Positional-Player Games

Orna Kupferman
School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel

Noam Shenwald
School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel

Abstract

In *reactive synthesis*, we transform a specification to a system that satisfies the specification in all environments. For specifications in linear-temporal logic, research on *bounded synthesis*, where the sizes of the system and the environment are bounded, captures realistic settings and has lead to algorithms of improved complexity and implementability. In the *game-based* approach to synthesis, the system and its environment are modeled by strategies in a two-player game with an ω-regular objective, induced by the specification. There, bounded synthesis corresponds to bounding the memory of the strategies of the players. The memory requirement for various objectives has been extensively studied. In particular, researchers have identified *positional objectives*, where the winning player can follow a memoryless strategy – one that needs no memory.

In this work we study bounded synthesis in the game setting. Specifically, we define and study *positional-player games*, in which one or both players are restricted to memoryless strategies, which correspond to *non-intrusive control* in various applications. We study positional-player games with Rabin, Streett, and Muller objectives, as well as with weighted multiple Büchi and reachability objectives. Our contribution covers their theoretical properties as well as a complete picture of the complexity of deciding the game in the various settings.

2012 ACM Subject Classification Theory of computation → Formal Languages and Automata Theory, Semantics and Reasoning

Keywords and phrases Games, ω-Regular Objectives, Memory, Complexity

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.

1 Introduction

Synthesis is the automated construction of a system from its specification [46]. A *reactive* system interacts with its environment and has to satisfy its specification in all environments [28]. A useful way to approach synthesis of reactive systems is to consider the situation as a *game* between the system and its environment. In each round of the game, the environment provides an assignment to the input signals and the system responds with an assignment to the output signals. The system wins if the generated computation satisfies the specification. The system and the environment are modeled by *transducers*, which direct them how to assign values to the signals given the history of the interaction so far.

Aiming to study realistic settings, researchers have studied *bounded synthesis* [50]. There, the input to the problem also contains bounds on the sizes of the transducers. Beyond modeling the setting more accurately, bounded synthesis has turned out to have practical advantages. Indeed, bounding the system enables a reduction of synthesis to SAT [19, 22]. Bounding both the system and the environment, a naive algorithm, which essentially checks all systems and environments, was shown to be optimal [38]. In richer settings, for example ones with concurrency, partial visibility, or probability, restricting the available memory is sometimes the key to decidability or to significantly improved complexity [9, 42, 16].

Algorithms for synthesis reduce the problem to deciding a two-player *graph game*. The arena of the game is a graph induced from the specification. The vertices are partitioned between the two players, namely the system and the environment. Starting from an initial vertex, the players jointly generate a *play*, namely a path in the graph, with each player deciding the successor vertex when the play reaches a vertex she owns. The objectives of the players refer to the infinite play that they generate. Each objective α defines a subset of V^ω [41], where V is the set of vertices of the game graph.[1] For example, in games with *Büchi* objectives, α is a subset of V, and a play satisfies α if it visits vertices in α infinitely often.

In the graph-game setting, a *strategy* for a player directs her how to proceed in vertices she owns, and it is *winning* if it guarantees the satisfaction of the player's objective. Winning strategies may need to choose different successors of a vertex in different visits to the vertex. Indeed, choices may depend on the history of the play so far. The number of histories is unbounded, and extensive research has concerned the *memory requirements* for strategies in games with ω-regular objectives, namely the minimal number of equivalence classes to which the histories can be partitioned [51, 18, 8, 10]. For example, it is well known that a winning strategy for a conjunction of k Büchi objectives requires memory k [18]. In practice, the strategies of the system and the environment are implemented by controllers whose state spaces correspond to the different memories that the strategies require. Clearly, we seek winning strategies whose controllers are of minimal size. Of special interest are *memoryless* strategies (also called *positional* strategies), which depend only on the current vertex. For them, all histories are in one equivalence class, leading to trivial controllers.

The need to design efficient controllers has led to extensive research on memoryless strategies. Researchers identified *positional objectives*, namely ones in which both players can use memoryless strategies (formally, an objective α is positional if in all games with objective α, the winner of the game has a memoryless winning strategy), and *half-positional objectives*, namely ones in which one of the players can use a memoryless strategy (formally, an objective α is 1-*positional* (2-*positional*) if in all games with objective α, if the system (environment, respectively) wins the game, then it has a memoryless winning strategy). For example, it is well known that *parity* (and hence, also *Büchi*) objectives are positional [21, 55, 26]. Then, *Rabin* objectives are 1-positional, and their dual *Streett* objectives are 2-positional [21, 35]. On the other hand, *Muller* objectives are not even half-positional. In addition, researchers study positional [25, 14, 45] and 1-positional [36, 44, 7] fragments of objectives that are in general not positional. For example, [11] identifies a class of parity automata that recognize general ω-regular positional objectives.

In this work we take a different view on the topic. Rather than studying the memory required for different types of objectives, we take the approach of bounded synthesis and study games in which the memory of the players is bounded, possibly to a level that prevents them from winning. We focus on controllers of size 1. Note that while transducers of size 1 are not of much interest, controllers of size 1 in the game setting correspond to memoryless strategies, and are thus of great interest. Indeed, in applications like *program repair* [33, 27], *supervisory control* [17] and *regret minimization* [31], researchers have studied *non-intrusive* control, which is modeled by memoryless strategies. Although controllers of size 1 do not always exist, due to the obvious implementation advantages, one can first try to find a controller of size 1, and extend the search if one does not exist. Since memoryless strategies amount to consistent behavior, restricting the memory of the environment models setting in which the system may *learn* an unknown yet static environment [47, 48].

We define and study *positional-player games* (*PPGs*, for short), in which both players are restricted to memoryless strategies, and *half-positional-player games* (*HPPGs*, for short),

[1] As we elaborate in Section 2, all our results apply also for *edge-based objectives*, namely when α defines a subset of E^ω, for the set E of edges of the graph.

in which only one player is restricted to memoryless strategies. We distinguish between *positional-Player 1 games* (*PP1Gs*, for short), in which only the system is restricted to memoryless strategies, and *positional-Player 2 games* (*PP2Gs*, for short), in which only the environment is restricted.

We study PPGs and HPPGs with Rabin, Streett, and Muller objectives, as well as with *weighted multiple objectives* [40]. Such objectives are of the form $\langle \alpha, w, t \rangle$, where $\alpha \subseteq 2^V$, is a set of objectives that are all Büchi, co-Büchi, *reachability*, or *avoid* objectives, $w : 2^\alpha \to \mathbb{N}$ is a non-decreasing, non-negative weight function that maps each subset S of α to a reward earned when exactly all the objectives in S are satisfied, and $t \geq 0$ is a threshold. An objective can be viewed as a *maximization* objective, in which case the goal is to maximize the earned reward (and t serves as a lower bound) or a *minimization* objective, in which case the goal is to minimize the earned reward (and t serves as an upper bound). Weighted multiple objectives can express *generalized objectives*, namely when α contains several objectives, all of which have to be satisfied. A weight function allows for a much richer reference to the underlying objectives: prioritizing them, referring to desired and less desired combinations, and addressing settings where we cannot expect all sub-specifications to be satisfied together.

Studying the theoretical properties of PPGs and HPPGs, we start with some easy observations on how the positionality of the objective type causes different variants of PPGs to coincide. We then study *determinacy* for PPGs and HPPGs; that is, whether there always exists a player that has a winning strategy. Clearly, if a player wins in a game but needs memory in order to win, she no longer wins when restricted to memoryless strategies. Can the other player always win in this case? We prove that the answer is positive for *prefix-independent* objectives (and only for them). On the other hand, when the other player is restricted too, the answer may be negative; thus PPGs of all objective types that are not 1-positional or not 2-positional need not be determined. Also, interestingly, even for objectives that are 1-positional, Player 1 may need memory in order to win against a positional Player 2. Indeed in games in which Player 2 wins when her memory is not bounded, Player 1 can win only by learning and remembering the memoryless strategy of Player 2.

We continue and study the complexity of deciding whether Player 1 or Player 2 win in a given PPG or HPPG. Our results are summarized in Tables 1 and 2. Since memoryless strategies are polynomial in the game graph, the problems for PPGs are clearly in Σ_2^{P}, namely they can be solved in NP using a co-NP oracle. Indeed, one can guess a strategy for the system, then guess a strategy for the environment, and finally check their outcome. Our main technical contribution here is to identify cases in which this naive algorithm is tight and cases where it can be improved. Moving to HPPGs, deciding the winner involves reasoning about the graph induced by a given strategy, and again, the complexity picture is diverse and depends on the half-positionality of the objective, the determinacy of HPPGs, and the succinctness of the objective type. In particular, handling Muller objectives, we have to cope with their complementation (a naive dualization may be exponential) and to introduce and study the *alternating vertex-disjoint paths* problem, which adds alternation to the graph.

2 Preliminaries

2.1 Two-player games

A *two-player game graph* is a tuple $G = \langle V_1, V_2, v_0, E \rangle$, where V_1, V_2 are finite disjoint sets of vertices, controlled by Player 1 and Player 2, respectively, and we let $V = V_1 \cup V_2$. Then, $v_0 \in V$ is an initial vertex, and $E \subseteq V \times V$ is a total edge relation, thus for every $v \in V$, there is $u \in V$ such that $(v,u) \in E$. The size of G, denoted $|G|$, is $|E|$, namely the number of edges in it.

In the beginning of a play in the game, a token is placed on v_0. Then, in each turn, the player that owns the vertex that hosts the token chooses a successor vertex and moves the token to it. Together, the players generate a *play* $\rho = v_0, v_1, \ldots$ in G, namely an infinite path that starts in v_0 and respects E: for all $i \geq 0$, we have that $(v_i, v_{i+1}) \in E$.

For $i \in \{1,2\}$, a *strategy* for Player i is a function $f_i : V^* \cdot V_i \to V$ that maps prefixes of plays that end in a vertex that belongs to Player i to possible extensions in a way that respects E. That is, for every $\rho \in V^*$ and $v \in V_i$, we have that $(v, f_i(\rho \cdot v)) \in E$. Intuitively, a strategy for Player i directs her how to move the token, and the direction may depend on the history of the game so far. The strategy f_i is *finite-memory* if it is possible to replace the unbounded histories in V^* by a finite number of memories. The strategy f_i is *memoryless* if it depends only on the current vertex,[2] thus for all $\rho, \rho' \in V^*$ and $v \in V_i$, we have that $f_i(\rho \cdot v) = f_i(\rho' \cdot v)$. Accordingly, a memoryless strategy is given by a function $f_i : V_i \to V$.

A *profile* is a tuple $\pi = \langle f_1, f_2 \rangle$ of strategies, one for each player. The *outcome* of a profile $\pi = \langle f_1, f_2 \rangle$ is the play obtained when the players follow their strategies in π. Formally, $\mathsf{outcome}(\pi) = v_0, v_1, \ldots \in V^\omega$ is such that for all $j \geq 0$, we have that $v_{j+1} = f_i(v_0, v_1, \ldots, v_j)$, where $i \in \{1,2\}$ is such that $v_j \in V_i$.

A *two-player game* is a pair $\mathcal{G} = \langle G, \psi \rangle$, where $G = \langle V_1, V_2, v_0, E \rangle$ is a two-player game graph, and ψ is a winning condition for Player 1, specifying a subset of V^ω, namely the set of plays in which Player 1 wins. The game is zero-sum, thus Player 2 wins when the play does not satisfy ψ. A strategy f_1 is a *winning strategy* for Player 1 if for every strategy f_2 for Player 2, Player 1 wins in the profile $\langle f_1, f_2 \rangle$, thus $\mathsf{outcome}(\langle f_1, f_2 \rangle)$ satisfies ψ. Dually, a strategy f_2 for Player 2 is a winning strategy for Player 2 if for every strategy f_1 for Player 1, we have that Player 2 wins in $\langle f_1, f_2 \rangle$. We say that Player i *wins* in $\mathcal{G}$ if she has a winning strategy. A game is *determined* if Player 1 or Player 2 wins in it.

2.2 Boolean objectives

For a play $\rho = v_0, v_1, \ldots$, we denote by *reach*(ρ) the set of vertices that are visited at least once along ρ, and we denote by *inf*(ρ) the set of vertices that are visited infinitely often along ρ. That is, *reach*$(\rho) = \{v \in V :$ there exists $i \geq 0$ such that $v_i = v\}$, and *inf*$(\rho) = \{v \in V :$ there are infinitely many $i \geq 0$ such that $v_i = v\}$. For a set of vertices $\alpha \subseteq V$, a play ρ satisfies the *reachability* objective α iff *reach*$(\rho) \cap \alpha \neq \emptyset$, and satisfies the *Büchi* objective α iff *inf*$(\rho) \cap \alpha \neq \emptyset$. The objectives dual to reachability and Büchi are *avoid* (also known as *safety*) and *co-Büchi*, respectively. Formally, a play ρ satisfies an avoid objective α iff *reach*$(\rho) \cap \alpha = \emptyset$, and satisfies a co-Büchi objective α iff *inf*$(\rho) \cap \alpha = \emptyset$.

A *Rabin* objective is a set $\alpha = \{\langle L_i, R_i \rangle\}_{i \in [k]} \subseteq 2^V \times 2^V$ of pairs of sets of vertices. A play ρ satisfies α iff there exists $i \in [k]$ such that ρ visits L_i infinitely often and visits R_i only finitely often. That is, *inf*$(\rho) \cap L_i \neq \emptyset$ and *inf*$(\rho) \cap R_i = \emptyset$, for some $i \in [k]$. The objective dual to Rabin is *Streett*. Formally, a play ρ satisfies a Streett objective $\alpha = \{\langle L_i, R_i \rangle\}_{i \in [k]} \subseteq 2^V \times 2^V$ iff *inf*$(\rho) \cap L_i = \emptyset$ or *inf*$(\rho) \cap R_i \neq \emptyset$, for every $i \in [k]$.

Finally, a *Muller* objective over a set of colors $[k]$ is a pair $\alpha = \langle \mathcal{F}, \chi \rangle$, where $\mathcal{F} \subseteq 2^{[k]}$ specifies desired subsets of colors and $\chi : V \to [k]$ colors the vertices in V. A play ρ satisfies α

[2] Memoryless strategies are sometimes termed *positional* strategies. We are going to use "positional" in order to describe objectives and games in which the players use memoryless strategies, and prefer to leave the adjective used for describing strategies different from the one used for describing objectives and games.

iff the set of colors visited infinitely often along ρ is in $\mathcal{F}$. That is, $\{i \in [k] : \mathit{inf}(\rho) \cap \chi^{-1}(i) \neq \emptyset\} \in \mathcal{F}$. The objective dual to α is the Muller objective $\langle 2^{[k]} \setminus \mathcal{F}, \chi\rangle$.

We define the *size* of an objective as the size of the sets in it. For Muller objectives, we also add the number k of colors. That is, the size of $\langle \mathcal{F}, \chi\rangle$ is $k + \sum_{F\in\mathcal{F}} |F|$.

▶ Remark 1. Objectives in two-player games can also be defined with respect to the edges (rather than vertices) traversed during plays. For some problems, the change is significant. For example, minimization of some types of automata is NP-complete in the state-based setting and can be solved in polynomial time in the edge-based setting [49, 1]. For our study here, all the results also apply for edge-based objectives. For example, for Muller objectives, by adding a new color, we can allow uncolored vertices, which enable a translation of games with colored-edges objectives to games with colored-vertices objectives in a way that preserves winning and memoryless winning strategies. Similar translations can be applied for other types of objectives. ◀

For two types of objectives γ and γ', we use $\gamma \preceq \gamma'$ to indicate that every set of plays that satisfy an objective of type γ can be specified also as an objective of type γ'. For example, Büchi $\preceq$ Rabin, as every Büchi objective α is equivalent to the Rabin objective $\{\langle \alpha, \emptyset\rangle\}$.

An objective type γ is *prefix-independent* if the satisfaction of any γ objective ψ in a play depends only on the infinite suffix of the play. That is, for every play ρ, we have that ρ satisfies ψ iff every infinite suffix ρ' of ρ satisfies ψ. An objective that is not prefix-independent is *prefix-dependent*. Note that objectives defined with respect to $\mathit{inf}(\rho)$ only are prefix-independent, whereas objectives defined with respect to $\mathit{reach}(\rho)$ are prefix-dependent.

2.3 Weighted multiple objectives

A *weighted objective* is a pair $\langle \alpha, w\rangle$, where $\alpha = \{\alpha_1, \ldots, \alpha_k\}$ is a set of k objectives, all of the same type, and $w : 2^\alpha \to \mathbb{N}$ is a weight function that maps subsets of objectives in α to natural numbers. We assume that w is *non-decreasing*: for every sets $S, S' \subseteq \alpha$, if $S \subseteq S'$, then $w(S) \leq w(S')$. In the context of game theory, non-decreasing functions are very useful, as they correspond to settings with *free disposal*, namely when satisfaction of additional objectives does not decrease the utility [43]. We also assume that $w(\emptyset) = 0$. A non-decreasing weight function is *additive* if for every set $S \subseteq \alpha$, the weight of S equals to the sum of weights of the singleton subsets that constitute S. That is, $w(S) = \sum_{\alpha_i \in S} w(\{\alpha_i\})$. An additive weight function is thus given by $w : \alpha \to \mathbb{N}$, and is extended to sets of objectives in the expected way, thus $w(S) = \sum_{\alpha_i \in S} w(\alpha_i)$, for every $S \subseteq \alpha$.

For a play ρ, let $\mathsf{sat}(\rho, \alpha) \subseteq \alpha$ be the set of objectives in α that are satisfied in ρ. The *satisfaction value* of $\langle \alpha, w\rangle$ in ρ, denoted $\mathsf{val}(\rho, \alpha, w)$, is then the weight of the set of objectives in α that are satisfied in ρ. That is, $\mathsf{val}(\rho, \alpha, w) = w(\mathsf{sat}(\rho, \alpha))$.

Weighted objectives can be viewed as either maximization or minimization objectives. That is, the goal is to maximize or minimize the weight of the set of objectives satisfied. [3] Note that when w is *uniform*, thus when $w(S) = |S|$ for all $S \subseteq \alpha$, the goal is to maximize or minimize the number of satisfied objectives. A special case of the latter, known in the literature as *generalized* conditions, is when we aim to satisfy all or at least one objective. We denote different classes of weighted objectives by acronyms in {MaxW, MinW, All, Exists} × {R, A, B, C}, describing the way we refer to the satisfaction value and the objectives type: reachability (R), avoid (A), Büchi (B), or co-Büchi (C).

[3] Note that adding a threshold to the objective makes satisfaction binary, and the corresponding optimization problem can be solved using a binary search.

Formally, for a play $\rho \in V^\omega$, an objective type $\gamma \in \{\text{R, A, B, C}\}$, a set $\alpha = \{\alpha_1, \ldots, \alpha_k\}$ of objectives, a weight function $w : 2^\alpha \to \mathbb{N}$, and a threshold $t \in \mathbb{N}$, we have the following winning conditions.

- ρ satisfies a MaxW-γ objective $\langle \alpha, w, t\rangle$ if $\mathsf{val}(\rho, \alpha, w) \geq t$.
- ρ satisfies a MinW-γ objective $\langle \alpha, w, t\rangle$ if $\mathsf{val}(\rho, \alpha, w) \leq t$.
- ρ satisfies an All-γ objective α if $|\mathsf{sat}(\rho, \alpha)| = |\alpha|$.
- ρ satisfies an Exists-γ objective α if $|\mathsf{sat}(\rho, \alpha)| \geq 1$.

For All-γ and Exists-γ objectives, we omit the weight function from the specification of the objective. Note that for all objective types γ, we have that All-$\gamma \preceq$ MaxW-γ and Exists-$\gamma \preceq$ MaxW-γ. Also, by [40], MaxWB $\preceq$ Muller. Indeed, it is easy to specify in a Muller objective all sets S such that $w(S) \geq t$. Since we assume non-decreasing weight functions, the other direction does not hold, thus Muller $\not\preceq$ MaxWB. In Section 8, we study dualities in weighted objectives and extend the above observations to minimization games and to underlying co-Büchi and avoid objectives. Finally, note that MaxW-γ and MinW-γ objectives are prefix-independent iff γ is prefix-independent.

3 Positional Objectives and PPGs

For $i \in \{1, 2\}$, an objective type γ is *i-positional* if for every γ-game $\mathcal{G}$, Player i wins in $\mathcal{G}$ iff she has a memoryless winning strategy in $\mathcal{G}$. Then, γ is *positional* iff it is both 1-positional and 2-positional, thus the winner of every γ-game has a memoryless winning strategy. If γ is neither 1-positional nor 2-positional, we say that it is *non-positional*.[4] It is known that the objective types reachability, Büchi, and parity (and thus also avoid and co-Büchi) are positional [21], Rabin is 1-positional (and thus Streett is 2-positional) [35], and Muller is non-positional [18]. As for weighted multiple objectives, MaxWB and MinWB are 2- and 1-positional, respectively, whereas MaxWR and MinWR are non-positional [40].

We lift the notion of positionality to games, studying settings in which one or both players are restricted to memoryless strategies. For $i \in \{1, 2\}$, a game is a *positional-Player i game* if Player i is restricted to memoryless strategies, and is a *positional-player game* (PPG, for short) if both players are restricted to memoryless strategies. Positional-Player 1 and positional-Player 2 games are also called *half-positional-player games* (HPPG, for short).[5]

For a game $\mathcal{G}$, we use $\mathsf{Pos}\text{-}\mathcal{G}$, $\mathsf{1Pos}\text{-}\mathcal{G}$, and $\mathsf{2Pos}\text{-}\mathcal{G}$ to denote the positional-player and half-positional-player variants of $\mathcal{G}$, respectively. Formally, for a game $\mathcal{G} = \langle G, \psi\rangle$, we have the following.

- Player 1 wins $\mathsf{Pos}\text{-}\mathcal{G}$ iff she has a memoryless strategy f_1 such that for every memoryless strategy f_2 for Player 2, we have that $\mathsf{outcome}(\langle f_1, f_2\rangle)$ satisfies ψ.
- Player 1 wins $\mathsf{1Pos}\text{-}\mathcal{G}$ iff she has a memoryless strategy f_1 such that for every strategy f_2 for Player 2, we have that $\mathsf{outcome}(\langle f_1, f_2\rangle)$ satisfies ψ.
- Player 1 wins $\mathsf{2Pos}\text{-}\mathcal{G}$ iff she has a strategy f_1 such that for every memoryless strategy f_2 for Player 2, we have that $\mathsf{outcome}(\langle f_1, f_2\rangle)$ satisfies ψ.

[4] In the literature, 1-positional and 2-positional objectives are sometimes termed *positional*, and positional objectives are sometimes termed *bipositional* [11].

[5] An automaton is *determinizable by pruning* if it embodies an equivalent deterministic automaton, thus if nondeterminism can be resolved in a memoryless manner [3]. This may hint on a relation to PP1Gs. However, the labels on the transitions of automata make the setting different, and reasoning about determinization by pruning is different from reasoning about PP1Gs [37, 2].

▶ **Example 2.** Consider the AllB game $\mathcal{G} = \langle G, \alpha \rangle$, for G that appears in Fig. 1, and $\alpha = \{\{u_1, u_2\}, \{d_1, d_2\}\}$. Since the objective of Player 1 is to satisfy both Büchi objectives in α, she wins iff for infinitely many traversals of G, the choices of the players from the vertices v_1 and v_2 as to whether they go up or down do not match.

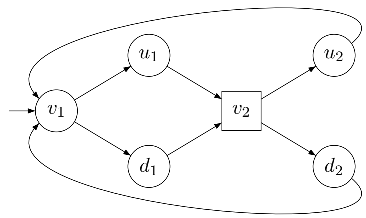

Figure 1 The game graph G. Drawing game graphs, vertices owned by Player 1 are circles, and vertices owned by Player 2 are squares.

It is easy to see that Player 1 wins in $\mathcal{G}$. Indeed, a winning strategy f_1 for Player 1 can move the token up to u_1 after visits of the token in d_2, and move the token down to d_1 after visits of the token in u_2. Since $2\mathsf{Pos}\text{-}\mathcal{G}$ only restricts the strategies that Player 2 may use, the strategy f_1 is a winning strategy for Player 1 also in $2\mathsf{Pos}\text{-}\mathcal{G}$.

On the other hand, Player 1 does not win in $1\mathsf{Pos}\text{-}\mathcal{G}$. Indeed, for every memoryless strategy f_1 for Player 1, Player 2 has a strategy f_2, in fact a memoryless one, that matches the choice Player 1 makes in v_1. That is, if $f_1(v_1) = u_1$, then $f_2(v_2) = u_2$, and if $f_1(v_1) = d_1$, then $f_2(v_2) = d_2$. Since f_2 is memoryless, Player 1 does not win in $\mathsf{Pos}\text{-}\mathcal{G}$ either.

As for Player 2, she clearly does not win in $\mathcal{G}$ and $2\mathsf{Pos}\text{-}\mathcal{G}$. In $1\mathsf{Pos}\text{-}\mathcal{G}$, a winning strategy f_2 for Player 2 can move the token up to u_2 after visits in u_1, and move the token down to d_2 after visits in d_1. The strategy f_2 requires memory, and Player 2 does not have a winning strategy in $\mathsf{Pos}\text{-}\mathcal{G}$. Indeed, for every memoryless strategy for Player 2, Player 1 has a memoryless strategy in which the choice in v_1 does not match the choice Player 2 makes in v_2, causing both Büchi objectives to be satisfied. Thus, interestingly, even though AllB objectives are 2-positional, Player 2 needs memory in order to win in $1\mathsf{Pos}\text{-}\mathcal{G}$. ◀

We conclude with some easy observations about PPGs and HPPGs and positional and half-positional objectives. The first follows immediately from the fact that the restriction to memoryless strategies only reduces the set of possible strategies. For a player $i \in \{1, 2\}$, we use $\tilde{i}$ to denote the other player; thus, if $i = 1$, then $\tilde{i} = 2$, and if $i = 2$, then $\tilde{i} = 1$.

▶ **Theorem 3.** *For every game $\mathcal{G}$ and $i \in \{1, 2\}$, all the following hold.*

- *If Player i wins $i\mathsf{Pos}\text{-}\mathcal{G}$, then she also wins $\mathcal{G}$ and $\mathsf{Pos}\text{-}\mathcal{G}$.*
- *If Player i wins $\mathsf{Pos}\text{-}\mathcal{G}$, then she also wins $\tilde{i}\mathsf{Pos}\text{-}\mathcal{G}$.*

We continue with observations that use the positionality of the objective type in order to relate different variants of PPGs. The proof of Theorem 4 follows immediately from the definitions and the proof of Theorem 5 can be found in the full version.

▶ **Theorem 4.** *Consider $i \in \{1, 2\}$ and an i-positional objective type γ. For every γ-game $\mathcal{G}$, Player i wins $\mathcal{G}$ iff Player i wins $i\mathsf{Pos}\text{-}\mathcal{G}$. In particular, if γ is positional, then Player i wins $\mathcal{G}$ iff Player i wins $\mathsf{Pos}\text{-}\mathcal{G}$, $1\mathsf{Pos}\text{-}\mathcal{G}$, and $2\mathsf{Pos}\text{-}\mathcal{G}$.*

▶ **Theorem 5.** *Consider $i \in \{1, 2\}$, and an $\tilde{i}$-positional objective type γ. For every γ-game $\mathcal{G}$, Player i wins $i\mathsf{Pos}\text{-}\mathcal{G}$ iff Player i wins $\mathsf{Pos}\text{-}\mathcal{G}$.*

4 Determinacy of PPGs and HPPGs

Games with ω-regular objectives enjoy determinacy: in every game, one player wins [41]. In this section we study the determinacy of PPGs and HPPGs. As expected, restricting the strategies of both players makes some games undetermined. Surprisingly, we are able to prove that for every prefix-independent objective ψ that is not positional, there is an undetermined PPG with the objective ψ. On the other hand, all HPPGs with prefix-independent objectives are determined. Thus, if a player needs memory in order to win a game with a prefix-independent objective, then restricting her to use only memoryless strategies without restricting her opponent, not only prevents her from winning, but also makes the opponent winning.

▶ **Theorem 6.** *For every objective ψ that is prefix-independent, not positional, and requires finite memory, there is an undetermined PPG with objective ψ. In particular, AllB and ExistsC PPGs need not be determined.*

Proof. Since ψ is not positional, then it is not 1-positional or not 2-positional. We consider the case ψ is not 1-positional. The case ψ is not 2-positional follows, as Player 2 has an objective that is not 1-positional. In the full version, we prove that for every objective ψ as above, there is a set V of vertices, a finite graph G over V consisting of a vertex $v \in V$ and two simple and vertex-disjoint paths $p, q \in V^*$, such that $v \cdot p$ and $v \cdot q$ are cycles, and either $(v \cdot p \cdot v \cdot q)^\omega$ satisfies ψ, and $(v \cdot p)^\omega$ and $(v \cdot q)^\omega$ do not satisfy ψ, or $(v \cdot p \cdot v \cdot q)^\omega$ does not satisfy ψ, and $(v \cdot p)^\omega$ and $(v \cdot q)^\omega$ satisfy ψ. We then take two copies of G in order to construct a PPG with the same objective in which no player wins. ◀

We continue to HPPGs. Note that for $\mathcal{G}$ in Example 2, allowing a single player to use memory makes this player win. Indeed, as the other player uses a memoryless strategy, she commits on her strategy in vertices visited along the play. As we formalize in Theorem 7 below, the player that uses memory can then learn these commitments, and follow a strategy that is tailored for them. Since learning is performed during a traversal of a prefix of the play, this works only for objectives that are prefix-independent.

▶ **Theorem 7.** *Consider a prefix-independent objective type γ. For every γ-game $\mathcal{G}$ and $i \in \{1, 2\}$, we have that iPos-$\mathcal{G}$ is determined.*

Proof. Consider a γ-game $\mathcal{G} = \langle G, \psi \rangle$. Let $G = \langle V_1, V_2, v_0, E \rangle$. We show that Player 1 wins in 1Pos-$\mathcal{G}$ iff Player 2 does not win 1Pos-$\mathcal{G}$. The proof for 2Pos-$\mathcal{G}$ is similar.

Clearly, if Player 1 wins 1Pos-$\mathcal{G}$, then Player 2 does not win 1Pos-$\mathcal{G}$. For the second direction, assume that Player 1 does not win 1Pos-$\mathcal{G}$. We show that then, Player 2 has a strategy that wins against all memoryless strategies of Player 1. Let $V' \subseteq V$ be the set of vertices from which Player 1 does not have a memoryless winning strategy in G, and let G' be the sub-graph of G that contains only vertices in V'. That is, $G' = \langle V_1 \cap V', V_2 \cap V', v_0, E \cap (V' \times V') \rangle$. Note that indeed $v_0 \in V'$ since Player 1 does not win 1Pos-$\mathcal{G}$.

Let $f_1, \ldots, f_k$ be all the memoryless strategies for Player 1 in G'. Since all of them are not winning strategies, there exist strategies $g_1, \ldots, g_k$ for Player 2 in G' such that $\mathsf{outcome}(\langle f_i, g_i \rangle)$ does not satisfy ψ, from every vertex in V', for every $i \in [k]$. Note that for every $i \in [k]$, such a strategy g_i exists only because f_i is restricted to vertices in V'. We construct from $g_1, \ldots, g_k$ a winning strategy g for Player 2 in 1Pos-$\mathcal{G}$.

Intuitively, starting with $i = 1$, the strategy g follows g_i, and whenever Player 1 takes a transition from a vertex $v \in V_1 \cap V'$ to a vertex u, it checks that the transition is consistent with f_i; that is, whether $f_i(v) = u$. If this is not the case, the strategy g updates i to the minimal $j > i$ such that $f_j(v) = u$. In the full version, we define g formally and prove that an index j as above always exists, and that the process eventually stabilizes, simulating a game in which Player 1 uses some memoryless strategy f_i, and Player 2 uses a strategy that wins against f_i, making g a winning strategy for Player 2. ◀

As for prefix-dependent objective types, here the vertices traversed while the unrestricted player learns the memoryless strategy of the restricted player may play a role in the satisfaction of the objective, and the picture is different (see detailed proof in the full version):

▶ **Theorem 8.** *For every objective type γ such that AllR $\preceq \gamma$ or ExistsA $\preceq \gamma$, both γ-PPGs and γ-HPPGs need not be determined.*

5 The Complexity of PPGs and HPPGs

Given a game $\mathcal{G}$, we would like to decide whether Player 1 wins in $\mathsf{1Pos}$-$\mathcal{G}$, $\mathsf{2Pos}$-$\mathcal{G}$, and Pos-$\mathcal{G}$. The complexity of the problem depends on the type of objective in $\mathcal{G}$. In this section we present general complexity results for the problem. Then, in Sections 6, 7, and 8, we provide an analysis for the different objective types. Note that since not all PPGs and HPPGs are determined, the results for Player 2 do not follow immediately from the results for Player 1. They do, however, follow from results about the dual objective.

5.1 General upper bound results

We start with upper bounds. We say that an objective type γ is *path-efficient* if given a lasso shape path $\rho_1 \cdot (\rho_2)^\omega$, for $\rho_1 \in V^*$ and $\rho_2 \in V^+$, checking whether $\rho_1 \cdot (\rho_2)^\omega$ satisfies a γ objective ψ can be done in time polynomial in $|\rho_1|, |\rho_2|$, and $|\psi|$. We say that an objective type γ is *all-path-efficient* if given a graph G, checking whether all the infinite paths in G satisfy a γ objective ψ can be done in time polynomial in $|G|$ and $|\psi|$.

The complexity class Σ_2^{P} contains all the problems that can be solved in polynomial time by a nondeterministic Turing machine augmented by a co-NP oracle. Since memoryless strategies are of polynomial size, the complexity class Σ_2^{P} is a natural class in the context of PPGs. Formally, we have the following (see proof in the full version).

▶ **Theorem 9.** *Consider a path-efficient objective type γ. For every γ-game $\mathcal{G}$, deciding whether Player 1 wins in* Pos-$\mathcal{G}$ *can be done in Σ_2^{P}.*

For HPPGs, once a memoryless strategy for Player 1 is guessed, one has to check all the paths in the graph that are consistent with it. Formally, consider a γ-game $\mathcal{G} = \langle G, \psi \rangle$ and a memoryless strategy f_1 for Player 1. Let G_{f_1} be the graph obtained from G by removing edges that leave vertices in V_1 that do not agree with f_1. Clearly, f_1 is a winning strategy iff all the paths in G_{f_1} satisfy ψ. Hence, we have the following (see proof in the full version).

▶ **Theorem 10.** *Consider an all-path-efficient objective type γ. For every γ-game $\mathcal{G}$, deciding whether Player 1 wins* $\mathsf{1Pos}$-$\mathcal{G}$ *can be done in NP.*

Note that beyond the fact that not all objective types are known to be all-path-efficient, the upper bounds that follow from Theorems 9 and 10 need not be tight.

5.2 General lower bound constructions

For our lower bounds, we use reductions from two well-known problems: QBF (Quantified Boolean Formulas) and its special case 2QBF, which are standard problems for proving hardness in PSPACE and Σ_2^{P}, respectively. Here, we define these problems and set some infrastructure for the reductions. Consider a set $X = \{x_1, \ldots, x_n\}$ of variables. Let $\overline{X} = \{\overline{x_1}, \ldots, \overline{x_n}\}$. A QBF formula is of the form $\Phi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \varphi$, where $Q_1, \ldots, Q_n \in \{\exists, \forall\}$ are existential and universal quantifiers, and φ is a propositional formula over $X \cup \overline{X}$. The QBF problem is to decide whether Φ is valid.

A QBF formula Φ is in 2QBF if there is only one alternation between existential and universal quantification in Φ, and the external quantification is existential. In 2QBF, we can assume that φ is given in 3DNF. For nicer presentation, we use $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ for the sets of existentially and universally quantified variables, respectively. That is, a 2QBF formula is of the form $\Phi = \exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \varphi$, where $\varphi = C_1 \vee \ldots \vee C_k$, for some $k \geq 1$, and for every $1 \leq i \leq k$, we have $C_i = (l_i^1 \wedge l_i^2 \wedge l_i^3)$, with $l_i^1, l_i^2, l_i^3 \in X \cup \overline{X} \cup Y \cup \overline{Y}$.

Below we describe two game graphs that are used in our reductions.

Consider a set of variables $X = \{x_1, \ldots, x_n\}$, and a QBF formula $\Phi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \varphi$, where $Q_1, \ldots, Q_n \in \{\exists, \forall\}$. The two-player game graph G_Φ lets Player 1 choose assignments to the existentially-quantified variables and Player 2 choose assignments to the universally-quantified variables. The idea is that when the players use memoryless strategies, they repeat the same choices. For the special case of 2QBF (see Figure 2), the order of the quantifiers on the variables corresponds to the way the strategies of the players are quantified. For QBF, the quantification on the variables is arbitrary, and G_Φ is used in the context of HPPGs. In the full version, we define G_Φ formally.

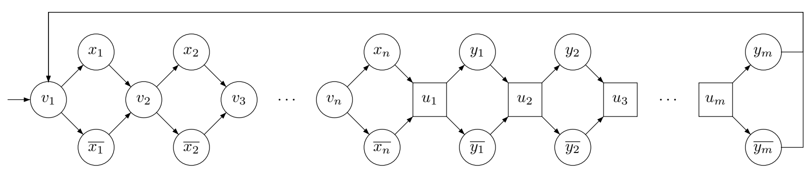

Figure 2 The game graph G_Φ for a 2QBF formula $\Phi = \exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m \varphi$.

In G_Φ, the vertices that correspond to the assignment to the last variable go back to the initial vertex. We use $Reach(G_\Phi)$ to denote the game graph obtained from G_Φ by replacing these edges by self-loops. Note that G_Φ and $Reach(G_\Phi)$ are independent of φ and only depend on the partition of X in Φ to existentially and universally quantified variables.

Next, we define a game graph F_Φ, induced by a QBF formula $\Phi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \varphi$, for φ in 3DNF. The game proceeds in two phases that are repeated infinitely often and are described as follows (see the full version for the full details, and Figure 3 below for an example). In the assignment phase, the players choose an assignment to the variables in X. This is done as in G_Φ. Then, the game continues to a checking phase, in which Player 2 tries to refute the chosen assignment by showing that every clause has a literal that is evaluated to **false**. For this, the game sequentially traverses, for every clause C_i, a vertex from which Player 2 can visit *refute-literal vertices*, associated with $\overline{l_i^1}$, $\overline{l_i^2}$, and $\overline{l_i^3}$, and continues to the next clause. Thus, for every clause, Player 2 chooses a vertex that corresponds to the negation of one of its literals.

Intuitively, when the players use memoryless strategies, the assignment phase induces an assignment to the variables. An assignment satisfies φ iff there exists a clause all whose literals are evaluated to **true**, which holds iff Player 2 is forced to choose a refute-literal vertex that corresponds to a literal evaluated to **false**. Then, Φ is valid iff there exists a memoryless strategy for Player 1 such that for every memoryless strategy for Player 2, there exists a literal l_i^j such that both l_i^j and a refute-literal vertex that corresponds to $\overline{l_i^j}$ are visited infinitely often.

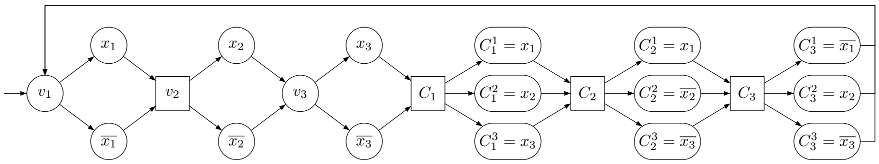

Figure 3 The game graph F_Φ, for $\Phi = \exists x_1 \forall x_2 \exists x_3 (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3})$.

We use $Reach(F_\Phi)$ to denote the game graph obtained from F_Φ by replacing the edges from the rightmost refute-literal vertices to the initial vertex by self-loops.

6 Rabin and Streett PPGs and HPPGs

In this section we study Rabin and Streett (and their respective special cases ExistsC and AllB) PPGs and HPPGs. Our results are summarized in Table 1 below.

<table>
<tr><th>Type</th><th>Positionality</th><th>P1 wins $\mathcal{G}$</th><th>P1 wins Pos-$\mathcal{G}$</th><th>P1 wins 1Pos-$\mathcal{G}$</th><th>P1 wins 2Pos-$\mathcal{G}$</th></tr>
<tr><td>ExistsC</td><td>1-positional</td><td>PTIME [12]</td><td rowspan="2">Σ_2^{P}-complete (Theorem 12)</td><td>PTIME (Theorem 11)</td><td rowspan="2">co-NP-complete (Theorem 11)</td></tr>
<tr><td>Rabin</td><td>1-positional</td><td>NP-complete [20]</td><td>NP-complete (Theorem 11)</td></tr>
<tr><td>AllB</td><td>2-positional</td><td>PTIME [12]</td><td rowspan="2">NP-complete (Theorem 13)</td><td rowspan="2">NP-complete (Theorem 11)</td><td>PTIME (Theorem 11)</td></tr>
<tr><td>Streett</td><td>2-positional</td><td>co-NP-complete [20]</td><td>co-NP-complete (Theorem 11)</td></tr>
<tr><td>Muller</td><td>non-positional</td><td>PSPACE-complete [30]</td><td>Σ_2^{P}-complete (Theorem 15)</td><td>NP-complete (Theorem 17)</td><td>co-NP-complete (Theorem 17)</td></tr>
</table>

Table 1 Complexity results for ω-regular PPGs and HPPGs. For positional objectives, the problems coincide with deciding usual games. Accordingly, they can be solved in PTIME for reachability, avoid, Büchi and co-Büchi objectives [5, 32, 55, 53], and in UP $\cap$ co-UP for parity objectives [34].

We start with HPPGs (see proof in the full version).

▶ **Theorem 11.** *Deciding whether Player 1 wins:*
- *an ExistsC PP1G or a AllB PP2G can be done in polynomial time.*
- *a γ-PP1G is NP-complete, for $\gamma \in \{$Rabin, AllB, Streett$\}$.*
- *a γ-PP2G is co-NP-complete, for $\gamma \in \{$Streett, ExistsC, Rabin$\}$.*

▶ **Theorem 12.** *Deciding whether Player 1 wins a Rabin PPG is Σ_2^{P}-complete. Hardness in Σ_2^{P} applies already for ExistsC PPGs.*

Proof. The upper bound follows from Theorem 9. For the lower bound, we describe a reduction from 2QBF. That is, given a 2QBF formula Φ, we construct an ExistsC game $\mathcal{G}_\Phi$ such that $\Phi =$ **true** iff Player 1 wins Pos-$\mathcal{G}_\Phi$.

Consider a 2QBF formula $\Phi = \exists X \forall Y \varphi$ such that $\varphi = C_1 \vee \ldots \vee C_k$ and $C_i = (l_i^1 \wedge l_i^2 \wedge l_i^3)$. Recall the game graph G_Φ defined in Section 5.2, and recall that memoryless strategies for the players induce assignments to the variables in X and Y in a way that corresponds to their quantification in Φ. Thus, we only need to define an objective that captures the satisfaction of some clause in φ. For this, we define the ExistsC objective $\alpha = \{\{\overline{l_i^1}, \overline{l_i^2}, \overline{l_i^3}\} : i \in [k]\}$. Thus, each clause C_i of φ contributes to α a set with the negations of the literals in C_i. Since for each variable $z \in X \cup Y$, a play visits exactly one of the literal vertices z and $\overline{z}$ infinitely often, the play satisfies a co-Büchi objective in α iff the chosen assignment satisfies φ, and so $\Phi = \mathbf{true}$ iff Player 1 wins in Pos-$\langle G_\Phi, \alpha\rangle$ (see proof in the full version). ◀

Theorem 12 shows that Rabin PPGs are strictly more complex than general Rabin games. For the dual Streett objective, the 1-positionality of Rabin objectives does make the problem easier. Formally, we have the following (see proof in the full version).

▶ **Theorem 13.** *Deciding whether Player 1 wins a Streett PPG is NP-complete. Hardness in NP applies already for AllB PPGs.*

7 Muller PPGs and HPPGs

In this section, we study the complexity of Muller PPGs and HPPGs. Our results are summarized in Table 1. We start with PPGs. It is not hard to see that Muller is path-efficient, and so Theorem 9 implies that the problem of deciding whether Player 1 wins a Muller PPG is in Σ_2^{P}. In Theorem 12, we proved that the problem is Σ_2^{P}-hard for Rabin and even ExistsC PPGs, and as ExistsC $\preceq$ Muller, it may seem that a similar lower bound would be easy to obtain. The translation from an ExistsC objective to a Muller objective may, however, be exponential [6], which is in particular the case for the objective used in the lower bound proof in Theorem 12. Note that the AllB objective used in the lower bound proof in Theorem 13 can be translated with no blow-up to a Muller objective, but deciding whether Player 1 wins an AllB PPG is NP-complete.

Accordingly, our first heavy technical result in the context of Muller PPGs is a proof of their Σ_2^{P}-hardness. For this, we add an alternation to the problem of vertex-disjoint paths, used in [33] in order to prove NP-hardness for AllB PPGs. We show that a Muller objective can capture the alternation, which lifts the complexity from NP to Σ_2^{P}.

We first need some definitions and notations. Consider a directed graph $G = \langle V, E\rangle$. A path p in G is *simple* if each vertex appears in p at most once. Two simple paths p and q in G are *vertex-disjoint* iff they do not have vertices in common, except maybe their first and last vertices. The *vertex-disjoint-paths* (VDP, for short) problem is to decide, given G and two vertices $s, t \in V$, whether there exist two vertex-disjoint paths from s to t and from t to s in G. The complementing problem, termed NVDP, is to decide whether there do not exist two vertex-disjoint paths from s to t and from t to s in G.

Now, the *alternating NVDP* problem (ANVDP, for short) is to decide, given a two-player game graph $G = \langle V_1, V_2, E\rangle$ and two vertices $s, t \in V$, whether there exists a memoryless strategy f_1 for Player 1 in G such that the $\langle G_{f_1}, s, t\rangle$ is in NVDP. That is, there do not exist two vertex-disjoint paths from s to t and from t to s in G_{f_1}.

▶ **Lemma 14.** *ANVDP is Σ_2^{P}-complete.*

Proof. VDP is known to be NP-complete [24, 33], making NVDP co-NP-complete, and inducing a Σ_2^{P} upper bound. NP-hardness is shown in [24] by a reduction from 3SAT (see full version for details): given a 3CNF formula φ over variables in X, a graph G_φ with vertices s and t is constructed such that paths from s to t correspond to choosing an assignment to the variables in X, and then choosing one literal in every clause in φ. Then, a path p from s to t has a path q from t to s that is vertex-disjoint from p iff all the chosen literals are evaluated to **true** in the chosen assignment. Accordingly, $\langle G_\varphi, s, t\rangle$ is in VDP iff φ is satisfiable.

For ANVDP, we use a similar reduction, but from 2QBF. Given a 2QBF formula $\Phi = \exists X \forall Y \varphi$ where φ is in 3DNF, we construct a game graph similar to $G_{\overline{\varphi}}$ for the 3CNF formula $\overline{\varphi}$ with vertices s and t, where memoryless strategies for Player 1 correspond to assignments to the variables in X. Every such assignment defines a sub-graph that is in VDP iff there exists an assignment to the variables in Y that, together with the assignment Player 1 chose, satisfies $\overline{\varphi}$. Accordingly, there exists a memoryless strategy for Player 1 such that the corresponding sub-graph, together with s and t, is in NVDP iff $\Phi = \mathbf{true}$. ◀

▶ **Theorem 15.** *Deciding whether Player 1 wins a Muller PPG is Σ_2^{P}-complete.*

Proof. The upper bound follows from Theorem 9. For the lower bound, we describe a reduction from ANVDP, which, by Lemma 14, is Σ_2^{P}-hard. Consider a two-player game graph $G = \langle V_1, V_2, s, E\rangle$, and a vertex $t \in V \setminus \{s\}$. We define a Muller game $\mathcal{G} = \langle G, \langle \mathcal{F}, \chi\rangle\rangle$ such that Player 1 wins $\mathsf{Pos}\text{-}\mathcal{G}$ iff $\langle G, s, t\rangle$ is in ANVDP.

Consider the set of colors $\{1, 2, 3\}$. We define $\chi : V \to \{1, 2, 3\}$, where $\chi(s) = 1$, $\chi(t) = 2$, and $\chi(v) = 3$ for every $v \in V \setminus \{s, t\}$. Then, $\mathcal{F} = \{\{1\}, \{2\}, \{3\}, \{1, 3\}, \{2, 3\}\}$. Thus, Player 1 wins in a play iff it does not visit both s and t infinitely often. For the correctness of the construction, note that for every memoryless strategy f_1 for Player 1 in G, we have that f_1 is winning in $\mathsf{Pos}\text{-}\mathcal{G}$ iff Player 2 does not have a memoryless winning strategy in G_{f_1}, which she has iff there exists a simple cycle in G_{f_1} that visits both s and t. Such a simple cycle exists iff there exist two vertex-disjoint paths from s to t and from t to s in G_{f_1}. That is, if $\langle G_{f_1}, s, t\rangle$ is in VDP. Accordingly, Player 1 wins $\mathsf{Pos}\text{-}\mathcal{G}$ iff $\langle G, s, t\rangle$ is in ANVDP. ◀

We continue to Muller HPPGs. Note that beyond implying membership in NP for the problem of deciding whether Player 1 wins a Muller PP1G, establishing the all-path efficiency of the Muller objective also implies that the universality problem for universal Muller word automata can be solved in polynomial time.[6]

▶ **Theorem 16.** *Muller objectives are all-path-efficient.*

Proof. Consider a graph G, and a Muller objective $\alpha = \langle \mathcal{F}, \chi\rangle$ defined over a set of colors $[k]$. Deciding whether every infinite path in G satisfies α can be reduced to deciding whether there is an infinite path in G that satisfies the dual Muller objective $\tilde{\alpha} = \langle 2^{[k]} \setminus \mathcal{F}, \chi\rangle$. Since the size of $\tilde{\alpha}$ need not be polynomial in $|\mathcal{F}|$, a naive algorithm that checks the existence of a path that satisfies $\tilde{\alpha}$ does not run in polynomial time.

The key point in our algorithm is to complement a given Muller objective $\mathcal{F}$ not to another Muller objective, but rather to view $\mathcal{F}$ as a DNF formula $\varphi_{\mathcal{F}}$ over $[k]$, dualize it to a CNF formula $\overline{\varphi_{\mathcal{F}}}$, and then convert $\overline{\varphi_{\mathcal{F}}}$ to an equivalent DNF formula. Specifically, $\varphi_{\mathcal{F}} = \bigvee_{F\in\mathcal{F}}((\bigwedge_{i\in F} i) \wedge (\bigwedge_{i\in[k]\setminus F} \bar{i}))$, with the semantics that a literal $i \in [k]$ ($\bar{i}$, respectively) requires vertices with the color i to be visited infinitely (finitely, respectively) often [4]. The dual Muller objective then corresponds to the complementing CNF formula; thus it corresponds to the *Emerson-Lei* objective [30] $\overline{\varphi_{\mathcal{F}}} = \bigwedge_{F\in\mathcal{F}}((\bigvee_{i\in F} \bar{i}) \vee (\bigvee_{i\in[k]\setminus F} i))$.

Note that deciding whether a graph G has a path that satisfies a DNF formula can be done in polynomial time (for example by checking whether there exists a path in G that respects the requirements induced by one clause in the formula). On the other hand, deciding whether a graph G has a path that satisfies a CNF formula is hard, which is why we convert

[6] An alternative proof to Theorem 16 can use the known polynomial translation of Muller objectives to *Zielonka DAGs* [30] and the fact Zielonka-DAG automata can be complemented in polynomial time [29]. In the full version, we describe this approach in detail. We find our direct proof useful, as it shows that translating full CNF formulas to equivalent DNF formulas can be done in polynomial time.

$\overline{\varphi_{\mathcal{F}}}$ to an equivalent DNF formula. Converting CNF to DNF is in general exponential. The CNF formula $\overline{\varphi_{\mathcal{F}}}$, however, is *full*: for every variable i, every conjunct contains the literal i or the literal $\bar{i}$. In the full version, we show that full CNF formulas can be converted to DNF in polynomial time. The conversion is based on the equality $\varphi \equiv (\varphi \wedge x_n) \vee (\varphi \wedge \overline{x_n})$, recursively applied to $(\varphi \wedge x_n)$ and $(\varphi \wedge \overline{x_n})$. The blow-up is kept polynomial by minimizing $(\varphi \wedge x_n)$ to include only clauses of φ that contain x_n, after removing x_n from them, and similarly for $(\varphi \wedge \overline{x_n})$. ◀

We can now conclude with the complexity of Muller HPPGs (see proof in the full version).

▶ **Theorem 17.** *Deciding whether Player 1 wins a Muller PP1G is NP-complete, and deciding whether Player 1 wins a Muller PP2G is co-NP-complete.*

8 Weighted Multiple Objectives PPGs and HPPGs

In this section we study PPGs and HPPGs with weighted multiple objectives. We focus on games with underlying Büchi or reachability objectives. In the full version, we prove that the results for underlying co-Büchi or avoid objectives follow. Essentially, this follows from the fact that weighted objectives may be dualized by complementing either the type of the objective or the way we refer to the satisfaction value. Specifically, for an objective type $\gamma \in \{\text{R, A, B, C}\}$, let $\tilde{\gamma}$ be the dual objective, thus $\tilde{R} = A$ and $\tilde{B} = C$. Then, as shown in [40], for every $\gamma \in \{\text{R, A, B, C}\}$, every MaxW-$\gamma$ objective has an equivalent MinW-$\tilde{\gamma}$ objective of polynomial size, and vise versa. Our results are summarized in Table 2 below.

Type	Positionality	P1 wins $\mathcal{G}$	P1 wins Pos-$\mathcal{G}$	P1 wins 1Pos-$\mathcal{G}$	P1 wins 2Pos-$\mathcal{G}$
MaxWB MinWC	2-positional [40]	co-NP-complete [40]	Σ_2^{P}-complete (Theorem 18)	Σ_2^{P}-complete (Theorem 20)	co-NP-complete (Theorem 20)
MinWB MaxWC	1-positional [40]	NP-complete [40]	Σ_2^{P}-complete (Theorem 18)	NP-complete (Theorem 20)	Π_2^{P}-complete (Theorem 20)
MaxWR MinWA	non-positional [40]	PSPACE-complete [40]	Σ_2^{P}-complete (Theorem 19)	Σ_2^{P}-complete (Theorem 21)	PSPACE-complete (Theorem 21)
MinWR MaxWA	non-positional [40]			Σ_2^{P}-complete (Theorem 21)	PSPACE-complete (Theorem 21)

Table 2 Complexity results for PPGs with weighted multiple objectives.

We start with PPGs with underlying Büchi and reachability objectives.

▶ **Theorem 18.** *Deciding whether Player 1 wins a MinWB or MaxWB PPG is Σ_2^{P}-complete. Hardness in Σ_2^{P} applies already for games with uniform weight functions.*

Proof. Both upper bounds follow from Theorem 9. Since the ExistsC objective used in the lower bound in the proof of Theorem 12 can be specified as a MinWB objective with a uniform weight function, a matching lower bound for MinWB PPGs is easy.

A lower bound for MaxWB PPGs is less easy. Consider a 2QBF formula $\Phi = \exists X \forall Y \varphi$ such that φ is in 3DNF. We construct a MaxWB game $\mathcal{G}_\Phi$ over the game graph F_Φ defined in Section 5.2 such that $\Phi = \textbf{true}$ iff Player 1 wins Pos-$\mathcal{G}_\Phi$.

For every literal l in φ, we define a Büchi objective α_l as the set of vertices associated with l. That is, the literal vertex l from the assignment phase of the game, and refute-literal vertices in the checking phase that originate from $\bar{l}$ appearing in some clause. The objective of Player 1 is to satisfy at least $|X| + |Y| + 1$ different Büchi objectives, which can be expressed with a uniform weight function. Formally, $\mathcal{G}_\Phi = \langle F_\Phi, \alpha, |X| + |Y| + 1 \rangle$, where

$\alpha = \{\alpha_l : l \in X \cup \overline{X} \cup Y \cup \overline{Y}\}$, with $\alpha_l = \{l\} \cup \{C_i^j : i \in [k], j \in [3], \text{ and } l_i^j = \bar{l}\}$, for every $l \in X \cup \overline{X} \cup Y \cup \overline{Y}$.

Intuitively (see proof in the full version), the literal vertices that a play visits infinitely often are exactly those that correspond to literals evaluated to **true** in the chosen assignment, and thus the play satisfies $|X| + |Y|$ Büchi objectives during the assignment phase – one for every literal evaluated to **true**. Then, the game satisfies an additional Büchi objective in the checking phase iff Player 2 chooses a refute-literal vertex that corresponds to a literal evaluated to **false**. Since Player 2 is forced to choose such a refute-literal vertex iff there exists a clause all whose literals are evaluated to **true**, Player 2 is forced to satisfy an additional Büchi objective iff the chosen assignment satisfies φ. Therefore, $\Phi = \textbf{true}$ iff Player 1 can force the satisfaction of at least $|X| + |Y| + 1$ Büchi objectives. ◀

▶ **Theorem 19.** *Deciding whether Player 1 wins a MinWR or MaxWR PPG is Σ_2^{P}-complete. Hardness in Σ_2^{P} applies already for uniform weight functions.*

Proof. The upper bounds follow from Theorem 9. The reductions from 2QBF to MinWB and MaxWB PPGs in the proof of Theorem 18 are also valid for MinWR and MaxWR PPGs. Indeed, when the players are restricted to memoryless strategies in G_Φ and F_Φ, every vertex is visited infinitely often iff it is reached. Thus, a Büchi (co-Büchi) objective α is satisfied iff the reachability (avoid, respectively) objective α is satisfied. ◀

Moving to HPPGs, we start with underlying Büchi objectives, where things are easy (see proof in the full version):

▶ **Theorem 20.** *Deciding whether Player 1 wins:*

- *a MinWB PP1G is NP-complete.*
- *a MaxWB PP1G is Σ_2^{P}-complete.*
- *a MaxWB PP2G is co-NP-complete.*
- *a MinWB PP2G is Π_2^{P}-complete.*

In all cases, hardness holds already for games with a uniform weight function.

For HPPGs with underlying reachability objectives, things are more complicated. First, recall that MinWR and MaxWR HPPGs are undetermined, thus the results for PP2Gs cannot be inferred from the results for PP1Gs. In addition, since MinWR and MaxWR objectives are not half-positional, the memory requirements for the unrestricted player adds to the complexity.

▶ **Theorem 21.** *Deciding whether Player 1 wins:*

- *a MinWR or MaxWR PP1G is Σ_2^{P}-complete.*
- *a MinWR or MaxWR PP2G is PSPACE-complete.*

In all cases, hardness holds already for games with a uniform weight function.

Proof. (sketch, see full details in the full version). We start with PP1Gs. For the upper bounds, consider a MinWR or MaxWR game $\mathcal{G} = \langle G, \psi \rangle$. A memoryless strategy f_1 for Player 1 in $\mathcal{G}$ is winning iff the objectives that are satisfied in ψ are reached within polynomially many rounds of the game. Hence, an NP algorithm that uses a co-NP oracle guesses a memoryless strategy f_1 for Player 1, and checks that ψ is satisfied in every guessed path of the appropriate length in G_{f_1}.

For the lower bounds, we argue that the reductions from 2QBF to MinWR and MaxWR PPGs in the proof of Theorem 19 are valid also for MinWR and MaxWR PP1Gs when replacing the game graphs G_Φ and F_Φ by $Reach(G_\Phi)$ and $Reach(F_\Phi)$, respectively. Essentially, this follows from the fact that in these games each vertex is visited at most once.

We continue to the PP2Gs. Proving a PSPACE upper bound, we describe an ATM T that runs in polynomial time and accepts a MinWR or a MaxWR game $\mathcal{G} = \langle G, \psi \rangle$ iff Player 1 wins 2Pos-$\mathcal{G}$. The alternation of T is used in order to simulate the game, and we show that, for MaxWR, we can require the reachability objectives to be satisfied within $|V| \cdot |\alpha|$, and for MinWR, we can require the reachability objectives to not be satisfied while traversing a cycle. Accordingly, we can bound the number of rounds in the simulation, with T maintaining on the tape the set of reachability objectives satisfied so far, and information required to detect when it may terminate and to ensure that Player 2 uses a memoryless strategy.

For the lower bounds, we describe reductions from QBF. That is, given a QBF formula $\Phi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$, we construct a MinWR and a MaxWR game $\mathcal{G}_\Phi = \langle \mathit{Reach}(G_\Phi), \alpha \rangle$ such that $\Phi = \textbf{true}$ iff Player 1 wins 2Pos-$\mathcal{G}_\Phi$. For MinWR, we assume that φ is in DNF and the winning objective α is similar to the ExistsC objective in Theorem 12. For MaxWR, we assume that φ is in CNF, and define an AllR objective in which each clause induces the set of its literals. It is easy to see that the reduction is valid when the strategies of the players are not restricted. We argue that since each vertex in $\mathit{Reach}(G_\Phi)$ is visited in each outcome at most ones, Player 1 wins $\mathcal{G}_\Phi$ iff Player 1 wins 2Pos-$\mathcal{G}_\Phi$, and so we are done. ◀

9 Discussion

We introduced and studied positional-player games, where one or both players are restricted to memoryless strategies. Below we discuss two directions for future research.

The focus on memoryless strategies corresponds to *non-intrusiveness* in applications like program repair [33, 27], supervisory control [17], regret minimization [31], and more. More intrusive approaches involve executing controllers of bounded size in parallel with the system or environment, leading to *bounded-player games*. There, the input to the problem contains, in addition to the game $\mathcal{G}$, also bounds m_1 and m_2 to the sizes of the system and the environment. Player 1 wins (m_1, m_2)-$\mathcal{G}$ iff she has a strategy with memory of size at most m_1 that wins against all strategies of size at most m_2 of Player 2.

Note that the games $(1,1)$-$\mathcal{G}$, $(1,\infty)$-$\mathcal{G}$, and $(\infty,1)$-$\mathcal{G}$ coincide with Pos-$\mathcal{G}$, 1Pos-$\mathcal{G}$, and 2Pos-$\mathcal{G}$, respectively, and so our complexity lower bounds here apply also to the bounded setting. Moreover, if we assume, as in work about bounded synthesis [38], that m_1 and m_2 are given in unary, then many of our upper bounds here extend easily to the bounded setting. Indeed, upper bounds that guess memoryless strategies for a player can now guess memory structures of the given size, and reason about the game obtained by taking the product with the structures. Some of our results, however, are not extended easily. For example, the NP upper bound for Streett PPGs relies on the fact that for every memoryless strategy f_1 for Player 1, we have that Player 2 has a winning strategy in G_{f_1} iff she has a memoryless winning strategy in G_{f_1}. When $m_2 < m_1$, the latter is not helpful for Player 2, and we conjecture that the problem in the bounded setting is more complex.

The second direction concerns *positional-player non-zero-sum games*, namely multi-player games in which the objectives of the players may overlap [13, 52]. There, typical questions concern the stability of the game and equilibria the players may reach [54]. In particular, in *rational synthesis*, we seek an equilibrium in which the objective of the system is satisfied [23, 15]. The study of positional strategies in the non-zero-sum setting is of particular interest, as it also restricts the type of deviations that players may perform, and our ability to incentivize or block such deviations [39].

References

1 B. Abu Radi and O. Kupferman. Minimization and canonization of GFG transition-based automata. *Log. Methods Comput. Sci.*, 18(3), 2022.

2 B. Abu Radi, O. Kupferman, and O. Leshkowitz. A hierarchy of nondeterminism. In *46th Int. Symp. on Mathematical Foundations of Computer Science*, volume 202 of *LIPIcs*, pages 85:1–85:21, 2021.

3 B. Aminof, O. Kupferman, and R. Lampert. Formal analysis of online algorithms. In *9th Int. Symp. on Automated Technology for Verification and Analysis*, volume 6996 of *Lecture Notes in Computer Science*, pages 213–227. Springer, 2011.

4 T. Babiak, F. Blahoudek, A. Duret-Lutz, J. Klein, J. Kretínský, D. Müller, D. Parker, and J. Strejcek. The hanoi omega-automata format. In *Proc. 27th Int. Conf. on Computer Aided Verification*, volume 9206 of *Lecture Notes in Computer Science*, pages 479–486. Springer, 2015.

5 C. Beeri. On the membership problem for functional and multivalued dependencies in relational databases. *ACM Trans. on Database Systems*, 5:241–259, 1980.

6 U. Boker. On the (in)succinctness of muller automata. In *Proc. 26th Annual Conf. of the European Association for Computer Science Logic*, volume 82 of *LIPIcs*, pages 12:1–12:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

7 P. Bouyer, A. Casares, M Randour, and P. Vandenhove. Half-Positional Objectives Recognized by Deterministic Büchi Automata. In *Proc. 33rd Int. Conf. on Concurrency Theory*, volume 243 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

8 P. Bouyer, Fijalkow N, M. Randour, and P. Vandenhove. How to play optimally for regular objectives?, 2023.

9 N. Bulling and V. Goranko. Combining quantitative and qualitative reasoning in concurrent multi-player games. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, page 1926?1928. International Foundation for Autonomous Agents and Multiagent Systems, 2022.

10 A. Casares. On the minimisation of transition-based rabin automata and the chromatic memory requirements of muller conditions. In *Proc. 30th Annual Conf. of the European Association for Computer Science Logic*, volume 216 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

11 A. Casares and P. Ohlmann. Positional ω-regular languages. In *Proc. 39th ACM/IEEE Symp. on Logic in Computer Science*. Association for Computing Machinery, 2024.

12 K. Chatterjee, W. Dvorak, M. Henzinger, and V. Loitzenbauer. Conditionally optimal algorithms for generalized büchi games. In *41st Int. Symp. on Mathematical Foundations of Computer Science*, volume 58 of *LIPIcs*, pages 25:1–25:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.

13 K. Chatterjee, R. Majumdar, and M. Jurdzinski. On Nash equilibria in stochastic games. In *Proc. 13th Annual Conf. of the European Association for Computer Science Logic*, volume 3210 of *Lecture Notes in Computer Science*, pages 26–40. Springer, 2004.

14 T. Colcombet and D. Niwiński. On the positional determinacy of edge-labeled games. *Theoretical Computer Science*, 352(1):190–196, 2006.

15 R. Condurache, E. Filiot, R. Gentilini, and J.-F. Raskin. The complexity of rational synthesis. In *Proc. 43th Int. Colloq. on Automata, Languages, and Programming*, volume 55 of *LIPIcs*, pages 121:1–121:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

16 F. Delgrange, J.P. Katoen, T. Quatmann, and M. Randour. Simple strategies in multi-objective mdps. In *Proc. 26th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, pages 346–364. Springer International Publishing, 2020.

17 J. Dubreil, Ph. Darondeau, and H. Marchand. Supervisory control for opacity. *IEEE Transactions on Automatic Control*, 55(5):1089–1100, 2010.

18 S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. How much memory is needed to win infinite games. In *Proc. 12th ACM/IEEE Symp. on Logic in Computer Science*, pages 99–110, 1997.

19 R. Ehlers. Symbolic bounded synthesis. In *Proc. 22nd Int. Conf. on Computer Aided Verification*, volume 6174 of *Lecture Notes in Computer Science*, pages 365–379. Springer, 2010.

20 E.A. Emerson and C. Jutla. The complexity of tree automata and logics of programs. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 328–337, 1988.

21 E.A. Emerson and C. Jutla. Tree automata, μ-calculus and determinacy. In *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, pages 368–377, 1991.

22 E. Filiot, N. Jin, and J.-F. Raskin. An antichain algorithm for LTL realizability. In *Proc. 21st Int. Conf. on Computer Aided Verification*, volume 5643, pages 263–277, 2009.

23 D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In *Proc. 16th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010.

24 S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111 – 121, 1980.

25 H. Gimbert and W. Zielonka. Games where you can play optimally without any memory. In *Proc. 16th Int. Conf. on Concurrency Theory*, volume 3653 of *Lecture Notes in Computer Science*, pages 428–442, 2005.

26 E. Grädel. Positional determinacy of infinite games. In *Proc. 21st Symp. on Theoretical Aspects of Computer Science*, volume 2996 of *Lecture Notes in Computer Science*, pages 4–18, 2004.

27 D. Harel, G. Katz, A. Marron, and G. Weiss. Non-intrusive repair of reactive programs. In *17th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS 2012, Paris, France, July 18-20, 2012*, pages 3–12. IEEE Computer Society, 2012.

28 D. Harel and A. Pnueli. On the development of reactive systems. In K. Apt, editor, *Logics and Models of Concurrent Systems*, volume F-13 of *NATO Advanced Summer Institutes*, pages 477–498. Springer, 1985.

29 C. Hugenroth. Zielonka dag acceptance and regular languages over infinite words. In *Developments in Language Theory*, pages 143–155. Springer Nature Switzerland, 2023.

30 P. Hunter and A. Dawar. Complexity bounds for regular games. In *30th Int. Symp. on Mathematical Foundations of Computer Science*, volume 3618, pages 495–506. Springer, 2005.

31 P. Hunter, G.A. Pérez, and J.F. Raskin. Reactive synthesis without regret. In *Proc. 26th Int. Conf. on Concurrency Theory*, volume 42 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 114–127. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015.

32 N. Immerman. Number of quantifiers is better than number of tape cells. *Journal of Computer and Systems Science*, 22(3):384–406, 1981.

33 B. Jobstmann, A. Griesmayer, and R. Bloem. Program repair as a game. In *Proc. 17th Int. Conf. on Computer Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 226–238, 2005.

34 M. Jurdzinski. Deciding the winner in parity games is in UP $\cap$ co-UP. *Information Processing Letters*, 68(3):119–124, 1998.

35 N. Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. *Annals of Pure and Applied Logic*, 69(2):243–268, 1994.

36 E. Kopczyński. Half-positional determinacy of infinite games. In *Proc. 33rd Int. Colloq. on Automata, Languages, and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 336–347. Springer, 2006.

37 D. Kuperberg and A. Majumdar. Width of non-deterministic automata. In *Proc. 35th Symp. on Theoretical Aspects of Computer Science*, volume 96 of *LIPIcs*, pages 47:1–47:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

38 O. Kupferman, Y. Lustig, M.Y. Vardi, and M. Yannakakis. Temporal synthesis for bounded systems and environments. In *Proc. 28th Symp. on Theoretical Aspects of Computer Science*, pages 615–626, 2011.

39 O. Kupferman and N. Shenwald. Games with trading of control. In *Proc. 34th Int. Conf. on Concurrency Theory*, volume 279 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

40 O. Kupferman and N. Shenwald. Games with weighted multiple objectives. In *22nd Int. Symp. on Automated Technology for Verification and Analysis*, Lecture Notes in Computer Science. Springer, 2024.

41 D.A. Martin. Borel determinacy. *Annals of Mathematics*, 65:363–371, 1975.

42 S. Nain and M.Y. Vardi. Solving partial-information stochastic parity games. In *Proc. 28th ACM/IEEE Symp. on Logic in Computer Science*, pages 341–348. IEEE Computer Society, 2013.

43 N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

44 P. Ohlmann. Characterizing positionality in games of infinite duration over infinite graphs. In *Proc. 37th ACM/IEEE Symp. on Logic in Computer Science*. Association for Computing Machinery, 2022.

45 P. Ohlmann and M. Skrzypczak. Positionality in Σ_0^2 and a completeness result, 2024. `arXiv:2309.17022`.

46 A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th ACM Symp. on Principles of Programming Languages*, pages 179–190, 1989.

47 D. Raju, R. Ehlers, and U. Topcu. Playing against opponents with limited memory. *CoRR*, abs/2002.07274, 2020.

48 D. Raju, R. Ehlers, and U. Topcu. Adapting to the behavior of environments with bounded memory. In *Proc. 12th International Symposium on Games, Automata, Logics and Formal Verification*, 2021.

49 S. Schewe. Minimising good-for-games automata is NP-complete. In *Proc. 40th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 182 of *LIPIcs*, pages 56:1–56:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

50 S. Schewe and B. Finkbeiner. Bounded synthesis. In *5th Int. Symp. on Automated Technology for Verification and Analysis*, volume 4762 of *Lecture Notes in Computer Science*, pages 474–488. Springer, 2007.

51 W. Thomas. On the synthesis of strategies in infinite games. In *Proc. 12th Symp. on Theoretical Aspects of Computer Science*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995.

52 M. Ummels. The complexity of Nash equilibria in infinite multiplayer games. In *Proc. 11th Int. Conf. on Foundations of Software Science and Computation Structures*, pages 20–34, 2008.

53 M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and Systems Science*, 32(2):182–221, 1986.

54 J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1953.

55 W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.