Positional ω-regular languages

Antonio Casares
University of Warsaw
Poland
antoniocasares@mimuw.edu.pl

Pierre Ohlmann
CNRS, LIS, Université Aix-Marseille
France
pierre.ohlmann@lis-lab.fr

ABSTRACT

In the context of two-player games over graphs, a language L is called positional if, in all games using L as winning objective, the protagonist can play optimally using positional strategies, that is, strategies that do not depend on the history of the play. In this work, we describe the class of parity automata recognising positional languages, providing a characterisation of positionality for ω-regular languages. As corollaries, we establish decidability of positionality in polynomial time, finite-to-infinite and 1-to-2-players lifts, and show the closure under union of prefix-independent positional objectives, answering a conjecture by Kopczyński in the ω-regular case.

CCS CONCEPTS

• **Theory of computation → Automata over infinite objects**.

KEYWORDS

Infinite duration games; Positionality; Strategy complexity; Parity automata

ACM Reference Format:
Antonio Casares and Pierre Ohlmann. 2024. Positional ω-regular languages. In *39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '24), July 8–11, 2024, Tallinn, Estonia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3661814.3662087

This document contains hyperlinks. Each occurrence of a notion is linked to its *definition*. On an electronic device, the reader can click on words or symbols (or just hover over them on some PDF readers) to see their definition.

1 INTRODUCTION

1.1 Context: Strategy complexity in infinite duration games

We study games in which two antagonistic players, that we call Eve and Adam, take turns in moving a token along the edges of a given (potentially infinite) edge-coloured directed graph. Vertices of the graph are partitioned into those belonging to Eve and those belonging to Adam; when the token lands in a vertex, it is the owner of this vertex who chooses where to move next. This interaction goes on in a non-terminating mode, producing an infinite path in the graph called a play. The winner of such a play is determined according to a language of infinite sequences of colours W, called the objective of the game; plays producing a sequence of colours in W are winning for Eve, and plays that do not satisfy the objective W are winning for the opponent Adam.

One of the central applications of games on graphs is the problem of reactive synthesis: given a system interacting with its environment and a formal specification, we want to obtain a controller for the system ensuring that the specification is met. The interaction between the system and the environment can be modelled as a game where a winning strategy corresponds to a correct implementation of a controller [4, 12, 45].

In this context, a crucial parameter is the complexity of strategies required by the players to play optimally. Games admitting simple strategies are both easier to solve algorithmically, and the controllers obtained for them can be represented succinctly.

Positional strategies. The simplest strategies are positional ones, those that depend only on the current vertex, and not on the history of the play. In this work, we are interested in the following question: Given a fixed objective W, is it the case that players can play optimally using positional strategies in all games that have W as winning objective? We can ask this question just for one player (player Eve) – we say in the affirmative case that W is positional[1] – or for both players – we say that W is bipositional. Also, it might be relevant to consider the question for subclasses of games, in particular, for finite games, or for 1-player games.

Bipositionality. The class of bipositional objectives, both over finite and infinite games, is already well understood. A characterisation of bipositionality over finite games was obtained by Gimbert and Zielonka [28], using two properties called *monotonicity* and *selectivity*. An important and useful corollary of their result is what is commonly known as a *1-to-2-player lift*: an objective W is bipositional over finite games if and only if both players can play optimally using positional strategies in finite 1-player games.

Over infinite games, a very simple and elegant characterisation of bipositionality was given by Colcombet and Niwiński for prefix-independent objectives [24]: a prefix-independent objective W is bipositional if and only if it is the parity objective. In particular, these objectives are necessarily ω-regular. No such characterisation is known for non-prefix-independent objectives (although a generalisation of this result for finite memory without the prefix-independent assumption is studied in [10]).

Positionality. Although positionality is arguably more relevant than bipositionality in the context of reactive synthesis (the controller is built basing on Eve's strategies), much less is known for

[1]Sometimes in the literature the term "half-positional" or "Eve-positional" is used to stress the asymmetric nature of this notion.

LICS '24, July 8–11, 2024, Tallinn, Estonia

ACM ISBN 979-8-4007-0660-8/24/07
https://doi.org/10.1145/3661814.3662087

this class. During the 90s, positionality of some central objectives was proved, notably of parity [27] and Rabin languages [30], but the first thorough study of positionality was conducted by Kopczyński in his PhD thesis [32]. There, he provides some sufficient conditions for positionality (which were generalised in [3]) and introduces an important set of conjectures that have greatly influenced research in the area in the recent years. However, no general characterisation was found for positionality.

Recently, Ohlmann made a step forward by providing a characterisation of positionality by means of the existence of graph-theoretical structures known as monotone universal graphs [40, 41]. While this characterisation is a valuable tool for proving positionality, it is not constructive and does not directly yield decidability results. Also, Ohlmann's result comes with a caveat: necessity of the existence of universal graphs for positional objectives is only guaranteed for those containing a neutral letter (a letter that does not change membership to W after its removal). He conjectures that this restriction is not essential, as the addition of a neutral letter to any objective should not break positionality.

ω-regular languages. A central class of languages over infinite words is the class of ω-regular languages, which admits several alternative definitions: these are the languages recognised by deterministic parity automata, by non-deterministic Büchi automata, definable using ω-expressions, or using monadic second order logic [13, 38, 39].

One of the main contributions of Kopczyński was to show decidability of positionality over finite games for prefix-independent ω-regular objectives [31, Theorem 2]. His procedure works by enumerating all possible game graphs where positionality might fail (up to a sufficiently large size); it runs in $O(n^{O(n^2)})$ time, and does not reveal much about the structure of automata recognising positional languages.

Regarding positionality over arbitrary games and for non-prefix-independent objectives, characterisations have been found for some subclasses of ω-regular objectives. For closed objectives (objectives recognised by safety automata), positionality was characterised by Colcombet, Fijalkow and Horn in 2014 [23].

Recently, a characterisation of positionality for languages recognised by deterministic Büchi automata was provided by Bouyer, Casares, Randour and Vandenhove [7] (see also Proposition 4.2). As a corollary, they establish polynomial-time decidability of positionality for deterministic Büchi automata. However, the conditions they provide are not necessary for positionality in general, for instance, for languages recognised by coBüchi automata.

Finite-to-infinite and 1-to-2-player lifts. As mentioned above, a consequence of Gimbert and Zielonka's result [28] is that, in order to check bipositionality over finite games, it suffices to check whether players can win positionally in 1-player games. Recently, generalisations of 1-to-2-player lifts have been studied in the setting of finite memory by Kozachinskiy [33] and Vandenhove [10, 11, 46]. Vandenhove conjectures that if W is positional over Eve-games (resp. over finite games), then W is positional over all games [46, Conjecture 9.1.1]. This conjecture has been shown to hold in the case of languages recognised by deterministic Büchi automata [7].

Closure under union. One of the recurring themes in Kopczyński's PhD thesis [32] is the following question.

Conjecture 1.1 (*Kopczyński's conjecture* [32, Conjecture 7.1]). *Let $W_1, W_2 \subseteq \Sigma^\omega$ be two prefix-independent positional objectives. Then $W_1 \cup W_2$ is positional.*

Very recently, Kozachinskiy [34] disproved this conjecture, but only for positionality over *finite games*. Also, the counter-example he gives is not ω-regular. On the positive side, Kopczyński's conjecture has been proved to hold for objectives recognised by deterministic Büchi automata [7], as well as for the family of Σ_2^0 objectives (objectives recognised by infinite coBüchi automata) [41, 42]. Kopczyński's conjecture and this latter result have been generalised to the setting of finite memory [19]. Solving Kopczyński's conjecture over infinite games is one of the driving open questions for the field.

1.2 Contributions and organisation

Our main contribution is a characterisation of positionality for ω-regular languages, stated in Theorem 3.1. We propose a syntactic description of a family of deterministic parity automata, so that any automaton in this class recognises a positional language, and any positional language can be recognised by such an automaton. In fact, we describe two slightly different such families, called, respectively, fully progress consistent signature automata and ε-completable automata. These families offer distinct advantages and complement our intuitions on positionality.

From this characterisation, we derive multiple corollaries that address the majority of open questions related to positionality in the case of ω-regular languages:

(1) **Decidability in polynomial time.** Given a deterministic parity automaton $\mathcal{A}$, we can decide in polynomial time whether $\mathcal{L}(\mathcal{A})$ is positional or not (Theorem 3.3).
(2) **Finite-to-infinite and 1-to-2-players lift.** An ω-regular objective W is positional over arbitrary games if and only if it is positional over finite, ε-free Eve-games (Theorem 3.4). This answers a question raised by Vandenhove [46, Conjecture 9.1.1].
(3) **Closure under union.** The union of two ω-regular positional objectives is positional, provided that one of them is prefix-independent (Theorem 3.5). This solves a stronger variant of Kopczyński's conjecture in the case of ω-regular languages.
(4) **Closure under addition of a neutral letter.** If W is ω-regular and positional, the objective obtained by adding a neutral letter to W is positional too (Theorem 3.6). This solves Ohlmann's conjecture in the case of ω-regular languages.

We obtain some additional results pertaining to classes of objectives that are not necessarily ω-regular. We relax the ω-regularity hypothesis in two orthogonal ways.

(5) **Characterisation of bipositionality of all objectives.** We extend the characterisation of bipositionality of Colcombet and Niwiński [24] to all objectives, getting rid of the prefix-independence assumption (Theorem 6.1).

(6) **Characterisation of positionality of closed and open objectives.** We characterise positionality for closed and open objectives (Theorem 6.3). We also obtain as corollaries 1-to-2 players lifts and closure under addition of a neutral letter for these classes of objectives.

Technical tools. We would like to highlight some technical tools that take primary importance in our proofs.

Universal graphs. In general, showing that a given objective is positional can be challenging, as we need to show that *for every game* Eve can play optimally using positional strategies. Ohlmann's characterisation using monotone universal graphs provides a painless path to prove positionality (see Proposition 2.2). We rely on this result to show that parity automata satisfying the syntactic conditions imposed in Theorem 3.1 do indeed recognise positional languages.

History-deterministic automata. History-deterministic automata are a model in between deterministic and non-deterministic ones; we refer to [6, 36] for detailed expositions on them. Although the statements of our results do not mention history-determinism, they appear naturally in two different parts of our proofs:

- Establishing necessity of the syntactic conditions from our main characterisation requires a very fine control of the structure of automata. We develop a technique for decomposing automata, for which we need to use and generalise the methods introduced by Abu Radi and Kupferman [1] for the minimisation of HD coBüchi automata.
- To show the sufficiency of these conditions, we build a monotone universal graph from a signature automaton. To facilitate this process, we first "saturate" automata, adding as many transitions as possible without modifying the languages they recognise. This procedure generates non-determinism, but preserves history-determinism, the key property that allows us to prove universality of the obtained graph (see Proposition 5.2).

We believe that this use of history-deterministic automata showcases their usefulness and canonicity.

Normal form of parity automata. In our central proof, we rely on a normal form of parity automata, as defined in [17, Section 6.2]. Automata in normal form present a set of properties that simplify manipulating them and reasoning about their runs. We make consistent use of these properties in our combinatorial arguments. The use of this normal form, or variants of it, is common in the literature, and finds application in diverse contexts, such as the study of history-deterministic coBüchi automata [1, 26, 35] or automata learning [5].

Congruences for parity automata. Since the beginning of the theory of finite automata, the notion of congruence has played a fundamental role [2, 37, 43]. Here, we propose a notion of congruences for parity automata that make it possible to build quotient automata that are compatible with the acceptance condition. This newly introduced vocabulary allows us to formalise the details of the proof of Theorem 3.1 in a simpler way. We believe that it will be useful for the study of parity automata in other contexts.

Organisation of the paper. This paper is an extended abstract aimed at presenting the main contributions and central proof ideas of this work. The full version [20] contains all proofs, as well as numerous extra examples and a detailed warm-up section is attached as an appendix.

After introducing some general definitions and terminology used throughout the paper, we begin Section 3 by stating the characterisation result (Theorem 3.1) and its main consequences. A substantial portion of that section is devoted to explain the two central automata-theoretic concepts used in the statement of the theorem: signature automata and ε-complete automata. In Section 4, we exemplify these notions by instantiating them in the subclasses of languages recognised by deterministic Büchi and coBüchi automata, respectively. We provide proofs for these cases, which already illustrate several important ideas used in the general setting. In Section 5 we explain how to generalise these ideas to the full class of ω-regular languages. To conclude, we include some discussions about future research directions in Section 6, and state some further corollaries from our results pointing in those directions.

2 PRELIMINARIES

2.1 Games and positionality

2.1.1 Games on graphs. A *Σ-graph* G is given by a (potentially infinite) set of vertices V together with a set of coloured directed edges $E \subseteq V \times \Sigma \times V$. We write $v \xrightarrow{c} v'$ to refer to an edge in G with source v, target v', and colour c. We use notations $v \overset{w}{\rightsquigarrow} v'$ for finite paths from v to v' labelled by $w \in \Sigma^*$ and $v \overset{w}{\rightsquigarrow}$ for infinite paths from v labelled by $w \in \Sigma^\omega$. The *size* of a graph G is defined to be the cardinality of V. We assume throughout the paper that Σ-graphs do not contain *sinks*, that is, every vertex has at least one outgoing edge.

A *game* $\mathcal{G} = (V, E, V_{\mathrm{Eve}}, W)$ is given by an $\Sigma \cup \{\varepsilon\}$-graph $G = (V, E)$ together with an objective $W \subseteq \Sigma^\omega$ and a partition $V = V_{\mathrm{Eve}} \sqcup V_{\mathrm{Adam}}$ of the vertices into those controlled by *Eve* and by her opponent *Adam*. Letter ε is a fresh element used to represent *uncoloured edges*; we impose that no infinite path in G is composed exclusively of ε-edges. Games not containing uncoloured edges are called *ε-free*. An *Eve-game* is a game $\mathcal{G}$ in which all the vertices are controlled by Eve, that is, $V = V_{\mathrm{Eve}}$. Unless stated otherwise, we take the point of view of player Eve; expressions as "winning" will implicitly stand for "winning for Eve", and strategies will be defined for her.

In a game, the players move a pebble from one vertex to another for an infinite duration. The player who owns the vertex v where the pebble is placed chooses an edge $v \xrightarrow{c} v'$ and the pebble travels through this edge to its target, producing colour c. In this way, they produce a path $\rho = v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} v_2 \xrightarrow{c_2} \cdots \in E^\omega$, that we call a *play*. Such a play is *winning* if the sequence $w \in \Sigma^\omega$ obtained by removing from $c_0c_1c_2\dots$ the occurrences of ε belongs to W.

A *strategy* is a function $\mathrm{strat} : E^* \to E$, that tells Eve which move to choose after any possible finite play. We say that a play $\rho \in E^\omega$ is *consistent with the strategy* strat if after each finite prefix $\rho' \sqsubseteq \rho$ ending in a vertex controlled by Eve, the next edge in ρ is $\mathrm{strat}(\rho')$. We say that the strategy strat is *winning from* a vertex $v \in V$ if all infinite plays starting in v consistent with strat are winning. If such a strategy exists, we say that Eve *wins $\mathcal{G}$ from v*.

The *winning region* of a game $\mathcal{G}$, written $\mathit{Win}_{\mathrm{Eve}}(\mathcal{G})$, is the set of vertices $v \in V$ such that Eve wins $\mathcal{G}$ from v.

2.1.2 Positionality. We say that a strategy strat : $E^* \to E$ is *positional* if there exists a mapping $\sigma\colon V_{\mathrm{Eve}} \to E$ such that for every finite play $\rho = v_0 \xrightarrow{c_0} \dots \xrightarrow{c_{n-1}} v_n$ ending in a vertex v_n controlled by Eve we have $\mathrm{strat}(\rho) = \sigma(v_n)$.

An objective $W \subseteq \Sigma^\omega$ is *positional* if for every W-game, Eve has a positional strategy which is winning from every vertex of her winning region.[2] We say that W is *bipositional* if both W and $\Sigma^\omega \setminus W$ are positional. If $\mathcal{X}$ is a subclass of W-games (notably, finite, ε-free and Eve-games), we say that W is *positional over $\mathcal{X}$ games* if for every W-game in $\mathcal{X}$, Eve has a positional strategy winning from her winning region.

Remark 2.1. Our notion of positionality uses what sometimes are called *uniform strategies*, that is, we require that a single positional strategy suffices to win independently of the initial vertex.

2.1.3 Universal graphs. We now introduce monotone universal graphs, which is our main tool for deriving positionality results.

Given two Σ-graphs $G = (V, E)$ and $G' = (V', E')$, a *morphism of Σ-graphs* ϕ from G to G' is a map $\phi : V \to V'$ such that for each edge $v \xrightarrow{c} v'$ in G, it holds that $\phi(v) \xrightarrow{c} \phi(v')$ defines an edge in G'.

Given a Σ-graph G, a vertex v of G and an objective $W \subseteq \Sigma^\omega$, we say that *$v$ satisfies* W in G if for any infinite path $v \overset{w}{\rightsquigarrow}$ in G, it holds that $w \in W$. Given a cardinal κ, a Σ-graph U is *(κ, W)-universal* if all graphs G of size $< \kappa$ admit a morphism $\phi : G \to U$ such that any vertex v that satisfies W in G is mapped to a vertex $\phi(v)$ that satisfies W in U.

A Σ-graph G together with an ordering (resp. well-ordering) of its vertices is called an *ordered graph* (resp. *well-ordered* graph). An ordered graph $(G, \leq)$ is called *monotone* if

$$v \geq u \xrightarrow{c} u' \geq v' \implies v \xrightarrow{c} v' \text{ in } G.$$

A letter $c \in \Sigma$ is *neutral for* an objective W if, for all $w_1, w_2, \dots \in \Sigma^+$ and $n_1, n_2, \dots \in \mathbb{N}$:

- $c^{n_1} w_1 c^{n_2} w_2 \dots \in W \iff w_1 w_2 \dots \in W$, and
- $w_1 c^\omega \in W \iff {w_1}^{-1} W \neq \emptyset$ (that is, there is $w \in \Sigma^\omega$ such that $w_1 w \in W$).

We now state our main tool for proving positionality.

Proposition 2.2 ([41, Theorem 3.1]). *Let $W \subseteq \Sigma^\omega$ be an objective. If for all cardinals κ there exists a (κ, W)-universal well-ordered monotone graph, then W is positional over all games. Moreover, the converse holds if W admits a neutral letter.*

2.2 Automata

2.2.1 Parity automata. A *(non-deterministic) parity automaton* over the alphabet Σ is given by a tuple $\mathcal{A} = (Q, \Sigma, q_{\mathsf{init}}, \Delta, \mathsf{p})$, where Q is a finite set of states, Σ is a set of letters called the *input alphabet*, q_{init} is the *initial state*, $\Delta \subseteq Q \times \Sigma \times Q$ is a set of transitions, and $\mathsf{p}\colon \Delta \to [d_{\min}, d_{\max}] \subseteq \mathbb{N}$ is a function assigning integer numbers to transitions; we refer to these numbers as *priorities*. We write $q \xrightarrow{a:x} q'$ to indicate that there is a transition $e = (q, a, q') \in \Delta$ with $\mathsf{p}(e) = x$.

For a state $q \in Q$, we write $\mathcal{A}_q$ to denote the automaton obtained by setting q as initial state. We assume in this paper that all automata are *complete*, that is if for every $q \in Q$ and $a \in \Sigma$, there is at least one transition $q \xrightarrow{a:s}$.

We write $q \overset{w:x}{\rightsquigarrow} p$ if there exists a path with minimal priority x from q to p labelled w. We write $q \overset{w:\geq x}{\rightsquigarrow} p$ to denote that there exists such a path producing no priority strictly smaller than x.

A *run over* an infinite word $w = a_0 a_1 a_2 \dots \in \Sigma^\omega$ in $\mathcal{A}$ is a path

$$\rho = q_{\mathsf{init}} \xrightarrow{a_0:x_0} q_1 \xrightarrow{a_1:x_1} q_2 \xrightarrow{a_2:x_2} \dots \in \Delta^\omega.$$

It is *accepting* if

$$\min\{x \in \mathbb{N} \mid x = x_i \text{ for infinitely many } i\} \text{ is even,}$$ [3]

and *rejecting* otherwise. A word $w \in \Sigma^\omega$ is *accepted* by $\mathcal{A}$ if there exists an accepting run over w. The *language recognised* by an automaton $\mathcal{A}$ is the set

$$\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^\omega \mid w \text{ is accepted by } \mathcal{A}\}.$$

Two automata recognising the same language are said to be *equivalent*. In this paper, "language" and "*objective*" are synonyms.

We say that an automaton $\mathcal{A}$ is *deterministic* if for every $q \in Q$ and $a \in \Sigma$, there is only one a-transition $q \xrightarrow{a:x}$ outgoing q. Any parity automaton admits an equivalent deterministic one [39]. A language is called *ω-regular* if it can be recognised by a parity automaton.

Remark 2.3 (Transition-based acceptance). We emphasise that in our definition, the acceptance condition is put over the *transitions* of the automata. This will be a crucial element in our characterisation. We refer to [16, Chapter VI] for further discussions on the comparison between transition-based and state-based automata.

History-determinism. An automaton $\mathcal{A}$ is *history-deterministic* (shortened HD) if there is function $\mathsf{r}\colon \Delta^* \times \Sigma \to \Delta$, such that for every word $w = a_0 a_1 \dots \in \mathcal{L}(\mathcal{A})$, the sequence $e_0 e_1 \dots \in \Delta^\omega$ defined by $e_i = \mathsf{r}(e_0 \dots e_{i-1}, a_i)$ is an accepting run over w in $\mathcal{A}$. In other words, r should be able to construct an accepting run in $\mathcal{A}$ letter-by-letter with only the knowledge of the word so far, for all words in $\mathcal{L}(\mathcal{A})$.

Büchi and coBüchi automata. A *Büchi automaton* is a parity automaton using $[0, 1]$ as set of priorities. In this case, transitions carrying priority 0 are called *Büchi transitions*. Parity automata using $[1, 2]$ as set of priorities are called *coBüchi*, and transitions carrying priority 1 are called *coBüchi transitions* in this case.

We say that a language W is *Büchi recognisable* (resp. *coBüchi recognisable*) if it can be recognised by a deterministic Büchi automaton (resp. deterministic coBüchi automaton). These are incomparable strict subclasses of the ω-regular languages.

[2] We remark that the question of positionality is independent from that of determinacy of the game. Nevertheless, all games with ω-regular objectives are determined [12].

[3] Note that we use the *min-parity* condition, that is, smaller priorities are the most important ones (in some parts of the literature the *max-parity* condition is used).

Automata with ε-transitions. An *automaton with ε-transitions* is defined just as an automaton over the alphabet $\Sigma \cup \{\varepsilon\}$, where $\varepsilon \notin \Sigma$ is a distinguished letter. In particular, transitions labelled by ε (called ε-transitions) also carry priorities. The language of an automaton $\mathcal{A}$ with ε-transitions is the set of words $w \in \Sigma^\omega$ such that there exists $w' \in (\Sigma \cup \{\varepsilon\})^\omega$ which is accepted by $\mathcal{A}$ and such that w is obtained from w' by removing all occurrences of ε.

2.2.2 *Residuals.* Let $W \subseteq \Sigma^\omega$ be a language of infinite words and let $u \in \Sigma^*$. We define the *residual of W with respect to u* by

$$u^{-1}W = \{w \in \Sigma^\omega \mid uw \in W\}.$$

We denote $\mathrm{Res}(W)$ the set of residuals of W, which we will always order by inclusion.

We now state a key monotonicity property for residuals; its proof is a direct check.

Lemma 2.4. *For any language $W \subseteq \Sigma^\omega$ and for any finite words $u, u', w \in \Sigma^*$, if $u^{-1}W \subseteq u'^{-1}W$ then $(uw)^{-1}W \subseteq (u'w)^{-1}W$.*

If $\mathcal{A}$ is a deterministic automaton recognising W, for every state q, and any word u reaching q from the initial state, $u^{-1}W$ coincides with $\mathcal{L}(\mathcal{A}_q)$. We say that $u^{-1}W$ is the *residual corresponding to q* in this case. We say that two states q and p are *equivalent* if $\mathcal{L}(\mathcal{A}_q) = \mathcal{L}(\mathcal{A}_p)$, and write $q \sim_{\mathcal{A}} p$.

Prefix-independence. We say that a language $W \subseteq \Sigma^\omega$ is *prefix-independent* if for all $w \in \Sigma^\omega$ and $u \in \Sigma^*$, $uw \in W$ if and only if $w \in W$. Equivalently, W is prefix-independent if $\mathrm{Res}(W)$ is a singleton.

3 POSITIONALITY OF ω-REGULAR OBJECTIVES: STATEMENT OF THE RESULTS

In this section, we state the central result of the paper: a full characterisation of deterministic parity automata recognising positional ω-regular languages (Theorem 3.1). In Section 3.3 we introduce and explain the technical terminology used in this Theorem, and in Section 3.2 we state the main consequences of this characterisation.

3.1 The characterisation theorem

We state our main characterisation theorem. Items are ordered following the sequence of logical implications in its proof.

Theorem 3.1. *Let $W \subseteq \Sigma^\omega$ be an ω-regular objective. The following are equivalent:*

1. *W is positional over finite ε-free Eve-games.*
2. *There is a deterministic fully progress consistent signature automaton recognising W.*
3. *There is a deterministic ε-completable automaton recognising W.*
4. *For all cardinals κ, there is a well-ordered monotone (κ, W)-universal graph.*
5. *W is positional over all games (potentially infinite and containing ε-edges).*

This is an automata-oriented characterisation of positionality: we identify two classes of deterministic parity automata (fully progress consistent signature and ε-completable, defined below), such that any positional language can be recognised by automata in these classes. Each of them presents some formal advantages that make them suitable for different kind of proofs. On one hand, signature automata can be built recursively from any given automaton recognising a positional language. On the other, ε-completable automata are closer to monotone universal graphs, allowing to prove positionality of a language recognised by such an automaton. We explain these notions in Section 3.3.

In fact, regarding ε-completable automata we obtain a stronger result: *any* parity automaton recognising a positional language is ε-completable (including non-deterministic ones). However, the proof of this result relies on Theorem 3.1 and its consequences. This is made more precise in the proposition below.

Proposition 3.2. *Let W be an ω-regular objective. The following are equivalent:*

1. *There is a deterministic ε-completable automaton recognising W.*
2. *There is a history-deterministic ε-complete automaton recognising W.*
3. *W is positional over all games.*
4. *Any automaton recognising W is ε-completable.*

3.2 Main consequences on positionality

Before defining fully progress consistent signature automata and ε-completable automata, we discuss consequences of Theorem 3.1.

Decidability of positionality in polynomial time.

Theorem 3.3. *Given a deterministic parity automaton $\mathcal{A}$, we can decide in polynomial time whether $\mathcal{L}(\mathcal{A})$ is positional.*

Although this result is not directly implied by Theorem 3.1, we prove (see Section 5.3) that we can decide in polynomial time whether a deterministic parity automaton admits an equivalent fully progress consistent signature automaton, or, also, if it is ε-completable. This provides two conceptually different polynomial-time procedures for checking positionality.

Finite-to-infinite and 1-to-2 player lifts. The following result simply restates the implication (1) $\implies$ (5) from Theorem 3.1.

Theorem 3.4. *If an ω-regular objective is positional over finite, ε-free Eve-games, then it is positional over all games (possibly infinite and containing ε-edges).*

Closure under union of prefix-independent positional languages. We now show that Kopczyński's conjecture holds for ω-regular languages: prefix-independent positional languages are closed under union. In fact, we show a stronger result: it suffices to suppose that only one of the objectives is prefix-independent.

Theorem 3.5. *Let $W_1, W_2 \subseteq \Sigma^\omega$ be two positional ω-regular objectives, and suppose that W_1 is prefix-independent. Then, $W_1 \cup W_2$ is positional.*

The result is first easily proved for Eve-games, where prefix-independence of W_1 guarantees a uniform positional strategy. Then Theorem 3.5 follows thanks to the 1-to-2 player lift (Theorem 3.4).

Kopczyński's conjecture and its stronger version in which only one of the objectives is required to by prefix-independent remain open for arbitrary objectives.

Closure of positionality under addition of neutral letters. As mentioned in the introduction, Ohlmann recently characterised positional objectives by means of the existence of universal graphs [41] (stated in Proposition 2.2). However, one direction of the proof requires a further hypothesis, W has to contain a neutral letter. He conjectures that this hypothesis is superfluous, as we could add neutral letters without breaking positionality. Using our characterisation, we settle this question in the case of ω-regular objectives.

For an objective $W \subseteq \Sigma^\omega$, we let W^ε be the unique objective obtained by adding a fresh neutral letter ε to W. That is, we define

$$W^\varepsilon \subseteq (\Sigma \cup \{\varepsilon\})^\omega = \left\{ \begin{array}{l} w_1\varepsilon^\omega \text{ such that } {w_1}^{-1}W \neq \emptyset, \text{ and} \\ \varepsilon^{n_1} w_1 \varepsilon^{n_2} w_2 \dots \text{ such that } w_1 w_2 \dots \in W \end{array} \right\}.$$

Theorem 3.6. *If an ω-regular objective W is positional over ε-free games, then W^ε is positional.*

3.3 Signature and ε-complete automata

We now present the two central automata-theoretic notions of our characterisation: signature automata and ε-completable automata. Before giving formal definitions, let us explain some of the intuitions behind these notions. Both of these models are deterministic parity automata together with a collection of nested total preorders over its states, and the finest of these preorders will define a total order with some nice monotonicity properties.

The coarsest of this preorders, $\leq_0$, is going to correspond to the inclusion of residuals: $q \leq_0 p$ if $\mathcal{L}(\mathcal{A}_q) \subseteq \mathcal{L}(\mathcal{A}_p)$; positionality of the objective will imply that this preorder is indeed total. We note that if $q <_0 p$, we can add a transition $p \xrightarrow{\varepsilon} q$ to the automaton without changing its language (going to a "worse state" will not lead to new accepted words). Positionality of the objective will allow us to not only add this transition, but also to output priority 0 on it: $p \xrightarrow{\varepsilon:0} q$. Note also that, if $q \leq_0 p$, we cannot add a transition $q \xrightarrow{\varepsilon:0} p$ without modifying the language accepted by the automaton.[4] Therefore, $q <_0 p$ if and only if we can add a transition $p \xrightarrow{\varepsilon:0} q$.

In general, we will have multiple equivalent states for $\leq_0$ (for instance, if the language is prefix-independent all states will be $\sim_0$-equivalent). We consider subsequent preorders to refine $\leq_0$, one for each even priority. The approach we take with ε-completable automata is to define $q <_2 p$ if we can add a transition $p \xrightarrow{\varepsilon:2} q$ without augmenting the language of the automaton; this is described more precisely in Section 3.3.1 below. We then present in Section 3.3.2 the definition of signature automata, which are a variation around the same ideas, where each preorder is built according to properties of the paths in the automaton.

A mental image that might be useful to the reader is to picture the states of such an automaton as the leaves of an ordered tree, where different levels in the tree correspond to even priorities. Leaves below a node at level x in the tree will be states that are $\sim_x$-equivalent (see Figures 1 and 2). Then, roughly speaking, when reading a transition with priority x from q, one only requires knowledge about $[q]_x$, the equivalence classes of the states of the automaton at level x. In the case of ε-completable automata, this intuition is formalised by the presence of ε-transitions which allow to reach equivalent states (at the cost of reading an odd priority); in the case of signature automata, this is enforced in the definition.

3.3.1 ε-completable automata.

Definition 3.7 (ε-complete automata). An *ε-complete* automaton $\mathcal{A}$ is a non-deterministic parity automaton (with ε-transitions), with priorities ranging between 0 and $d+1$, where d is even, such that:

- The relations $\xrightarrow{\varepsilon:1}, \xrightarrow{\varepsilon:3}, \dots, \xrightarrow{\varepsilon:d+1}$ all define total preorders, each refining the previous one.
- For each even $x \in \{0, 2, \dots, d\}$, the relation $\xrightarrow{\varepsilon:x}$ is the strict variant of $\xrightarrow{\varepsilon:x+1}$, which means that $q \xrightarrow{\varepsilon:x} q'$ if and only if $q' \xrightarrow{\varepsilon:x+1} q$ does not hold.

An automaton $\mathcal{A}$ is *ε-completable* if one may add ε-transitions making it ε-complete without augmenting its language. We call the obtained automaton an *ε-completion* of $\mathcal{A}$.

On an intuitive level, if x is even, $q \xrightarrow{\varepsilon:x} q'$ means that "q is much better than q'", since one may, at any point, move from q to q' and be rewarded with an even priority x on the way. On the contrary, $q' \xrightarrow{\varepsilon:x+1} q$ means that "q' is not much worse than q", since one may at any point move from q' to q at the cost of reading an odd priority $x+1$. In other words, in a ε-complete automaton, one may say that q and q' are comparable for priority x.

Example 3.8. Consider the automaton in Figure 1, which recognises the language W of words with either infinitely many a's, or with no occurrence of a and finitely many occurrences of the factor bb. It turns out that the following ε-transitions can be added all together to the automaton without modifying its language (only some of them are depicted in the figure):

- $q_2, q_3 \xrightarrow{\varepsilon:0} q_1$,
- $q_2, q_3 \xrightarrow{\varepsilon:1} q_1$, $q_2 \xrightarrow{\varepsilon:1} q_3$, $q_3 \xrightarrow{\varepsilon:1} q_2$, and $q \xrightarrow{\varepsilon:1} q$ for all q,
- $q_j \xrightarrow{\varepsilon:2} q_i$ for $i < j$, and
- $q_j \xrightarrow{\varepsilon:3} q_i$ for $i \leq j$.

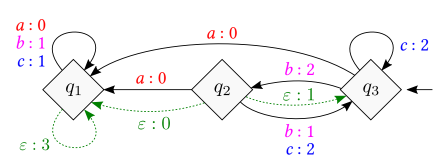


Figure 1: Automaton $\mathcal{A}$ recognising $W = \mathtt{Inf}(a) \cup (\mathtt{No}(a) \wedge \mathtt{Fin}(bb))$. Many ε-transitions can be added to $\mathcal{A}$, some of them are depicted in dotted green.

This is maximal: no further ε-transition can be added. For x an even priority, we define $q <_x q'$ if $q' \xrightarrow{\varepsilon:x} q$ is possible, and $q \sim_x q'$ if both $q' \xrightarrow{\varepsilon:x+1} q$ and $q \xrightarrow{\varepsilon:x+1} q'$ are possible. We obtain then:

- $q_1 <_0 q_2, q_3$ and $q_2 \sim_0 q_3$,
- $q_1 <_2 q_2 <_2 q_3$.

We can represent the obtained structure of nested preorders as a tree with two levels, as displayed in Figure 2. For instance, the

[4] For this to be correct when $q \sim_0 p$, we need to assume that if $q \sim_0 p$, then q and p occur in a same SCC of the automaton, which can always be assumed w.l.o.g.

fact that $q_2 \sim_0 q_3$ is captured by the fact that these states have the same ancestor at level 0 of the tree. All transitions on this tree may be added to the automaton as ε-transitions (like the three dashed ones) without increasing the language; therefore the automata is ε-completable. This implies that W is positional (Theorem 3.1).

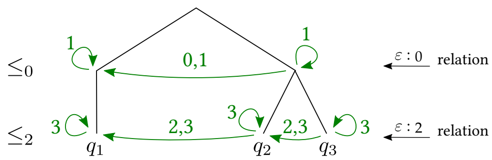


Figure 2: Tree corresponding to the 2-level-decomposition of the automaton $\mathcal{A}$ from Figure 1.

3.3.2 *Fully progress consistent signature automata.*

For this subsection we let $\mathcal{A}$ be a deterministic parity automaton over Σ with states Q and using priorities in $[0, d_{\max}]$.

An equivalence relation $\sim$ over Q is a *congruence* if for all pair of equivalent states $q \sim q'$ and for all $a \in \Sigma$, if $q \xrightarrow{a} p$ and $q' \xrightarrow{a} p'$, then $p \sim p'$ (regardless of the priority produced by the transitions). We say that it is a $[0, x]$*-faithful congruence* if, moreover, if $q \xrightarrow{a:y} p$ with $y \le x$ then $q' \xrightarrow{a:y} p'$ (both transitions produce the same output priority).

We note that a preorder $\le_x$ over Q induces the equivalence relation $\sim_x$ defined by

$$q \sim_x q' \iff q \le_x q' \text{ and } q' \le_x q.$$

Definition 3.9 (Signature automaton). A *signature automaton* is a deterministic parity automaton $\mathcal{A}$ together with a collection of nested total preorders $\le_0, \le_1, \le_2, \dots, \le_{d_{\max}}$ over Q ($\le_{x+1}$ refines $\le_x$) such that:

I) Preorder $\le_0$ is given by the inclusion of residuals.
II) For x even, the equivalence relation $\sim_x$ is a $[0, x]$-faithful congruence.
III) For $2 \le x \le d_{\max}$, x even, and $q \sim_{x-2} p$:
$$q <_{x-1} p \implies \text{there is no path } q \overset{w:\ge x}{\rightsquigarrow} p.$$
IV) For x even, transitions using priorities $\ge x$ are monotone for $\le_x$ over each $\sim_{x-1}$ class. That is, for $q \sim_{x-1} q'$, if $q \le_x q'$:
$$q \xrightarrow{a:\ge x} p \implies q' \xrightarrow{a:\ge x} p', \text{ and } p \le_x p'.[5]$$

The definition of odd equivalence classes $\sim_{x-1}$ and condition (III) serve to restrict the domain to which the local monotonicity of condition (IV) applies. We also note that Condition (IV) is always satisfied for $x = 0$ by monotonicity of the residuals (we interpret $\sim_{-1}$ to be the trivial relation).

The existence of a signature automaton recognising an objective W does not suffice to ensure positionality of W; we need to impose an additional restriction which is the following.

[5] Note that, as $\mathcal{A}$ is assumed deterministic, p and p' are the unique a-successors of q and q', respectively.

Definition 3.10 (Full progress consistency). We say that a signature automaton $\mathcal{A}$ is *fully progress consistent* if, for each x even and every finite word $w \in \Sigma^*$:

$$q <_x p \text{ and } q \overset{w:\ge x}{\rightsquigarrow} p \implies w^\omega \in \mathcal{L}(\mathcal{A}_q).$$

Example 3.11. Consider the automaton from Figure 1. We define nested preorders over it as follows:

- Preorder $\le_0$ is given by the inclusion of residuals: $q_1 <_0 q_2, q_3$, and $q_2 \sim_0 q_3$.
- Preorder $\le_1$ coincides with preorder $\le_0$.
- Preorder $\le_2$ is a total order: $q_1 <_2 q_2 <_2 q_3$.

We note that $\le_0$ and $\le_2$ are defined as in Example 3.8 (they are given by the tree in Figure 2). It is a direct check to verify that this preorders satisfy all the requirements of a signature automaton, which is moreover fully progress consistent.

4 SPECIAL CASES AND EXAMPLES

In this section, we instantiate the characterisation of Theorem 3.1 for the classes of languages recognised by deterministic Büchi and coBüchi automata. Our goal is to both shed some light on the definition of signature automata, and to explain the proofs on how we obtain these automata assuming positionality (implication (1) $\implies$ (2) in Theorem 3.1).

4.1 Büchi recognisable objectives

The characterisation of positionality for languages recognised by deterministic Büchi automata that we present here (Proposition 4.2) was first obtained in [7]. However, we have adapted and simplified the existing proof in order to generalise it to all parity automata.

4.1.1 *Signature automata: the Büchi case.*

Assume that $\mathcal{A}$ is a Büchi automaton. Then in order to be able to define the structure of a signature automaton over it it should satisfy:

I) The relation $\le_0$ defined by $q \le_0 p$ if $\mathcal{L}(\mathcal{A}_q) \subseteq \mathcal{L}(\mathcal{A}_p)$ is a total preorder.
II) If $q \sim_0 q'$ and $q \xrightarrow{a:0}$, then $q' \xrightarrow{a:0}$. (Uniformity of 0-transitions).

Since $\mathcal{A}$ does not use priority 2, condition (III) of the definition of signature automaton is vacuous, and therefore the order $\le_1$ is irrelevant. Also, condition (IV) is satisfied by monotonicity of the residuals (Lemma 2.4).

Observe that if $\mathcal{A}$ satisfies the previous condition (II), we can safely merge equivalent states and obtain an equivalent automaton that only has one state per residual.

In this case, the full progress consistency condition only applies to the preorder $\le_0$ corresponding to the inclusion of residuals, and therefore $\mathcal{A}$ is fully progress consistent if and only if:

$$\text{For all } u, w \in \Sigma^*,\ u^{-1}W \subsetneq (uw)^{-1}W \implies uw^\omega \in W, \qquad (1)$$

where $W = \mathcal{L}(\mathcal{A})$. An objective W satisfying property (1) is called *progress consistent*.

Example 4.1 (Non-progress consistent objective). Consider the objective

$$W = \{w \in \Sigma^\omega \mid w \text{ contains the factor } aa\}.$$

Its three residuals are totally ordered by inclusion:

$$\varepsilon^{-1}W \subseteq a^{-1}W \subseteq (aa)^{-1}W.$$

However, it is not progress consistent, as by taking $u = \varepsilon$ and $w = ba$ we have $u^{-1}W \subsetneq (uw)^{-1}W$, but $(ba)^{\omega} \notin W$. The objective W is not positional, as Eve wins the game in which she can alternate between two self-loops labelled 'ba' and 'ab' (by producing $(baab)^{\omega}$), but she cannot win positionally.

All in all, we obtain the following characterisation (first stated in [7, Theorem 10]).

Proposition 4.2 (Positionality for Büchi recognisable objectives [7]). *Let $W \subseteq \Sigma^{\omega}$ be a Büchi recognisable language. Then, W is positional if and only if:*

- *$\mathrm{Res}(W)$ is totally ordered by inclusion,*
- *W is progress consistent, and*
- *W can be recognised by a Büchi automaton having a single state per residual of W.*

Example 4.3 (Appearing in [7, Example 7]). In Figure 3 we show a Büchi automaton $\mathcal{A}$ recognising the objective W of words that either contain the factor aa, or contain letter a infinitely often. This objective has three different residuals, ordered by inclusion:

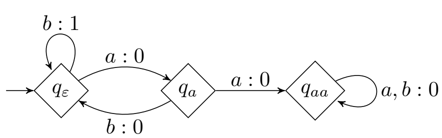


Figure 3: Büchi automaton recognising the objective $W = \mathtt{Inf}(a) \cup \mathrm{Reach}(aa)$.

$\varepsilon^{-1}W \subseteq a^{-1}W \subseteq (aa)^{-1}W$. It is easy to verify that W is progress consistent, and $\mathcal{A}$ has a state per residual, so by Proposition 4.2, W is positional.

We devote the rest of the subsection to explain the proof of the necessity of these conditions for positionality. We first prove necessity of the two first conditions (which hold for any objective). Obtaining the third condition is more technical and specific to the case of Büchi recognisable languages; we first prove it in the case of prefix-independent languages, and then show how to generalise it to any Büchi recognisable language.

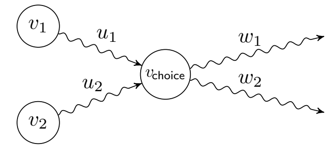
4.1.2 Total order on residuals and progress consistency.

Lemma 4.4 (Total order of residuals). *If an objective $W \subseteq \Sigma^{\omega}$ is positional, then $\mathrm{Res}(W)$ is totally ordered by inclusion.*

Proof. Suppose that W has two incomparable residuals, ${u_1}^{-1}W$ and ${u_2}^{-1}W$. Take $w_1 \in u_1^{-1}W \setminus u_2^{-1}W$ and $w_2 \in u_2^{-1}W \setminus u_1^{-1}W$. Stated differently, we have:

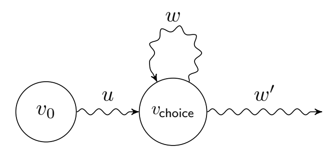
$u_1 w_1 \in W, \quad u_1 w_2 \notin W,$
$u_2 w_1 \notin W, \quad u_2 w_2 \in W.$

We conclude with a contradicion: in the Eve-game depicted above, she wins but not positionally. □



Lemma 4.5 (Necessity of progress consistency). *Any positional objective is progress consistent.*

Proof. Let W be an objective that is not progress consistent, that is, there are $u, w \in \Sigma^*$ such that $u^{-1}W \subsetneq (uw)^{-1}W$ and $uw^{\omega} \notin W$. Let $w' \in (uw)^{-1}W \setminus u^{-1}W$.



In the Eve-game depicted above, she wins from vertex v_0 by producing the play $v_0 \overset{u}{\rightsquigarrow} v_{\text{choice}} \overset{w}{\rightsquigarrow} v_{\text{choice}} \overset{w'}{\rightsquigarrow}$. However, she cannot win positionally from v_0 since positional strategies produce either uw^{ω} or uw', both of which are losing. □

4.1.3 Prefix-independent case: existence of super letters.

Let $\mathcal{A}$ be a deterministic Büchi automaton recognising a prefix-independent objective W. Then, W can be recognised by an automaton with a single state if and only if it is of the form

$$\mathrm{Buchi}(B) = \{w \in \Sigma^{\omega} \mid \text{ letters of } B \text{ appear infinitely often in } w\},$$

for some $B \subseteq \Sigma$. We prove that if W is positional, this is necessarily the case.

We say that $u \in \Sigma^+$ is a *super word* (for W) if every $w \in \Sigma^{\omega}$ containing u as a factor infinitely often belongs to W. If u is a letter, we say that it is a *super letter*. Let $B_W \subseteq \Sigma$ be the set of super letters for W. It is clear that $\mathrm{Buchi}(B_W) \subseteq W$. We will show that if W is positional this is in fact an equality.

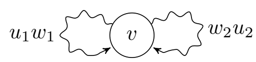
Lemma 4.6 (Existence of super letters). *A non-empty prefix-independent positional Büchi recognisable objective W admits a super letter.*

One may easily conclude using this lemma. Indeed, the restriction of W to non-super letters is a prefix-independent positional Büchi recognisable objective which contains no super letter. Thus, Lemma 4.6 tells us that it is empty, and therefore $W = \mathrm{Buchi}(B_W)$.

Proof sketch for Lemma 4.6. First, we show that if W is non-empty it admits some super word (this crucially relies on the fact that W is recognised by a Büchi automaton $\mathcal{A}$). From every state q in $\mathcal{A}$, we can find a word $w_q \in \Sigma^*$ such that $q \overset{w_q:0}{\rightsquigarrow}$ visits some Büchi transition. Concatenating all these words in a suitable way yields a word $w \in \Sigma^*$ satisfying that it will visit a Büchi transition from any state of the automaton; w is necessarily a super word.

Then, we show (using positionality) that super words can be chopped into smaller super words: if $w = w_1 w_2 \in \Sigma^+$ is a super word, then either w_1 or w_2 is a super word. This allows us to conclude by repeatedly chopping a super word until obtaining a super letter. The proof of this chopping lemma relies in a *normal form* for Büchi automata: we can assume that any path $q \xrightarrow{w:1} p$ not visiting a Büchi transition can be completed into a returning path $p \xrightarrow{w':1} q$ not visiting a Büchi transition.

Suppose by contradiction that neither w_1 nor w_2 are super words. Then, there are states q_1 and q_2 such that $q_1 \xrightarrow{w_1:1} q_1'$ and $q_2 \xrightarrow{w_2:1} q_2'$. By normality, we obtain returning paths $q_1' \xrightarrow{u_1:1} q_1$ and $q_2' \xrightarrow{u_2:1} q_2$. Therefore, $(w_1u_1)^\omega \notin W$ and $(w_2u_2)^\omega \notin W$.



Then in the Eve-game above, Eve can win by producing the output $(u_1w_1w_2u_2)^\omega$, since w_1w_2 is a super word, but she cannot win positionally. □

4.1.4 General case: Uniformity of 0*-transitions.* Our objective is to derive the following result, from which Proposition 4.2 easily follows.

Lemma 4.7 (Uniformity of 0-transitions). *Let $W \subseteq \Sigma^\omega$ be a positional Büchi recognisable objective. There is a deterministic Büchi automaton $\mathcal{A}$ recognising W such that for every pair $q \sim_{\mathcal{A}} q'$ of equivalent states and for every letter a, transition $q \xrightarrow{a}$ produces priority 0 if and only if transition $q' \xrightarrow{a}$ produces priority 0.*

One of the main ideas to prove this lemma is to reduce to the prefix-independent case, for which we introduce the *localisations of W to a residual*, as defined next.

For each residual $u^{-1}W$, we define the *local alphabet at $u^{-1}W$* to be:

$$\Sigma_{[u]} = \{w \in \Sigma^+ \mid u^{-1}W = (uw)^{-1}W\}.$$

Seeing words in $\Sigma_{[u]}^\omega$ as words in Σ^ω, define the *localisation of W to $u^{-1}W$* to be the objective

$$W_{[u]} = \{w \in \Sigma_{[u]}^\omega \mid uw \in W\}.$$

It is easy to check that these definitions do not depend on the representative u.

That is, $\Sigma_{[u]}$ corresponds to the set of words connecting states corresponding to $u^{-1}W$ in $\mathcal{A}$, and $W_{[u]}$ is the concatenation of those words that produce an accepting run.

These languages allow us to reduce to the prefix-independent case, as it is easy to check that, for every residual $u^{-1}W$:

- $W_{[u]}$ is a prefix-independent Büchi recognisable objective.
- If W is positional, so is $W_{[u]}$.

Therefore, if W is positional, there is a subset $B_{W_{[u]}} \subseteq \Sigma_{[u]}$ such that $W_{[u]}$ is exactly the set of words $w \in \Sigma_{[u]}^\omega$ containing infinitely often factors in $B_{W_{[u]}}$.

Then, to obtain Lemma 4.7, the idea is to show that we can simplify any Büchi automaton recognising W so that for all states q corresponding to a residual $u^{-1}W$, a transition $q \xrightarrow{a:0}$ produces priority 0 if and only if all words in $\Sigma_{[u]}$ starting by a are in $B_{W_{[u]}}$. This step is rather technical, we refer to Section 4.3 in the full version [20]. for details.

4.2 CoBüchi recognisable objectives

We now consider a language W recognised by a deterministic coBüchi automaton $\mathcal{A}$. For simplicity, we will focus on the case where we further assume that W is prefix-independent.

We first introduce some vocabulary concerning coBüchi automata. We say that a path $q \xrightarrow{w} q'$ in a coBüchi automaton $\mathcal{A}$ is *safe* if no coBüchi transition appears on it. A *safe component* of $\mathcal{A}$ is a strongly connected component of the automaton obtained by removing from $\mathcal{A}$ all coBüchi transitions.

We define the *safe language of a state q* as:

$$\mathsf{Safe}_{<2}(q) = \{w \in \Sigma^\omega \mid \text{ there is a safe run } q \xrightarrow{w:2} \}.$$

We assume that automata are in *normal form*, that is, transitions between different safe components are coBüchi transitions. Equivalently, any safe path can be closed in a safe cycle, that is, if $q \xrightarrow{w:2} p$ then there exists a path $p \xrightarrow{w':2} q$. Note that any automaton can be put in normal form in polynomial time [14].

Proposition 4.8 (Positionality for prefix-independent coBüchi recognisable objectives). *A prefix-independent coBüchi recognisable objective W is positional if and only if it can be recognised by a deterministic coBüchi automaton satisfying that within each safe component, states are totally ordered by inclusion of safe languages.*

Example 4.9. Consider the coBüchi automaton from Figure 4, which recognises the prefix-independent language W of words that contain either finitely often the factor ac, or finitely often the factor bb, over the alphabet $\Sigma = \{a, b, c\}$.

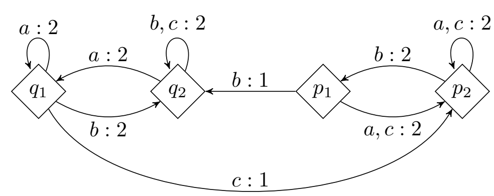


Figure 4: Deterministic coBüchi automaton recognising objective $W = \mathtt{Fin}(ac) \cup \mathtt{Fin}(bb)$ from Example 4.9.

This automaton has two safe components: $S_1 = \{q_1, q_2\}$ and $S_2 = \{p_1, p_2\}$. The states of each component are totally ordered by inclusion of safe languages, as we have $\mathsf{Safe}_{<2}(q_1) \subseteq \mathsf{Safe}_{<2}(q_2)$ and $\mathsf{Safe}_{<2}(p_1) \subseteq \mathsf{Safe}_{<2}(p_2)$. Therefore, it satisfies the hypothesis from Proposition 4.8, so W is positional.

Remark 4.10. Kopczyński [32] proposed a class of so-called *monotonic automata* over finite words, and showed that if $L \subseteq \Sigma^*$ is recognised by such an automaton, then the (prefix-independent) objective $\Sigma^\omega \setminus L^\omega$ is positional [32, Prop 6.6]. It turns out that these correspond exactly to the objectives characterised in Proposition 4.8.

4.2.1 Signature automata: The coBüchi case. We first comment how the previous characterisation matches the definition of signature automata in this case. Let $\mathcal{A}$ be a deterministic coBüchi automaton recognising a prefix-independent language such that within each safe component, states are totally ordered by inclusion of safe languages. We can define $\leq_0$ to be the trivial relation (all states are $\sim_0$-equivalent). We fix an arbitrary order $S_1 < S_2 < \cdots < S_k$ on the safe components of $\mathcal{A}$ and let $q <_1 p$ if q is in a smaller safe component for this order. Finally, for two states q, p in the same safe component, we let $q \leq_2 p$ if $\mathsf{Safe}_{<2}(q) \subseteq \mathsf{Safe}_{<2}(p)$. We claim that the obtained automaton is a fully progress consistent signature automaton:

I) Since W is prefix-independent, $\leq_0$ corresponds to inclusion of residuals.
II) As $q \sim_2 p$ implies $\mathsf{Safe}_{<2}(q) = \mathsf{Safe}_{<2}(p)$, transitions $q \xrightarrow{a} q'$ and $p \xrightarrow{a} p'$ produce the same priority.
III) As $q <_1 p$ implies that q and p are in different safe components, and $\mathcal{A}$ is assumed in normal form, there is no safe path from p to q.
IV) Follows from monotonicity of safe languages: if $\mathsf{Safe}_{<2}(q) \subseteq \mathsf{Safe}_{<2}(q')$ and $q \xrightarrow{a:2} p$, then $q' \xrightarrow{a:2} p'$ and it must be the case that $\mathsf{Safe}_{<2}(p) \subseteq \mathsf{Safe}_{<2}(p')$.

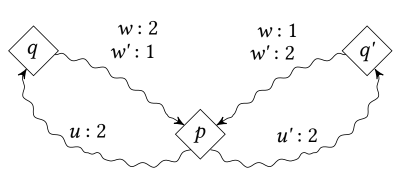
Let us check full progress consistency. As all states are $\sim_0$-equivalent, no strict inequality $q <_0 p$ occurs. Assume now that $q <_2 p$ (that is, $\mathsf{Safe}_{<2}(q) \subsetneq \mathsf{Safe}_{<2}(p)$) and $q \overset{w:2}{\rightsquigarrow} p$. Then, by inclusion of safe languages, $p \overset{w:2}{\rightsquigarrow} p_2$ is also a safe path, and by monotonicity of safe languages, $\mathsf{Safe}_{<2}(p) \subseteq \mathsf{Safe}_{<2}(p_2)$. We conclude by induction that the run over w^ω from q is safe, and therefore accepting.

4.2.2 Obtaining total orders in safe components. We now explain how we prove the necessity of the conditions from Proposition 4.8. We say that a coBüchi automaton has the *synchronising words property* if, for all q, q', p in the same safe component such that $\mathsf{Safe}_{<2}(q) \nsubseteq \mathsf{Safe}_{<2}(q')$, there is w such that $q \overset{w:2}{\rightsquigarrow} p$ and $q' \overset{w:1}{\rightsquigarrow} p$.

Lemma 4.11. *Let $\mathcal{A}$ be a deterministic coBüchi automaton with the synchronising words property recognising a prefix-independent positional languages. Then, the states of each safe component of $\mathcal{A}$ are totally preordered by inclusion of safe languages.*

Proof. Suppose by contradiction that there are two states q, q' in a same safe component S such that $\mathsf{Safe}_{<2}(q) \nsubseteq \mathsf{Safe}_{<2}(q')$ and $\mathsf{Safe}_{<2}(q') \nsubseteq \mathsf{Safe}_{<2}(q)$. Let p be any state in S, and let $u, u' \in \Sigma^*$ be such that $p \overset{u:2}{\rightsquigarrow} q$ and $p \overset{u':2}{\rightsquigarrow} q'$ (which exist by definition of safe component). By the synchronising words property, there are $w, w' \in \Sigma^\omega$ such that:

$q \overset{w:2}{\rightsquigarrow} p$, $q \overset{w':1}{\rightsquigarrow} p$,
$q' \overset{w:1}{\rightsquigarrow} p$, $q' \overset{w':2}{\rightsquigarrow} p$, as in the figure below.



We obtain that:

- $(w'u)^\omega \notin W$,
- $(wu')^\omega \notin W$,
- $(wu'w'u)^\omega \in W$.

Thus consider the game in which Eve controls a vertex with two self loops labelled $w'u$ and wu'; she can win by alternating both loops, but fails to win positionally. ◻

The remaining issue is, of course, that not all coBüchi automata have the synchronising words property. However, there is an equivalent automaton satisfying (a version of) this property: the minimal history-deterministic coBüchi automaton of Abu Radi and Kupferman [1]. However, this detour requires introducing non-determinism, and then proving that in the case of positional languages, determinism can be recovered over this minimal automaton. We refer to Section 4.4 in the full version [20] for details.

5 PRESENTATION OF THE PROOF

5.1 From positionality to signature automata

We now provide a general overview for the proof of the implication (1) $\implies$ (2) in Theorem 3.1. We refer to Section 5.2 in the full version [20] for details.

The main idea is, given a deterministic parity automaton $\mathcal{A}$ recognising a positional objective W, to successively apply the transformations discussed for the Büchi and coBüchi cases in Section 4 to define the preorders making $\mathcal{A}$ a signature automaton layer-by-layer. The base case of this recursion consists in showing that the preorder $\leq_0$ given by the inclusion of residuals is total, and ensuring Item (II) for this preorder (uniformity of 0-transitions). For the recursion step, we suppose that we have defined preorders $\leq_0, \leq_1, \ldots, \leq_{x-2}$, for x even, satisfying the properties from the definition of a signature automaton. To define preorders $\leq_{x-1}$ and $\leq_x$ we apply the following procedure:

i) **Definition of $\leq_{x-1}$.** We let $q \sim_{x-1} p$ if $q \sim_{x-2} p$ and these states are in a same $(<x)$-safe component of $\mathcal{A}$ (SCC of the restriction of $\mathcal{A}$ to transitions with priority $\geq x$). We order $\sim_{x-1}$-classes arbitrarily to define $\leq_{x-1}$. This ensures Item (III) from the definition of signature automaton.
ii) **Canonisation of automata.** We generalise the minimisation procedure for HD coBüchi automata from [1] to odd levels of parity automata. In this way, we obtain an equivalent automaton satisfying (a generalisation of) the synchronising words property at level $x-1$, while preserving the previous structure of a signature automaton. However, this is done at the cost of introducing some non-determinism.
iii) **Total order in safe components.** We prove that the states of each $(<x)$-safe component are totally ordered by inclusion of $(<x)$-safe languages. This shows that the preorder $\leq_x$ given by the inclusion of safe languages is total. Moreover, this order satisfies the monotonicity properties from Item (IV).
iv) **Re-determinisation.** We determinise automaton $\mathcal{A}$, while preserving previously obtained properties. For this, the fact that $\leq_x$ is total is key.
v) **Uniformity of x-transitions.** Finally, we show that we can trim the automaton $\mathcal{A}$ so that it satisfies Item (II) ($q \sim_x p$ implies $q \xrightarrow{a:y} \iff p \xrightarrow{a:y}$, for $y \leq x$).

This establishes that an ω-regular positional objective W can be recognised by a deterministic signature automaton. Finally, we prove that such an automaton must necessarily be fully progress consistent.

5.2 Back to positionality through ε-complete automata

We now want to prove that any objective recognised by a fully progress consistent signature automaton is indeed positional (implication (2) $\implies$ (5) in Theorem 3.1). We achieve this using the intermediate model of ε-complete automata, which also provide further intuition on what are the automata recognising positional languages.

5.2.1 From signature to ε-complete automata.

Proposition 5.1. *Any deterministic fully progress consistent signature automaton is ε-completable.*

Proof sketch. Let $\mathcal{A}$ be a deterministic fully progress consistent signature automaton with nested preorders $\leq_0, \leq_1, \ldots, \leq_d$ and let $W = \mathcal{L}(\mathcal{A})$. Consider the automaton $\mathcal{A}'$ obtained from $\mathcal{A}$ by adding, for all even priorities $x \in [0, d]$, transitions $q \xrightarrow{\varepsilon:x+1} q'$ whenever $q' \leq_x q$ and $q \xrightarrow{\varepsilon:x} q'$ whenever $q' <_x q$.

Since by definition, $q' <_x q$ is the negation of $q \leq_x q'$, it follows immediately that $\mathcal{A}'$ is ε-complete. Moreover, as $\mathcal{A}$ is a subautomaton of $\mathcal{A}'$, the inclusion $W \subseteq \mathcal{L}(\mathcal{A}')$ is trivial. The difficulty lies in showing that $\mathcal{L}(\mathcal{A}') \subseteq W$.

The key property that allows to show this inclusion is full progress consistency. Assume that a run in $\mathcal{A}'$ takes infinitely often a transition $q \xrightarrow{\varepsilon:x} p$, for x the least even priority occurring in the run. Then, $p \leq_x q$ and there must be infinitely many factors of the run of the form $p \overset{w:\geq x}{\rightsquigarrow} q$. Full progress consistency ensures that the repetition of infinitely many factors that go upstream in this way produce a word in W.

The formal details are quite technical, and make use of the rest of the properties of a signature automaton to combine the parts of the run occurring between different "ε-jumps". We refer the reader to Section 5.3 in the full version [20] for details. ◻

5.2.2 From ε-completable automata to positionality. Let $\mathcal{A}$ be a deterministic automaton that admits an ε-completion $\mathcal{A}'$ (which, by definition satisfies $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$). Then, the resulting automaton $\mathcal{A}'$ is history-deterministic (it is in fact what is sometimes called determinisable by pruning): Given an input word $w \in \Sigma^\omega$, we can follow the run on $\mathcal{A}$, disregarding ε-transitions; if $w \in \mathcal{L}(\mathcal{A}')$, this run will be accepting. We claim that this is a sufficient hypothesis to obtain positionality of $W = \mathcal{L}(\mathcal{A})$.

Proposition 5.2. *Let $\mathcal{A}$ be a history-deterministic ε-complete automaton. Then, $\mathcal{L}(\mathcal{A})$ is positional (over all games).*

We prove this result by constructing a monotone universal graph out of $\mathcal{A}$, and then concluding thanks to Proposition 2.2. To do so, we rely on an already existing universal graph U_{parity} for the parity condition [22, Section 6] (this graph can be seen as a reformulation of previous proofs of positionality of the parity condition [27, 29, 47]). The graph U_{parity} has the shape of a tree; nodes are the leaves, and each even priority x defines a preorder $\leq_x$ such that $\leq_{x+2}$ refines $\leq_x$.

Given an HD ε-complete automaton $\mathcal{A}$ recognising W, we build an ordered graph U_W by taking an interleaved product of U_{parity} with $\mathcal{A}$. Universality is easily proved as a consequence of history-determinism, while monotonicity follows from the fact that $\mathcal{A}$ is ε-complete.

5.3 Decision methods

We now establish decidability of positionality of ω-regular languages in polynomial time, as stated in Theorem 3.3. We propose two conceptually different decision procedures. We refer to Section 6 in the full version [20] for details.

5.3.1 Procedure 1: Recursive decomposition. The first method consists in, given a deterministic parity automaton $\mathcal{A}$, applying the construction described in Section 5.1 to attempt to build a deterministic signature automaton. All the steps mentioned in Section 5.1 can be carried out in polynomial time, and this will either end up with a failure, indicating that W is not positional, or return a deterministic signature automaton. If such an automaton is successfully obtained, it suffices to check full progress consistency, which can also be done in polynomial time (generalising methods from [7, Lemma 25]).

5.3.2 Procedure 2: ε-completion. The second procedure is simpler to describe: we give a direct proof that any (non-deterministic) automata recognising a positional language is ε-completable. However, the proof itself relies on Theorem 3.1, and more specifically its consequence Theorem 3.5 about the closure under union.

Theorem 5.3. *Let $\mathcal{A}$ be a non-deterministic parity automaton recognising a positional language W. Then for each pair of states $q, q' \in Q$, and for each even priority x, one may add one of the transitions*

$$q \xrightarrow{\varepsilon:x} q' \qquad \text{or} \qquad q' \xrightarrow{\varepsilon:x+1} q$$

without augmenting the language.

The proof is based on the choice game technique introduced in [22, 40].

Proof sketch. We build a game $\mathcal{G}$ based on the automaton $\mathcal{A}$. Roughly speaking, Adam controls a run in $\mathcal{A}$, however, whenever the run visits q or q', Eve has the choice of going either to q, producing priority $x + 1$, or to q', producing priority x. The game $\mathcal{G}$ uses as objective W' a disjunction of W with some parity conditions; it is positional (which is crucial to this proof), thanks to Theorem 3.5. Eve can then ensure to win simply by simulating a run in $\mathcal{A}$ (this strategy is not positional). Positionality of W' implies the existence of a positional strategy, which either always chooses q or q'. In the first case, we show that one can add transition $q' \xrightarrow{\varepsilon:x+1} q$ to $\mathcal{A}$, and in the latter, that transition $q \xrightarrow{\varepsilon:x} q'$ can be added. ◻

As a corollary, we obtain that any non-deterministic parity automaton recognising a positional language is ε-completable (Proposition 3.2).

We show how to obtain decidability of positionality in polynomial time from this result. Let $\mathcal{A}_0$ be a deterministic parity automaton recognising a language W. We build a completion of $\mathcal{A}_0$ as follows. At each step, pick a pair of states (q, q') such that neither

$q \xrightarrow{\varepsilon:x} q'$ nor $q' \xrightarrow{\varepsilon:x+1} q$ belongs to the current automaton. Then try to add one of these transitions, and see if the language increases (checking whether $\mathcal{L}(\mathcal{A}) \subseteq W$ can be done in polynomial time since W is recognised by the deterministic automaton $\mathcal{A}_0$ [21]). If the language does not increase for one of the two transitions, then proceed to the next pair of states; otherwise conclude that W is not positional thanks to Theorem 5.3.

After polynomially-many steps, we obtain an automaton such that for each pair of states (q, q') and for each even x, either $q \xrightarrow{\varepsilon:x} q'$ or $q' \xrightarrow{\varepsilon:x+1} q$. Now one may additionally add transitions $q \xrightarrow{\varepsilon:y'} q'$ whenever $q \xrightarrow{\varepsilon:y} q'$ and $y' \preccurlyeq y$ (where $d+1 \preccurlyeq \ldots \preccurlyeq 3 \preccurlyeq 1 \preccurlyeq 0 \preccurlyeq 2 \preccurlyeq \ldots \preccurlyeq d$), and close the relations $\xrightarrow{\varepsilon:y}$ by transitivity without increasing the language. It is an easy check that the obtained automaton is ε-complete which implies that W is positional thanks to Theorem 3.1.

6 FURTHER RESULTS AND FUTURE WORK

We have provided a complete characterisation of positionality for ω-regular languages, based on two families of parity automata. As a consequence, we were able to provide polynomial-time procedures for checking positionality and solve many open question about positionality of ω-regular languages.

However, one drawback of our approach is its conceptual complexity and its exclusive focus on automata. In this regard, it would be interesting to find a characterisation centered on the language-theoretical properties of the objectives (closer to the existing one for Büchi recognisable languages, see Proposition 4.2).

On the technical side, proving Theorem 3.1 required introducing new notions concerning congruences for parity automata, as well as different transformations and decomposition techniques. We expect that these techniques will prove valuable in the study of parity automata in various contexts, particularly for obtaining canonical representations for ω-regular languages. In particular, we extended the methods introduced by Abu Radi and Kupferman [1] for minimising history-deterministic coBüchi automata to other classes of parity automata. We see the use of history-deterministic automata in this work as further evidence of the usefulness of this model in the theoretical study of ω-regular languages.

We now discuss some further results obtained from our characterisation, and future lines of work.

6.1 Bipositionality

Using our results, we can extend the characterisation of bipositionality from Colcombet and Niwiński to non-prefix-independent objectives (without ω-regularity assumptions). (See Section 7 in the full version [20].)

Theorem 6.1 (Characterisation of bipositionality). *An objective $W \subseteq \Sigma^\omega$ is bipositional (over all games) if and only if:*

1. *W has a finite number of residuals, totally ordered by inclusion, and*
2. *Both W and $\Sigma^\omega \setminus W$ are progress consistent, and*
3. *W can be recognised by a parity automaton with one state per residual.*

We note that, in particular, a bipositional objective is necessarily ω-regular. A more general result was obtained by Bouyer, Randour and Vandenhove: ω-regular objectives are exactly those objectives for which both players can play with finite arena-independent memory [10].

This characterisation only holds for infinite game graphs, as there are non ω-regular objectives that are bipositional over finite games, as, for example, energy objectives [8] and their generalisation [34]. However, we deduce from Theorem 3.4 that in the case of ω-regular objectives these conditions do also characterise bipositionality over finite games. We also obtain from Theorem 3.4 1-to-2 player and finite-to-infinite lifts for bipositionality.

Corollary 6.2 (1-to-2 player and finite-to-infinite lift). *An objective $W \subseteq \Sigma^\omega$ is bipositional (over all games) if and only if it is bipositional over Eve-games and Adam-games.*

An ω-regular objective $W \subseteq \Sigma^\omega$ is bipositional (over all games) if and only if it is bipositional over finite Eve-games and finite Adam-games.

We note that a 1-to-2 player lift was obtained for objectives that are bipositional over finite game graphs by Gimbert and Zielonka [28, 48] (even in the more general setting of qualitative objectives). However, their proof consisted in an induction over the size of the game graph, so it does not generalise to infinite games. Indeed, as remarked above, bipositionality over finite and infinite graphs behaves in a completely different manner. In this respect, Corollary 6.2 and the result of Gimbert and Zielonka are incomparable.

6.2 Positionality of objectives defined by topological properties

A potential research direction consists in investigating positionality for broader classes of objectives, namely those defined by topological properties, such as those corresponding to a given class in the Borel hierarchy.

We initiate this path in the full version of this paper (see [20, Section 8]) by characterising positionality for *open* and *closed objectives*, that is, those of the form $W_L = \{w \in \Sigma^\omega \mid w \text{ contains a prefix in } L\}$ for a given $L \subseteq \Sigma^*$ and their complements, respectively. We state these characterisations next.

An objective W is *reset-stable* if for any sequence of finite words $u_0, u_1, \ldots$ and any sequence of residuals $s_0^{-1}W, s_1^{-1}W, \ldots$ of W such that for all i, $s_{i+1}^{-1}W < (s_i u_i)^{-1}W$, we have $u_0 u_1 \cdots \in s_0^{-1}W$. This is a generalisation of progress consistency which is better suited to objectives with infinitely many residuals.

Theorem 6.3. *A closed objective is positional if and only if its residuals are well-ordered by inclusion. An open objective is positional if and only if its residuals are well-ordered by inclusion and it is reset-stable.*

As some interesting corollaries, we obtain that the 1-to-2-players lift and the closure under addition of a neutral letter hold for these classes of objectives. Kopczyński's conjecture trivially holds here, as the only open prefix-independent languages are $\emptyset$ and Σ^ω.

Corollary 6.4 (1-to-2-player lift and neutral letters for open and closed objectives). *Let $W \subseteq \Sigma^\omega$ be an open or closed objective. If W is positional over ε-free Eve-games, then W^ε is positional over all game graphs.*

Natural continuations would be objectives in Σ_2^0 and Π_2^0, which are, respectively, unions of closed objectives and intersections of open objectives, or, more generally, to $\mathcal{BC}(\Sigma_2^0)$ (boolean combinations of objectives in Σ_2^0).[6] Recently, Ohlmann and Skrzypczak [42] proposed a characterisation of positionality for prefix-independent objectives in Σ_2^0; however the cases of non-prefix-independent Σ_2^0 objectives or Π_2^0 objectives are open. We hope to be able to give characterisations for these classes, as some of the constructions introduced in this work seem to generalise to automata with infinite states. We believe that some properties of positional ω-regular objectives could be lifted to $\mathcal{BC}(\Sigma_2^0)$. In particular, we conjecture that the 1-to-2 player lift holds for $\mathcal{BC}(\Sigma_2^0)$-objectives. Other important steps forward in our understanding of positionality would involve proving Kopczyński's conjecture or Ohlmann's neutral letter conjecture for the class $\mathcal{BC}(\Sigma_2^0)$.

6.3 Memory requirements

An orthogonal research direction would be to maintain the focus on ω-regular languages, but attempt to characterise their memory requirements rather than just their positionality. A notable effort has already been made in this direction [9, 10, 15, 18, 23, 25], however, only characterisations for fairly simple classes of languages are known (Muller [15, 18, 25] and closed languages [9, 23]). To this day, deciding whether winning in games with a given ω-regular objective (even an open one) is possible with $\leq k$ states of memory is not known to be decidable.

The generalisation of the theory of monotone universal graphs to characterise memory, presented in [19], offers a promising tool to establish tight upper bounds on memory requirements, which could be valuable to overcome impediments in the advancement of the study of memory for ω-regular objectives.

ACKNOWLEDGMENTS

We want to thank Hugo Gimbert, Damian Niwiński and Pierre Vandenhove for interesting discussions on the subject.
Antonio Casares: supported by the Polish National Science Centre (NCN) grant "Polynomial finite state computation" (2022/46/A/ST6/00072).

REFERENCES

[1] Bader Abu Radi and Orna Kupferman. Minimization and canonization of GFG transition-based automata. *Log. Methods Comput. Sci.*, 18(3), 2022. doi:10.46298/lmcs-18(3:16)2022.
[2] André Arnold. A syntactic congruence for rational omega-language. *Theor. Comput. Sci.*, 39:333–335, 1985. doi:10.1016/0304-3975(85)90148-3.
[3] 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.
[4] 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 International Publishing, 2018. doi:10.1007/978-3-319-10575-8_27.
[5] León Bohn and Christof Löding. Constructing deterministic parity automata from positive and negative examples. *CoRR*, abs/2302.11043, 2023. doi:10.48550/arXiv.2302.11043.
[6] Udi Boker and Karoliina Lehtinen. When a little nondeterminism goes a long way: An introduction to history-determinism. *ACM SIGLOG News*, 10(1):24–51, 2023. doi:10.1145/3584676.3584682.
[7] Patricia Bouyer, Antonio Casares, Mickael Randour, and Pierre Vandenhove. Half-positional objectives recognized by deterministic Büchi automata. In *CONCUR*, volume 243, pages 20:1–20:18, 2022. doi:10.4230/LIPIcs.CONCUR.2022.20.
[8] Patricia Bouyer, Ulrich Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jirí Srba. Infinite runs in weighted timed automata with energy constraints. In *FORMATS*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47, 2008. doi:10.1007/978-3-540-85778-5_4.
[9] Patricia Bouyer, Nathanaël Fijalkow, Mickael Randour, and Pierre Vandenhove. How to play optimally for regular objectives? In *ICALP*, volume 261 of *LIPIcs*, pages 118:1–118:18, 2023. doi:10.4230/LIPIcs.ICALP.2023.118.
[10] Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs. *TheoretiCS*, 2, 2023. doi:10.46298/theoretics.23.1.
[11] Patricia Bouyer, Stéphane Le Roux, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. Games where you can play optimally with arena-independent finite memory. In *CONCUR*, volume 171, pages 24:1–24:22, 2020. doi:10.4230/LIPIcs.CONCUR.2020.24.
[12] J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969. URL: http://www.jstor.org/stable/1994916.
[13] 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.
[14] Olivier Carton and Ramón Maceiras. Computing the Rabin index of a parity automaton. *RAIRO*, pages 495–506, 1999. doi:10.1051/ita:1999129.
[15] 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.
[16] Antonio Casares. *Structural properties of automata over infinite words and memory for games (Propriétés structurelles des automates sur les mots infinis et mémoire pour les jeux).* Phd thesis, Université de Bordeaux, France, 2023. URL: https://theses.hal.science/tel-04314678.
[17] Antonio Casares, Thomas Colcombet, Nathanaël Fijalkow, and Karoliina Lehtinen. From Muller to Parity and Rabin Automata: Optimal Transformations Preserving (History) Determinism. *TheoretiCS*, Volume 3, April 2024. doi:10.46298/theoretics.24.12.
[18] 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 *ICALP*, volume 229, pages 117:1–117:20, 2022. doi:10.4230/LIPIcs.ICALP.2022.117.
[19] Antonio Casares and Pierre Ohlmann. Characterising memory in infinite games. In *ICALP*, volume 261 of *LIPIcs*, pages 122:1–122:18, 2023. doi:10.4230/LIPIcs.ICALP.2023.122.
[20] Antonio Casares and Pierre Ohlmann. Positional ω-regular languages. *CoRR*, abs/2401.15384, 2024. arXiv:2401.15384, doi:10.48550/ARXIV.2401.15384.
[21] 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.
[22] Thomas Colcombet, Nathanaël Fijalkow, Pawel Gawrychowski, and Pierre Ohlmann. The theory of universal graphs for infinite duration games. *Log. Methods Comput. Sci.*, 18(3), 2022. doi:10.46298/lmcs-18(3:29)2022.
[23] Thomas Colcombet, Nathanaël Fijalkow, and Florian Horn. Playing safe. In *FSTTCS*, volume 29, pages 379–390, 2014. doi:10.4230/LIPIcs.FSTTCS.2014.379.
[24] 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.
[25] Stefan Dziembowski, Marcin Jurdziński, and Igor Walukiewicz. How much memory is needed to win infinite games? In *LICS*, pages 99–110, 1997. doi:10.1109/LICS.1997.614939.
[26] Rüdiger Ehlers and Sven Schewe. Natural colors of infinite words. In *FSTTCS*, volume 250, pages 36:1–36:17, 2022. doi:10.4230/LIPIcs.FSTTCS.2022.36.
[27] E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *FOCS*, pages 368–377, 1991. doi:10.1109/SFCS.1991.185392.
[28] Hugo Gimbert and Wieslaw Zielonka. Games where you can play optimally without any memory. In *CONCUR*, volume 3653, pages 428–442, 2005. doi:10.1007/11539452_33.
[29] Marcin Jurdzinski. Small progress measures for solving parity games. In *STACS*, volume 1770 of *Lecture Notes in Computer Science*, pages 290–301, 2000. doi:10.1007/3-540-46541-3_24.
[30] Nils Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. *Annals of Pure and Applied Logic*, 69(2):243–268, 1994. doi:10.1016/0168-0072(94)90086-8.

[6] These classes admit automata-oriented definitions: Σ_2^0 are the objectives recognised by deterministic infinite Büchi automata, and Π_2^0 those recognised by infinite coBüchi automata [44]. $\mathcal{BC}(\Sigma_2^0)$ coincides with the class of objectives recognised by infinite deterministic parity automata [44] (this is a strict subclass of $\Delta_3^0 = \Sigma_3^0 \cap \Pi_3^0$).

[31] Eryk Kopczyński. Omega-regular half-positional winning conditions. In *CSL*, volume 4646 of *Lecture Notes in Computer Science*, pages 41–53, 2007. doi:10.1007/978-3-540-74915-8_7.

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

[33] Alexander Kozachinskiy. One-to-two-player lifting for mildly growing memory. In *STACS*, volume 219 of *LIPIcs*, pages 43:1–43:21, 2022. doi:10.4230/LIPIcs.STACS.2022.43.

[34] Alexander Kozachinskiy. Energy games over totally ordered groups. In *CSL*, volume 288, pages 34:1–34:12, 2024. doi:10.4230/LIPICS.CSL.2024.34.

[35] Denis Kuperberg and Michał Skrzypczak. On determinisation of good-for-games automata. In *ICALP*, pages 299–310, 2015. doi:10.1007/978-3-662-47666-6_24.

[36] Orna Kupferman. Using the past for resolving the future. *Frontiers Comput. Sci.*, 4, 2022. doi:10.3389/fcomp.2022.1114625.

[37] Oded Maler and Ludwig Staiger. On syntactic congruences for ω-languages. *Theoretical Computer Science*, 183(1):93–112, 1997. Formal Language Theory. doi:10.1016/S0304-3975(96)00312-X.

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

[39] Andrzej W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In *SCT*, pages 157–168, 1984. doi:10.1007/3-540-16066-3_15.

[40] Pierre Ohlmann. *Monotonic graphs for parity and mean-payoff games. (Graphes monotones pour jeux de parité et à paiement moyen)*. PhD thesis, Université Paris Cité, France, 2021. URL: https://tel.archives-ouvertes.fr/tel-03771185.

[41] Pierre Ohlmann. Characterizing positionality in games of infinite duration over infinite graphs. *TheoretiCS*, 2, 2023. doi:10.46298/theoretics.23.3.

[42] Pierre Ohlmann and Michal Skrzypczak. Positionality in Σ^0_2 and a completeness result. In *STACS*, volume 289, pages 54:1–54:18, 2024. doi:10.4230/LIPICS.STACS.2024.54.

[43] Bertrand Le Saëc. Saturating right congruences. *RAIRO*, 24:545–559, 1990. doi:10.1051/ita/1990240605451.

[44] Michał Skrzypczak. Topological extension of parity automata. *Information and Computation*, 228-229:16–27, 2013. doi:10.1016/j.ic.2013.06.004.

[45] Wolfgang Thomas. On the synthesis of strategies in infinite games. In *STACS*, pages 1–13, 1995. doi:10.1007/3-540-59042-0_57.

[46] Pierre Vandenhove. *Strategy complexity of zero-sum games on graphs. (Complexité des stratégies des jeux sur graphes à somme nulle)*. PhD thesis, University of Mons, Belgium, 2023. URL: https://tel.archives-ouvertes.fr/tel-04095220.

[47] Igor Walukiewicz. Pushdown processes: Games and model checking. In *CAV*, volume 1102 of *Lecture Notes in Computer Science*, pages 62–74, 1996. doi:10.1007/3-540-61474-5_58.

[48] Wieslaw Zielonka. An invitation to play. In *MFCS*, volume 3618, pages 58–70, 2005. doi:10.1007/11549345_7.