Half-Positional Objectives Recognized by Deterministic Büchi Automata

Patricia Bouyer
Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, 91190, Gif-sur-Yvette, France

Antonio Casares
LaBRI, Université de Bordeaux, France

Mickael Randour
F.R.S.-FNRS & UMONS – Université de Mons, Belgium

Pierre Vandenhove
F.R.S.-FNRS & UMONS – Université de Mons, Belgium
Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, 91190, Gif-sur-Yvette, France

Abstract

A central question in the theory of two-player games over graphs is to understand which objectives are *half-positional*, that is, which are the objectives for which the protagonist does not need memory to implement winning strategies. Objectives for which *both* players do not need memory have already been characterized (both in finite and infinite graphs); however, less is known about half-positional objectives. In particular, no characterization of half-positionality is known for the central class of *ω-regular objectives*.

In this paper, we characterize objectives recognizable by deterministic Büchi automata (a class of ω-regular objectives) that are half-positional, in both finite and infinite graphs. Our characterization consists of three natural conditions linked to the language-theoretic notion of *right congruence*. Furthermore, this characterization yields a polynomial-time algorithm to decide half-positionality of an objective recognized by a given deterministic Büchi automaton.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases two-player games on graphs, half-positionality, memoryless optimal strategies, Büchi automata, ω-regularity

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.20

Related Version *Full Version*: `https://arxiv.org/abs/2205.01365` [8]

Funding This work has been partially supported by the ANR Project MAVeriQ (ANR-ANR-20-CE25-0012). Mickael Randour is an F.R.S.-FNRS Research Associate and a member of the TRAIL Institute. Pierre Vandenhove is an F.R.S.-FNRS Research Fellow.

Acknowledgements We would like to thank Igor Walukiewicz for suggesting a simplification of the proof of Lemma 17 and Pierre Ohlmann for interesting discussions on the subject.

1 Introduction

Graph games and reactive synthesis. We study *zero-sum turn-based games on graphs* confronting two players (a protagonist and its opponent). They interact by moving a pebble in turns through the edges of a graph for an infinite amount of time. Each vertex belongs to a player, and the player controlling the current vertex decides on the next state of the game. Edges of the graph are labeled with *colors*, and the interaction of the two players therefore produces an infinite sequence of them. The objective of the game is specified by a subset of infinite sequences of colors, and the protagonist wins if the produced sequence belongs to this set. We are interested in finding a *winning strategy* for the protagonist, that is, a function indicating how the protagonist should move in any situation, guaranteeing the achievement of the objective.

This game-theoretic model is particularly fitted to study the *reactive synthesis problem* [7]: a system (the protagonist) wants to satisfy a specification (the objective) while interacting continuously with its environment (the opponent). The goal is to build a controller for the system satisfying the specification, whenever possible. This comes down to finding a winning strategy for the protagonist in the derived game.

Half-positionality. In order to obtain a controller for the system that is simple to implement, we are interested in finding the simplest possible winning strategy. Here, we focus on the amount of information that winning strategies have to remember. The simplest strategies are then arguably *positional* (also called *memoryless*) strategies, which do not remember anything about the past and base their decisions solely on the current state of the game. We intend to understand for which objectives positional strategies suffice for the protagonist to play optimally (i.e., to win whenever it is possible) – we call these objectives *half-positional*. We distinguish half-positionality from *bipositionality* (or *memoryless-determinacy*), which refers to objectives for which positional strategies suffice to play optimally for *both* players.

Many natural objectives have been shown to be bipositional over games on finite and sometimes infinite graphs: e.g., discounted sum [53], mean-payoff [28], parity [29], total payoff [31], energy [9], or average-energy games [11]. Bipositionality can be established using general criteria and characterizations, over games on both finite [31, 32, 3] and infinite [26] graphs. Yet, there exist many objectives and combinations thereof for which one player, but not both, has positional optimal strategies (Rabin conditions [35, 34], mean-payoff parity [22], energy parity [20], some window objectives [21, 14], energy mean-payoff [15]...), and to which these results do not apply.

Various attempts have been made to understand common underlying properties of half-positional objectives and provide sufficient conditions [36, 37, 38, 6], but little more was known until the recent work of Ohlmann [48] (discussed below). These conditions are not general enough to prove half-positionality of some very simple objectives, even in the well-studied class of *ω-regular objectives* [6, Lemma 13]. Furthermore, multiple questions concerning half-positionality remain open. For instance, in [38], Kopczyński conjectured that *prefix-independent* half-positional objectives are closed under finite union (this conjecture was recently refuted for games on finite graphs [39], but is still unsolved for games on infinite graphs). Also, Kopczyński showed that given a deterministic parity automaton recognizing a prefix-independent objective W, we can decide if W is half-positional [37]. However, the time complexity of his algorithm is $\mathcal{O}(n^{\mathcal{O}(n^2)})$, where n is the number of states of the automaton. It is unknown whether this can be done in polynomial time, and no algorithm exists in the non-prefix-independent case.

ω-regular objectives and deterministic Büchi automata. A central class of objectives, whose half-positionality is not yet completely understood, is the class of *ω-regular objectives*. There are multiple equivalent definitions for them: they are the objectives defined, e.g., by ω-regular expressions, by non-deterministic Büchi automata [45], and by deterministic parity automata [46]. These objectives coincide with the class of objectives defined by monadic second-order formulas [17], and they encompass linear-time temporal logic (LTL) specifications [50]. Part of their interest is due to the landmark result that finite-state machines are sufficient to implement optimal strategies in ω-regular games [16, 33], implying the decidability of the monadic second-order theory of natural numbers with the successor relation [17] and the decidability of the synthesis problem under LTL specifications [51].

In this paper, we focus on the subclass of ω-regular objectives recognized by *deterministic Büchi automata* (DBA), that we call *DBA-recognizable*. DBA-recognizable objectives correspond to the ω-regular objectives that can be written as a countable intersection of open objectives (for the Cantor topology, that is, that are G_δ-sets of the Borel hierarchy); or equivalently, that are the limit of a regular language of finite words [42, 49]. Deciding the winner of a game with a DBA-recognizable objective is doable in polynomial time in the size of the arena and the DBA (by solving a Büchi game on the product of the arena and the DBA [7]).

We now discuss two technical tools at the core of our approach: *universal graphs* and *right congruences*.

Universal graphs. One recent breakthrough in the study of half-positionality is the introduction of *well-monotonic universal graphs*, combinatorial structures that can be used to provide a witness of winning strategies in games with a half-positional objective. Recently, Ohlmann [48] has shown that the existence of a *well-monotonic universal graph* for an objective W exactly characterizes half-positionality (under minor technical assumptions on W). Moreover, under these assumptions, a wide class of algorithms, called *value iteration algorithms*, can be applied to solve any game with a half-positional objective [24, 48].

Although it brings insight on the structure of half-positional objectives, showing half-positionality through the use of universal graphs is not always straightforward, and has not yet been applied in a systematic way to ω-regular objectives.

Right congruence. Given an objective W, the *right congruence* $\sim_W$ *of* W is an equivalence relation on finite words: two finite words w_1 and w_2 are equivalent for $\sim_W$ if for all infinite continuations w, $w_1w \in W$ if and only if $w_2w \in W$. There is a natural automaton classifying the equivalence classes of the right congruence, which we refer to as the *prefix-classifier* [54, 44].

In the case of languages of *finite* words, a straightforward adaptation of the right congruence recovers the known Myhill-Nerode congruence. This equivalence relation characterizes the regular languages (a language is regular if and only if its congruence has finitely many equivalence classes), and the prefix-classifier is exactly the smallest deterministic finite automaton recognizing a language – this is the celebrated Myhill-Nerode theorem [47].

Objectives are languages of *infinite* words, for which the situation is not so clear-cut. In particular, an ω-regular objective may not always be recognized by its prefix-classifier along with a natural acceptance condition (Büchi, coBüchi, parity, Muller…) [44, 4].

Contributions. Our main contribution is a *characterization* of half-positionality for DBA-recognizable objectives through a conjunction of three easy-to-check conditions (Theorem 10).

(1) The equivalence classes of the right congruence are *totally* ordered w.r.t. inclusion of their winning continuations.

(2) Whenever the set of winning continuations of a finite word w_1 is a proper subset of the set of winning continuations of a concatenation w_1w_2, the word $w_1(w_2)^\omega$ produced by repeating infinitely often w_2 is winning.

(3) The objective has to be recognizable by a DBA using the structure of its prefix-classifier.

A few examples of simple DBA-recognizable objectives that were not encompassed by previous half-positionality criteria [36, 6] are, e.g., reaching a color twice [6, Lemma 13] and weak parity [55]. We also refer to Example 7, which is half-positional but not bipositional, and whose half-positionality is straightforward using our characterization.

Various corollaries with practical and theoretical interest follow from our characterization.

- We obtain a painless path to show (by checking each of the three conditions) that given a deterministic Büchi automaton, the half-positionality of the objective it recognizes is decidable in time $\mathcal{O}(k^2 \cdot n^4)$, where k is the number of colors and n is the number of states of the DBA (Section 3.3).
- Prefix-independent DBA-recognizable half-positional objectives are exactly the very simple *Büchi conditions*, which consist of all the infinite words seeing infinitely many times some subset of the colors (Proposition 11). In particular, Kopczyński's conjecture trivializes for DBA-recognizable objectives (the union of Büchi conditions is a Büchi condition).
- We obtain a *finite-to-infinite* and *one-to-two-player* lift result (Proposition 14): in order to check that a DBA-recognizable objective is half-positional over arbitrary – possibly two-player and infinite – graphs, it suffices to check the existence of positional optimal strategies over *finite* graphs where all the vertices are controlled by the protagonist.

Other related works. We have discussed the relevant literature on half-positionality [36, 37, 6, 48] and bipositionality [31, 32, 26, 3]. A more general quest is to understand *memory requirements* when positional strategies are not powerful enough: e.g., [43, 10, 12, 13].

Memory requirements have been precisely characterized for some classes of ω-regular objectives (not encompassing the class of DBA-recognizable objectives), such as Muller conditions [27, 57, 18, 19] and safety specifications, i.e., objectives that are closed for the Cantor topology [25]. The latter also uses the order of the equivalence classes of the right congruence as part of its characterization.

Recently, a link between the prefix-classifier, the memory requirements, and the recognizability of ω-regular objectives was established [13]. However, this result does not provide optimal bounds on the strategy complexity, and is thereby insufficient to study half-positionality.

Structure of the paper. Notations and definitions are introduced in Section 2. Our main contributions are presented in Section 3: we introduce and discuss the three conditions used in our results, then we state our main characterization (Theorem 10) and some corollaries, and we end with an explanation on how to use the characterization to decide half-positionality of DBA-recognizable objectives in polynomial time. Due to space constraints, we only provide high-level details about proofs in this version of the article: a proof sketch for Theorem 10 is provided in Section 4. Complete proofs, as well as additional details and examples, can be found in the extended version of the article [8].

2 Preliminaries

In the whole article, letter C refers to a (finite or infinite) non-empty set of *colors*. Given a set A, we write respectively A^*, A^+, and A^ω for the set of finite, non-empty finite, and infinite sequences of elements of A. We denote by ε the empty word.

2.1 Games and positionality

Graphs. An *(edge-colored) graph* $\mathcal{G} = (V, E)$ is given by a non-empty set of *vertices* V (of any cardinality) and a set of *edges* $E \subseteq V \times C \times V$. We write $v \xrightarrow{c} v'$ if $(v, c, v') \in E$. We assume graphs to be *non-blocking*: for all $v \in V$, there exists $(v', c, v'') \in E$ such that $v = v'$. We allow graphs with infinite branching. For $v \in V$, an *infinite path of* $\mathcal{G}$ *from* v is an infinite sequence of edges $\pi = (v_0, c_1, v'_1)(v_1, c_2, v'_2)\ldots \in E^\omega$ such that $v_0 = v$ and for all $i \geq 1$, $v'_i = v_i$. A *finite path of* $\mathcal{G}$ *from* v is a finite prefix in E^* of an infinite path of $\mathcal{G}$ from

v. For convenience, we assume that there is a distinct *empty path* λ_v for every $v \in V$. If $\gamma = (v_0, c_1, v_1) \dots (v_{n-1}, c_n, v_n)$ is a non-empty finite path of $\mathcal{G}$, we define $\mathsf{last}(\gamma) = v_n$. For an empty path λ_v, we define $\mathsf{last}(\lambda_v) = v$. An infinite (resp. finite) path $(v_0, c_1, v_1)(v_1, c_2, v_2) \dots$ is sometimes represented as $v_0 \xrightarrow{c_1} v_1 \xrightarrow{c_2} \dots$. A graph $\mathcal{G} = (V, E)$ is *finite* if both V and E are finite. A graph is *strongly connected* if for every pair of vertices $(v, v') \in V \times V$ there is a path from v to v'. A *strongly connected component* of $\mathcal{G}$ is a maximal strongly connected subgraph.

Arenas and strategies. We consider two players $\mathcal{P}_1$ and $\mathcal{P}_2$. An *arena* is a tuple $\mathcal{A} = (V, V_1, V_2, E)$ such that (V, E) is a graph and V is the disjoint union of V_1 and V_2. Intuitively, vertices in V_1 are controlled by $\mathcal{P}_1$ and vertices in V_2 are controlled by $\mathcal{P}_2$. An arena $\mathcal{A} = (V, V_1, V_2, E)$ is a *one-player arena of* $\mathcal{P}_1$ (resp. *of* $\mathcal{P}_2$) if $V_2 = \emptyset$ (resp. $V_1 = \emptyset$). Finite paths of (V, E) are called *histories of* $\mathcal{A}$. For $i \in \{1, 2\}$, we denote by $\mathsf{Hists}_i(\mathcal{A})$ the set of histories γ of $\mathcal{A}$ such that $\mathsf{last}(\gamma) \in V_i$.

Let $i \in \{1, 2\}$. A *strategy of* $\mathcal{P}_i$ *on* $\mathcal{A}$ is a function $\sigma_i \colon \mathsf{Hists}_i(\mathcal{A}) \to E$ such that for all $\gamma \in \mathsf{Hists}_i(\mathcal{A})$, the first component of $\sigma_i(\gamma)$ coincides with $\mathsf{last}(\gamma)$. Given a strategy σ_i of $\mathcal{P}_i$, we say that an infinite path $\pi = e_1 e_2 \dots$ is *consistent with* σ_i if for all finite prefixes $\gamma = e_1 \dots e_i$ of π such that $\mathsf{last}(\gamma) \in V_i$, $\sigma_i(\gamma) = e_{i+1}$. A strategy σ_i is *positional* (also called *memoryless* in the literature) if its outputs only depend on the current vertex and not on the whole history, i.e., if there exists a function $f \colon V_i \to E$ such that for $\gamma \in \mathsf{Hists}_i(\mathcal{A})$, $\sigma_i(\gamma) = f(\mathsf{last}(\gamma))$.

Objectives. An *objective* is a set $W \subseteq C^\omega$ (subsets of C^ω are sometimes also called *languages of infinite words*, *ω-languages*, or *winning conditions* in the literature). When an objective W is clear in the context, we say that an infinite word $w \in C^\omega$ is *winning* if $w \in W$, and *losing* if $w \notin W$. We write $\overline{W}$ for the complement $C^\omega \setminus W$ of an objective W. An objective W is *prefix-independent* if for all $w \in C^*$ and $w' \in C^\omega$, $w' \in W$ if and only if $ww' \in W$. An objective that we will often consider is the *Büchi condition*: given a subset $F \subseteq C$, we denote by $\mathsf{Büchi}(F)$ the set of infinite words seeing infinitely many times a color in F. Such an objective is prefix-independent. A *game* is a tuple $(\mathcal{A}, W)$ of an arena $\mathcal{A}$ and an objective W.

Optimality and half-positionality. Let $\mathcal{A} = (V, V_1, V_2, E)$ be an arena, $(\mathcal{A}, W)$ be a game, and $v \in V$. We say that *a strategy* σ_1 *of* $\mathcal{P}_1$ *is winning from* v if for all infinite paths $v_0 \xrightarrow{c_1} v_1 \xrightarrow{c_2} \dots$ from v consistent with σ_1, $c_1 c_2 \dots \in W$.

A strategy of $\mathcal{P}_1$ is *optimal for* $\mathcal{P}_1$ *in* $(\mathcal{A}, W)$ if it is winning from all the vertices from which $\mathcal{P}_1$ has a winning strategy. We often write *optimal for* $\mathcal{P}_1$ *in* $\mathcal{A}$ if the objective W is clear from the context. We stress that this notion of optimality requires a *single* strategy to be winning from *all* the winning vertices (a property sometimes called *uniformity*).

An objective W is *half-positional* if for all arenas $\mathcal{A}$, there exists a positional strategy of $\mathcal{P}_1$ on $\mathcal{A}$ that is optimal for $\mathcal{P}_1$ in $\mathcal{A}$. We sometimes only consider half-positionality on a restricted set of arenas (typically, finite and/or one-player arenas). For a class of arenas $\mathcal{X}$, an objective W is *half-positional over* $\mathcal{X}$ if for all arenas $\mathcal{A} \in \mathcal{X}$, there exists a positional strategy of $\mathcal{P}_1$ on $\mathcal{A}$ that is optimal for $\mathcal{P}_1$ in $\mathcal{A}$.

2.2 Büchi automata

Automaton structures and Büchi automata. A *non-deterministic automaton structure (on C)* is a tuple $\mathcal{S} = (Q, C, Q_{\mathsf{init}}, \Delta)$ such that Q is a finite set of *states*, $Q_{\mathsf{init}} \subseteq Q$ is a non-empty set of *initial states* and $\Delta \subseteq Q \times C \times Q$ is a set of *transitions*. We assume that all states of automaton structures are reachable from an initial state in Q_{init} by taking transitions in Δ.

A *(transition-based) non-deterministic Büchi automaton* (NBA) is an automaton structure $\mathcal{S}$ together with a set of transitions $\alpha \subseteq \Delta$. The transitions in α are called *Büchi transitions.*

Given an NBA $\mathcal{B} = (Q, C, Q_{\mathsf{init}}, \Delta, \alpha)$, a *(finite or infinite) run of $\mathcal{B}$ on a (finite or infinite) word* $w = c_1c_2\ldots \in C^* \cup C^\omega$ is a sequence $(q_0, c_1, q_1)(q_1, c_2, q_2)\ldots \in \Delta^* \cup \Delta^\omega$ such that $q_0 \in Q_{\mathsf{init}}$. An infinite run $(q_0, c_1, q_1)(q_1, c_2, q_2)\ldots \in \Delta^\omega$ of $\mathcal{B}$ is *accepting* if for infinitely many $i \geq 0$, $(q_i, c_{i+1}, q_{i+1}) \in \alpha$. A word $w \in C^\omega$ is *accepted* by $\mathcal{B}$ if there exists an accepting run of $\mathcal{B}$ on w – if not, it is *rejected.* We denote the set of infinite words accepted by $\mathcal{B}$ by $\mathcal{L}(\mathcal{B})$, and we then say that $\mathcal{L}(\mathcal{B})$ is the objective *recognized by* $\mathcal{B}$. Here, we take the definition of an *ω-regular objective* as an objective that can be recognized by an NBA. Given an automaton structure $\mathcal{S} = (Q, C, Q_{\mathsf{init}}, \Delta)$, we say that an NBA $\mathcal{B}$ is *built on top of* $\mathcal{S}$ if there exists $\alpha \subseteq \Delta$ such that $\mathcal{B} = (Q, C, Q_{\mathsf{init}}, \Delta, \alpha)$.

Deterministic automata. An automaton structure $\mathcal{S} = (Q, C, Q_{\mathsf{init}}, \Delta)$ is *deterministic* if $|Q_{\mathsf{init}}| = 1$ and, for each $q \in Q$ and $c \in C$, there is exactly one $q' \in Q$ such that $(q, c, q') \in \Delta$ (we remark that in this paper deterministic automata are *complete*). A *deterministic Büchi automaton* (DBA) is an NBA whose underlying automaton structure is deterministic. For a DBA $\mathcal{B} = (Q, C, \{q_{\mathsf{init}}\}, \Delta, \alpha)$, we denote by q_{init} the unique initial state (and we will drop the braces around q_{init} in the tuple), and by $\delta\colon Q \times C \to Q$ the *update function* that associates to $(q, c) \in Q \times C$ the only $q' \in Q$ such that $(q, c, q') \in \Delta$. We denote by δ^* the natural extension of δ to finite words. As transitions are uniquely determined by their first two components, we also assume for brevity that $\alpha \subseteq Q \times C$.

For a DBA $\mathcal{B}$, a state $q \in Q$ and a word $w = c_1c_2\ldots \in C^* \cup C^\omega$, we denote by $\mathcal{B}(q, w) = (q, c_1, q_1)(q_1, c_2, q_2)\ldots \in \Delta^* \cup \Delta^\omega$ the only run on w starting from q.

An objective W is *DBA-recognizable* if there exists a DBA $\mathcal{B}$ such that $W = \mathcal{L}(\mathcal{B})$. For $F \subseteq C$, notice that $\mathsf{Büchi}(F)$ is DBA-recognizable: it is recognized by the DBA $(\{q_{\mathsf{init}}\}, C, q_{\mathsf{init}}, \Delta, \alpha)$ with a *single* state such that $(q_{\mathsf{init}}, c) \in \alpha$ if and only if $c \in F$.

▶ Remark 1. The fact that a single state suffices for recognizing $\mathsf{Büchi}(F)$ relies on the assumption that our DBA are *transition-based* and not *state-based* (α is a set of transitions, not of states). Indeed, apart from the trivial cases $F = \emptyset$ and $F = C$, a state-based DBA recognizing $\mathsf{Büchi}(F)$ requires two states. The third condition of our upcoming characterization (Theorem 10) would therefore not apply to this simple example if we only considered state-based DBA. ⌟

▶ Remark 2. DBA recognize a proper subset of the ω-regular objectives [56]. ⌟

Let $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \Delta, \alpha)$ be a DBA. We say that a finite run $\varrho \in \Delta^*$ of $\mathcal{B}$ is *α-free* if it does not contain any transition from α. For $q \in Q$, we define

$$\alpha\text{-}\mathsf{Free}_{\mathcal{B}}(q) = \{w \in C^* \mid \mathcal{B}(q, w) \text{ is } \alpha\text{-free}\},$$
$$\alpha\text{-}\mathsf{FreeCycles}_{\mathcal{B}}(q) = \{w \in C^* \mid w \in \alpha\text{-}\mathsf{Free}_{\mathcal{B}}(q) \text{ and } \delta^*(q, w) = q\}.$$

We call the words in the first set the *α-free words from q*, and the words in the second set the *α-free cycles from q*.

Right congruence. Let $W \subseteq C^\omega$ be an objective. For a finite word $w \in C^*$, we write $w^{-1}W = \{w' \in C^\omega \mid ww' \in W\}$ for the set of *winning continuations of* w. We define the *right congruence* $\sim_W \subseteq C^* \times C^*$ *of* W as $w_1 \sim_W w_2$ if $w_1^{-1}W = w_2^{-1}W$. Relation $\sim_W$ is an equivalence relation. When W is clear from the context, we write $\sim$ for $\sim_W$. For $w \in C^*$, we denote by $[w] \subseteq C^*$ its equivalence class for $\sim$.

When $\sim$ has finitely many equivalence classes, we can associate a natural deterministic automaton structure $\mathcal{S}_\sim = (Q_\sim, C, \widetilde{q}_{\mathsf{init}}, \Delta_\sim)$ to $\sim$ such that $Q_\sim$ is the set of equivalence classes of $\sim$, $\widetilde{q}_{\mathsf{init}} = [\varepsilon]$, and $\delta_\sim([w], c) = [wc]$ [54, 44]. The transition function $\delta_\sim$ is well-defined since if $w_1 \sim w_2$, then for all $c \in C$, $w_1c \sim w_2c$. We call the automaton structure $\mathcal{S}_\sim$ the *prefix-classifier of* W.

▶ Remark 3. Equivalence relation $\sim_W$ has only one equivalence class if and only if W is prefix-independent. In particular, an objective has a prefix-classifier with a single state if and only if it is prefix-independent. ⌟

We define the *prefix preorder* $\preceq_W$ *of* W: for $w_1, w_2 \in C^*$, we write $w_1 \preceq_W w_2$ if $w_1^{-1}W \subseteq w_2^{-1}W$ (meaning that any continuation that is winning after w_1 is also winning after w_2). Intuitively, $w_1 \preceq_W w_2$ means that a game starting with w_2 is always preferable to a game starting with w_1 for $\mathcal{P}_1$, as there are more ways to win after w_2. When W is clear from the context, we write $\preceq$ for $\preceq_W$. Relation $\preceq \subseteq C^* \times C^*$ is a preorder. Notice that $\sim$ is equal to $\preceq \cap \succeq$. We also define the strict preorder $\prec = \preceq \setminus \sim$.

Given a DBA $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \Delta, \alpha)$ recognizing the objective W, observe that for $w, w' \in C^*$ such that $\delta^*(q_{\mathsf{init}}, w) = \delta^*(q_{\mathsf{init}}, w')$, we have $w \sim w'$. In this case, equivalence relation $\sim$ has at most $|Q|$ equivalence classes. For $q \in Q$, we write abusively $q^{-1}W$ for the objective recognized by the DBA $(Q, C, q, \Delta, \alpha)$. Objective $q^{-1}W$ equals $w^{-1}W$ for any word $w \in C^*$ such that $\delta^*(q_{\mathsf{init}}, w) = q$. We extend the equivalence relation $\sim$ and preorder $\preceq$ to elements of Q (we sometimes write $\sim_\mathcal{B}$ and $\preceq_\mathcal{B}$ to avoid any ambiguity).

3 Half-positionality characterization for DBA-recognizable objectives

In this section, we present our main contribution in Theorem 10, by giving three conditions that exactly characterize half-positional DBA-recognizable objectives. These conditions are presented in Section 3.1. Theorem 10 and several consequences of it are stated in Section 3.2. In Section 3.3, we use this characterization to show that we can decide the half-positionality of a DBA in polynomial time. Missing proofs for this section are in [8, Section 3], except for the proof of Theorem 10, which is in [8, Sections 4 & 5].

3.1 Three conditions for half-positionality

We define the three conditions on objectives at the core of our characterization.

▶ **Condition 1** (Total prefix preorder)**.** We say that an objective $W \subseteq C^\omega$ has a *total prefix preorder* if for all $w_1, w_2 \in C^*$, $w_1 \preceq_W w_2$ or $w_2 \preceq_W w_1$.

An objective W recognized by a DBA $\mathcal{B}$ has a total prefix preorder if and only if the (reachable) states of $\mathcal{B}$ are totally ordered for $\preceq_\mathcal{B}$.

▶ **Example 4** (Not total prefix preorder)**.** Let $C = \{a, b\}$. We consider the objective W recognized by the DBA $\mathcal{B}$ depicted in Figure 1 (left). It consists of the infinite words starting with aa or bb. This objective does not have a total prefix preorder: words a and b are incomparable for $\preceq_W$. Indeed, a^ω is winning after a but not after b, and b^ω is winning after b but not after a. In terms of automaton states, we have that q_a and q_b are incomparable for $\preceq_\mathcal{B}$. This objective is not half-positional, as witnessed by the arena on the right of Figure 1. In this arena, $\mathcal{P}_1$ is able to win when the game starts in v_1 by playing a in v_3, and when the game starts in v_2 by playing b. However, no positional strategy wins from both v_1 and v_2. ⌟

▶ Remark 5. The prefix preorder of an objective W is total if and only if the prefix preorder of its complement $\overline{W}$ is total. ⌟

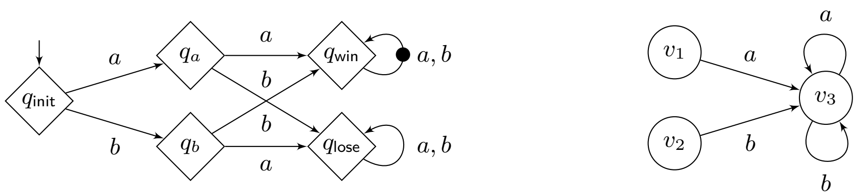

Figure 1 DBA $\mathcal{B}$ recognizing objective $W = (aa + bb)C^\omega$ (left), and an arena in which positional strategies do not suffice for $\mathcal{P}_1$ to play optimally for this objective (right). Transitions labeled with a • symbol are the Büchi transitions. In figures, diamonds represent automaton states and circles represent arena vertices controlled by $\mathcal{P}_1$.

▶ Remark 6. Having a total prefix preorder is equivalent to the *strong monotony* notion [6] in general, and equivalent to *monotony* [32] for ω-regular objectives. We discuss in more depth the relation between the conditions appearing in the characterization and other properties from the literature studying half-positionality in [8, Appendix A]. ⌟

▶ **Condition 2** (Progress-consistency)**.** We say that an objective W is *progress-consistent* if for all $w_1 \in C^*$ and $w_2 \in C^+$ such that $w_1 \prec w_1 w_2$, we have $w_1(w_2)^\omega \in W$.

Intuitively, this means that whenever a word w_2 can be used to make progress after seeing a word w_1 (in the sense of getting to a position in which more continuations are winning), then repeating this word has to be winning.

▶ **Example 7** (Progress-consistent objective)**.** We consider the DBA in Figure 2. This DBA recognizes the objective $W = \mathsf{Büchi}(\{a\}) \cup C^* aa C^\omega$: W contains the words seeing a infinitely often, or that see a twice in a row at some point. The equivalence classes for $\sim_W$ are $q_{\mathsf{init}}^{-1}W = W$, $q_a^{-1}W = aC^\omega \cup W$ and $q_{aa}^{-1}W = C^\omega$. This objective is progress-consistent: any word reaching q_{aa} is straightforwardly accepted when repeated infinitely often, and any word w such that $\delta^*(q_{\mathsf{init}}, w) = q_a$ necessarily contains at least one a, and thus is accepted when repeated infinitely often. Objective W is half-positional, which will be readily shown with our upcoming characterization (Theorem 10).

Here, notice that the complement $\overline{W}$ of W is not progress-consistent. Indeed, $a \prec_{\overline{W}} a(bab)$, but $a(bab)^\omega \notin \overline{W}$. Unlike having a total prefix preorder, progress-consistency can hold for an objective but not its complement.

Note that half-positionality of W cannot be shown using existing half-positionality criteria [36, 6] (it is neither prefix-independent nor *concave*) nor bipositionality criteria, as it is simply not bipositional. ⌟

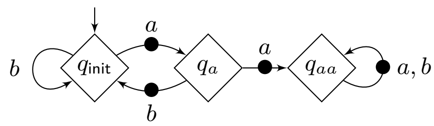

Figure 2 A DBA recognizing the set of words seeing a infinitely many times, or aa at some point.

▶ **Condition 3** (Recognizability by the prefix-classifier)**.** Being recognized by a Büchi automaton built on top of the prefix-classifier is our third condition. In other words, for a DBA-recognizable objective $W \subseteq C^\omega$ and its prefix-classifier $\mathcal{S}_\sim = (Q_\sim, C, \widetilde{q}_{\mathsf{init}}, \Delta_\sim)$, this condition requires that there exists $\alpha_\sim \subseteq Q_\sim \times C$ such that W is recognized by DBA $(Q_\sim, C, \widetilde{q}_{\mathsf{init}}, \Delta_\sim, \alpha_\sim)$.

We show an example of a DBA-recognizable objective that satisfies the first two conditions (having a total prefix preorder and progress-consistency), but not this third condition, and which is not half-positional.

▶ **Example 8** (Not recognizable by the prefix-classifier)**.** Let $C = \{a, b\}$. We consider the objective $W = \mathsf{Büchi}(\{a\}) \cap \mathsf{Büchi}(\{b\})$ recognized by the DBA in Figure 3. This objective is prefix-independent: as such (Remark 3), there is only one equivalence class for $\sim$. This implies that the prefix preorder is total, and that W is progress-consistent (the premise of the progress-consistency property can never be true). This objective is not half-positional, as witnessed by the arena in Figure 3 (right): $\mathcal{P}_1$ has a winning strategy from v, but it needs to take infinitely often both a and b.

Any DBA recognizing this objective has at least two states, but all their (reachable) states are equivalent for $\sim$ – no matter the state we choose as an initial state, the recognized objective is the same (by prefix-independence). As it is prefix-independent, its prefix-classifier $\mathcal{S}_\sim$ has only one state. ⌟

Figure 3 DBA recognizing the objective $\mathsf{Büchi}(\{a\}) \cap \mathsf{Büchi}(\{b\})$ (left), and an arena in which positional strategies do not suffice for $\mathcal{P}_1$ to play optimally for this objective (right).

As will be shown formally, being recognized by a DBA built on top of the prefix-classifier is necessary for half-positionality of *DBA-recognizable* objectives over finite one-player arenas. The first two conditions are actually necessary for half-positionality of general objectives, but this third condition is not, even for objectives recognized by other standard classes of ω-automata.

▶ **Example 9.** We consider the complement $\overline{W}$ of the objective $W = \mathsf{Büchi}(\{a\}) \cap \mathsf{Büchi}(\{b\})$ of Example 8, which consists of the words ending with a^ω or b^ω. Objective $\overline{W}$ is not DBA-recognizable (a close proof can be found in [5, Theorem 4.50]). Still, it is recognizable by a *deterministic coBüchi automaton* similar to the automaton in Figure 3, but which accepts infinite words that visit transitions labeled by • only finitely often. This objective is half-positional, which can be shown using [27, Theorem 6]. However, its prefix-classifier has just one state, and there is no way to recognize $\overline{W}$ by building a coBüchi (or even parity) automaton on top of it. ⌟

3.2 Characterization and corollaries

We have now defined the three conditions required for our characterization.

▶ **Theorem 10.** *Let $W \subseteq C^\omega$ be a DBA-recognizable objective. Objective W is half-positional (over all arenas) if and only if*

- *its prefix preorder $\preceq$ is total,*
- *it is progress-consistent, and*
- *it can be recognized by a Büchi automaton built on top of its prefix-classifier $\mathcal{S}_\sim$.*

High-level details about the proof of this theorem are provided in Section 4. The complete proof of the necessity of the three conditions can be found in [8, Section 4]; the proof of the sufficiency of the conjunction of the three conditions can be found in [8, Section 5].

This characterization is valuable to prove (and disprove) half-positionality of DBA-recognizable objectives. Examples 4 and 8 are not half-positional, and they falsify respectively the first and the third condition from the statement. On the other hand, Example 7 is half-positional. We have already discussed its progress-consistency, but it is also straightforward to verify that its prefix preorder is total and that it is recognizable by its prefix-classifier: the right congruence has three totally ordered equivalence classes corresponding to the states of the automaton of Figure 2.

We state two notable consequences of Theorem 10 and of its proof technique. The first one is the specialization of Theorem 10 to objectives that are prefix-independent, a frequent assumption in the literature [36, 26, 30, 24] – under this assumption, half-positionality of DBA-recognizable objectives is very easy to understand and characterize.

▶ **Proposition 11.** *Let $W \subseteq C^\omega$ be a prefix-independent, DBA-recognizable objective. Objective W is half-positional if and only if there exists $F \subseteq C$ such that $W = \mathsf{Büchi}(F)$.*

▶ Remark 12. A corollary of this result is that when W is prefix-independent, DBA-recognizable and half-positional, we also have that $\overline{W}$ is half-positional. Indeed, the complement of objective $W = \mathsf{Büchi}(F)$ is a so-called *coBüchi objective*, which is also known to be half-positional (it is a special case of a parity objective [29]). This statement does not hold in general when W is not prefix-independent, as was shown in Example 7. Moreover, the reciprocal of the statement also does not hold, as was shown in Example 9. ⌟

▶ Remark 13. A second corollary is that prefix-independent DBA-recognizable half-positional objectives are closed under finite union (since a finite union of Büchi conditions is a Büchi condition). This settles Kopczyński's conjecture for DBA-recognizable objectives. ⌟

A second consequence of Theorem 10 and its proof technique shows that half-positionality of DBA-recognizable objectives can be reduced to half-positionality over the restricted class of *finite, one-player arenas*. Results reducing strategy complexity in two-player arenas to the easier question of strategy complexity in one-player arenas are sometimes called *one-to-two-player lifts* and appear in multiple places in the literature [32, 10, 40, 13].

▶ **Proposition 14** (One-to-two-player and finite-to-infinite lift)**.** *Let $W \subseteq C^\omega$ be a DBA-recognizable objective. If objective W is half-positional over finite one-player arenas, then it is half-positional over all arenas (of any cardinality).*

One-to-two-player lifts from the literature all require an assumption on the strategy complexity of *both* players, and are either stated solely over finite arenas, or solely over infinite arenas. Proposition 14, albeit set in the more restricted context of DBA-recognizable objectives, therefore displays stronger properties than the known one-to-two-player lifts.

3.3 Deciding half-positionality in polynomial time

In this section, we assume that C is finite. We show that the problem of deciding, given a DBA $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \Delta, \alpha)$ as an input, whether $\mathcal{L}(\mathcal{B})$ is half-positional can be solved in polynomial time, and more precisely in time $\mathcal{O}(|C|^2 \cdot |Q|^4)$.

We investigate how to verify each property used in the characterization of Theorem 10. Let $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \Delta, \alpha)$ be a DBA (we assume w.l.o.g. that all states in Q are reachable from q_{init}) and $W = \mathcal{L}(\mathcal{B})$ be the objective it recognizes. Our algorithm first verifies that the prefix preorder is total and recognizability by $\mathcal{S}_\sim$, and then, under these first two assumptions, progress-consistency. For each condition, we sketch an algorithm to decide it, and we discuss the time complexity of this algorithm.

Total prefix preorder. To check that W has a total prefix preorder, it suffices to check that the states of $\mathcal{B}$ are totally preordered by $\preceq_{\mathcal{B}}$. We start by computing, for each pair of states $q, q' \in Q$, whether $q \preceq_{\mathcal{B}} q'$, $q' \preceq_{\mathcal{B}} q$, or none of these. This can be rephrased as an *inclusion problem* for two DBA-recognizable objectives: if $\mathcal{B}_q = (Q, C, q, \Delta, \alpha)$ and $\mathcal{B}_{q'} = (Q, C, q', \Delta, \alpha)$, we have that $q \preceq_{\mathcal{B}} q'$ if and only if $\mathcal{L}(\mathcal{B}_q) \subseteq \mathcal{L}(\mathcal{B}_{q'})$. Such a problem can be solved in time $\mathcal{O}(|C|^2 \cdot |Q|^2)$ [23]. We can therefore know for all $|Q|^2$ pairs $q, q' \in Q$ whether $q \preceq_{\mathcal{B}} q'$, $q' \preceq_{\mathcal{B}} q$, $q' \sim_{\mathcal{B}} q$ (as $\sim_{\mathcal{B}} = \preceq_{\mathcal{B}} \cap \succeq_{\mathcal{B}}$), or none of these in time $\mathcal{O}(|Q|^2 \cdot (|C|^2 \cdot |Q|^2)) = \mathcal{O}(|C|^2 \cdot |Q|^4)$. In particular, the prefix preorder is total if and only if for all $q, q' \in Q$, we have $q \preceq_{\mathcal{B}} q'$ or $q' \preceq_{\mathcal{B}} q$.

Recognizability by the prefix-classifier. After all the relations $\preceq_{\mathcal{B}}$ and $\sim_{\mathcal{B}}$ between pairs of states are computed in the previous step, we can compute the states and transitions of the prefix-classifier $\mathcal{S}_\sim = (Q_\sim, C, \widetilde{q}_{\mathsf{init}}, \Delta_\sim)$ by merging all the equivalence classes for $\sim_{\mathcal{B}}$. We assume for simplicity that $Q_\sim = Q/_{\sim_{\mathcal{B}}}$.

We now wonder whether it is possible to recognize W by carefully selecting a set $\alpha_\sim$ of Büchi transitions in $\mathcal{S}_\sim$. After a simple transformation of $\mathcal{B}$ (called *saturation* [8, Section 2]), it actually suffices to try with the specific, easy-to-compute set of transitions of the prefix-classifier such that all corresponding transitions in the original DBA were Büchi:

$$\alpha_\sim = \{([q], c) \in Q_\sim \times C \mid \forall q' \in [q], (q', c) \in \alpha\}.$$

We then simply check whether $W = \mathcal{L}((Q_\sim, C, \widetilde{q}_{\mathsf{init}}, \Delta_\sim, \alpha_\sim))$, an equivalence query which, as discussed above, can be performed in time $\mathcal{O}(|C|^2 \cdot |Q|^2)$.

Progress-consistency. We assume that we have already checked that W is recognizable by a Büchi automaton built on top of $\mathcal{S}_\sim$, and that we know the (total) ordering of the states. We show that checking progress-consistency, under these two hypotheses, can be done in polynomial time. We state a lemma reducing the search for words witnessing that W is not progress-consistent to a known problem on regular languages.

▶ **Lemma 15.** *We assume that $\mathcal{B}$ is built on top of the prefix-classifier $\mathcal{S}_\sim$ and that the prefix preorder of W is total. Then, W is progress-consistent if and only if for all $q, q' \in Q$ with $q \prec_{\mathcal{B}} q'$, $\{w \in C^+ \mid \delta^*(q, w) = q'\} \cap \alpha\text{-}\mathsf{FreeCycles}_{\mathcal{B}}(q') = \emptyset$.*

Notice that for each pair of states $q, q' \in Q$, the sets $\{w \in C^+ \mid \delta(q, w) = q'\}$ and $\alpha\text{-}\mathsf{FreeCycles}_{\mathcal{B}}(q')$ are both regular languages recognized by deterministic finite automata with at most $|Q|$ states. The emptiness of their intersection can be decided in time $\mathcal{O}(|C|^2 \cdot |Q|^2)$ [52]. By Lemma 15, we can therefore decide whether $\mathcal{B}$ is progress-consistent in time $\mathcal{O}(|Q|^2 \cdot (|C|^2 \cdot |Q|^2)) = \mathcal{O}(|C|^2 \cdot |Q|^4)$: for all $|Q|^2$ pairs of states $q, q' \in Q$, if $q \prec q'$, we test the emptiness of the intersection of these two regular languages.

4 Technical sketch

We discuss each direction of the proof of Theorem 10 (necessity and sufficiency of the conditions), for which complete arguments can be found in [8, Sections 4 & 5].

4.1 Necessity of the three conditions

The necessity of the first two conditions (total prefix preorder and progress-consistency) is relatively straightforward: by contrapositive, we can use the words witnessing that these properties are not satisfied to build finite one-player arenas in which positional strategies do not suffice to play optimally.

For the third condition, we need to show the following statement.

▶ **Proposition 16.** *Let $W \subseteq C^\omega$ be a DBA-recognizable objective that is half-positional over finite one-player arenas. Then, W is recognized by a Büchi automaton built on top of $\mathcal{S}_\sim$.*

The proof of Proposition 16 makes extensive use of a "normal form" of Büchi automata verifying that any α-free path can be extended to an α-free cycle [8, Section 2]. Such a normal form can be produced by *saturating* a given DBA $\mathcal{B}$ with Büchi transitions [41, 1, 2]. To do so, we add to α all transitions that do not appear in an α-free cycle of $\mathcal{B}$. This can be done by decomposing into strongly connected components the structure obtained by removing the Büchi transitions from $\mathcal{B}$.

In Figure 4, we show an intuitive example of the saturation process.

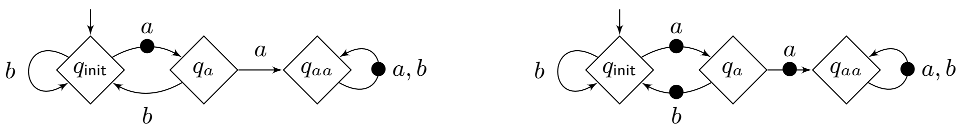

Figure 4 A DBA (left) and its unique saturation (right).

The proof of Proposition 16 is split in two steps: we first give a proof for prefix-independent objectives, and then build on it for the general case.

▶ **Lemma 17.** *Let $W \subseteq C^\omega$ be a prefix-independent DBA-recognizable objective that is half-positional over finite one-player arenas. Then, there exists $F \subseteq C$ such that $W = \mathsf{Büchi}(F)$.*

Proof sketch of Lemma 17. We assume that the objective W is recognized by a DBA $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \Delta, \alpha)$ (which has been saturated) and is prefix-independent, so all the states of $\mathcal{B}$ are equivalent for $\sim$. The goal is to find a suitable definition for F, so that $W = \mathsf{Büchi}(F)$. To do so, we exhibit a state q_{max} of $\mathcal{B}$ that is "the most rejecting state of the automaton": it satisfies that the set of α-free words from q_{max} contains the α-free words from all the other states (q_{max} is then called an *α-free-maximum*) and that the set of α-free cycles on q_{max} contains the α-free cycles on all the other states (it is also an *α-free-cycle-maximum*). We define F as the set of colors c such that $(q_{\mathsf{max}}, c) \in \alpha$.

We first show that if an α-free-maximum exists, we can assume w.l.o.g. that it is unique. Then, we show the existence of an α-free-cycle-maximum. This part of the proof relies on the half-positionality over finite one-player arenas of W, as well as on the saturation of $\mathcal{B}$. Finally, defining F using q_{max} as explained above, we prove that $W = \mathsf{Büchi}(F)$. ◀

We show how to reduce the general case to the prefix-independent case.

Proof sketch of Proposition 16. We now relax the prefix-independence assumption on W. If $\mathcal{B}$ has exactly one state per equivalence class of $\sim$, it means that it is built on top of $\mathcal{S}_\sim$, and we are done. If not, let $q_\sim \in Q$ be a state such that $|[q_\sim]| \geq 2$. Our proof shows how to modify $\mathcal{B}$ by "merging" all states in equivalence class $[q_\sim]$ into a single state, while still recognizing the same objective W. The main technical argument is to build a variant $W_{[q_\sim]}$ of objective W on a new set of colors $C_{[q_\sim]}$, that turns out to also be half-positional over finite one-player arenas and DBA-recognizable, but which is *prefix-independent*. We can therefore use the prefix-independent case and find $F_{[q_\sim]} \subseteq C_{[q_\sim]}$ such that $W_{[q_\sim]} = \mathsf{Büchi}(F_{[q_\sim]})$. Then, we exhibit a state $q_{\mathsf{max}} \in [q_\sim]$ whose α-free words are tightly linked to the elements of $F_{[q_\sim]}$. Finally, we show that it is still possible to recognize W while keeping only state q_{max} in $[q_\sim]$.

Once we know how to merge the equivalence class $[q_\sim]$ into a single state, we can simply repeat the operation for each equivalence class with multiple states, until we obtain a DBA built on top of $\mathcal{S}_\sim$. ◀

4.2 Sufficiency of the conditions

We now focus on the other direction of the proof of Theorem 10. We want to show the following statement.

▶ **Proposition 18.** *Let $W \subseteq C^\omega$ be an objective that has a total prefix preorder, is progress-consistent, and is recognizable by a DBA built on top of $\mathcal{S}_\sim$. Then, W is half-positional.*

The main technical tool to prove Proposition 18 is the notion of well-monotonic universal graph for an objective W, whose existence is sufficient to prove the half-positionality of W [48]. We will show how to build such a graph in our case.

Well-monotonic universal graphs. Let $\mathcal{G} = (V, E)$ be a graph and $W \subseteq C^\omega$ be an objective. A vertex v of $\mathcal{G}$ *satisfies* W if for all infinite paths $v_0 \xrightarrow{c_1} v_1 \xrightarrow{c_2} \dots$ from v, we have $c_1c_2\dots \in W$.

Given two graphs $\mathcal{G} = (V, E)$ and $\mathcal{G}' = (V', E')$, a *(graph) morphism from $\mathcal{G}$ to $\mathcal{G}'$* is a function $\phi\colon V \to V'$ such that $(v_1, c, v_2) \in E$ implies $(\phi(v_1), c, \phi(v_2)) \in E'$. A morphism ϕ from $\mathcal{G}$ to $\mathcal{G}'$ is *W-preserving* if for all $v \in V$, v satisfies W implies that $\phi(v)$ satisfies W. Notice that if $\phi(v)$ satisfies W, then v satisfies W, as any path $v \xrightarrow{c_1} v_1 \xrightarrow{c_2} \dots$ of $\mathcal{G}$ implies the existence of a path $\phi(v) \xrightarrow{c_1} \phi(v_1) \xrightarrow{c_2} \dots$ of $\mathcal{G}'$ – there are "more paths" in $\mathcal{G}'$.

A graph $\mathcal{U}$ is *(κ, W)-universal* if for all graphs $\mathcal{G}$ of cardinality $\le \kappa$, there is a W-preserving morphism from $\mathcal{G}$ to $\mathcal{U}$.

We consider a graph $\mathcal{G} = (V, E)$ along with a total order $\le$ on its vertex set V. We say that $\mathcal{G}$ is *monotonic* if for all $v, v', v'' \in V$, for all $c \in C$,

- $(v \xrightarrow{c} v' \text{ and } v' \ge v'') \implies v \xrightarrow{c} v''$, and
- $(v \ge v' \text{ and } v' \xrightarrow{c} v'') \implies v \xrightarrow{c} v''$.

This means that (i) whenever there is an edge $v \xrightarrow{c} v'$, there is also an edge with color c from v to all states smaller than v' for $\le$, and (ii) whenever $v \ge v'$, then v has at least the same outgoing edges as v'. Graph $\mathcal{G}$ is *well-monotonic* if it is monotonic and the total order $\le$ is a well-order (i.e., any set of vertices has a minimum). Graph $\mathcal{G}$ is *completely well-monotonic* if it is well-monotonic and there exists a vertex $\top \in V$ maximum for $\le$ such that for all $v \in V$, $c \in C$, $\top \xrightarrow{c} v$.

▶ **Theorem 19** (Consequence of [48, Theorem 1.1]). *Let $W \subseteq C^\omega$ be an objective. If for all cardinals κ, there exists a completely well-monotonic (κ, W)-universal graph, then W is half-positional (over all arenas).*

The exact result [48, Theorem 1.1] can actually be instantiated on more precise classes of arenas. However, we use it to prove here half-positionality of a family of objectives over *all* arenas, so the above result turns out to be sufficient.

Universal graphs for Büchi automata. We show that for a DBA-recognizable objective W, the three conditions from Theorem 10 imply half-positionality of W by providing a completely well-monotonic (κ, W)-universal graph for any κ.

Let $W \subseteq C^\omega$ be an objective with a total prefix preorder, that is progress-consistent, and that is recognized by a $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \Delta, \alpha)$ built on top of $\mathcal{S}_\sim$. We assume w.l.o.g. that $\mathcal{B}$ is *saturated*, as in Section 4.1. For θ an ordinal, we build a graph $\mathcal{U}_{\mathcal{B},\theta}$ as follows.

- We set the vertices as $U_{\mathcal{B},\theta} = \{(q, \lambda) \mid q \in Q, \lambda < \theta\} \cup \{\top\}$.
- For every transition $\delta(q, c) = q'$ of $\mathcal{B}$,
  - if $(q, c) \in \alpha$, then for all ordinals λ, λ', we define an edge $(q, \lambda) \xrightarrow{c} (q', \lambda')$;
  - if $(q, c) \notin \alpha$, then for all ordinals λ, λ' s.t. $\lambda' < \lambda$, we define an edge $(q, \lambda) \xrightarrow{c} (q', \lambda')$.
  - for $q'' \prec q'$, then for all ordinals λ, λ'', we define an edge $(q, \lambda) \xrightarrow{c} (q'', \lambda'')$.
- For all $c \in C$, $v \in U_{\mathcal{B},\theta}$, we define an edge $\top \xrightarrow{c} v$.

We order the vertices lexicographically: $(q, \lambda) \le (q', \lambda')$ if $q \prec q'$ or ($q = q'$ and $\lambda \le \lambda'$), and we define $\top$ as the maximum for $\le$ ($(q, \lambda) < \top$ for all $q \in Q$, $\lambda < \theta$).

Graph $\mathcal{U}_{\mathcal{B},\theta}$ is built such that on the one hand, it is sufficiently large and has sufficiently many edges so that there is a morphism from any graph $\mathcal{G}$ (of cardinality smaller than some function of $|\theta|$) to $\mathcal{U}_{\mathcal{B},\theta}$. On the other hand, for the morphism to be W-preserving, at least some vertices of $\mathcal{U}_{\mathcal{B},\theta}$ need to satisfy W, which imposes a restriction on the infinite paths from vertices. Graph $\mathcal{U}_{\mathcal{B},\theta}$ is actually built so that for any automaton state $q \in Q$ and ordinal $\lambda < \theta$, the vertex (q, λ) satisfies $q^{-1}W$ [8, Section 5]. The intuitive idea is that for a non-Büchi transition $(q, c) \notin \alpha$ of the automaton such that $\delta(q, c) = q'$, a c-colored edge from a vertex (q, λ) in the graph either (i) reaches a vertex with first component q', in which case the ordinal must decrease on the second component, or (ii) reaches a vertex with first component $q'' \prec q'$, with no restriction on the second component, but therefore with fewer winning continuations. Using progress-consistency and the fact that there is no infinitely decreasing sequence of ordinals, we can show that this implies that no infinite path in $\mathcal{U}_{\mathcal{B},\theta}$ corresponds to an infinite run in the automaton visiting only non-Büchi transitions.

We give an example of this construction.

▶ **Example 20.** We consider again the DBA $\mathcal{B}$ from Example 7, recognizing the words seeing a infinitely many times, or a twice in a row at some point. We represent the graph $\mathcal{U}_{\mathcal{B},\theta}$, with $\theta = \omega$ in Figure 5. ⌟

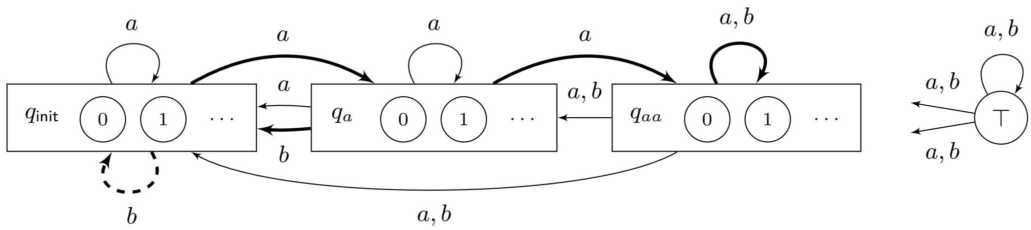

Figure 5 The graph $\mathcal{U}_{\mathcal{B},\omega}$, where $\mathcal{B}$ is the automaton from Example 7 ($\mathcal{L}(\mathcal{B}) = \mathsf{Büchi}(\{a\}) \cup C^*aaC^\omega$). The dashed edge with color b indicates that $(q_{\mathsf{init}}, \lambda) \xrightarrow{b} (q_{\mathsf{init}}, \lambda')$ if and only if $\lambda' < \lambda$ (it corresponds to the only non-Büchi transition in $\mathcal{B}$). Elsewhere, an edge between two rectangles labeled q, q' with color c means that for all ordinals λ, λ', $(q, \lambda) \xrightarrow{c} (q, \lambda')$. Thick edges correspond to the original transitions of $\mathcal{B}$. There are edges from $\top$ to all vertices of the graph with colors a and b. Vertices are totally ordered from left to right.

We show that the graph $\mathcal{U}_{\mathcal{B},\theta}$ is completely well-monotonic (Lemma 21) and, for any cardinal κ, it is (κ, W)-universal for sufficiently large θ (Proposition 22) [8, Section 5].

▶ **Lemma 21.** *Graph $\mathcal{U}_{\mathcal{B},\theta}$ is completely well-monotonic.*

▶ **Proposition 22.** *Let κ be a cardinal, and θ' be an ordinal such that $\kappa < |\theta'|$. Let $\theta = |Q| \cdot \theta'$. Graph $\mathcal{U}_{\mathcal{B},\theta}$ is (κ, W)-universal.*

Thanks to these two results, we can show Proposition 18 using Theorem 19.

References

1 Bader Abu Radi and Orna Kupferman. Minimizing GFG transition-based automata. In *ICALP*, volume 132, pages 100:1–100:16, 2019. doi:10.4230/LIPIcs.ICALP.2019.100.

2 Bader Abu Radi and Orna Kupferman. Canonicity in GFG and transition-based automata. In *GandALF*, volume 326, pages 199–215, 2020. doi:10.4204/EPTCS.326.13.

3 Benjamin Aminof and Sasha Rubin. First-cycle games. *Inf. Comput.*, 254:195–216, 2017. doi:10.1016/j.ic.2016.10.008.

4 Dana Angluin and Dana Fisman. Regular ω-languages with an informative right congruence. *Inf. Comput.*, 278:104598, 2021. doi:10.1016/j.ic.2020.104598.

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

6 Alessandro Bianco, Marco Faella, Fabio Mogavero, and Aniello Murano. Exploring the boundary of half-positionality. *Ann. Math. Artif. Intell.*, 62(1-2):55–77, 2011. doi:10.1007/s10472-011-9250-1.

7 Roderick Bloem, Krishnendu Chatterjee, and Barbara Jobstmann. Graph games and reactive synthesis. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 921–962. Springer, 2018. doi:10.1007/978-3-319-10575-8_27.

8 Patricia Bouyer, Antonio Casares, Mickael Randour, and Pierre Vandenhove. Half-positional objectives recognized by deterministic Büchi automata. *CoRR*, abs/2205.01365, 2022. arXiv:2205.01365, doi:10.48550/arXiv.2205.01365.

9 Patricia Bouyer, Ulrich Fahrenberg, Kim G. Larsen, Nicolas Markey, and Jirí Srba. Infinite runs in weighted timed automata with energy constraints. In Franck Cassez and Claude Jard, editors, *Formal Modeling and Analysis of Timed Systems, 6th International Conference, FORMATS 2008, Saint Malo, France, September 15-17, 2008. Proceedings*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2008. doi:10.1007/978-3-540-85778-5_4.

10 Patricia Bouyer, Stéphane Le Roux, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. Games where you can play optimally with arena-independent finite memory. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPIcs*, pages 24:1–24:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CONCUR.2020.24.

11 Patricia Bouyer, Nicolas Markey, Mickael Randour, Kim G. Larsen, and Simon Laursen. Average-energy games. *Acta Inf.*, 55(2):91–127, 2018. doi:10.1007/s00236-016-0274-1.

12 Patricia Bouyer, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. Arena-independent finite-memory determinacy in stochastic games. In Serge Haddad and Daniele Varacca, editors, *32nd International Conference on Concurrency Theory, CONCUR 2021, August 24-27, 2021, Virtual Conference*, volume 203 of *LIPIcs*, pages 26:1–26:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CONCUR.2021.26.

13 Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)*, volume 219 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:16, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2022.16.

14 Véronique Bruyère, Quentin Hautem, and Mickael Randour. Window parity games: an alternative approach toward parity games with time bounds. In Domenico Cantone and Giorgio Delzanno, editors, *Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016, Catania, Italy, 14-16 September 2016*, volume 226 of *EPTCS*, pages 135–148, 2016. doi:10.4204/EPTCS.226.10.

15 Véronique Bruyère, Quentin Hautem, Mickael Randour, and Jean-François Raskin. Energy mean-payoff games. In Wan Fokkink and Rob van Glabbeek, editors, *30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands*, volume 140 of *LIPIcs*, pages 21:1–21:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CONCUR.2019.21.

16 J. Richard Büchi and Lawrence H. Landweber. Definability in the monadic second-order theory of successor. *J. Symb. Log.*, 34(2):166–170, 1969. doi:10.2307/2271090.

17 J. Richard Büchi. On a decision method in restricted second order arithmetic. *Proc. Internat. Congr. on Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.

18 Antonio Casares. On the minimisation of transition-based Rabin automata and the chromatic memory requirements of Muller conditions. In *CSL*, volume 216, pages 12:1–12:17, 2022. doi:10.4230/LIPIcs.CSL.2022.12.

19 Antonio Casares, Thomas Colcombet, and Karoliina Lehtinen. On the size of good-for-games Rabin automata and its link with the memory in Muller games. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229, pages 117:1–117:20, 2022. doi:10.4230/LIPIcs.ICALP.2022.117.

20 Krishnendu Chatterjee and Laurent Doyen. Energy parity games. *Theor. Comput. Sci.*, 458:49–60, 2012. doi:10.1016/j.tcs.2012.07.038.

21 Krishnendu Chatterjee, Laurent Doyen, Mickael Randour, and Jean-François Raskin. Looking at mean-payoff and total-payoff through windows. *Inf. Comput.*, 242:25–52, 2015. doi:10.1016/j.ic.2015.03.010.

22 Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdzinski. Mean-payoff parity games. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings*, pages 178–187. IEEE Computer Society, 2005. doi:10.1109/LICS.2005.26.

23 Edmund M. Clarke, I. A. Draghicescu, and Robert P. Kurshan. A unified approch for showing language inclusion and equivalence between various types of omega-automata. *Inf. Process. Lett.*, 46(6):301–308, 1993. doi:10.1016/0020-0190(93)90069-L.

24 Thomas Colcombet, Nathanaël Fijalkow, Pawel Gawrychowski, and Pierre Ohlmann. The theory of universal graphs for infinite duration games. *CoRR*, abs/2104.05262, 2021. arXiv:2104.05262.

25 Thomas Colcombet, Nathanaël Fijalkow, and Florian Horn. Playing safe. In Venkatesh Raman and S. P. Suresh, editors, *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India*, volume 29 of *LIPIcs*, pages 379–390. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014. doi:10.4230/LIPIcs.FSTTCS.2014.379.

26 Thomas Colcombet and Damian Niwiński. On the positional determinacy of edge-labeled games. *Theor. Comput. Sci.*, 352(1-3):190–196, 2006. doi:10.1016/j.tcs.2005.10.046.

27 Stefan Dziembowski, Marcin Jurdzinski, and Igor Walukiewicz. How much memory is needed to win infinite games? In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 – July 2, 1997*, pages 99–110. IEEE Computer Society, 1997. doi:10.1109/LICS.1997.614939.

28 Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979. doi:10.1007/BF01768705.

29 E. Allen Emerson and Charanjit 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. doi:10.1109/SFCS.1991.185392.

30 Hugo Gimbert and Edon Kelmendi. Submixing and shift-invariant stochastic games. *CoRR*, abs/1401.6575, 2014. arXiv:1401.6575.

31 Hugo Gimbert and Wieslaw Zielonka. When can you play positionally? In Jirí Fiala, Václav Koubek, and Jan Kratochvíl, editors, *Mathematical Foundations of Computer Science 2004, 29th International Symposium, MFCS 2004, Prague, Czech Republic, August 22-27, 2004, Proceedings*, volume 3153 of *Lecture Notes in Computer Science*, pages 686–697. Springer, 2004. doi:10.1007/978-3-540-28629-5_53.

32 Hugo Gimbert and Wieslaw Zielonka. Games where you can play optimally without any memory. In Martín Abadi and Luca de Alfaro, editors, *CONCUR 2005 – Concurrency Theory, 16th International Conference, CONCUR 2005, San Francisco, CA, USA, August 23-26, 2005, Proceedings*, volume 3653 of *Lecture Notes in Computer Science*, pages 428–442. Springer, 2005. doi:10.1007/11539452_33.

33 Yuri Gurevich and Leo Harrington. Trees, automata, and games. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 60–65. ACM, 1982. doi:10.1145/800070.802177.

34 Nils Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. *Ann. Pure Appl. Log.*, 69(2-3):243–268, 1994. doi:10.1016/0168-0072(94)90086-8.

35 Nils Klarlund and Dexter C. Kozen. Rabin measures and their applications to fairness and automata theory. In *LICS 1991*, pages 256–265, 1991.

36 Eryk Kopczyński. Half-positional determinacy of infinite games. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 336–347. Springer, 2006. doi:10.1007/11787006_29.

37 Eryk Kopczyński. Omega-regular half-positional winning conditions. In Jacques Duparc and Thomas A. Henzinger, editors, *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2007. doi:10.1007/978-3-540-74915-8_7.

38 Eryk Kopczyński. *Half-positional Determinacy of Infinite Games.* PhD thesis, Warsaw University, 2008.

39 Alexander Kozachinskiy. Energy games over totally ordered groups. *CoRR*, abs/2205.04508, 2022. doi:10.48550/arXiv.2205.04508.

40 Alexander Kozachinskiy. One-to-two-player lifting for mildly growing memory. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 43:1–43:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.43.

41 Denis Kuperberg and Michal Skrzypczak. On determinisation of good-for-games automata. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming – 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 299–310. Springer, 2015. doi:10.1007/978-3-662-47666-6_24.

42 Lawrence H. Landweber. Decision problems for omega-automata. *Math. Syst. Theory*, 3(4):376–384, 1969. doi:10.1007/BF01691063.

43 Stéphane Le Roux, Arno Pauly, and Mickael Randour. Extending finite-memory determinacy by Boolean combination of winning conditions. In Sumit Ganguly and Paritosh K. Pandya, editors, *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, volume 122 of *LIPIcs*, pages 38:1–38:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.FSTTCS.2018.38.

44 Oded Maler and Ludwig Staiger. On syntactic congruences for omega-languages. *Theor. Comput. Sci.*, 183(1):93–112, 1997. doi:10.1016/S0304-3975(96)00312-X.

45 Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Inf. Control.*, 9(5):521–530, 1966. doi:10.1016/S0019-9958(66)80013-X.

46 Andrzej Wlodzimierz Mostowski. Regular expressions for infinite trees and a standard form of automata. In Andrzej Skowron, editor, *Computation Theory – Fifth Symposium, Zaborów, Poland, December 3-8, 1984, Proceedings*, volume 208 of *Lecture Notes in Computer Science*, pages 157–168. Springer, 1984. doi:10.1007/3-540-16066-3_15.

47 A. Nerode. Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, 1958. doi:10.2307/2033204.

48 Pierre Ohlmann. *Monotonic graphs for parity and mean-payoff games*. PhD thesis, IRIF – Research Institute on the Foundations of Computer Science, 2021.

49 Dominique Perrin and Jean-Eric Pin. *Infinite words – automata, semigroups, logic and games*, volume 141 of *Pure and applied mathematics series*. Elsevier Morgan Kaufmann, 2004.

50 Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October – 1 November 1977*, pages 46–57. IEEE Computer Society, 1977. doi:10.1109/SFCS.1977.32.

51 Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *POPL*, pages 179–190, 1989. doi:10.1145/75277.75293.

52 Michael O. Rabin and Dana S. Scott. Finite automata and their decision problems. *IBM J. Res. Dev.*, 3(2):114–125, 1959. doi:10.1147/rd.32.0114.

53 Lloyd S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. doi:10.1073/pnas.39.10.1095.

54 Ludwig Staiger. Finite-state omega-languages. *J. Comput. Syst. Sci.*, 27(3):434–448, 1983. doi:10.1016/0022-0000(83)90051-X.

55 Wolfgang Thomas. Church's problem and a tour through automata theory. In Arnon Avron, Nachum Dershowitz, and Alexander Rabinovich, editors, *Pillars of Computer Science, Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday*, volume 4800 of *Lecture Notes in Computer Science*, pages 635–655. Springer, 2008. doi:10.1007/978-3-540-78127-1_35.

56 Klaus Wagner. On ω-regular sets. *Information and control*, 43(2):123–177, 1979.

57 Wieslaw Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998. doi:10.1016/S0304-3975(98)00009-7.