance to one property over another. This is particularly relevant when a system cannot satisfy the conjunction of all properties at once. For instance, consider a system that strives to ensure three properties, say $p_1$, $p_2$, $p_3$ in that order of importance. Then satisfying $p_1$ but not $p_2$ and $p_3$ would be considered as more desirable than satisfying $p_2$ and $p_3$ but not $p_1$. This problem can be modeled by two-player games with lexicographically ordered objectives. Lexicographic mean-payoff games are studied in, and among several orders, the lexicographic order is investigated in for concurrent games with $\omega$-regular objectives.

In this paper, we consider turn-based lexicographic games with the classical $\omega$-regular objectives: reachability, safety, Büchi, co-Büchi, parity, Rabin, Streett, explicit Muller, and Muller objectives. In this setting, we associate with every play a Boolean vector the $i$th value of which is 1 if the play satisfies the $i$th objective, and 0 otherwise. Player 1 aims at maximizing his payoff with respect to the lexicographic ordering, while player 2 aims at minimizing it. We study the threshold problem, which asks, given some threshold $\mu$, whether player 1 has a winning strategy to ensure a payoff greater than or equal to $\mu$, and the value problem, which asks, what is the largest threshold that player 1 can ensure.

Contributions. Our contributions are as follows. First, we show that lexicographic games with Borel objectives are determined, and have optimal strategies (Corollary). For each classical $\omega$-regular objective,
we give a full picture of the complexity of the threshold problem (see Table 2). While the complexity upper bounds follow from results of [5], we provide additional tight lower bounds (Theorem 16).

Second and more importantly, the algorithms in [5] are nondeterministic and guess witnesses that are polynomial in the size of the game graph. This means that their deterministic versions are exponential in the size of the graph. In stark contrast, we provide new deterministic algorithms to solve the threshold and the value problems that depend only polynomially on the size of the game graph. Our new algorithms are instrumental to a finer complexity analysis of the considered problems: games with lexicographically ordered $\omega$-regular objectives are fixed parameter tractable when the number and description of the objectives are taken as parameters (see Theorem 21). This is important as in practice, the number of different objectives to consider is small and usually much smaller than the size of the game graph. For parity, Rabin, Streett, and Muller objectives, our results rely on the recent breakthrough of Calude et al. that provides a quasipolynomial-time algorithm for parity games and fixed parameter tractability for those games [7].

Finally, we also study tight memory bounds for playing optimally in the studied games.

Structure of the article. The article is organized as follows. In Section 2, we recall some useful notions on games, introduce the lexicographic games, and state the studied problems. In Section 3, the threshold and the value problems are solved for lexicographic games with Borel objectives. We also recall several useful results on classical games with $\omega$-regular objectives. When the objectives are $\omega$-regular, the fine complexity analysis of the threshold problem is conducted in Section 4, and its fixed-parameter tractability is studied in Section 5. We give a short conclusion in the last section.

2 Preliminaries

We consider zero-sum turn-based games played by two players, $P_1$ and $P_2$, on a finite directed graph. We associate with each game several objectives, and with each play of this game a vector of bits called payoff, the components of which indicate which objectives are satisfied. $P_1$ wants to maximize this payoff, seen as a reward, with respect to the lexicographic order, while $P_2$ wants to minimize it. We are interested in strategies for $P_1$ that ensure him to have a payoff at least equal to a given threshold, as well as in the highest possible threshold that he can guarantee.

Game structures. Let us give all the useful notions by first recalling the notion of game structure.

Definition 1. A game structure is a tuple $G = (V_1, V_2, E)$ where

1. $(V, E)$ is a finite directed graph, with $V = V_1 \cup V_2$ the set of vertices and $E \subseteq V \times V$ the set of edges such that for each $v \in V$, there exists $(v, v') \in E$ for some $v' \in V$;
2. $(V_1, V_2)$ forms a partition of $V$ such that $V_i$ is the set of vertices controlled by player $P_i$ with $i \in \{1, 2\}$.

A play of $G$ is an infinite sequence of vertices $\pi = v_0v_1\ldots \in V^\omega$ such that $(v_k, v_{k+1}) \in E$ for all $k \in \mathbb{N}$. We denote by Plays$(G)$ the set of plays in $G$. Histories of $G$ are finite sequences $\rho = v_0\ldots v_k \in V^+$ defined in the same way. Given a play $\pi = v_0v_1\ldots$, the set Occ$(\pi)$ denotes the set of vertices that occur in $\pi$, and the set $\text{Inf}(\pi)$ denotes the set of vertices visited infinitely often along $\pi$, i.e., $\text{Occ}(\pi) = \{v \in V \mid \exists k \geq 0, v_k = v\}$ and $\text{Inf}(\pi) = \{v \in V \mid \forall k \geq 0, \exists l \geq k, v_l = v\}$. Given a set $U \subseteq V$ and a set $\Omega \subseteq V^\omega$, we denote by $U^c$ the set $V \setminus U$ and by $\Omega^c$ the set $V^\omega \setminus \Omega$.

Strategies. A strategy $\sigma_i$ for $P_i$ is a function $\sigma_i: V^*V_i \to V$ assigning to each history $\rho v \in V^*V_i$ a vertex $v' = \sigma_i(\rho v)$ such that $(v, v') \in E$. It is memoryless if $\sigma_i(\rho v) = \sigma_i(\rho' v)$ for all histories $\rho v, \rho' v$ ending with the same vertex $v$, that is, if $\sigma_i$ is a function $\sigma_i: V_i \to V$. It is finite-memory if it can be encoded by a deterministic Moore machine $M = (M, m_0, \alpha_u, \alpha_n)$ where $M$ is a finite set of states (the memory of the strategy), $m_0 \in M$ is the initial memory state, $\alpha_u: M \times V \to M$ is the update function, and $\alpha_n: M \times V_i \to V$.

3 This condition guarantees that there is no deadlock. It can be assumed w.l.o.g. for all the problems considered in this article.
is the next-action function. The Moore machine $M$ defines a strategy $\sigma_i$ such that $\sigma_i(\rho v) = \alpha_v(\hat{\alpha}_v(m_0, \rho), v)$ for all histories $\rho v \in V^*V_i$, where $\hat{\alpha}_v$ extends $\alpha_v$ to histories as expected. The size of the strategy $\sigma_i$ is the size $|M|$ of its machine $M$. Note that $\sigma_i$ is memoryless when $|M| = 1$.

The set of all strategies of $\mathcal{P}_i$ is denoted by $\Sigma_i$. Given a strategy $\sigma_i$ of $\mathcal{P}_i$, we say that a play $\pi = v_0v_1\ldots$ of $G$ is consistent with $\sigma_i$ if $v_{k+1} = \sigma_i(v_0\ldots v_k)$ for all $k \in \mathbb{N}$ such that $v_k \in V_i$. Given an initial vertex $v_0$, and a strategy $\sigma_i$ of each player $\mathcal{P}_i$, we have a unique play consistent with both strategies $\sigma_1, \sigma_2$. This play is called the outcome of the game and is denoted by $\text{Out}(v_0, \sigma_1, \sigma_2)$.

**Single objectives and lexicographic ordered objectives.** Let $G = (V_1, V_2, E)$ be a game structure. An objective for $\mathcal{P}_1$ is a set of plays $\Omega \subseteq \text{Plays}(G)$. A game $(G, \Omega)$ is composed of a game structure $G$ and an objective $\Omega$. A play $\pi$ is winning for $\mathcal{P}_1$ if $\pi \in \Omega$, and losing otherwise. As the studied games are zero-sum, $\mathcal{P}_2$ has the opposite objective $\overline{\Omega}$, meaning that a play $\pi$ is winning for $\mathcal{P}_1$ if and only if it is losing for $\mathcal{P}_2$.

Given a game structure $G$, an objective $\Omega$ and an initial vertex $v_0$, a strategy $\sigma_1$ for $\mathcal{P}_1$ is winning from $v_0$ if $\text{Out}(v_0, \sigma_1, \sigma_2) \in \Omega$ for all strategies $\sigma_2$ of $\mathcal{P}_2$. Vertex $v_0$ is thus called winning for $\mathcal{P}_1$. We also say that $\mathcal{P}_1$ is winning from $v_0$ or that he can ensure $\Omega$ from $v_0$. Similarly the winning vertices of $\mathcal{P}_1$ are those from which $\mathcal{P}_2$ can ensure his objective $\overline{\Omega}$ against all strategies of $\mathcal{P}_1$.

A game $(G, \Omega)$ is determined if each of its vertices is either winning for $\mathcal{P}_1$ or winning for $\mathcal{P}_2$. Martin's theorem [22] states that all games with Borel objectives are determined. The problem of solving a game $(G, \Omega)$ means to decide, given an initial vertex $v_0$, whether $\mathcal{P}_1$ is winning from $v_0$ (or dually whether $\mathcal{P}_2$ is winning from $v_0$ when the game is determined).

Instead of a single objective $\Omega$, one can consider several objectives $\Omega_1, \ldots, \Omega_n$ that are lexicographically ordered in the following way. We first define the payoff of a play as a vector of bits, seen as a reward, the components of which indicate which objectives are satisfied.

**Definition 2.** Given a game structure $G = (V_1, V_2, E)$, and $n$ objectives $\Omega_1, \ldots, \Omega_n \subseteq \text{Plays}(G)$, the payoff function $\text{Payoff}: \text{Plays}(G) \to \{0, 1\}^n$ assigns a vector of bits to each play $\pi \in \text{Plays}(G)$, where for all $k \in \{1, \ldots, n\}$, $\text{Payoff}_k(\pi) = 1$ if $\pi \in \Omega_k$ and 0 otherwise.

The aim of $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) is then to maximize (resp. minimize) this payoff with respect to the lexicographic order $\prec$. Given two tuples $x, y \in \{0, 1\}^n$, we recall that

$$x \prec y \iff (x_1 < y_1) \lor (x_1 = y_1 \land x_2 < y_2) \lor \ldots \lor (\forall k \in \{1, \ldots, n-1\}, x_k = y_k \land x_n < y_n).$$

This order $\prec$ is total, and $x \preceq y$ if and only if $\neg(y \prec x)$. We also use notation $y \succ x$ (resp. $y \succeq x$) for $x \prec y$ (resp. $x \preceq y$). We call lexicographic game the tuple $(G, \Omega_1, \ldots, \Omega_n)$, the payoff function of which is defined w.r.t. the objectives $\Omega_1, \ldots, \Omega_n$, and its values are ordered with $\prec$. In this context, we are interested in the following problem.

**Problem 3.** The threshold problem for lexicographic games $(G, \Omega_1, \ldots, \Omega_n)$ asks, given a threshold $\mu \in \{0, 1\}^n$ and an initial vertex $v_0 \in V$, to decide whether $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) has a strategy to ensure the objective $\Omega = \{\pi \in \text{Plays}(G) \mid \text{Payoff}(\pi) \succeq \mu\}$ from $v_0$ (resp. $\overline{\Omega} = \{\pi \in \text{Plays}(G) \mid \text{Payoff}(\pi) \prec \mu\}$).

In case $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) has such a winning strategy, we also say that he can ensure a payoff $\succeq \mu$ (resp. $\prec \mu$).

**Example 4.** Consider the game structure $G$ depicted on Figure 1, where circle vertices belong to $\mathcal{P}_1$ and square vertices belong to $\mathcal{P}_2$. We consider the lexicographic game $(G, \Omega_1, \ldots, \Omega_n)$ with $\Omega_i = \{\pi \in \text{Plays}(G) \mid v_i \in \text{Inf}(\pi)\}$ for $i = 1, 2$. Therefore the function $\text{Payoff}$ assigns to each play $\pi$ the reward 1 on the first (resp. second) bit if $\pi$ visits infinitely often vertex $v_1$ (resp. $v_2$) and 0 otherwise. In this lexicographic game, $\mathcal{P}_1$ has a strategy to ensure a payoff $\succeq 01$ from $v_0$. Indeed, consider the memoryless strategy $\sigma_1$ that loops in $v_1$ and in $v_2$. Then, from $v_0$, either $\mathcal{P}_2$ decides to go to $v_1$ leading to the payoff 10, or to $v_2$ leading to the payoff 01. As $10 \succ 01$, this shows that any play $\pi$ consistent with $\sigma_1$ satisfies $\text{Payoff}(\pi) \succeq 01$. Remark that $\mathcal{P}_2$ has no strategy ensuring a payoff $\prec 01$ from $v_0$, but rather a winning strategy for the single objective $\overline{\Omega}_1$ from $v_0$, and similarly for the objective $\overline{\Omega}_2$.

\footnote{Note that when $n = 1$, we recover the notion of single objective with the threshold $\mu = 1$.}
Values and optimal strategies. In a lexicographic game, one can define the best reward that \( P_1 \) can ensure from a given vertex, that is, the highest threshold \( \mu \) for which \( P_1 \) can ensure a payoff \( \geq \mu \). Dually, we can also define the worth reward that \( P_2 \) can ensure. Note that in the following definition, the infimum and supremum functions are applied with the lexicographic order \( \prec \).

**Definition 5.** Given a lexicographic game \( (G, \Omega_1, \ldots, \Omega_n) \), for every vertex \( v \in V \), the upper value \( \text{Val}(v) \) and the lower value \( \text{Val}(v) \) are defined as:

\[
\text{Val}(v) = \inf_{\sigma_2 \in \Sigma_2} \sup_{\sigma_1 \in \Sigma_1} \text{Payoff}(\text{Out}(v, \sigma_1, \sigma_2)),
\]

\[
\text{Val}(v) = \sup_{\sigma_1 \in \Sigma_1} \inf_{\sigma_2 \in \Sigma_2} \text{Payoff}(\text{Out}(v, \sigma_1, \sigma_2)).
\]

The lexicographic game \( (G, \Omega_1, \ldots, \Omega_n) \) is determined\(^5\) if, for every \( v \in V \), \( \text{Val}(v) = \text{Val}(v) \). In this case, we write \( \text{Val}(v) = \text{Val}(v) = \text{Val}(v) \) and we call \( \text{Val}(v) \) the value of \( v \). Note that the inequality \( \text{Val}(v) \geq \text{Val}(v) \) always holds. If \( P_1 \) (resp. \( P_2 \)) can ensure a payoff \( \geq \text{Val}(v) \) (resp. \( \leq \text{Val}(v) \)) from \( v \), his related winning strategy \( \sigma_1^* \) (resp. \( \sigma_2^* \)) is called optimal from \( v \):

\[
\text{Val}(v) \leq \text{Payoff}(\text{Out}(v, \sigma_1^*, \sigma_2)) \quad \forall \sigma_2 \in \Sigma_2.
\]

\[
\text{Payoff}(\text{Out}(v, \sigma_1, \sigma_2^*)) \leq \text{Val}(v) \quad \forall \sigma_1 \in \Sigma_1.
\]

Notice that in case of determinacy, the play \( \pi = \text{Out}(v, \sigma_1^*, \sigma_2^*) \) consistent with both optimal strategies has payoff \( \text{Payoff}(\pi) = \text{Val}(v) \).

**Studied problem.** In this article, given a lexicographic game \( (G, \Omega_1, \ldots, \Omega_n) \), we want to study whether it is determined, what is the complexity of computing the value of its vertices and what are the memory requirements of the related optimal strategies. In particular, we will see that this requires to solve the threshold problem mentioned before. We focus on **homogeneous objectives**, in the sense that all the objectives \( \Omega_1, \ldots, \Omega_n \) are of the same type, and taken in the following list of well-known \( \omega \)-regular objectives.

Given a game structure \( G = (V_1, V_2, E) \) and a subset \( U \) of \( V \) called target set:

- The **reachability objective** asks to visit a vertex of \( U \) at least once, i.e. \( \text{Reach}(U) = \{ \pi \in \text{Plays}(G) \mid \text{Occ}(\pi) \cap U \neq \emptyset \} \).
- The **safety objective** asks to always stay in the set \( U \), i.e. \( \text{Safe}(U) = \{ \pi \in \text{Plays}(G) \mid \text{Occ}(\pi) \cap U^c = \emptyset \} \).
- The **Büchi objective** asks to visit infinitely often a vertex of \( U \), i.e. \( \text{Buchi}(U) = \{ \pi \in \text{Plays}(G) \mid \text{Inf}(\pi) \cap U \neq \emptyset \} \).
- The **co-Büchi objective** asks to eventually always stay in the set \( U \), i.e. \( \text{CoBuchi}(U) = \{ \pi \in \text{Plays}(G) \mid \text{Inf}(\pi) \cap U^c = \emptyset \} \).

Given a family \( \mathcal{F} = (F_i)_{i=1}^k \) of subsets \( F_i \) of \( V \), and a family of pairs \( (\langle E_i, F_i \rangle)_{i=1}^k \), with \( E_i, F_i \subseteq V \):

---

\(^5\) The notion of determinacy is different from the one given previously for games \( (G, \Omega) \) with a single objective \( \Omega \).
The explicit Muller objective asks that the set of vertices seen infinitely often is exactly one among $\mathcal{F}$, i.e. $\text{ExplMuller}(\mathcal{F}) = \{ \pi \in \text{Plays}(G) \mid \exists i \in \{1, \ldots, k\}, \text{Inf}(\pi) = F_i \}$.

- The Rabin objective asks that there exists a pair $(E_i, F_i)$ such that a vertex of $F_i$ is visited infinitely often while no vertex of $E_i$ is visited infinitely often, i.e. $\text{Rabin}((E_i, F_i)_{i=1}^k) = \{ \pi \in \text{Plays}(G) \mid \exists i \in \{1, \ldots, k\}, \text{Inf}(\pi) \cap E_i = \emptyset \text{ and } \text{Inf}(\pi) \cap F_i \neq \emptyset \}$.

- The Streett objective asks that for each pair $(E_i, F_i)$, a vertex of $E_i$ is visited infinitely often or no vertex of $F_i$ is visited infinitely often, i.e. $\text{Streett}((E_i, F_i)_{i=1}^k) = \{ \pi \in \text{Plays}(G) \mid \forall i \in \{1, \ldots, k\}, \text{Inf}(\pi) \cap E_i \neq \emptyset \text{ or } \text{Inf}(\pi) \cap F_i = \emptyset \}$.

Given a coloring function $p: V \rightarrow \{0, \ldots, d\}$ that associates with each vertex a color, and $\mathcal{F} = (F_i)_{i=1}^k$ a family of subsets $F_i$ of $p(V)$:

- The parity objective asks that the minimum color seen infinitely often is even, i.e. $\text{Parity}(p) = \{ \pi \in \text{Plays}(G) \mid \min_{v \in \text{Inf}(\pi)} p(v) \text{ is even} \}$.

- The Muller objective asks that the set of colors seen infinitely often is exactly one among $\mathcal{F}$, i.e. $\text{Muller}(p, \mathcal{F}) = \{ \pi \in \text{Plays}(G) \mid \exists i \in \{1, \ldots, k\}, p(\text{Inf}(\pi)) = F_i \}$.

When the context is clear, we sometimes simply write Reach, Streett, Parity, ... without mentioning the related sets, families, or coloring functions. Moreover when the objectives of a lexicographic game are of kind $X$, we speak of lexicographic $X$ game. As already mentioned, when $n = 1$, a lexicographic game resumes to a classical game $(G, \Omega)$, i.e. with a single objective $\Omega$, that is traditionally called an $\Omega$ game. For instance, a lexicographic game $(G, \Omega_1, \ldots, \Omega_n)$ where $\Omega_1, \ldots, \Omega_n$ are reachability objectives is called a lexicographic reachability game, and when $n = 1$ $(G, \Omega_1)$ is called a reachability game.

Note that given a lexicographic game with $n$ non-homogeneous $\omega$-regular objectives, we can always construct a new equivalent lexicographic parity game.

3 Useful results

In this section, we aim at providing material useful for the study of lexicographic games. We begin by introducing some terminology on lexicographically ordered payoffs. We then show that solving the threshold problem for a lexicographic game $(G, \Omega_1, \ldots, \Omega_n)$ is equivalent to solving a game $(G, \Omega)$ where the objective $\Omega$ is of special form: a polynomial disjunction of conjunctions of objectives taken among $\Omega_i, i \in \{1, \ldots, n\}$. When the objectives are Borel, this yields determinacy and optimal strategies for the lexicographic game. We also give a procedure to compute the value of each vertex under the assumption that the threshold problem is decidable. Finally we recall a series of well-known results on classical games with $\omega$-regular objectives.

Useful terminology on payoffs and lexicographic order. Let us provide several useful terminology and comments on lexicographically ordered payoffs. Given a vector $x \in \{0, 1\}^n$, we denote by $\overline{x}$ the complement of $x$, i.e. $\overline{x}_i = 1 - x_i$, for all $i \in \{1, \ldots, n\}$. Suppose that $x \neq 0^n$, we denote by $x - 1$ the predecessor of $x$, that is, the greatest vector which is smaller than $x$ with respect to the lexicographic order $\prec$. Similarly, we can define the successor $x + 1$ of $x \neq 1^n$. The following definition is used in Theorem 9 for solving the threshold problem (see Problem 3).

Definition 6. To any vector $x \in \{0, 1\}^n$, we associate

- the set $\delta_x = \{ i \in \{1, \ldots, n\} \mid x_i = 1 \}$ containing all indices $i$ such that the $i^{\text{th}}$ bit of $x$ is equal to 1,
- the set $M(x) = \{ x \} \cup \{ y^j \in \{0, 1\}^n \mid x_j = 0 \}$, where for all $j \in \{1, \ldots, n\}$ such that $x_j = 0$, we define the vector $y^j \in \{0, 1\}^n$ as equal to $x_1 \ldots x_{j-1} 10^{n-j}$ ($x$ and $y^j$ share the same (possibly empty) prefix $x_1 \ldots x_{j-1}$).

Example 7. Consider the vector $x = 001100$. Then, $\overline{x} = 110011$, $x - 1 = 001011$, and $x + 1 = 001101$. The set $\delta_x$ is equal to $\{3, 4\}$ and the set $M(x)$ is equal to $\{x\} \cup \{100000, 010000, 001110, 001101\}$.
Note that by definition of the lexicographic order, we have \( x < y \) for all \( y \in M(x) \setminus \{x\} \). Moreover by definition of \( \delta_x \) and \( M(x) \), we have \( |\delta_x| \leq n \) and \( |M(x)| \leq n + 1 \) with \( |M(x)| \leq n \) if \( x \neq 0^n \).

The following remark concerns the threshold problem.

Remark 8. (i) In the sequel, as the threshold problem is trivial for \( \mu = 0^n \), we do not consider this threshold. By abuse of notation, we keep writing \( \mu \in \{0,1\}^n \) without mentioning that \( \mu \neq 0^n \). (ii) One could have studied the threshold problem using the comparison \( \succ \mu \) instead of \( \succeq \mu \). However as a vector \( x \in \{0,1\}^n \) satisfies \( x \succeq \mu \) (with \( \mu \neq 0^n \)) if and only if it satisfies \( x \succ \mu - 1 \), the considered statement of the threshold problem is not restrictive.

The threshold problem revisited. Our aim is now to show that solving the threshold problem for a lexicographic game \( (G, \Omega_1, \ldots, \Omega_n) \) is equivalent to solving a classical game \( (G, \Omega) \). The idea is the following one: \( P_1 \) can ensure a payoff \( \succeq \mu \) if and only if he can ensure a payoff \( = \nu \) for some \( \nu \succeq \mu \). As the payoff of a play equal to \( \nu \) means that it satisfies the conjunction of the objectives \( \Omega_i \) such that \( \nu_i = 1 \), the objective of \( P_1 \) is thus a disjunction (over \( \nu \succeq \mu \)) of conjunctions of objectives taken among \( \Omega_i, i \in \{1, \ldots, n\} \). In the following theorem, we show that this disjunction can be restricted to the set \( M(\mu) \) defined in Definition 6.

We will then derive that lexicographic games \( (G, \Omega_1, \ldots, \Omega_n) \) are Borel objectives, and will show how to compute values and optimal strategies when the threshold problem is decidable (see Corollary 10).

Theorem 9. Let \( (G, \Omega_1, \ldots, \Omega_n) \) be a lexicographic game, \( \mu \in \{0,1\}^n \) be a threshold, and \( v_0 \) be an initial vertex. Then, \( P_1 \) can ensure a payoff \( \succeq \mu \) from \( v_0 \) in \( (G, \Omega_1, \ldots, \Omega_n) \) if and only if \( P_1 \) has a winning strategy from \( v_0 \) in \( (G, \Omega) \) with

\[
\Omega = \cup_{\nu \in M(\mu)} \cap_{i \in \delta_\nu} \Omega_i
\]

and \( |M(\mu)| \leq n \) and \( |\delta_\nu| \leq n \) for each \( \nu \in M(\mu) \).

Proof. Recall that \( P_1 \) has a strategy to ensure a payoff \( \succeq \mu \) from \( v_0 \) means that \( P_1 \) has a winning strategy from \( v_0 \) for the objective \( \{ \in \text{Plays}(G) \mid \text{Payoff}(\pi) \succeq \mu \} \). We prove that (i) this objective is equal to \( \Omega^* = \cup_{\nu \succeq \mu} \cap_{i \in \delta_\nu} \Omega_i \), and (ii) that \( \Omega^* = \Omega \).

(i) Suppose that \( P_1 \) has a strategy such that \( \text{Payoff}(\pi) \succeq \mu \) for any consistent play \( \pi \) from \( v_0 \). Let \( \nu \) be such that \( \text{Payoff}(\pi) = \nu \succeq \mu \). Then in particular \( \pi \in \cap_{i \in \delta_\nu} \Omega_i \) by definition of the payoff function. In the other direction, if \( P_1 \) has a winning strategy for the objective \( \Omega^* \), then for any play \( \pi \) consistent with this strategy from \( v_0 \), there exists \( \nu \succeq \mu \) such that \( \pi \in \cap_{i \in \delta_\nu} \Omega_i \). Thus, \( P_1 \) can ensure a payoff \( \succeq \min_{\nu \succeq \mu} \nu = \mu \).

(ii) We easily see that \( \Omega \subseteq \Omega^* \) since \( \nu \succeq \mu \) for all \( \nu \in M(\mu) \). Let us show that \( \Omega^* \subseteq \Omega \). Let \( \nu \succeq \mu \) be a threshold. In case \( \nu \succ \mu \), let \( j \) be the first index such that \( \nu_j = 1 \) and \( \mu_j = 0 \). Note that \( \mu_1 \ldots \mu_{j-1} = \nu_1 \ldots \nu_{j-1} \) since \( \nu \succ \mu \). We naturally associate to \( \nu \) the vector \( \nu^0 = \nu_1 \ldots \nu_{j-1} 10^{n-j} \in M(\mu) \). Then, we have that \( \cap_{i \in \delta_\nu} \Omega_i \subseteq \cap_{i \in \delta_{\nu^0}} \Omega_i \). In case \( \nu = \mu \), recall that \( \mu \in M(\mu) \) by definition of \( M(\mu) \). This proves that \( \Omega^* \subseteq \Omega \).

This finishes the proof since the given bounds for \( |M(\mu)| \) and \( |\delta_\nu| \), \( \nu \in M(\mu) \), come from Definition 6. \( \square \)

This theorem yields the following corollary.

Corollary 10. Let \( (G, \Omega_1, \ldots, \Omega_n) \) be a lexicographic game. If \( \Omega_1, \ldots, \Omega_n \) are Borel sets, then \( P_1 \) has a strategy to ensure a payoff \( \succeq \mu \) from \( v_0 \) if and only if it is not the case that \( P_2 \) has a strategy to ensure a payoff \( \prec \mu \) from \( v_0 \), and the lexicographic game is determined and has optimal strategies. Moreover, if the threshold problem can be solved with an algorithm of complexity \( \mathbb{C} \), then for all \( v \in V \), the value \( \text{Val}(v) \) can be computed with an algorithm of complexity \( n \cdot \mathbb{C} \).

Proof. If \( \Omega_1, \ldots, \Omega_n \) are Borel sets, as there is a finite number of thresholds, then \( \Omega = \cup_{\nu \in M(\mu)} \cap_{i \in \delta_\nu} \Omega_i \) is also a Borel set. By Martin’s theorem [22], the classical game \( (G, \Omega) \) is thus determined, and \( P_1 \) has a strategy to ensure a payoff \( \succeq \mu \) from \( v_0 \) if and only if it is not the case that \( P_2 \) has a strategy to ensure a payoff \( \prec \mu \) from \( v_0 \). Let us show that the lexicographic game \( (G, \Omega_1, \ldots, \Omega_n) \) is also determined. To this end, we use the following Folk property: if there exists some \( \alpha \in \{0,1\}^n \) and two strategies \( \sigma_1^* \in \Sigma_1, \sigma_2^* \in \Sigma_2 \) such that

\[
\text{Val}(v) = \begin{cases} 
\alpha_1 & \text{if } \sigma_1^* \text{ is a winning strategy for } \alpha \in M(v) \text{ and } \sigma_2^* \text{ is a losing strategy for } \alpha \in \neg M(v) \\
\alpha_2 & \text{if } \sigma_2^* \text{ is a winning strategy for } \alpha \in M(v) \text{ and } \sigma_1^* \text{ is a losing strategy for } \alpha \in \neg M(v) 
\end{cases}
\]
Payoff(Out(v, σ₁, σ₂)) ≤ α ≤ Payoff(Out(v, σ₁', σ₂)) for all strategies σ₁ ∈ Σ₁, σ₂ ∈ Σ₂, then α = Val(v) and σ₁', σ₂' are optimal strategies from v. Let v be a vertex. The set of thresholds µ is partitioned between the two players according to whether \( P_1 \) (resp. \( P_2 \)) can ensure a payoff \( ≥ µ \) (resp. \( < µ + 1 \)) from v. Let α be the highest threshold that \( P_1 \) can ensure and \( σ₁' \) be the corresponding winning strategy. By definition of α, \( P_2 \) can ensure a payoff \( < α + 1 \) with a winning strategy \( σ₂' \). It follows that \( Payoff(Out(v, σ₁, σ₂)) ≤ α ≤ Payoff(Out(v, σ₁', σ₂)) \) for all strategies \( σ₁ ∈ Σ₁, σ₂ ∈ Σ₂ \), and therefore we have \( Val(v) = α \).

When the threshold problem is decidable, the procedure to compute the value \( Val(v) \) works as follows. The idea is to solve the threshold problem for different thresholds from vertex v in a way to compute the highest threshold \( µ \) for which \( P_1 \) can ensure a payoff \( ≥ µ \). This threshold \( µ \) is the value \( Val(v) \).

First, we test whether \( P_1 \) can ensure a payoff \( ≥ 10^{n-i} \) from v. If this is the case, we set bit \( µ_i \) to 1 and to 0 otherwise. Then, for \( i ∈ \{2, \ldots, n\} \), we successively test whether \( P_1 \) can ensure a payoff \( ≥ 10^{n-i} \) from v and we set bit \( µ_i \) to 1 if this is the case and to 0 otherwise. Thus, after those \( n \) solutions to the threshold problem, we obtain a threshold \( µ = µ_1 \ldots µ_n \) for which \( P_1 \) can ensure a payoff \( ≥ µ \) from v. The complexity of the algorithm computing \( µ \) is thus in \( n \cdot C \). By using again the previous Folk property with the computed \( µ \), we have that \( µ = Val(v) \).

This concludes the proof. 

Remark 11. When the procedure given in the proof of Corollary 10 computes the value \( µ \) of a given vertex v, it also computes at the same time an optimal strategy from this vertex for both players. Indeed, the optimal strategy \( σ₁' \) of \( P_1 \) from v is his winning strategy (for the threshold problem) that ensures a payoff \( ≥ µ \) and that the optimal strategy \( σ₂' \) of \( P_2 \) from v is his winning strategy that ensures a payoff \( < µ + 1 \). Notice that in this procedure \( σ₁' \) (resp. \( σ₂' \)) is the winning strategy of \( P_1 \) (resp. \( P_2 \)) for the last bit \( µ_n \) set to 1 (resp. to 0). Therefore, in order to study some properties on optimal strategies (such as memory requirements), it is sufficient to study winning strategies for the threshold problem.

Example 12. Let us consider the lexicographic reachability game \( (G, Ω_1, Ω_2, Ω_3) \) depicted on Figure 2 where \( Ω_1 = \text{Reach}\{v_1\} \), \( Ω_2 = \text{Reach}\{v_2, v_3\} \) and \( Ω_3 = \text{Reach}\{v_5\} \). We apply the procedure described in Corollary 10 to compute \( Val(v_0) \) and the corresponding optimal strategies. For this purpose, we begin by testing whether \( P_1 \) can ensure a payoff \( ≥ 100 \) from v₀. This is not the case since \( P_2 \) can prevent him from visiting vertex v₁ by going from v₀ to v₂. In particular, this strategy of \( P_2 \) ensures a payoff \( < 100 \). We fix \( µ_1 = 0 \) and we now test whether \( P_1 \) can ensure a payoff \( ≥ 010 \). This is the case since by going from v₂ to v₄, \( P_1 \) is guaranteed to visit vertex v₄. We thus set \( µ_2 = 1 \). The final test made is whether \( P_1 \) can ensure a payoff \( ≥ 011 \). This is possible with the strategy that consists in going from v₂ to v₅. Indeed, the two possible outcomes consistent with this strategy are \( v₀v₁v₂v₅^2 \) and \( v₀v₂v₃v₅^2 \). The payoff of the first play is \( 101 ≥ 011 \) while the latter payoff is \( 011 \). Hence we set \( µ_3 = 1 \) and we get \( Val(v₀) = 011 \). The corresponding optimal strategies are to go from v₂ to v₅ for \( P_1 \) (to ensure a payoff \( ≥ 011 \)) and to go from v₀ to v₂ for \( P_2 \) (to ensure a payoff \( < 100 \)). Note that the outcome from v₀ consistent with those strategies is \( v₀v₂v₃v₅^2 \) and that its payoff is indeed \( Val(v₀) \).

\[ \text{Fig. 2. The value of vertex } v₀ \text{ is equal to } 011. \]
The following proposition establishes the duality between a payoff defined with some objectives and the payoff defined with the opposite objectives.

**Proposition 13.** Let \((G, \Omega_1, \ldots, \Omega_n)\) be a lexicographic game, \(\mu \in \{0, 1\}^n\) be a threshold, and \(v_0\) be an initial vertex. Then \(P_1\) can ensure a payoff \(\geq \mu\) in the lexicographic game \((G, \Omega_1, \ldots, \Omega_n)\) if and only if \(P_1\) can ensure a payoff \(\leq \mu\) in the lexicographic game \((G, \overline{\Omega_1}, \ldots, \overline{\Omega_n})\).

**Proof.** Recall that \(P_1\) can ensure a payoff \(\geq \mu\) from \(v_0\) in the lexicographic game \((G, \Omega_1, \ldots, \Omega_n)\) if and only if he has a winning strategy from \(v_0\) for the objective \(\{\pi \mid \text{Payoff}(\pi) = \mu\}\). Moreover, for any \(i, \nu_i = 1\) if and only if \(\pi \in \Omega_i\), i.e. \(\pi \notin \overline{\Omega_i}\), and \(\nu_i = 0\) if and only if \(\pi \notin \Omega_i\), i.e. \(\pi \in \overline{\Omega_i}\). Thus, \(\nu_i = 1\) iff \(\pi \notin \overline{\Omega_i}\) and \(\nu_i = 0\) iff \(\pi \in \overline{\Omega_i}\). Then, as \(\nu \geq \mu\) iff \(\overline{\mu} \leq \overline{\nu}\), we have that \(P_1\) is winning from \(v_0\) for the objective \(\{\pi \mid \text{Payoff}(\pi) = \nu \geq \mu\}\) if and only if \(P_1\) has a strategy to ensure a payoff \(\leq \overline{\mu}\) from \(v_0\) in the lexicographic game \((G, \overline{\Omega_1}, \ldots, \overline{\Omega_n})\).

Some well-known properties on classical games with \(\omega\)-regular objectives. The lexicographic games studied in the article have \(n\) homogeneous \(\omega\)-regular objectives \(\Omega_1, \ldots, \Omega_n\) taken in the list of objectives given in Section 2. In the proofs, we will need to consider three additional \(\omega\)-regular objectives. Let \(G\) be a game structure \(G\) and \(U_1, \ldots, U_m \subseteq V\) be a family of target sets:

- The generalized reachability objective asks a play to visit a vertex of \(U_i\) at least once, for each \(i \in \{1, \ldots, m\}\), i.e. \(\text{GenReach}(U_1, \ldots, U_m) = \cap_{i=1}^m \text{Reach}(U_i)\).
- The generalized Büchi objective asks a play to visit a vertex of \(U_i\) infinitely often, for each \(i \in \{1, \ldots, m\}\), i.e. \(\text{GenBuchi}(U_1, \ldots, U_m) = \cap_{i=1}^m \text{Buchi}(U_i)\).

Let \(\phi\) be a Boolean formula over variables \(x_1, \ldots, x_m\). We say that a play \(\pi\) satisfies \((\phi, U_1, \ldots, U_m)\) if the truth assignment \((x_1 = 1\text{ if and only if }\text{Inf}(\pi) \cap U_i \neq \emptyset, \text{and } x_i = 0 \text{ otherwise})\) satisfies \(\phi\).

- The objective being a Boolean combination of Büchi objectives, or shortly the Boolean Büchi objective, asks a play to satisfy \((\phi, U_1, \ldots, U_m)\), i.e. \(\text{BooleanBuchi}(\phi, U_1, \ldots, U_m) = \{\pi \in \text{Plays}(G) \mid \pi\) satisfies \((\phi, U_1, \ldots, U_m)\}.

All operators \(\lor, \land, \neg\) are allowed in Boolean Büchi objectives. However we denote by \(|\phi|\) the size of \(\phi\) equal to the number of disjunctions and conjunctions inside \(\phi\), and we say that the Boolean Büchi objective \(\text{BooleanBuchi}(\phi, U_1, \ldots, U_m)\) is of size \(|\phi|\) and with \(m\) variables. The definition of \(|\phi|\) is not the classical one that usually counts the number of operators \(\lor, \land, \neg\) and variables. This is not a restriction since one can transform any Boolean formula \(\phi\) into one such that negations only apply on variables.

Let us recall some well-known results about classical games \((G, \Omega)\) with \(\omega\)-regular objectives \(\Omega\). All these games are determined by Martin’s theorem [22]. Moreover, the complexity class of solving those games is well-known, as well as the kind (memoryless, finite-memory) of winning strategies for both players. See Theorem 14 and Table 1 below. For each type of objective, the complexity of the algorithms is expressed in terms of the sizes \(|V|\) and \(|E|\) of the game structure \(G\), the number \(m\) of objectives in the intersection of objectives (for GenReach and GenBuchi), the number \(d\) of colors (for Parity and Muller), the number \(k\) of pairs (for Rabin and Streett), the size \(|\mathcal{F}|\) of the family \(\mathcal{F}\) (for ExplMuller and Muller) and the size \(|\phi|\) of the formula \(\phi\) (for BooleanBuchi).

**Theorem 14.** For games \((G, \Omega)\) with \(\omega\)-regular objectives, we have:

- Solving reachability or safety games is P-complete (with an algorithm in \(O(|V| + |E|) \text{ time})\) and both players have memoryless winning strategies [2,17,20].
- Solving Büchi or co-Büchi games is P-complete (with an algorithm in \(O(|V|^2) \text{ time})\) and both players have memoryless winning strategies [9,15,20].
- Solving generalized reachability games with \(m\) target sets is PSPACE-complete (with an algorithm in \(O(2^m \cdot (|V| + |E|)) \text{ time})\) and exponential memory strategies are necessary and sufficient for both players [16].
- Solving generalized Büchi games with \(m\) target sets is P-complete (with an algorithm in \(O(m \cdot |V|^2) \text{ time})\) and linear memory (resp. memoryless) strategies are necessary and sufficient for \(P_1\) (resp. \(P_2\)) [8].

8
– Solving explicit Muller games with a family $\mathcal{F}$ is $P$-complete (with an algorithm in $O(|\mathcal{F}| \cdot (|\mathcal{F}| + |V| \cdot |E|)^2)$ time) and exponential memory strategies are necessary and sufficient for both players [12,18].
– Solving parity games with $d$ colors is in UP $\cap$ co-UP (with an algorithm in $O(|V|^{\lfloor \log(d) \rfloor + 6})$ time [7]) and both players have memoryless winning strategies [21].
– Solving Rabin (resp. Streett) games with $k$ pairs is NP-complete (resp. co-NP-complete) [14] (with an algorithm in $O(|V|^{k+1} \cdot k!)$ time [24]). In Rabin games (resp. Streett games) memoryless strategies are sufficient for $P_1$ (resp. for $P_2$) [13] and exponential memory strategies are necessary and sufficient for $P_2$ (resp. $P_1$) [12].
– Solving Muller games is PSPACE-complete (with an algorithm in $O(|V|^2 \cdot |E| \cdot |V|!)$ time [23]) and exponential memory strategy are necessary and sufficient for both players [12,19].
– Solving Boolean Büchi games is PSPACE-complete (with an algorithm in $O(|\phi| \cdot 2^{|V|^2})$) time and exponential memory strategies are necessary for both players [1].

| Objectives          | Complexity class | $P_1$ memory | $P_2$ memory |
|---------------------|------------------|--------------|--------------|
| Reachability, safety| P-complete       | memoryless   |              |
| Büchi, co-Büchi     |                  |              |              |
| Explicit Muller     |                  | exponential  |              |
| Generalized Büchi   |                  | linear       | memoryless   |
| Generalized reachability | PSPACE-complete |              |              |
| Parity              | NP $\cap$ coNP   | memoryless   |              |
| Rabin               | NP-complete      | memoryless   | exponential  |
| Streett             | coNP-complete    | exponential  | memoryless   |
| Muller              | PSPACE-complete  |              | exponential  |

Table 1. Overview of known results on classical games with $\omega$-regular objectives. The last two columns indicate the tight memory requirements of the winning strategies.

In the sequel, we need some classical properties on $\omega$-regular objectives that we summarize in the following proposition.

Proposition 15. 1. A safety (resp. co-Büchi, Streett) objective is the complement of a reachability (resp. Büchi, Rabin) objective.
2. A parity objective is both a Rabin and a Streett objective.
3. Rabin and Streett objectives with one pair are parity objectives with 3 colors. Thus, a Rabin (resp. Streett) objective is the union (resp. intersection) of parity objectives with 3 colors.
4. The intersection (resp. union) of $m$ explicit Muller objectives $\text{ExplMuller}(\mathcal{F}_i)$ is an explicit Muller objective $\text{ExplMuller}(\mathcal{F})$ where $|\mathcal{F}| \leq \min_i\{|\mathcal{F}_i|\}$ (resp. $|\mathcal{F}| \leq \sum_i |\mathcal{F}_i|$).
5. A parity objective with $d$ colors (resp. Streett objective with $k$ pairs, Muller objective with $d$ colors and a family $\mathcal{F}$) is a Boolean Büchi objective of size at most $\frac{d^2}{2}$ (resp. $2 \cdot k$, $d \cdot |\mathcal{F}|$) and with $d$ (resp. $2 \cdot k$, $d$) variables.

Proof. First, items (1) immediately follow from the definitions and items (2, 3) are detailed in [10].
For Item 4, for the intersection we have $\bigcap_i \text{ExplMuller}(\mathcal{F}_i) = \text{ExplMuller}(\mathcal{F})$ where $\mathcal{F} = \bigcap_i \mathcal{F}_i$, and thus $|\mathcal{F}| \leq \min_i\{|\mathcal{F}_i|\}$. For the union we have $\bigcup_i \text{ExplMuller}(\mathcal{F}_i) = \text{ExplMuller}(\mathcal{F})$ with $\mathcal{F} = \bigcup_i \mathcal{F}_i$ with $|\mathcal{F}| \leq \sum_i |\mathcal{F}_i|$.

$^6$ The algorithm complexity and the memory requirements do not appear explicitly in [1] but can be deduced straightforwardly thanks to the proposed algorithm.
Let us prove the last item by beginning with Muller objectives. It suffices to note that a play belongs to Muller\((p, F)\) if and only if there exists an element \(F\) of \(F\) such that all colors of \(F\) are seen infinitely often along the play while no other color is seen infinitely often. This is obviously a Boolean B"uchi objective \(\text{BooleanBuchi}\((\phi, U_1, \ldots, U_m)\), where each \(U_i\) corresponds to a color, that is, \(U_i\) is the set of vertices labeled by this color. Note that, in this case, the size of the related formula \(\phi\) is at most \(d \cdot |F|\). The arguments are similar for parity and Streett objectives (for instance, a play belongs to \(\text{Parity}(p)\) if and only there exists an even color seen infinitely often along the play and no lower color seen infinitely often).

\[\square\]

4 Lexicographic \(\omega\)-regular games

In this section, we rely on Theorem 9 to study the complexity of the threshold problem for lexicographic reachability, safety, B"uchi, co-B"uchi, explicit Muller, parity, Rabin, Streett, and Muller games. Thanks to Corollary 10, we then deduce how to compute the value of each vertex and what are the tight memory requirements for the related optimal strategies. All results are summarized in Table 2.

| Threshold problem | Reachability | Safety | B"uchi | Co-B"uchi | Explicit Muller | Parity | Rabin | Streett | Muller |
|-------------------|--------------|--------|--------|-----------|-----------------|--------|-------|---------|--------|
| Values            | PSPACE-complete | P-complete | PSPACE-complete |
| \(P_1\) memory   | exponential   | polynomial | exponential |
| \(P_2\) memory   | exponential   | memoryless | linear |

Table 2. Overview of the results on lexicographic games with \(\omega\)-regular objectives. The second row indicates the complexity time of computing the values. The third and last rows indicate the tight memory requirements of the optimal strategies.

Threshold problem. The following theorem states the complexity class of the threshold problem for the lexicographic games considered in this article. It also states the tight memory requirements for the winning strategies of the players.

**Theorem 16.** The threshold problem is P-complete for lexicographic B"uchi, co-B"uchi and explicit Muller games, and is PSPACE-complete for lexicographic reachability, safety, parity, Rabin, Streett, and Muller games.

Moreover, linear memory strategies are necessary and sufficient for \(P_1\) (resp. \(P_2\)) while memoryless strategies are sufficient for \(P_2\) (resp. \(P_1\)) in lexicographic B"uchi (resp. co-B"uchi) games, and exponential memory strategies are both necessary and sufficient for both players in lexicographic reachability, safety, explicit Muller, parity, Rabin, Streett, and Muller games.

The proof of Theorem 16 is divided into two parts. The upper bounds are proved in Proposition 17 with the corresponding algorithmic complexities. The lower bounds are later proved in Proposition 18.

Upper bounds. The proof of Proposition 17 is based on Theorem 9. This requires the study of games \((G, \Omega)\) with \(\Omega\) equal to a union of homogeneous intersections of some \(\omega\)-regular objectives. Intuitively, we show that the union of intersections of co-B"uchi (resp. safety) objectives is simply the complementary of a generalized B"uchi (resp. reachability) objective, and that the union of intersections of explicit Muller (resp. parity, Streett, Muller) objectives is an explicit Muller objective (resp. a Boolean B"uchi objective). The remaining cases of reachability, B"uchi and Rabin objectives are obtained with Proposition 13. We also provide the size of all the resulting objectives.

**Proposition 17.** For lexicographic games \((G, \Omega_1, \ldots, \Omega_n)\), there is a polynomial reduction from the threshold problem of \((G, \Omega)\)
– lexicographic reachability and safety games to solving generalized reachability games with at most $n$ target sets,
– lexicographic Büchi and co-Büchi games to solving generalized Büchi games with at most $n$ target sets,
– lexicographic explicit Muller games to solving explicit Muller games with a family $F$ of size at most
  $\beta = \sum_{i=1}^{n} |F_i|$, where $\Omega = \text{ExpMuller}(F_i)$ for all $i$,
– lexicographic parity (resp. Rabin, Streett, and Muller) games to solving Boolean Büchi games with a
  formula $\phi$
  \begin{itemize}
    \item of size $\alpha_p = n \cdot \sum_{i=1}^{n} d_i^2$ (resp. $\alpha_R = n \cdot \sum_{i=1}^{n} 2 \cdot k_i$, $\alpha_S = \alpha_R$, and $\alpha_M = n \cdot \sum_{i=1}^{n} d_i \cdot |F_i|$),
    \item and with $\sum_{i=1}^{n} d_i$ (resp. $\sum_{i=1}^{n} 2 \cdot k_i$, $\sum_{i=1}^{n} 2 \cdot k_i$, $\sum_{i=1}^{n} d_i$) variables,
  \end{itemize}
where $\Omega$ uses $d_i$ colors (resp. $k_i$ pairs, $k_i$ colors and family $F_i$) for all $i$.

Moreover, the threshold problem is decidable in polynomial time with an algorithm in

– $O(n \cdot |V|^2)$ time for lexicographic Büchi and co-Büchi games,
– $O(\beta \cdot (|V| \cdot |E| + \beta)^2)$ time for lexicographic explicit Muller games,

and it is decidable in PSPACE with an algorithm in

– $O(2^n \cdot (|V| + |E|))$ time for lexicographic reachability and safety games,
– $O(\alpha \cdot 2|V|^2)$ time for lexicographic parity (resp. Rabin, Streett, and Muller) games with $\alpha = \alpha_p$ (resp. $\alpha_R$, $\alpha_S$, and $\alpha_M$).

Proof. Let $(G, \Omega_1, \ldots, \Omega_n)$ be a lexicographic game, $\mu \in \{0, 1\}^n$ be a threshold and $v_0$ be an initial vertex. By Theorem 9, solving the threshold problem for this lexicographic game is equivalent to solving a classical game $(G, \Omega)$ with $\Omega = \bigcup_{\nu \in M(\mu)} \cap_{i \in \delta_i} \Omega_i$. Let us recall that $|M(\mu)| \leq n$ and $|\delta_i| \leq n$ for each $\nu \in M(\mu)$.

For the following objectives, we repeatedly apply Theorem 14 and Proposition 15:

– If each $\Omega_i$ is a co-Büchi objective, then the intersection of co-Büchi objectives $\cap_{i \in \delta_i} \Omega_i$ is again a co-Büchi objective $\text{CoBuchi}(d_i)$ and $\bigcup_{\nu \in M(\mu)} \bigcap_{i \in \delta_i} \text{Buchi}(U_i')$ is a generalized Büchi objective with $\leq n$ target sets, and by Theorem 14 the threshold problem is thus decidable in $O(n \cdot |V|^2)$ time.
– If each $\Omega_i$ is a safety objective then $\Omega$ is a generalized reachability objective with $\leq n$ target sets with similar arguments as in the previous item. By Theorem 14 the threshold problem is thus decidable in PSPACE with an algorithm in $O(2^n \cdot (|V| + |E|))$ time.
– If each $\Omega_i$ is a parity objective $\text{Parity}(p_i)$ (resp. a Streett objective with $k_i$ pairs, a Muller objective $\text{Muller}(p_i, F_i)$) then $\Omega$ is a Boolean Büchi objective of polynomial size by Item 5 of Proposition 15. By Theorem 14 the threshold problem is thus decidable in PSPACE. The algorithm runs in $O(|\phi| \cdot 2|V|^2)$ time where $|\phi| \leq n \cdot \sum_{i=1}^{n} d_i^2$ (resp. $|\phi| \leq n \cdot \sum_{i=1}^{n} 2 \cdot k_i$, $|\phi| \leq n \cdot \sum_{i=1}^{n} d_i \cdot |F_i|$). Note that in those cases, the number of used variables is obtained thanks to Item 5 of Proposition 15 and the fact that $\bigcup_{\nu \in M(\mu)} \{\Omega_i \mid i \in \delta_i\} = \{\Omega_i \mid i \in \{1, \ldots, n\}\}$.
– If each $\Omega_i$ is an explicit Muller objective $\text{ExpMuller}(F_i)$ then $\Omega$ is again an explicit Muller objective by Item 4 of Proposition 15. By Theorem 14 the threshold problem is thus decidable in $O(|F| \cdot (|V| \cdot |E| + |F|)^2)$ time, where $|F| \leq \sum_{\nu \in M(\mu)} \min_{j \in \delta_i} |F_j| \leq \sum_{i=1}^{n} |F_i|$. The latter inequality is obtained as follows. Recall that the elements $\nu^i \in M(\mu) \setminus \{\mu\}$ such that $\mu_i = 0$ and $\nu^i = \mu_1 \ldots \mu_{i-1} 10^{n-i}$ (see Definition 6). Hence $\min_{j \in \delta_i} |F_j| \leq |F_i|$ since $i \in \delta_i$. For $\mu \in M(\mu)$, we have $\mu_i = 1$ for some $i$ by Remark 8. We have thus also $\sum_{i=1}^{n} |F_i| \leq |F|$. Therefore $|F| \leq \sum_{i=1}^{n} |F_i|$.

For the following objectives, we rather use Proposition 13, Corollary 10, and the previous items of the proof.

– If each $\Omega_i$ is a Büchi objective then by Proposition 13, $P_1$ can ensure a payoff $\geq \mu$ in the lexicographic Büchi game $(G, \Omega_1, \ldots, \Omega_n)$ if and only if $P_1$ can ensure a payoff $\leq \overline{\mu}$ in the lexicographic co-Büchi game $(\overline{G}, \overline{\Omega_1}, \ldots, \overline{\Omega_n})$. Therefore by Corollary 10, it is equivalent to solve the threshold problem in the lexicographic co-Büchi game $(G, \Omega_1, \ldots, \Omega_n)$ for the threshold $\overline{\mu} + 1$ (by exchanging the role of the two players). Thanks to the first item of the proof, this can be done in $O(n \cdot |V|^2)$ time.
Similarly, if each \( \Omega_i \) is a reachability objective, it is equivalent to solving the threshold problem in the lexicographic safety game \((G, \Omega_1, \ldots, \Omega_n)\). Thanks to the second item of the proof, this can be done in \( \text{PSPACE} \) with an algorithm running in \( O(2^n \cdot (|V| + |E|)) \) time.

Similarly, if each \( \Omega_i \) is a Rabin objective with \( k_i \) pairs, it is equivalent to solving the threshold problem in the lexicographic Streett game \((G, \Omega_1, \ldots, \Omega_n)\). Thanks to the third item of the proof, this can be done in \( \text{PSPACE} \) with an algorithm running in \( O(\|\phi\| \cdot 2^{|V|^2}) \) time where \( |\phi| \leq n \cdot \sum_{i=1}^n 2 \cdot k_i \).

\( \square \)

**Lower bounds.** We now establish the lower bounds of Theorem 16.

**Proposition 18.** The threshold problem is \( \mathcal{P} \)-hard for lexicographic Büchi, co-Büchi and explicit Muller games, and is \( \text{PSPACE} \)-hard for lexicographic reachability, safety, parity, Rabin, Streett, and Muller games.

Moreover, linear memory strategies are necessary for \( \mathcal{P}_1 \) (resp. \( \mathcal{P}_2 \)) for lexicographic Büchi (resp. co-Büchi) games, and exponential memory strategies are both necessary for both players in lexicographic reachability, safety, explicit Muller, parity, Rabin, Streett, and Muller games.

**Proof.** We begin to study the lower bounds on the complexities.

- Lexicographic Büchi (resp. co-Büchi, explicit Muller) games with \( n = 1 \) and \( \mu = 1 \) are a special case of classical Büchi (resp. co-Büchi, explicit Muller) games and solving the latter games is \( \mathcal{P} \)-complete by Theorem 14. Thus, the threshold problem for lexicographic Büchi (resp. co-Büchi, explicit Muller) games is \( \mathcal{P} \)-hard. With the same argument, the threshold problem for lexicographic Muller games is \( \text{PSPACE} \)-hard.

- The \( \text{PSPACE} \)-hardness of the threshold problem for lexicographic reachability (resp. safety) games is obtained thanks to a polynomial reduction from solving generalized reachability games which is \( \text{PSPACE} \)-complete by Theorem 14. Let \((G, \Omega)\) be a generalized reachability game with \( \Omega = \text{GenReach}(U_1, \ldots, U_n) \). Let \((G, \Omega_1, \ldots, \Omega_n)\) be the lexicographic reachability (resp. safety) game such that for all \( i, \Omega_i = \text{Reach}(U_i) \) (resp. \( \Omega_i = \text{Safe}(U_i^c) \)).

  - **Reachability:** Let us show that \( \mathcal{P}_1 \) is winning in \((G, \Omega)\) from an initial vertex \( v_0 \) if and only if \( \mathcal{P}_1 \) can ensure a payoff \( \geq \mu \) from \( v_0 \) in the lexicographic safety game \((G, \Omega_1, \ldots, \Omega_n)\) with \( \mu = 1^n \). Indeed, it is sufficient to note that a play \( \pi \) satisfies \( \text{Payoff}(\pi) \geq \mu \) if and only if \( \pi \) visits each target set \( U_i \).

  - **Safety:** We claim that \( \mathcal{P}_1 \) is winning in \((G, \Omega)\) from \( v_0 \) if and only if \( \mathcal{P}_1 \) can ensure a payoff \( \geq \mu \) from \( v_0 \) in the lexicographic safety game with \( \mu = 0^{n-1}1 \). First, note that a play \( \pi \) satisfies \( \text{Payoff}(\pi) \geq \mu \) if and only if \( \pi \) stays in \( U_i^c \) for some \( i \). Thus, \( \mathcal{P}_1 \) has a strategy to ensure a payoff \( \geq \mu \) from \( v_0 \) in the lexicographic safety game if and only if \( \mathcal{P}_2 \) has a winning strategy in the generalized reachability game \((G, \Omega)\) from \( v_0 \). Since generalized reachability games are determined, the claims follows.

- The hardness of the threshold problem for lexicographic parity games is obtained thanks to a polynomial reduction from solving games \((G, \Omega)\) the objective \( \Omega \) of which is a union of a Rabin objective and a Streett objective, which is known to be \( \text{PSPACE} \)-complete [1]. Let

  \[ \Omega = \text{Rabin}((E_i, F_i)_{i=1}^{n_1}) \cup \text{Streett}((E_i, F_i)_{i=n_1+1}^{n}). \]

By Item 3 of Proposition 15, we can rewrite \( \Omega \) as follows

\[ \Omega = \bigcup_{i=1}^{n_1} (\text{Parity}(p_i)) \cup \bigcap_{i=n_1+1}^{n} \text{Parity}(p_i), \]

where all \( p_i \) are coloring functions. Consider the lexicographic parity game \((G, \Omega_1, \ldots, \Omega_n)\) where \( \Omega_i = \text{Parity}(p_i) \) for all \( i \). We claim that \( \mathcal{P}_1 \) is winning in the game \((G, \Omega)\) from \( v_0 \) if and only if \( \mathcal{P}_1 \) can ensure a payoff \( \geq \mu \) from \( v_0 \) in the lexicographic parity game \((G, \Omega_1, \ldots, \Omega_n)\) where \( \mu = 0^{n-1}1^{n-\mu} \). Indeed, if a play \( \pi \) satisfies \( \text{Payoff}(\pi) \geq \mu \) then either \( \text{Payoff}(\pi) = \mu \) in which case \( \pi \in \bigcap_{i=n_1+1}^{n} \text{Parity}(p_i) \), i.e. \( \pi \) satisfies the Streett objective, or \( \text{Payoff}(\pi) > \mu \) in which case there exists \( 1 \in \{1, \ldots, n_1\} \) such that \( \pi \in \text{Parity}(p_i) \), i.e. \( \pi \) satisfies the Rabin objective. Conversely, if a play \( \pi \) satisfies the Streett or the Rabin objective then \( \text{Payoff}(\pi) \geq \mu \) since \( \text{Payoff}(\pi) \geq \mu \) (resp. \( \mu \)) as soon as \( \pi \) satisfies the Streett (resp. Rabin) objective.
– As parity objectives are a special case of Rabin (resp. Streett) objectives by Item 2 of Proposition 15, we immediately obtain the lower bound of the threshold problem (from the previous item) for both lexicographic Rabin and Streett games.

Based on the given previous reductions, let us now study the lower bounds on the memory size of the strategies.

– In lexicographic reachability and safety games (resp. explicit Muller games, Muller games), exponential memory is necessary for both players from Theorem 14 and the reduction from solving generalized reachability games (resp. explicit Muller games, Muller games) to the threshold problem for those lexicographic games.

– Exponential memory is also necessary for both players in lexicographic parity, Rabin and Streett games. Indeed we gave above a reduction from solving games the objective of which is a union of a Rabin and a Streett objective to the threshold problem for lexicographic parity games. Thus, the latter problem is harder than solving both Rabin and Streett games, which implies that exponential memory is necessary for both players in lexicographic parity games by Theorem 14. This is also the case for lexicographic Rabin and Streett games, since parity objectives are a special case of Rabin (resp. Streett) objectives.

– It remains to show that \( P_1 \) (resp. \( P_2 \)) needs linear memory in lexicographic Büchi (resp. co-Büchi) games. This is obtained thanks to Theorem 14 and the following reductions from generalized Büchi games. Let \( (G, \Omega) \) be a generalized Büchi game with \( \Omega = \text{GenBuchi}(U_1, \ldots, U_n) \)

  - Büchi: let us consider the lexicographic Büchi game \((G, \text{Buchi}(U_1), \ldots, \text{Buchi}(U_n))\). Then \( P_1 \) is winning in \((G, \Omega)\) from an initial vertex \( v_0 \) if and only if \( P_1 \) can ensure a payoff \( \geq 1^n \) from \( v_0 \) in this lexicographic Büchi game. Indeed, it is sufficient to note that a play \( \pi \) belongs to \( \text{GenBuchi}(U_1, \ldots, U_n) \) if and only if \( \pi \) satisfies \( \text{Payoff}(\pi) \geq 1^n \).

  - Co-Büchi: Consider now the lexicographic co-Büchi game \((G, \text{CoBuchi}(U^+_1), \ldots, \text{CoBuchi}(U^+_n))\). Note that any play \( \pi \) belongs to \( \text{GenBuchi}(U_1, \ldots, U_n) \) if and only if \( \pi \not\in \text{CoBuchi}(U^+_i) \) for all \( i \), i.e. \( \text{Payoff}(\pi) = 0^n \). Hence, \( P_1 \) is winning in \((G, \Omega)\) from \( v_0 \) if and only if, taking on the role of \( P_2 \), he can ensure a payoff \( \prec \mu \) in this lexicographic co-Büchi game where \( \mu = 0^{n-1}1 \).

\[ \square \]

**Proof of Theorem 16 and determinacy.** With the two previous propositions, we now prove Theorem 16.

Proof (of Theorem 16). The complexity upper bounds of the threshold problem follow from Proposition 17. From this proposition and Theorem 14, we can derive the upper bounds on the memory size of the winning strategies. The lower bounds on the complexity and on the memory size of the strategies are given in Proposition 18. \[ \square \]

Thanks to Proposition 17 and Corollary 10, we can directly derive the determinacy of the studied lexicographic games as well as the complexity time for computing the value of each vertex and the tight memory requirements for the related optimal strategies.

**Corollary 19.** The lexicographic games of Theorem 16 are determined, and the value of each vertex can be computed with a polynomial time algorithm for lexicographic Büchi, co-Büchi, and explicit Muller games, and with an exponential time algorithm for lexicographic reachability, safety, parity, Rabin, Streett, and Muller games. The memory requirements for the optimal strategies of the players are the same as for the winning strategies in Theorem 16.

Proof. By Proposition 17, there exists a polynomial (resp. exponential) algorithm for the threshold problem for lexicographic Büchi, co-Büchi and, explicit Muller games (resp. lexicographic reachability, safety, parity, Rabin, Streett, and Muller games). Recall that by Corollary 10 and Remark 11, an algorithm for computing the value of each vertex and the corresponding optimal strategies reduces in solving the threshold problem for \( n \) different thresholds. We thus have algorithms for computing the values with the same (polynomial or exponential) complexity as for the threshold problem, and as optimal strategies are in particular winning strategies for the threshold problem, we get the same memory requirements as in Theorem 16. \[ \square \]
5 Fixed parameter complexity

In this section, we study fixed parameter tractability of the threshold problem.

Parameterized complexity and known results on classical games. A parameterized language $L$ is a subset of $\Sigma^* \times \mathbb{N}$, where $\Sigma$ is a finite alphabet, the second component being the parameter of the language. It is called fixed parameter tractable (FPT) if there is an algorithm that determines whether $(x, t) \in L$ in time $f(t) \cdot |x|^c$ time, where $c$ is a constant independent of the parameter $t$ and $f$ is a computable function depending on $t$ only. We also say that $L$ belongs to (the class) FPT. Intuitively, a language is FPT if there is an algorithm running in polynomial time w.r.t the input size times some computable function on the parameter. In this framework, we do not rely on classical polynomial reductions but rather use so called FPT-reductions. An FPT-reduction between two parameterized languages $L \subseteq \Sigma^* \times \mathbb{N}$ and $L' \subseteq \Sigma'^* \times \mathbb{N}$ is a function $R : L \rightarrow L'$ such that

- $(x, t) \in L$ if and only if $(x', t') = R(x, t) \in L'$, where parameter $t'$ only depends on $t$, and
- $R$ is computable by an algorithm that takes $f(t) \cdot |x|^c$ time where $c$ is a constant.

Moreover, if $L'$ is in FPT, then $L$ is also in FPT. Note that a classical polynomial reduction is in particular an FPT-reduction. We refer to [11] for more details on parameterized complexity.

It is proved in [16] that generalized reachability games belong to FPT, with an algorithm running in $O(2^m \cdot (|V| + |E|))$ time where parameter $m$ is the number of reachability objectives. Parity, Rabin, Streett, and Muller games are shown to be FPT-interreducible in [3]. Very recently, Calude and al. provided a quasipolynomial time algorithm for parity games and showed that parity games are FPT [7]. It follows that Rabin, Streett, and Muller games also belong to FPT. The authors of [7] exhibited an algorithm running in $O(|V|^5) + g(d)$ time for parity games (for some computable function $g$), and an algorithm running in $O((d^2 \cdot |V|)^5)$ time for Muller games, where parameter $d$ is the number of colors. All these results are summarized in the next theorem.

Theorem 20. Solving generalized reachability, parity, Rabin, Streett, and Muller games is in FPT. Generalized reachability (resp. parity, Muller) games are solvable with an algorithm running in $O(2^m \cdot (|V| + |E|))$ (resp. $O(|V|^5) + g(d)$, $O((d^2 \cdot |V|)^5)$) time, where parameter $m$ is the number of reachability objectives, parameter $d$ is the number of colors, and $g$ is some computable function.

Parameterized complexities of lexicographic games. In the sequel, we do not focus on lexicographic Büchi, co-Büchi, and explicit Muller games since the corresponding threshold problem is solvable in polynomial time by Theorem 16. The following theorem states that the threshold problem is in FPT for the other lexicographic games of this article, see also Table 3.

Theorem 21. For lexicographic games $(G, \Omega_1, \ldots, \Omega_n)$, the threshold problem is in FPT with

- an algorithm in $O(2^n \cdot (|V| + |E|))$ time for lexicographic reachability and safety games,
- an algorithm in $O((M_{LJ} \cdot |V|)^5)$ time for lexicographic parity and Muller games,
- an algorithm in $O((M_{LJ} \cdot |V|)^5)$ time for lexicographic Rabin and Streett games,

where $M_j = 2^{K_j}$ for $j \in \{1, 2\}$, $K_1 = \sum_{i=1}^n d_i$, $K_2 = \sum_{i=1}^n 2 \cdot k_i$, and for $i \in \{1, \ldots, n\}$, $d_i$ (resp. $k_i$) is the number of colors (resp. pairs).

Before proving this theorem, we need the following result.

Proposition 22. Solving Boolean Büchi games $(G, \Omega)$ is in FPT, with an algorithm in $O(2^M \cdot |\phi| + (M^M \cdot |V|)^5)$ time with $M = 2^m$ such that $m$ is the number of variables of the Boolean Büchi objective $\Omega$.
Table 3. Fixed parameter tractability of lexicographic games $(G, \Omega_1, \ldots, \Omega_n)$: for $i \in \{1, \ldots, n\}$, $d_i$ denotes the number of colors of $p_i$, $k_i$ denotes the number of pairs. For $j \in \{1, 2\}$, $M_j = 2^{k_j}$, where $K_1 = \sum_{i=1}^n d_i$ and $K_2 = \sum_{i=1}^n 2 \cdot k_i$.

Proof. We are going to show the existence of an FPT-reduction from Boolean Büchi games to Muller games. Let $(G, \Omega)$ be a Boolean Büchi game with $\Omega = \texttt{BooleanBuchi}(\phi, U_1, \ldots, U_m)$, where $\phi$ is a Boolean formula over variables $x_1, \ldots, x_m$, and $m$ is seen as a parameter. Let us show how to define an adequate Muller game $(G, \text{Muller}(p, \mathcal{F}))$. This game uses the same game structure and is parameterized by the number of colors. The coloring function $p$ and the family $\mathcal{F}$ are constructed as follows.

To any vertex $v \in V$, we associate the vector $\mu^v \in \{0, 1\}^m$ such that $\mu_i^v = 1$ if $v \in U_i$ and 0 otherwise. Intuitively, we keep track for all $i$, whether a vertex belongs to $U_i$ or not. For each $j \in \{0, \ldots, 2^m - 1\}$, we denote by $\text{bin}(j)$ its binary encoding on $m$ bits (this notation is extended to subsets of integers). Consider the coloring function $p: V \to \{0, \ldots, 2^m - 1\}$ that associates with each vertex $v$ the color $p(v)$ such that $\text{bin}(p(v)) = \mu^v$. The number $M$ of colors is thus equal to $2^m$. One can notice that $(\ast)$ a play $\pi$ visits a vertex $v \in U_i$ if and only if $\pi$ visits a color $i$ with a binary encoding $\text{bin}(i) = \mu$ such that $\mu_i = 1$.

To any subset $F$ of $p(V)$, we associate the truth assignment $\chi(F) \in \{0, 1\}^m$ of variables $x_1, \ldots, x_m$ such that for all $i$, $\chi(F)_i = 1$ if there exists $\mu \in \text{bin}(F)$ with $\mu_i = 1$, and 0 otherwise. The idea (by $(\ast)$) is that the set $F$ of colors visited infinitely often by a play $\pi$ corresponds to the set $\text{Inf}(\pi)$ of vertices visited infinitely often, such that $\chi(F)_i = 1$ if and only if $\text{Inf}(\pi) \cap U_i \neq \emptyset$. We then define $\mathcal{F} = \{F \subseteq p(V) \mid \chi(F) = \phi\}$, that is, $\mathcal{F}$ corresponds to the set of all truth assignments satisfying $\phi$.

In this way we have the announced FPT-reduction: First, parameter $M = 2^m$ only depends on parameter $m$. Second $\mathcal{P}_1$ is winning in $(G, \text{BooleanBuchi}(\phi, U_1, \ldots, U_m))$ from an initial vertex $v_0$ if and only if he is winning in $(G, \text{Muller}(p, \mathcal{F}))$ from $v_0$. Indeed, a play $\pi$ satisfies $(\phi, U_1, \ldots, U_m)$ if and only if the truth assignment $(x_1 = 1 \text{ and only if } \text{Inf}(\pi) \cap U_1 \neq \emptyset)$ satisfies $\phi$. This is equivalent to have that $F = p(\text{Inf}(\pi))$ belongs to $\mathcal{F}$ (by definition of $\chi(F)$), that is, $\pi$ belongs to $\text{Muller}(p, \mathcal{F})$. Third, the construction of the Muller game is in $O(2^{2m} \cdot |\phi|)$ time since it requires $O(|V| + |E|)$ time for the game structure, $O(m \cdot |V|)$ time for the coloring function $p$, and $O(2^{2m} \cdot |\phi|)$ time for the family $\mathcal{F}$.

From this FPT-reduction and by Theorem 20, we have an algorithm solving the Boolean Büchi game in $O(2^{M} \cdot |\phi| + (M^{M} \cdot |V|)^5)$ time, where $M = 2^m$.  

The proof of Theorem 21 is based on Proposition 17, Theorem 20, and Proposition 22.

Proof (of Theorem 21). For each kind of objective, we use Proposition 17 to polynomially reduce the threshold problem of the lexicographic games $(G, \Omega_1, \ldots, \Omega_n)$ to solving some classical games being in FPT by Theorem 20 or Proposition 22. Thus, it follows that those lexicographic games are in FPT with the following algorithms:

- By Proposition 17, the threshold problem for lexicographic reachability and safety games reduces to solving generalized reachability games with at most $n$ target sets. Therefore the threshold problem parameterized by $n$ for lexicographic reachability and safety games is in FPT, with an algorithm running in $O(2^n \cdot (|V| + |E|))$ time by Theorem 20.
- By Proposition 17, the threshold problem for lexicographic parity games reduces to solving Boolean Büchi games of size $|\phi| \leq n \cdot \sum_{i=1}^{n} d_i^2 / 2$ and with $m = \sum_{i=1}^{n} d_i$ variables. Therefore the threshold problem parameterized by $n, d_1, \ldots, d_n$ for parity games is in FPT, with an algorithm running in $O(2^{M} \cdot |\phi| + (M^{M} \cdot |V|)^5)$ time with $M = 2^m$ by Proposition 22. The latter complexity is in $O((M^{M} \cdot |V|)^5)$ since $\sum_{i=1}^{n} d_i^2 \leq m^2$, $n \leq m$, and thus $|\phi| \leq m^3$.  


The arguments are similar for lexicographic Rabin, Streett, and Muller games. The only differences are the bound on size $|\phi|$ and the number $m$ of variables given in Proposition 17. For Rabin and Streett games, $|\phi| \leq n \cdot \sum_{i=1}^{n} 2^{-k_i}$ and $m = \sum_{i=1}^{n} 2^{-k_i}$, and then $|\phi| \leq m^2$. For Muller games, $|\phi| \leq n \cdot \sum_{i=1}^{n} |F_i| \cdot d_i$ and $m = \sum_{i=1}^{n} d_i$, and as $|F_i| \leq 2^{d_i} \leq 2^m$, it follows that $|\phi| \leq m^2 2^m$.

This proves the desired results.

\[ \square \]

6 Conclusion

In this paper, we have studied the computational complexity of the threshold problem for lexicographic games with $\omega$-regular objectives. We have established tight upper and lower bounds and shown fixed parameter tractability when the set of objectives is taken as a parameter. This latter result is particularly relevant as in practice, the number of objectives is usually restricted. We have also studied tight memory requirements for the winning strategies.

As future work, we would like to investigate notions of equilibria for those games, as well as subgame perfection: a strategy is subgame perfect if it ensures the maximal value that is achievable in every subgame. Also, we would like to study lexicographic games with quantitative objectives.

Acknowledgments. We would like to thank Antonia Lechner for useful discussions.

References

1. R. Alur, S. La Torre, and P. Madhusudan. Playing games with boxes and diamonds. In R. M. Amadio and D. Lugiez, editors, CONCUR Proceedings, volume 2761 of LNCS, pages 127–141. Springer, 2003.
2. C. Beeri. On the membership problem for functional and multivalued dependencies in relational databases. ACM Trans. Database Syst., 5(3):241–259, 1980.
3. H. Björklund, S. Sandberg, and S. Vorobyov. On fixed-parameter complexity of infinite games. Technical Report 2003-038, Department of Information Technology, Uppsala University, Aug. 2003.
4. R. Bloem, K. Chatterjee, T. A. Henzinger, and B. Jobstmann. Better quality in synthesis through quantitative objectives. In CAV Proceedings, volume 5643 of Lecture Notes in Computer Science, pages 140–156. Springer, 2009.
5. P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Concurrent games with ordered objectives. In FOSSACS Proceedings, volume 7213 of LNCS, pages 301–315. Springer, 2012.
6. V. Bruyère, Q. Hautem, and J. Raskin. On the complexity of heterogeneous multidimensional games. In 27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Quebec City, Canada, volume 59 of LIPIcs, pages 11:1–11:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
7. C. S. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan. Deciding parity games in quasipolynomial time. In STOC, pages 252–263. ACM, 2017.
8. K. Chatterjee, W. Dvorák, M. Henzinger, and V. Loitzenbauer. Conditionally optimal algorithms for generalized Büchi games. In MFCS, volume 58 of LIPIcs, pages 25:1–25:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
9. K. Chatterjee and M. Henzinger. Efficient and dynamic algorithms for alternating Büchi games and maximal end-component decomposition. J. ACM, 61(3):15:1–15:40, 2014.
10. K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In FOSSACS Proceedings, volume 4423 of LNCS, pages 153–167. Springer, 2007.
11. R. G. Downey and M. R. Fellows. Parameterized Complexity. Springer Publishing Company, Incorporated, 2012.
12. S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. How much memory is needed to win infinite games? In IEEE Proceedings, pages 99–110. IEEE, 1997.
13. E. A. Emerson. Automata, tableaux and temporal logics (extended abstract). In R. Parikh, editor, Logics of Programs, Conference, Brooklyn College, June 17-19, 1985, Proceedings, volume 193 of LNCS, pages 79–88. Springer, 1985.
14. E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs (extended abstract). In 29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988, pages 328–337. IEEE Computer Society, 1988.
15. E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991, pages 368–377. IEEE Computer Society, 1991.

16. N. Fijalkow and F. Horn. The surprizing complexity of generalized reachability games. CoRR, abs/1010.2420, 2010.

17. E. Grädel, W. Thomas, and T. Wilke, editors. Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001], volume 2500 of LNCS. Springer, 2002.

18. F. Horn. Explicit Muller games are PTIME. In FSTTCS, volume 2 of LIPIcs, pages 235–243. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2008.

19. P. Hunter and A. Dawar. Complexity bounds for regular games. In MFCS Proceedings, volume 3618 of LNCS, pages 495–506. Springer, 2005.

20. N. Immerman. Number of quantifiers is better than number of tape cells. J. Comput. Syst. Sci., 22(3):384–406, 1981.

21. M. Jurdzinski. Deciding the winner in parity games is in UP ∩ co-UP. Inf. Process. Lett., 68(3):119–124, 1998.

22. D. A. Martin. Borel determinacy. Annals of Mathematics, 102(2):363–371, 1975.

23. R. McNaughton. Infinite games played on finite graphs. Ann. Pure Appl. Logic, 65(2):149–184, 1993.

24. N. Piterman and A. Pnueli. Faster solutions of Rabin and Streett games. In LICS Proceedings, pages 275–284. IEEE Computer Society, 2006.

25. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In POPL, pages 179–190. ACM Press, 1989.

26. Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. M. Rabinovich, and J. Raskin. The complexity of multi-mean-payoff and multi-energy games. Inf. Comput., 241:177–196, 2015.

27. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. Theor. Comput. Sci., 158, 959:343–359, 1996.