The memory of ω-regular and $\mathrm{BC}(\Sigma_2^0)$ objectives

Antonio Casares
University of Warsaw, Poland

Pierre Ohlmann
CNRS, Laboratoire d'Informatique et des Systèmes (LIS), Marseille, France

Abstract

In the context of 2-player zero-sum infinite duration games played on (potentially infinite) graphs, the memory of an objective is the smallest integer k such that in any game won by Eve, she has a strategy with $\leq k$ states of memory. For ω-regular objectives, checking whether the memory equals a given number k was not known to be decidable. In this work, we focus on objectives in $\mathrm{BC}(\mathbf{\Sigma}_2^0)$, i.e. recognised by a potentially infinite deterministic parity automaton. We provide a class of automata that recognise objectives with memory $\leq k$, leading to the following results:

- for ω-regular objectives, the memory over finite and infinite games coincides and can be computed in **NP**;
- given two objectives W_1 and W_2 in $\mathrm{BC}(\mathbf{\Sigma}_2^0)$ and assuming W_1 is prefix-independent, the memory of $W_1 \cup W_2$ is at most the product of the memories of W_1 and W_2.

Our results also apply to chromatic memory, the variant where strategies can update their memory state only depending on which colour is seen.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Infinite duration games, memory, omega-regular

Funding *Antonio Casares*: Supported by the Polish National Science Centre (NCN) grant "Polynomial finite state computation" (2022/46/A/ST6/00072).

Acknowledgements We thank Nathan Lhote for his participation in the scientific discussions which led to this paper. We also thank Pierre Vandenhove for pointing us to references concerning lifting results for memory.

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

Context: Strategy complexity in infinite duration games

We study infinite duration games on graphs in which two players, called Eve and Adam, interact by moving a token along the edges of a (potentially infinite) edge-coloured directed graph. Each vertex belongs to one player, who chooses where to move next during a play. This interaction goes on for an infinite duration, producing an infinite path in the graph. The winner is determined according to a language of infinite sequences of colours W, called the objective of the game; Eve aims to produce a path coloured by a sequence in W, while Adam tries to prevent this. This model is widespread for its use in verification and synthesis [CHVB18].

In order to achieve their goal, players use strategies, which are representations of the course of all possible plays together with instructions on how to act in each scenario. In this work, we are interested in optimal strategies for Eve, that is, strategies that guarantee a victory whenever this is possible. More precisely, we are interested in the complexity of such strategies, or in other words, in the succinctness of the representation of the space of plays.

Positionality. The simplest strategies are those that assign in advance an outgoing edge to each vertex owned by Eve, and always play along this edge, disregarding all the other features of the play. All the information required to implement such a strategy appears in the game graph itself. Objectives for which such strategies are sufficient to play optimally are called *positional* (or memoryless). Understanding positionality has been the object of a long line of research. The landmark results of Gimbert and Zielonka [GZ05] and Colcombet and Niwiński [CN06] gave a good understanding of which objectives are bi-positional, i.e. positional for both players.

More recently, Ohlmann proposed to use universal graphs as a tool for studying positionality (taking Eve's point of view) [Ohl23]. This led to many advances in the study of positionality [BCRV24, OS24], and most notably, a characterisation of positional ω-regular objectives by Casares and Ohlmann [CO24a], together with a polynomial time decision procedure (and some other important corollaries, more discussion below).

Strategies with memory. In many scenarios, playing optimally requires distinguishing plays that end in the same vertex. A seminal result of Büchi and Landweber [BL69] states that in finite games where the objective is an ω-regular language, the winner has a winning strategy that can be implemented by a finite automaton processing the edges of the game; this result was later extended to infinite game graphs by Gurevich and Harrington [GH82]. Here, the states of the automaton are interpreted as memory states of the strategy, and a natural measure of the complexity of a strategy is the number of such states. More precisely, the memory of an objective W is the minimal k such that whenever Eve wins a game with objective W, she has a winning strategy with k states of memory. For ω-regular objectives, this is always finite [BL69, GH82], while the case of positionality discussed above corresponds to memory $k = 1$.

Characterising the memory of objectives has been a recurrent research subject since the 1990s. Notable results include the characterisation for Muller objectives by Dziembowski, Jurdziński, and Walukiewicz [DJW97], or for closed objectives by Colcombet, Fijalkow and Horn [CFH14]. However, these are all rather restricted classes of ω-regular objectives. The problem of deciding the memory of ω-regular objectives has been raised in many occasions (see for instance [Kop08, Sect. 9.2], [BRV22, Sect. 4], [CFH22, Conclusions], or [BFRV23]), but prior to this work, even computing the memory of open ω-regular objectives was not known to be decidable.

Chromatic memory. In the special case where the automata implementing strategies are only allowed to read the colours on the edges of the game graph, instead of the edges themselves, we speak of chromatic memory. In his PhD thesis, Kopczyński showed that, for prefix-independent ω-regular objectives and over finite game graphs, the chromatic memory can be computed in exponential time [Kop08, Theorem 8.14]. Recently, it was shown that computing the chromatic memory of some restricted subclasses of ω-regular objectives is in fact **NP**-complete: for Muller objectives [Cas22] and for topologically open or closed objectives [BFRV23]. However, the chromatic and the unconstrained memory of objectives may differ, even exponentially [Cas22, CCL22].

Finite-to-infinite lift. In general, the memory of an objective may differ if we consider only finite game graphs or arbitrary ones (a well-known such example is the mean-payoff objective[1], or the unboundedness objective $\{w_0w_1\cdots \in \{-1,1\}^\omega \mid \forall N, \exists k, \sum_{i=0}^{k} w_i \geq N\}$). In his PhD thesis, Vandenhove conjectured that this is not the case for ω-regular objectives [Van23, Conjecture 9.1.2][2]: their memory is the same over finite and infinite games. This conjecture has recently been proved in the case of positional (memoryless) objectives, i.e., those with memory equal to one [CO24a, Theorem 3.4].

Unions of objectives. The driving question in Kopczyński's PhD thesis [Kop08] is whether prefix-independent positional objectives are closed under union, which has become known as Kopczyński's conjecture. Recently, Kozachinskiy [Koz24] disproved this conjecture, but only for positionality over finite game graphs, and using non-ω-regular objectives. In fact, the conjecture is now known to hold for ω-regular objectives [CO24a] and $\mathbf{\Sigma}_2^0$ objectives [OS24]. Casares and Ohlmann proposed a generalisation of this conjecture from positional objectives to objectives requiring memory [CO25, Conjecture 7.1] (see also [Kop08, Proposition 8.11]):

▶ **Conjecture 1** (*Generalised Kopczyński's conjecture*)**.** *Let $W_1, W_2 \subseteq \Sigma^\omega$ be two prefix-independent objectives with memory k_1 and k_2, respectively. Then $W_1 \cup W_2$ has memory at most $k_1 \cdot k_2$.*

Using the characterisation of [DJW97], it is not hard to verify that the conjecture holds for Muller objectives.

BC($\mathbf{\Sigma}_2^0$) languages. The results in this work apply not only to ω-regular languages, but to the broader class of BC($\mathbf{\Sigma}_2^0$) languages. These are boolean combinations of languages in $\mathbf{\Sigma}_2^0$ (countable unions of closed languages), or equivalently, recognised by deterministic parity automata with possibly infinitely many states. This class includes typical non-ω-regular examples such as energy or mean-payoff objectives, but also broader classes such as unambiguous ω-petri nets [FSJ$^+$22] and deterministic ω-Turing machines (Turing machines with a Muller condition).

Contributions

Our main contribution is a characterisation of BC($\mathbf{\Sigma}_2^0$) objectives with memory $\leq k$, stated in Theorem 7. It captures both the memory and the chromatic memory of objectives over infinite game graphs. The characterisation is based on the notion of k-wise ε-completable automata, which are parity automata with states partitioned in k chains, where each chain is endowed with a tight hierarchical structure encoded in the ε-transitions of the automaton.

From this characterisation, we derive the following corollaries:

1. **Decidability in NP.** Given a deterministic parity automaton $\mathcal{A}$, the memory (resp. chromatic memory) of $L(\mathcal{A})$ can be computed in **NP**.
2. **Finite-to-infinite lift.** If an ω-regular objective has memory (resp. chromatic memory) $\leq k$ over finite games, then the same is true over arbitrary games.

[1] Ohlmann and Skrzypczak recently showed that if defined as $\{w \in \mathbb{Z}^\omega \mid \limsup_k \sum_{i=0}^{k-1} w_i < 0\}$, the mean-payoff objective is even positional over infinite games [OS24]. However, the complement of this objective is only positional on finite game graphs.

[2] Formally, the conjecture is stated for arena-independent memories (a slightly different setting).

3. **Generalised Kopczyński's conjecture.** We establish (and strengthen) Conjecture 1 in the case of $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ objectives: if W_1 and W_2 are $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ objectives with memory k_1 and k_2, and one of them is prefix-increasing, then the memory of $W_1 \cup W_2$ is $\leq k_1 \cdot k_2$.

Our toolbox: Universal graphs and ε-completable automata. As mentioned above, Ohlmann proposed a characterisation of positionality by means of universal graphs [Ohl23]. In 2023, Casares and Ohlmann extended this characterisation to objectives with memory $\leq k$ by considering partially ordered universal graphs [CO25]. Until now, universal graphs have been mainly used to show that certain objectives have memory $\leq k$ (usually for $k = 1$); this is done by constructing a universal graph for the objective. One technical novelty of this work is to exploit both directions of the characterisation, as we also rely on the existence of universal graphs to obtain decidability results.

Our characterisation is based on the notion of k-wise ε-completable automata, which extends the key notion of [CO24a] from positionality to finite memory.

Comparison with [CO24a]. In 2024, Casares and Ohlmann characterised positional ω-regular objectives [CO24a], establishing decidability of positionality in polynomial time, and settling Kopczyński's conjecture for ω-regular objectives. Although the current paper generalises most of these results to the case of memory, as well as potentially infinite automata, the proof techniques are significantly different: while [CO24a] is based on intricate successive transformations of parity automata, the proof we present here is based on an extraction method in the infinite and manipulates ordinal numbers. Though somewhat less elementary, the new proof is notably shorter, and probably easier to read.

Still, when instantiated to the case of memory 1, our results recover and extend several of those from [CO24a]:

- The finite-to-infinite lift of positionality for ω-regular objectives is recovered.
- Kopczyński's conjecture is extended to $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ objectives.
- Although our computability results only talk about **NP**, decidability of positionality in polynomial time can be recovered using a relatively simple greedy argument presented in [CO24a, Theorem 5.3], which relies on the closure under union, which we do re-establish (previous item).

Results that we do not recover include the 1-to-2-player lift and the closure of positionality under addition of neutral letters. However, the fact that we do not obtain the 1-to-2-player lift is not surprising since it does not hold for memory $k > 1$ (see Proposition 42).

2 Preliminaries

We let Σ be a countable alphabet[3] and $\varepsilon \notin \Sigma$ be a fresh symbol that should be interpreted as a neutral letter. Given a word $w \in (\Sigma \cup \{\varepsilon\})^\omega$ we write $\pi_\Sigma(w)$ for the (finite or infinite) word obtained by removing all ε's from w; we call $\pi_\Sigma(w)$ the projection of w on Σ. An *objective* is a set $W \subseteq \Sigma^\omega$. Given an objective $W \subseteq \Sigma^\omega$, we let W^ε denote $\pi_\Sigma^{-1}(W) \subseteq (\Sigma \cup \{\varepsilon\})^\omega$.

Throughout the paper, we use Von Neumann's notation for ordinals: λ denotes the set of ordinals $< \lambda$. This also applies to finite numbers, e.g. $k = \{0, \dots, k-1\}$. As is standard, we also identify cardinals with their initial ordinals.

[3] We restrict our study to countable alphabets, as if Σ is uncountable, the topological space Σ^ω is not Polish and the class $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ is not as well-behaved.

2.1 Graphs, games and memory

We introduce notions pertaining to games and strategy complexity, as they will be central in the statement of our results. Nevertheless, we note that all our technical proofs will use these definitions through Theorem 5 below, and will not explicitly use games.

Graphs. A *Σ-graph* G is given by a set of vertices $V(G)$ and a set of coloured, directed edges $E(G) \subseteq V(G) \times \Sigma \times V(G)$. We write $v \xrightarrow{c} v'$ for edges (v, c, v'). A path is a sequence of edges with matching endpoints $(v_0 \xrightarrow{c_0} v_1)(v_1 \xrightarrow{c_0} v_2)\dots$ which we write as $v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} \dots$. Paths can be empty, finite, or infinite, and have a label $c_0 c_1 \dots$. Throughout the paper, graphs are implicitly assumed to be without dead-end: every vertex has an outgoing edge.

We say that a vertex v in a Σ-graph (resp. a $(\Sigma \cup \{\varepsilon\})$-graph) *satisfies* an objective $W \subseteq \Sigma^\omega$ if the label of any infinite path from v belongs to W (resp. to W^ε). A *pointed graph* is a graph with an identified initial vertex. A pointed graph satisfies an objective $W \subseteq \Sigma^\omega$ if the initial vertex satisfies W; a non-pointed graph satisfies an objective if all its vertices do. An *infinite tree* is a sinkless pointed graph whose initial vertex is called the root, and with the property that every vertex admits a unique path from the root.

A *morphism* from a Σ-graph G to a Σ-graph H is a map $\phi : V(G) \to V(H)$ such that for any edge $v \xrightarrow{c} v'$ in G, it holds that $\phi(v) \xrightarrow{c} \phi(v')$ is an edge in H. Morphisms between pointed graphs should moreover send the initial vertex of G to the initial vertex of H. Morphisms need not be injective. We write $G \to H$ when there exists a morphism $\phi\colon G \to H$.

Games and strategies. A *game* is given by a pointed $(\Sigma \cup \{\varepsilon\})$-graph G together with an objective $W \subseteq \Sigma^\omega$, and a partition of the vertex set $V(G) = V_{\text{Eve}} \sqcup V_{\text{Adam}}$ into the vertices controlled by Eve and those controlled by Adam. A *strategy*[4] (for Eve) is a pointed graph together with a morphism π towards G, satisfying that for every edge $v \xrightarrow{c} v'$ in the game, where $v \in V_{\text{Adam}}$, and for all $u \in \pi^{-1}(v)$, there is an edge $u \xrightarrow{c} u'$ such that $u' \in \pi^{-1}(v')$. A strategy is *winning* if it satisfies the objective W of the game. We say that Eve wins if there exists a winning strategy.

Memory. A *finite-memory strategy*[5] is a strategy with vertices $V(G) \times k$ and projection $\pi(v, m) = v$, with the additional requirement that for every edge $(v, m) \xrightarrow{\varepsilon} (v', m')$, it holds that $m = m'$. We say that k is the *memory* of the strategy, and numbers $1, \dots, k$ are called memory states. Informally, the requirement above says that when reading an ε-transition in the game, we are not allowed to change the memory state; this is called ε-memory in [CO25] (to which we refer for more discussion), but since it is the main kind of memory in this paper, we will simply call it the memory.

A finite-memory strategy is called *chromatic* if there is a map $\chi : k \times (\Sigma \cup \{\varepsilon\}) \to k$ such that for every edge $(v, m) \xrightarrow{c} (v', m')$ in the strategy, it holds that $m' = \chi(m, c)$. We say that χ is the chromatic update. Note that necessarily, we have $\chi(m, \varepsilon) = m$ for every memory state m.

The (*chromatic*) *memory* of an objective W is the minimal k such that for every game with objective W, if Eve has a winning strategy, she has a winning (chromatic) strategy with

[4] We follow the terminology from [CO25]. The classical notion of a strategy as a function $f\colon E(G)^* \to V(G)$ can be recovered by considering the graph with vertices $E(G)^*$, and edges $\rho \xrightarrow{e} \rho e$.

[5] It is common to define a memory structure as an automaton reading the edges of a game graph. This notion can be recovered by taking k as the states of the automaton.

memory $\leq k$.

2.2 Automata

A *parity automaton* $\mathcal{A}$ (or just automaton in the following) with *index* d – an even number – and alphabet Σ, is a pointed $((\Sigma \cup \{\varepsilon\}) \times d)$-graph. Vertices are called states, edges are called transitions and written $q \xrightarrow{c:y} q'$, where $c \in \Sigma \cup \{\varepsilon\}$ and $y \in d$. Elements in d are called priorities. Generally, we use the convention that even priorities are denoted with letter x, whereas y can be used to denote any priority. Transitions of the form $q \xrightarrow{\varepsilon:y} q'$ are called *ε-transitions*; note that they also carry priorities.

Infinite paths from the initial state q_0 are called runs. A run is accepting if the projection of its label on the second coordinate belongs to

$$\mathrm{Parity}_d = \{y_0 y_1 \cdots \in d^\omega \mid \liminf(y) \text{ is even}\}. \quad \text{(Note the use of } \textit{min}\text{-parity.)}$$

The *language* $L(\mathcal{A})$ of $\mathcal{A}$ is $\pi_\Sigma(L')$, where $L' \subseteq (\Sigma \cup \{\varepsilon\})^\omega$ is the set of projections on the first coordinate of runs which are accepting. We require that all these projections are infinite words; stated differently, there is no accepting run from q_0 labelled by a word in $(\Sigma \cup \{\varepsilon\})^* \varepsilon^\omega$. An automaton is *deterministic* if there are no ε-transitions and for any state $q \in V(\mathcal{A})$ and any letter $a \in \Sigma$ there is at most one transition $q \xrightarrow{a:\_} \_$. We say that an automaton is *determinisable by pruning* if one can make it deterministic by removing some transitions, without modifying its language. A language belongs to $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ if it is the language of a deterministic automaton, and it is *ω-regular* if the automaton is moreover finite.

We will often identify pointed graphs with automata of index 2 whose transitions are all labelled with priority 0. Note that in this case, all runs are accepting. This requires making sure that there is no accepting run labelled by a word from $\Sigma^* \varepsilon^\omega$, which, up to assuming that all vertices are accessible from the initial one, amounts to saying that there is no infinite path of $\xrightarrow{\varepsilon}$. We say that such a graph is *well-founded*.

Blowups and k-automata. A *k-blowup* $\mathcal{B}$ of an automaton $\mathcal{A}$ is any automaton with $V(\mathcal{B}) \subseteq V(\mathcal{A}) \times \{1, \dots, k\}$, with initial state in $\{q_0\} \times k$ and such that

1. for each transition $q \xrightarrow{c:y} q'$ in $\mathcal{A}$ and each $m \in k$, there is a transition $(q, m) \xrightarrow{c:y} (q', m')$ in $\mathcal{B}$ for some $m' \in k$, and
2. all ε-transitions are of the form $(q, m) \xrightarrow{\varepsilon:y} (q', m)$ for some $m \in k$.

Note that the first item implies that $L(\mathcal{A}) \subseteq L(\mathcal{B})$. However there may be additional transitions (e.g. ε-transitions) that are not of the form described in the first item.

A *k-automaton* is just an automaton whose states are a subset of $Q \times k$, for some set Q; for instance, k-blowups are k-automata. Equivalently, these are automata with a partition of their states in k identified subsets. For a state (q, m) in a k-automaton, m is called its *memory state*. A k-automaton is called *chromatic* if there is a map $\chi : \{1, \dots, k\} \times (\Sigma \cup \{\varepsilon\}) \to \{1, \dots, k\}$ such that for all transition $(q, m) \xrightarrow{c:y} (q', m')$ it holds that $m' = \chi(m, c)$.

Cascade products. Let $\mathcal{A}$ be an automaton with alphabet Σ and index d, and let S be a d-graph. We define their *cascade product* $\mathcal{A} \ltimes S$ to be the $(\Sigma \cup \{\varepsilon\})$-graph with vertices $V(\mathcal{A}) \times V(S)$ and edges

$$(q, s) \xrightarrow{c} (q', s') \quad \iff \quad \exists y, [q \xrightarrow{c:y} q' \text{ and } s \xrightarrow{y} s'].$$

If S is a pointed graph with intial vertex s_0, then $\mathcal{A} \ltimes S$ is pointed with initial vertex (q_0, s_0). It is easy to check that we then have the following lemma.

▶ **Lemma 2.** *Let $\mathcal{A}$ be an automaton with index d and S be a d-graph satisfying* Parity_d. *Then $\mathcal{A} \ltimes S$ is well-founded and satisfies $L(\mathcal{A})$.*

2.3 ε-completability and universal graphs

We now introduce and discuss the key notion used in our main characterisation, which adapts the notion from [CO24a] from positionality to finite memory.

k-wise ε-completability. A k-automaton $\mathcal{A}$ with index d is called *k-wise ε-complete* if for each even $x \in d$, for each memory state m and each ordered pair of states $(q, m), (q', m)$:

$$\text{either} \quad (q,m) \xrightarrow{\varepsilon:x} (q',m) \quad \text{or} \quad (q',m) \xrightarrow{\varepsilon:x+1} (q,m).$$

Intuitively, having an edge $(q, m) \xrightarrow{\varepsilon:x} (q', m)$ means that "(q, m) is much better than (q', m)", as one may freely go from (q, m) to (q', m) and even see a good priority on the way. Similarly, $(q', m) \xrightarrow{\varepsilon:x+1} (q, m)$ means that "(q', m) is not much worse than (q, m)".

It is also useful for the intuition to apply the definition to both ordered pairs $((q, m), (q', m))$ and $((q', m), (q, m))$. Since automata exclude accepting runs which are ultimately comprised of ε-transitions, we cannot have $(q, m) \xleftrightarrow{\varepsilon:x} (q', m)$, and therefore ε-completability rewrites as: for each x, each memory state m and each unordered pair $(q, m), (q', m)$ of states,

$$\text{either} \quad (q,m) \xrightarrow[\varepsilon:x+1]{\varepsilon:x} (q',m) \quad \text{or} \quad (q,m) \xleftrightarrow{\varepsilon:x+1} (q',m).$$

Hence an alternative, maybe more useful view (this is the point of view adopted in [CO24a]) is that, up to applying some adequate closure properties, a k-wise ε-complete automaton is endowed with the following structure: for each even priority x and each memory state m, the states with memory state m are totally preordered by the relation $\xrightarrow{x+1}$, and the relation $\xrightarrow{x}$ is the strict version of this preorder. Moreover, for $x' > x$, the x'-preorder is a refinement of the x-preorder.

A k-automaton $\mathcal{A}$ is called *k-wise ε-completable* if one may add ε-transitions to it so as to turn it into a k-wise ε-complete automaton $\mathcal{A}^\varepsilon$ satisfying $L(\mathcal{A}^\varepsilon) = L(\mathcal{A})$. In this case, we call $\mathcal{A}^\varepsilon$ an *ε-completion*. We simply say *ε-complete* (resp. "*ε-completable*") when k is clear from the context.

▶ **Example 3.** Let $\Sigma = \{a, b, c\}$ and

$$W = \mathtt{No}(b) \vee \mathtt{Fin}(aa) \vee \mathtt{Inf}(cc).$$

We show a 2-wise ε-complete automaton recognising W in Figure 1. By Theorem 7, the memory of W is ≤ 2 (and it is easy to see that this bound is tight).

In the important special case of well-founded graphs viewed as automata, since all non-ε-transitions are labelled with priority 0, transitions $\xrightarrow{\varepsilon:0}$ and $\xrightarrow{\varepsilon:1}$ play the same role, and we simply write them $\xrightarrow{\varepsilon}$. Therefore, (1) can be seen as totality of the $\xrightarrow{\varepsilon}$ relation.

The following theorem, a key result in [CO25] where it is called the structuration lemma, will also be crucial to this work. Recall that we see well-founded pointed graphs as automata with only 0-transitions, and we apply the terms k-blowup and ε-completable to them accordingly.

▶ **Theorem 4** (Adapted from Lemma 3.4 in [CO25])**.** *Let G be a well-founded pointed graph satisfying an objective W which is assumed to have (chromatic) memory $\leq k$ over games of size $\leq 2^{|G|}$. There is a (chromatic) k-blowup G' of G which is well-founded, k-wise ε-complete, and satisfies W.*

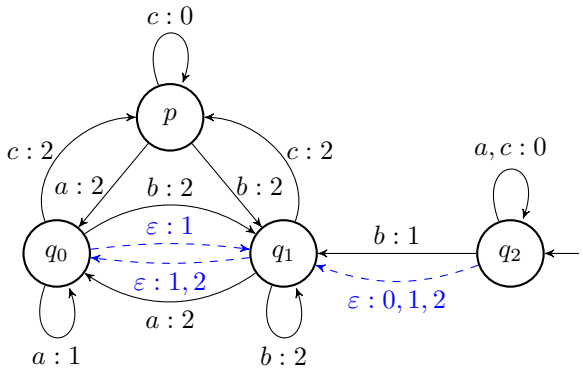

Figure 1 A 2-automaton recognising $W = \mathtt{No}(b) \vee \mathtt{Fin}(aa) \vee \mathtt{Inf}(cc)$, where p is assumed to have a different memory state than q_0, q_1 and q_2. It is 2-wise ε-completable by adding the indicated ε-transitions. A completion also contains the transitions $q_2 \xrightarrow{\varepsilon:0,1,2} q_0$, as well as all transitions $q_i \xrightarrow{\varepsilon:3} q_j$ for $i > j$, and transitions $p \xrightarrow{\varepsilon:y} p$ for y odd; these are omitted for ease of reading. However, it is not chromatic since reading c may or may not switch the memory state.

For completeness, we give a proof of Theorem 4 in Appendix A.

Universal graphs. Given an objective W and a cardinal κ, we say that a graph U is (κ, W)-*universal* if for any infinite tree T of cardinality $|V(T)| < \kappa$ satisfying W, there is a morphism $\phi : T \to U$ such that $\phi(t_0)$ satifies W in U, where t_0 is the root of T. We may now rephrase the main theorem of [CO25] in terms of ε-complete universal graphs.

▶ **Theorem 5** (Theorem 3.1 in [CO25]). *Let W be an objective. Then W has (chromatic) memory $\leq k$ if and only if for every cardinal κ there exists a (κ, W)-universal graph which is (chromatic and) k-wise ε-complete.*

We now give an explicit definition of a $(\kappa, \mathrm{Parity}_d)$-universal graph S^κ_d which is ε-complete. These ideas date back to the works of Streett and Emerson, who coined the name signatures [SE89], and were made more explicit by Walukiewicz [Wal96]. Vertices are tuples of ordinals $< \kappa$, indexed by odd priorities in d and ordered lexicographically (with the smaller indices being the most significant). For a tuple s and index y, we let $s_{<y}$ be the tuple obtained from s by restricting to coordinates with index $< y$. Edges are given by

$$s \xrightarrow{y} s' \quad \iff \quad \begin{cases} s_{<y} \geq s'_{<y} \text{ and } y \text{ is even; or} \\ s_{\leq y} > s'_{\leq y} \text{ and } y \text{ is odd.} \end{cases}$$

In particular, note that $s \xrightarrow{d-1} s'$ if and only if $s > s'$.

▶ **Lemma 6** ([CO24b, Lemma 2.7]). *The graph S^κ_d is $(\kappa, \mathrm{Parity}_d)$-universal.*

Signature trees. We will work extensively with the graph S^κ_d defined above, and manipulations of its vertices which are tuples of ordinals indexed by odd priorities up to d. For a number x, we use $x_{\mathrm{odd}} = \{1, 3, \ldots, x-1\}$ to denote the set of odd priorities $< x$.

For readability, we use subscripts to indicate which (odd) coordinates are concerned, for instance $s_{<x}$ will be our notation for tuples of ordinals $< \kappa$ indexed with odd priorities $< x$, and similarly for $s_{>x}$. Therefore we often use $s_{<x}$ and $s_{>x}$ as two different variables (and not necessarily the projections of some given variable s). Concatenation of tuples is written

like for words, therefore $s_{<x}s_{>x}$ denotes a tuple indexed by all odd priorities (i.e. a vertex of S_d^κ).

By a slight abuse (since these do not correspond to trees are defined above), we use the terminology *signature trees* to refer to subsets of $\kappa^{d_{\text{odd}}} = V(S_d^\kappa)$. Elements of the subsets should be thought of as leaves of the tree, while their (non-proper) prefixes correspond to nodes. More precisely, a *node at level x*, where x is an even priority from d, in a signature tree T is a tuple $s_{<x} \in \kappa^{x_{\text{odd}}}$ such that there exists $s_{>x}$ satisfying $s_{<x}s_{>x} \in T$. In particular, elements of T are nodes at level d (i.e. leaves). The *subtree rooted at* a node $s_{<x}$ of level x is defined to be $\{s_{>x} \mid s_{<x}s_{>x} \in T\}$.

The children of a node $s_{<x}$ are the nodes of the form $s_{<x}s_{x+1}$. The branching of a node is its number of children, and we say that a signature tree has *branching* b if every node has branching exactly b. Notably, $\kappa^{d_{\text{odd}}}$ has branching κ.

3 Characterisation of objectives in $\mathrm{BC}(\mathbf{\Sigma}_2^0)$ with memory $\leq k$

We state our main characterisation theorem and its decidability consequences for ω-regular languages. We assume that the alphabet Σ is countable, therefore automata can also be taken with countable sets of states.

▶ **Theorem 7** (Main characterisation). *Let W be a $\mathrm{BC}(\mathbf{\Sigma}_2^0)$ objective and let $k \in \mathbb{N}$. The following are equivalent:*

- **(i.)** *W has memory $\leq k$ (resp. chromatic memory $\leq k$) on games of size $\leq 2^{2^{\aleph_0}}$.*
- **(ii.)** *For any automaton $\mathcal{A}$ recognising W, there is a (chromatic) k-blowup $\mathcal{B}$ of $\mathcal{A}$ which is k-wise ε-complete and recognises W.*
- **(iii.)** *There is a deterministic (chromatic) k-automaton $\mathcal{A}$ which is k-wise ε-completable and recognises W. If W is recognised by a deterministic automaton of size n, then $\mathcal{A}$ can be taken of size kn.*
- **(iv.)** *For every cardinal κ, there is a (chromatic) (κ, W)-universal graph which is well-founded and k-wise ε-complete.*
- **(v.)** *W has (chromatic) memory $\leq k$ on arbitrary games.*

Moreover in the case where W is ω-regular and recognised by an automaton with n states and index d, this is also equivalent to:

- **(i'.)** *W has memory $\leq k$ (resp. chromatic memory $\leq k$) on games of size $\leq f(k, |\Sigma|, n, d)$, where f is some triply exponential function explicited in Proposition 19 below.*

This immediately gives the finite-to-infinite lift for both memory and chromatic memory.

▶ **Corollary 8** (Finite-to-infinite lift). *An ω-regular objective has memory (resp. chromatic memory) $\leq k$ over finite games if and only if it has memory (resp. chromatic memory) $\leq k$ over all games.*

For ω-regular W given by a deterministic automaton $\mathcal{B}$ of size n, this also allows to compute the (chromatic) memory in **NP**. First, we note that the memory of $L(\mathcal{B})$ is at most n, as the automaton itself can serve as a (chromatic) memory. Therefore, we can guess $k \leq n$, a deterministic automaton $\mathcal{A}$ of size $\leq kn$ and a (chromatic) k-wise ε-completion $\mathcal{A}^\varepsilon$, and check if $L(\mathcal{B}) \subseteq L(\mathcal{A})$ and if $L(\mathcal{A}^\varepsilon) \subseteq L(\mathcal{B})$, which can be done in polynomial time, since $\mathcal{A}$ and $\mathcal{B}$ are deterministic [CDK93]. Prior to our work, computing the memory was not known

to be decidable, and computing the chromatic memory was only known[6] do be doable in exponential time [Kop08].

▶ **Corollary 9** (Decidability in **NP**). *Given an integer k and a deterministic automaton $\mathcal{A}$, the problem of deciding if $L(\mathcal{A})$ has (chromatic) memory $\leq k$ belongs to **NP**.*

A third important consequence of Theorem 7 is a closure under union – a strong form of the generalised Kopczyński conjecture – stated below. The proof of this result will be the object of Section 4. An objective is *prefix-increasing* if for all $a \in \Sigma$ and $w \in \Sigma^\omega$, it holds that if $w \in W$ then $aw \in W$. It is *prefix-independent* if the converse also holds, that is, $w \in W$ if and only if $aw \in W$.

▶ **Theorem 10** (Union has bounded memory). *Let $W_1, W_2 \subseteq \Sigma^\omega$ be two $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ objectives over the same alphabet, such that W_2 is prefix-increasing. Assume that W_1 has memory $\leq k_1$ and W_2 has memory $\leq k_2$. Then $W_1 \cup W_2$ has memory $\leq k_1 k_2$.*

Our main technical contribution are the implications from (i) to (ii) and from (i') to (ii) in the ω-regular case, which are the objects of Sections 3.1 and 3.2. We proceed in Section 3.3 to show that (ii) implies (iii) which is straightforward. The implication (iii) $\implies$ (iv) is adapted from [CO24a] and presented in Section 3.4. Finally, the implication (iv) $\implies$ (v) is the result of [CO25] (Theorem 5), and the remaining one is trivial.

3.1 Existence of k-wise ε-complete automata: infinitary proof

We start with the more challenging and innovative implication: how to obtain a k-wise ε-complete automaton given an objective in $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ with memory k (that is, (i) $\implies$ (ii)). The finitary version (that is, (i') $\implies$ (ii)) follows the same lines but requires some additional insights, it is the object of the next section (Section 3.2).

▶ **Proposition 11.** *Let W be an objective recognised by an automaton $\mathcal{A}$, and assume that W has (chromatic) memory $\leq k$ on games of size $\leq 2^{2^{\aleph_0}}$. Then there is a (chromatic) k-blowup $\mathcal{B}$ of $\mathcal{A}$ recognising W which is k-wise ε-complete.*

We start with a detailed proof overview (Section 3.1.1), before moving on to the formal proof (Sections 3.1.2, 3.1.3 and 3.1.4).

3.1.1 Proof overview

Let $\kappa = 2^{\aleph_0}$. We assume that W has memory $\leq k$ on games of size $\leq 2^\kappa$; we discuss the chromatic case at the end of the section. Let $\mathcal{A}$ be an automaton recognising W; we aim to construct a k-blowup $\mathcal{B}$ of $\mathcal{A}$ which is ε-complete. We let S denote S^κ_d, the $(\kappa, \mathrm{Parity}_d)$-universal graph defined in the preliminaries.

We consider the cascade product $\mathcal{A} \ltimes S$; this is a $(\Sigma \cup \{\varepsilon\})$-graph which intuitively encodes all possible accepting behaviours in $\mathcal{A}$. Then we apply the structuration result (Theorem 4) to $\mathcal{A} \ltimes S$ which yields a k-blowup G of $\mathcal{A} \ltimes S$ which is well-founded and k-wise ε-complete (as a graph). Stated differently, up to blowing the graph $\mathcal{A} \ltimes S$ into k copies, we have been able to endow it with many ε-transitions, so that over each copy, $\xrightarrow{\varepsilon}$ defines a well-order. Note that the states of G are of the form (q, m, s), with $q \in V(\mathcal{A})$, $m \in k$ and $s \in V(S)$.

[6] In fact, Kopczyński proved that the chromatic memory *over finite games* is computable. By Corollary 8, this coincides with the chromatic memory over arbitrary games.

The states of $\mathcal{B}$ will be $V(\mathcal{B}) = V(\mathcal{A}) \times k$. The challenge lies in defining the transitions in $\mathcal{B}$, based on those of G.

Given a state $(q, m) \in V(\mathcal{B})$ and a transition $q \xrightarrow{c:y} q'$ in $\mathcal{A}$, where $c \in \Sigma \cup \{\varepsilon\}$, by applying the definitions we get transitions of the form $(q, m, s) \xrightarrow{c} (q', m', s')$ in G, for different values of m', whenever $s \xrightarrow{y} s'$ in S. We will therefore define transition $(q, m) \xrightarrow{c:y} (q', m')$ in $\mathcal{B}$ if m' matches suitably many transitions as above; for now, we postpone the precise definition.

We should then verify that the obtained automaton $\mathcal{B}$:

- is a k-blowup of $\mathcal{A}$,
- is ε-complete, and
- recognises W.

The first two items above state that $\mathcal{B}$ should have many transitions: at least those inherited from $\mathcal{A}$, and in addition a number of ε-transitions. This creates a tension with the third item, which states that even with all these added transitions, the automaton $\mathcal{B}$ should not accept too many words.

Let us focus on the third item for now, which will lead to a correct definition for $\mathcal{B}$. Take an accepting run

$$(q_0, m_0) \xrightarrow{c_0:y_0} (q_1, m_1) \xrightarrow{c_1,y_1} \dots$$

in $\mathcal{B}$, where $x = \liminf_i y_i$ is even. For the sake of simplicity, assume that all y_i's are $\geq x$. We should show that its labelling $w = c_0 c_1 \dots$ belongs to W. To this end, we will decorate the run with labels $s_0, s_1, \dots \in S$ so that

$$(q_0, m_0, s_0) \xrightarrow{c_0} (q_1, m_1, s_1) \xrightarrow{c_1} \dots$$

defines a path in G, which concludes since G satisfies W.

Recall that the elements of S are tuples of ordinals $< \kappa$ indexed by odd priorities up to d, and that we use $s_{<x}$ (resp. $s_{>x}$) to refer to a tuple indexed by odd priorities up to $x - 1$ (resp. from $x + 1$). To construct the s_i's, we fix a well chosen prefix $s_{<x} \in \kappa^{x_{\text{odd}}}$ which will be constant, and proceed as follows.

(a.) If $y_i = x$, then we set $s_i = s_{<x} s_{>x}$, for some $s_{>x}$ which depends only on c_i.
(b.) If $y_i > x$, then we set $s_i = s_{<x} s_{>x}$, for some $s_{>x}$ which depends on c_i as well as s_{i+1}.

At this stage the reader may be worried that the backward induction underlying the above definition is not well-founded; however, since the first case occurs infinitely often, the backward induction from the second item is only performed over finite blocks (see also Figure 2 in Section 3.1.4).

This leads to the following definition for $\mathcal{B}$, where x is an even priority:

$$\begin{aligned}
(q, m) \xrightarrow{c:x} (q', m') \text{ in } \mathcal{B} &\iff \exists s_{<x} \exists s_{>x} \forall s'_{>x}, (q, m, s_{<x} s_{>x}) \xrightarrow{c} (q', m', s_{<x} s'_{>x}) \text{ in } G, \\
(q, m) \xrightarrow{c:x+1} (q', m') \text{ in } \mathcal{B} &\iff \exists s_{<x} \forall s'_{>x} \exists s_{>x}, (q, m, s_{<x} s_{>x}) \xrightarrow{c} (q', m', s_{<x} s'_{>x}) \text{ in } G.
\end{aligned}$$

The first line corresponds to point (a.) above, where $s_{>x}$ can be chosen independently of $s'_{>x}$, whereas the second line corresponds to point (b.), since the choice of $s_{>x}$ is conditioned on the value of $s'_{>x}$. (For priorities $> x + 1$, we may apply either the first or second line, depending on the parity, to get the required conclusion.)

The remaining issue is that the choice of the fixed common prefix $s_{<x}$ should be made uniformly, regardless of the transition. This is achieved thanks to an adequate extraction lemma (which extends the pigeonhole principle to the case at hands), which finds a large

enough subset T of $\kappa^{d_{\mathrm{odd}}}$, so that transitions $(q,m,s) \to (q',m',s')$ are similar for different choices of $s, s' \in T$. This ensures that $s_{<x}$ can be chosen uniformly.

There remains to verify that $\mathcal{B}$ is indeed a k-blowup of $\mathcal{A}$ and that it is ε-complete, which will follow easily from the definitions and ε-completeness of G (this is because, after removing "$\exists s_{<x}$" from the definition above, the second line resemble the negation of the first).

For the chromatic case, the proof is exactly the same, we should simply check that if G is chromatic (which is guaranteed by Theorem 4), then so is the obtained automaton $\mathcal{B}$.

We now present the full details of the proof, starting with the extraction lemma.

3.1.2 A combinatorial lemma: Extracting homogeneous subtrees

As an important part of our proof, we will take the graph S, whose set of vertices is $\kappa^{d_{\mathrm{odd}}}$, and extract from it a large enough subgraph which is homogeneous.

We say that a tree T is *everywhere cofinal* if for each node $s_{<x}$, the subtree rooted at x is cofinal in $\kappa^{(d-x)_{\mathrm{odd}}}$. An *inner labelling* of a tree T by L is a map λ assigning a label in L to every node in T. We say that an inner labelling is *constant per level* if for every $x \in d_{\mathrm{odd}}$, λ is constant over nodes of level x in T.

We are now ready to state the extraction lemma. Recall that $\kappa = 2^{\aleph_0}$.

▶ **Lemma 12.** *Let λ be an inner labelling of $\kappa^{d_{\mathrm{odd}}}$ by L, where L is countable. There is an everywhere cofinal tree T such that at every level x, $\lambda|_T$ is constant per level.*

Proof. We prove the lemma by induction on d. For $d = 0$ there is nothing to prove since d_{odd} is empty; let $d \geq 1$ and assume the result known for $d-2$. For each node $s_{<2}$ at level 2, apply the induction hypothesis on the subtree of $\kappa^{d_{\mathrm{odd}}}$ rooted at $s_{<2}$, which gives an everywhere cofinal tree $T_{s_{<2}}$ such that λ is constant per level over $T_{s_{<2}}$. Let $\ell_2^{s_{<2}}, \dots, \ell_d^{s_{<2}}$ denote the constant values of λ on the corresponding levels of $T_{s_{<2}}$, and define a new auxiliary labelling of the nodes s at level 2 of S by the tuple $\ell = (\lambda(s), \ell_2^{s_{<2}}, \dots, \ell_d^{s_{<2}})$.

Now since there are at most countably-many new labellings, and there are $\kappa = 2^{\aleph_0}$ nodes at level 2, there is one of the auxiliary labels ℓ such that cofinaly-many nodes have this new label. We conclude by taking T to be the union of $\{s\} \times T'_s$, where s ranges over nodes at level 2 with the new labelling ℓ, and T'_s are the corresponding everywhere cofinal trees. ◀

We are now ready to formalise the definition of $\mathcal{B}$.

3.1.3 Definition of $\mathcal{B}$

Without loss of generality, we assume that $\mathcal{A}$ contains transition $q \xrightarrow{\varepsilon:d-1} q'$ for every state q (these can be added without risk). Consider the cascade product $\mathcal{A} \ltimes S$. Note that thanks to the assumption above and the definition of S, we have transitions $(q,s) \xrightarrow{\varepsilon} (q',s')$ in $\mathcal{A} \ltimes S$ whenever $s > s'$.

By Lemma 2, $\mathcal{A} \ltimes S$ satisfies W and is well-founded. Moreover it has size $\leq \kappa$, so by our assumption on W, we may apply Theorem 4. This yields a k-blowup G of $\mathcal{A} \ltimes S$ which is k-wise ε-complete. Let us write $V(G) = V(\mathcal{A}) \times k \times V(S)$. We close G by transitivity, meaning that we add transitions $(q,m,s) \xrightarrow{c} (q',m',s')$, for $c \in \Sigma \cup \{\varepsilon\}$, whenever $(q,m,s) \xrightarrow{\varepsilon^* c \varepsilon^*} (q',m,s')$; the obtained graph $\overline{G}$ still satisfies W.

Here comes the important definition: say that (q,m) strongly c-dominates (q',m') at node $s_{<x}$ if

$$\exists s_{>x} \forall s'_{>x} \qquad (q,m,s_{<x}s_{>x}) \xrightarrow{c} (q',m',s_{<x}s'_{>x}) \text{ in } \overline{G},$$

and that (q, m) weakly c-dominates (q', m') at $s_{<x}$ if

$$\forall s'_{>x} \exists s_{>x} \qquad (q, m, s_{<x}s_{>x}) \xrightarrow{c} (q', m', s_{<x}s'_{>x}) \text{ in } \overline{G},$$

where $c \in \Sigma \cup \{\varepsilon\}$. Note that strong domination implies weak domination. The type of a node $s_{<x}$ is the information, for each q, q', m, m' and c, of whether (q, m) strongly or weakly (or not at all) c-dominates (q', m'). This gives finitely-many possibilities for fixed q, q', m, m' and c, and therefore there are in total a countable number of possible types. Thus Lemma 12 yields a tree $T \subseteq \kappa^\alpha$ which is everywhere cofinal and such that for all x, nodes at level x in T all have the same type t_x.

We are now ready to define $\mathcal{B}$. We put $V(\mathcal{B}) = V(\mathcal{A}) \times k$, and for each even $x \in d$ and $c \in \Sigma \cup \{\varepsilon\}$, we define transitions by

$$\begin{aligned}
(q, m) &\xrightarrow{c:x} (q', m') && \text{if } (q, m) \text{ strongly } c\text{-dominates } (q', m') \text{ in } t_x,\\
(q, m) &\xrightarrow{c:x+1} (q', m') && \text{if } (q, m) \text{ weakly } c\text{-dominates } (q', m') \text{ in } t_x.
\end{aligned}$$

Here is the main lemma, which proves the direct implication in Theorem 7.

▶ **Lemma 13.** *Automaton $\mathcal{B}$ is a k-blowup of $\mathcal{A}$, it is k-wise ε-complete and it recognises W.*

The remainder of the section is devoted to proving Lemma 13.

3.1.4 Correctness of $\mathcal{B}$: Proof of Lemma 13

There are a few things to show. The interesting argument is the one that shows that $\mathcal{B}$ recognises W (Lemma 18 below). We should also prove there is no accepting run over words in $\Sigma^*\varepsilon^\omega$, which will be done below as part of Lemma 18.

$\mathcal{B}$ is a k-blowup of $\mathcal{A}$. We should prove the following.

▷ Claim 14. For all transitions $q \xrightarrow{c:y} q'$ in $\mathcal{A}$, and any $m \in k$ there is some $m' \in k$ such that $(q, m) \xrightarrow{c:y} (q', m')$ in $\mathcal{B}$.

Proof. Let $q \xrightarrow{c:y} q'$ be a transition in $\mathcal{A}$ and let $m \in k$. Since G is a k-blowup of $\mathcal{A} \times S$, for all edges $s \xrightarrow{y} s'$ in S there is $m' \in k$ such that $(q, m, s) \xrightarrow{c} (q', m', s')$ in G. Although both proofs are similar, we distinguish two cases.

- If $y = x$ is even. We prove that for all nodes $s_{<x}$ at level x, (q, m) strongly c-dominates (q', m') for some m'. Therefore the same is true in t_x which implies the wanted result. We let $s_{>x} = 0_{>x}$, the zero sequence in $\kappa^{(d-x)_{\text{odd}}}$. Now for all $s'_{>x} \in \kappa^{(d-x)_{\text{odd}}}$, it holds that $s_{<x}s_{>x} = s_{<x}0_{>x} \xrightarrow{x} s_{<x}s'_{>x}$ in S, so there is m' such that $(q, m, s_{<x}s_{>x}) \xrightarrow{c} (q, m', s_{<x}s'_{>x})$ in G and thus also in $\overline{G}$; in this case say that m' is good for $s'_{>x}$.
  Now we claim that if m' is good for $\tilde{s}'_{>x} \geq s'_{>x}$, then it is also good for $s'_{>x}$. Indeed, as observed at the beginning of the section, we have $(q', s_{<x}\tilde{s}'_{>x}) \xrightarrow{\varepsilon} (q', s_{<x}s'_{>x})$ in $\mathcal{A} \times S$ therefore since ε-transitions preserve the memory state in G (by definition of a blowup) we have $(q', m', s_{<x}\tilde{s}'_{>x}) \xrightarrow{\varepsilon} (q', m', s_{<x}s'_{>x})$ in G thus $(q, m, s_{<x}s_{>x}) \xrightarrow{c} (q', m', s_{<x}s'_{>x})$ in $\overline{G}$.
  Therefore the sets $M'_{s'_{>x}}$ of m' which are good for $s'_{>x}$ form a decreasing chain of non-empty subsets of k and thus their intersection is non-empty: there is some m' which is good for all $s'_{>x}$, as required.
- If $y = x + 1$ is odd. We now prove that for all nodes $s_{<x}$ at level x, (q, m) weakly c-dominates (q', m') for some m'. Let $s'_{>x} \in \kappa^{(d-x)_{\text{odd}}}$. Then for any $s_{>x}$ such that

$s_x > s'_x$, it holds that $s_{<x}s_{>x} \xrightarrow{x+1} s'_{<x}s_{>x}$ in S, so there is some m' such that $(q, m, s_{<x}s_{>x}) \xrightarrow{c} (q, m', s_{<x}s'_{>x})$ in G. Hence there is some m' such that, for cofinitely many $s_{>x}$, $(q, m, s_{<x}s_{>x}) \xrightarrow{c} (q, m', s_{<x}s'_{>x})$ is an edge in G and thus also in $\overline{G}$; say that such an m' is good for $s'_{>x}$.
Now, we claim that if m' is good for $\tilde{s}'_{>x} \geq s'_{>x}$, then it is also good for $s'_{>x}$. Indeed, as in the first case, we have $(q', m', \tilde{s}_{<x}s'_{>x}) \xrightarrow{\varepsilon} (q', m', s_{<x}s'_{>x})$ in G thus for cofinitely many $s_{>x}$ we have $(q, m, s_{<x}s_{>x}) \xrightarrow{c} (q', m', s_{<x}s'_{>x})$ in $\overline{G}$.
We conclude just as above. ◁

$\mathcal{B}$ is k-wise ε-complete. We should prove the following claim.

▷ Claim 15. For every even x, memory state m and states q, q', either $(q, m) \xrightarrow{\varepsilon:x} (q', m)$ or $(q', m) \xrightarrow{\varepsilon:x+1} (q, m)$.

Proof. Assume that $(q, m) \xrightarrow{\varepsilon:x+1} (q', m)$ is not a transition in $\mathcal{B}$. Consider a node $s_{<x}$ at level x in T: it has type t_x and thus (q, m) does not weakly ε-dominate (q', m) at $s_{<x}$. This rewrites as

$$\exists s'_{>x} \forall s_{>x} \qquad (q, m, s_{<x}s_{>x}) \xrightarrow{\varepsilon} (q', m, s_{<x}s'_{>x}) \text{ is not an edge in } \overline{G}.$$

Now since $\overline{G}$ is ε-complete, we get

$$\exists s'_{>x} \forall s_{>x} \qquad (q', m, s_{<x}s'_{>x}) \xrightarrow{\varepsilon} (q, m, s_{<x}s_{>x}) \text{ in } \overline{G},$$

therefore (q', m) strongly ε-dominates (q, m) at $s_{<x}$, which concludes. ◁

$\mathcal{B}$ recognises W. We now turn to the more involved part. We start by proving the following two technical lemmas.

▶ **Lemma 16.** *Assume that $(q, m) \xrightarrow{c:y} (q', m')$ in $\mathcal{B}$ for some y. There is a map $f : T \to T$ such that for all $s \in T$,*

- *there is an edge $(q, m, f(s)) \xrightarrow{c} (q', m', s)$ in $\overline{G}$; and*
- *it holds that $f(s)_{<y} = s_{<y}$.*

Proof. Let $x + 1$ be the smallest odd priority $\geq y$. Since strong domination implies weak domination, and $(q, m) \xrightarrow{c:y} (q', m')$, we have in any case that (q, m) weakly c-dominates (q', m') in t_x. This means that for any $s' = s'_{<x}s'_{>x} \in T$, there exists $s_{>x}$ such that $(q, m, s'_{<x}s_{>x}) \xrightarrow{c} (q', m', s'_{<x}s'_{>x})$ in $\overline{G}$. Now since T is everywhere cofinal, there exists $\tilde{s}_{>x} \geq s_{>x}$ such that $s'_{<x}\tilde{s}_{>x} \in T$, and we let $f(s') = s'_{<x}\tilde{s}_{>x}$. Clearly $f(s')_{<y} = s'_{<x} = s'_{<y}$, and also, we have $(q, m, s'_{<x}\tilde{s}_{>x}) \xrightarrow{\varepsilon} (q, m, s'_{<x}s_{>x})$ in G so the result follows from ε-transitivity. ◀

▶ **Lemma 17.** *Assume that $(q, m) \xrightarrow{c:x} (q', m')$ in $\mathcal{B}$ for some even x, and let $s_{<x}$ be a node at level x in T. There is $s_{>x} \in \kappa^{(d-x)_{\mathrm{odd}}}$ such that $s_{<x}s_{>x} \in T$ and for any $s'_{>x} \in \kappa^{(d-x)_{\mathrm{odd}}}$, $(q, m, s_{<x}s_{>x}) \xrightarrow{c} (q', m', s_{<x}s'_{>x})$ in $\overline{G}$.*

Proof. From the definition of strong domination, there is $\tilde{s}_{>x}$ such that for all $s'_{>}x$, it holds that $(q, m, s_{<x}\tilde{s}_{>x}) \xrightarrow{c} (q', m', s_{<x}s'_{>x})$ in $\overline{G}$. By everywhere confinality of T, there is $s_{>x} \geq \tilde{s}_{>x}$ such that $s_{<x}s_{>x} \in T$. We conclude (as previously) using ε-transitivity. ◀

We are now ready for the main argument.

▶ **Lemma 18.** *The language of $\mathcal{B}$ is contained in W. Moreover, there is no accepting run labelled by words in $\Sigma^*\varepsilon^\omega$.*

Proof. Take an accepting run

$$(q_0, m_0) \xrightarrow{c_0:y_0} (q_1, m_1) \xrightarrow{c_1:y_1} \dots$$

in $\mathcal{B}$. Let $x = \liminf_i y_i$ (it is even since the run is accepting), and let i_0 be such that $y_i \geq x$ for $i \geq i_0$.

As explained in the proof overview, our goal will be to endow each (q_i, m_i) with some $s_i \in T$ such that for all i, $(q_i, m_i, s_i) \xrightarrow{c_i} (q_{i+1}, m_{i+1}, s_{i+1})$ in $\overline{G}$. This implies the result since $\overline{G}$ satisfies W, and since it does not have paths labelled by words in $\Sigma^*\varepsilon^\omega$ by well-foundedness. We pick an arbitrary node $s_{<x}$ at level x in T and proceed as follows:

- for each $i \geq i_0$ such that $y_i = x$, we let $s_{>x}$ be obtained from Lemma 17 and set $s_i = s_{<x}s_{>x}$;
- for any other i, we proceed by backwards induction within each block (see Figure 2) and let $s_i = f(s_{i+1})$, where f is obtained by applying Lemma 16 to transition $(q_i, m_i) \xrightarrow{c_i:y_i} (q_{i+1}, m_{i+1})$.

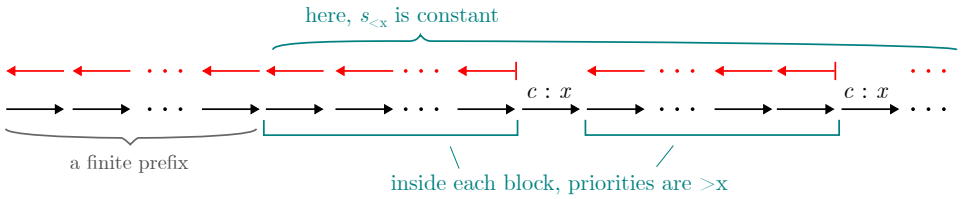

Figure 2 The run in $\mathcal{B}$ (in black), and the order in which the s_i's are computed (in red).

For i's as in the second item, it follows from Lemma 16 that $(q_i, m_i, s_i) \xrightarrow{c_i} (q_{i+1}, m_{i+1}, s_{i+1})$, and moreover, assuming $i \geq i_0$, that $(s_i)_{<x} = (s_{i+1})_{<x}$. Thus for all $i \geq i_0$ we have $(s_i)_{<x} = s_{<x}$ hence we also have, by Lemma 17, that $(q_i, m_i, s_i) \xrightarrow{c_i} (q_{i+1}, m_{i+1}, s_{i+1})$ in $\overline{G}$ for i's as in the first item. ◀

3.2 Existence of ε-completable automata: finitary proof

We now adapt Proposition 11 to the finitary setting, for ω-regular objectives.

▶ **Proposition 19.** *Let W be an ω-regular objective recognised by a finite automaton $\mathcal{A}$ of index d. Assume that W has (chromatic) memory $\leq k$ on games of size $\leq 2^\kappa$, where*

$$\kappa = k^{|Q|^2(|\Sigma|+1)}(3^{k^2|Q|^2(|\Sigma|+1)^{d/2+1}}2|Q|^2)^{k2^{|Q|^2}}.$$

Then there exists a k-wise ε-complete automaton of size $\leq k|Q|$ recognising W.

Note that the size of κ is doubly exponential in the size of $\mathcal{A}$, and therefore the bound on the size of the game graphs is triply exponential. This section is devoted to proving the proposition. The general idea of the proof is similar to the previous section (see Section 3.1.1 for a detailed overview), except that we now manipulate finite objects. In particular, the notion of being (everywhere) cofinal is now irrelevant, and we will need more careful extraction mechanisms.

We will rely on the following standard lemma, which allows to bound the size of each block in Figure 2. We say that an accepting run in a given automaton is *ℓ-simple* if its accepting priority x occurs in every infix of size $\leq \ell$.

▶ **Lemma 20.** *Let $\mathcal{A}$ and $\mathcal{B}$ be automata of size $\leq n$. Assume that every $(2n^2)$-simple accepting run in $\mathcal{B}$ is labelled by a word from $L(\mathcal{A})$. Then $L(\mathcal{B}) \subseteq L(\mathcal{A})$.*

Proof. Let $\mathcal{A} \times \mathcal{B}$ be the product automaton of $\mathcal{A}$ and $\mathcal{B}$. That is, $\mathcal{A} \times \mathcal{B}$ is the graph with states $Q^{\mathcal{A}} \times Q^{\mathcal{B}}$ and transitions $(q,p) \xrightarrow{a:(y,z)} (q',p')$ if $q \xrightarrow{a:y} q'$ in $\mathcal{A}$ and $p \xrightarrow{a:z} p'$ in $\mathcal{B}$. We say that a run $(q_1,p_1) \xrightarrow{a_1:(y_1,z_1)} (q_2,p_2) \xrightarrow{a_2:(y_2,z_2)} \dots$ in $\mathcal{A} \times \mathcal{B}$ is $\mathcal{A}$-accepting if $y_1y_2\cdots \in \mathrm{Parity}$, and $\mathcal{B}$-accepting if $z_1z_2\cdots \in \mathrm{Parity}$.

Assume by contradiction that there is a run ρ as above that is $\mathcal{B}$-accepting but $\mathcal{A}$-rejecting. Let $y_{\min} = \liminf y_i$ (which is odd) and $z_{\min} = \liminf z_i$ (which is even). In the following, the term *subrun* means a subword of a run (that is, a path obtained by removing edges of the given run) that is also a run. We will build a subrun of ρ having as accepting priorities $y_{\min}$ and $z_{\min}$, in each component, but such that the projection into $\mathcal{B}$ is a $(2n^2)$-simple run, leading to a contradiction.

Note that for every non-empty finite run $\upsilon = (q,p) \overset{u}{\rightsquigarrow} (q',p')$ in $\mathcal{A} \times \mathcal{B}$, there exists a subrun $\upsilon' = (q,p) \overset{u'}{\rightsquigarrow} (q',p')$ with same starting and ending point and of size $\leq n^2$. This is obtained just by recursively removing internal cycles (i.e. proper infixes starting and ending in the same state). Also, if $\upsilon = (q,p) \overset{u}{\rightsquigarrow} (q',p')$ is a finite run containing one occurrence of priority $y_{\min}$ in the first component, there exists a subrun $\tilde{\upsilon} = (q,p) \overset{u'}{\rightsquigarrow} (q',p')$ also containing some occurrence of $y_{\min}$ and of size $\leq 2n^2$. Indeed, it suffices to pick one occurrence of $y_{\min}$ in υ , and apply the previous remark to the prefix and the suffix of the run split in the occurrence of $y_{\min}$. That is, if $\upsilon = \upsilon_1 e \upsilon_2$, with $y_{\min}$ appearing in the first component of e, then $\tilde{\upsilon} = \upsilon_1' e \upsilon_2'$. As we can assume that e does not appear in υ_1' nor in υ_2' (by the previous argument), we have $|\tilde{\upsilon}| \leq 2(n^2-1)+1$.

Consider the decomposition of the run $\rho = \upsilon_1 e_1 \upsilon_2 e_2 \dots$, where each transition e_i carries priority $z_{\min}$ in the second component. The run $\rho' = \tilde{\upsilon_1} e_1 \tilde{\upsilon_2} e_2 \dots$ is as desired. ◀

As in the infinitary proof, we let S denote S_d^κ (which is finite), and apply Theorem 4 to the product $\mathcal{A} \ltimes S$. This gives a k-blowup G of $\mathcal{A} \ltimes S$ which is k-wise ε-complete and satisfies W. As previously, we assume without loss of generality that $\mathcal{A}$ contains all transitions of the form $q \xrightarrow{\varepsilon:d-1} q$, and therefore for $s > s'$ in S, since $s \xrightarrow{d-1} s'$ in S we have edges $(q,m,s) \xrightarrow{\varepsilon} (q,m,s')$ in G.

We again let $\overline{G}$ be obtained by closing G by transitivity, i.e. adding transitions $(q,m,s) \xrightarrow{c} (q',m',s')$, for $c \in \Sigma \cup \{\varepsilon\}$, whenever $(q,m,s) \xrightarrow{\varepsilon^* c \varepsilon^*} (q',m,s')$ in G. Note that $\overline{G}$ also satisfies W. Without loss of generality (up to removing some ε-edges while preserving being k-wise ε-complete), we assume that for each m, $\xrightarrow{\varepsilon}$ defines a linear order over $Q \times \{m\} \times S$ in $\overline{G}$.

3.2.1 Domination and definition of the automaton

As in the previous proof, in order to define $\mathcal{B}$, we rely on a notion of domination; however we now need a more precise definition. Consider a signature[7] tree $T \subseteq \kappa^{d_{\mathrm{odd}}}$ with branching b; the following definitions are with respect to T, but we omit T in the notations for clarity. Let $s_{<x}$ be a node at level $x < d$ in T and let $s_{<x}s^{(0)}_{x+1}, \dots, s_{<x}s^{(b-1)}_{x+1}$ denote its children. Recall that $0_{>x}$ denotes the sequence with 0s indexed at odd positions $> x$. Then we say that (q,m) strongly c-dominates (q',m') at $(s_{<x},i)$ if

$$(q,m,s_{<x}s^{(0)}_{x+1}0_{>x+2}) \xrightarrow{c} (q',m',s_{<x}s^{(i)}_{x+1}0_{>x+2}) \text{ in } \overline{G}$$

[7] Throughout the proof, we drop the word "signature" and only talk of "trees" since these are the only kind of trees we consider here.

and that (q, m) weakly c-dominates (q', m') at $(s_{<x}, i)$ if

$$(q, m, s_{<x}s_{x+1}^{(i+1)}0_{>x+2}) \xrightarrow{c} (q', m', s_{<x}s_{x+1}^{(i)}0_{>x+2}) \text{ in } \overline{G}.$$

We say that (q, m) strongly (resp. weakly) c-dominates (q', m') at $s_{<x}$ if this is the case for all $i \in b$, and that (q, m) strongly (resp. weakly) c-dominates (q', m') at level x if this is the case for all nodes $s_{<x}$ at level x.

Given a tree $T \subseteq \kappa^{d_{\text{odd}}}$, we define an automaton $\mathcal{B}_T$ in the same way as in the previous section, with the adapted definitions of domination: We put $V(\mathcal{B}_T) = Q \times k$, and for each even $x \in d$ and $c \in \Sigma \cup \{\varepsilon\}$, we define transitions by

$$\begin{aligned}
(q, m) &\xrightarrow{c:x} (q', m') && \text{if } (q, m) \text{ strongly } c\text{-dominates } (q', m') \text{ at level } x \text{ in } T,\\
(q, m) &\xrightarrow{c:x+1} (q', m') && \text{if } (q, m) \text{ weakly } c\text{-dominates } (q', m') \text{ at level } x \text{ in } T.
\end{aligned}$$

To obtain a correct automaton $\mathcal{B}_T$, the challenge lies in extracting a tree T whose branching matches the size $2|Q|^2$ of blocks given by Lemma 20, and such that the following holds.

1. For every $(q, m) \in V(\mathcal{B}_T)$, every transition $q \xrightarrow{c:x} q'$ in $\mathcal{A}$ (resp. $\xrightarrow{c:x+1}$) and every node $s_{<x}$ at level x, there is m' such that (q, m) strongly (resp. weakly) c-dominates (q', m') at $s_{<x}$. This will ensure that $\mathcal{B}_T$ is indeed a k-blowup of $\mathcal{A}$.
2. For every memory state m, every node $s_{<x}$ at level x and every pair of states (q, q'), either (q, m) strongly ε-dominates (q', m) or (q', m) weakly ε-dominates (q, m). This will ensure that $\mathcal{B}_T$ is ε-complete.
3. The fact that (q, m) strongly (or weakly) c-dominates (q', m') at $s_{<x}$ depends only on the level x, but not on the choice of the node $s_{<x}$. That is, in the sought tree T, (q, m) strongly (or weakly) c-dominates (q', m') at level x if this is the case *for some* node in level x.

A first extraction allows to guarantee item 1; this is shown in Section 3.2.2 below. However, choosing T such that item 2 holds requires some combinatorics which are presented in Section 3.2.3. We then adapt the previous cofinal extraction lemma (Lemma 12) to the finitary setting, and apply it to obtain the third item in Section 3.2.4. Proposition 19 is finally established in Section 3.2.5.

3.2.2 Guaranteeing that $\mathcal{B}$ is a k-blowup

We prove the first item from the explanation above.

▶ **Lemma 21.** *There exists a tree $T \subseteq \kappa^{d_{\text{odd}}}$ with branching*

$$b_1 = \kappa / k^{|Q|^2(|\Sigma|+1)} = (3^{k^2|Q|^2(|\Sigma|+1)^{d/2+1}} 2|Q|^2)^{k2^{|Q|^2}}$$

such that for every $(q, m) \in Q \times k$ and every transition $q \xrightarrow{c:x} q'$ in $\mathcal{A}$ (resp. $\xrightarrow{c:x+1}$) and every node $s_{<x}$ at level x in T, there is m' such that (q, m) strongly (resp. weakly) c-dominates (q', m') at $s_{<x}$.

Proof. First, observe that if $j \geq i$ and $i' \geq j'$, and

$$(q, m, \underbrace{s_{<x}s_{x+1}^{(i)}0_{>x+2}}_{s^i}) \xrightarrow{c} (q', m', \underbrace{s_{<x}s_{x+1}^{(i')}0_{>x+2}}_{s^{i'}}) \text{ in } \overline{G}$$

then also

$$(q, m, \underbrace{s_{<x} s^{(j)}_{x+1} 0_{>x+2}}_{s^j}) \xrightarrow{c} (q', m', \underbrace{s_{<x} s^{(j')}_{x+1} 0_{>x+2}}_{s^{j'}}),$$

because $s^j > s^{i'}$ and $s^{i'} > s^{j'}$.

We prove the following claim, and then explain how the lemma follows.

▷ Claim 22. Let T be a tree of branching b, let $(q, m) \in Q \times k$, let $q \xrightarrow{c:y} q'$ be a transition in $\mathcal{A}$ and let $s_{<x}$ be a node at level x in T, where either $y = x$ or $y = x + 1$.

- If y is even then there is $m' \in k$ such that (q, m) strongly c-dominates (q', m') at $s_{<x}$ (with respect to T).
- If y is odd, then there is m' and b/k children of $s_{<x}$ such that in any subtree of T where the children of $s_{<x}$ are among these ones, it holds that (q, m) weakly c-dominates (q', m') at $s_{<x}$.

Proof. Let $s_{<x}s^{(0)}_x, \dots, s_{<x}s^{(b-1)}_x$ denote the b children of $s_{<x}$. There are two cases according to the parity of y.

- If $y = x$ is even. First, we prove that there is m' such that (q, m) strongly c-dominates (q', m') at $(s_{<x}, b-1)$, which means

  $$(q, m, \underbrace{s_{<x} s^{(0)}_{x+1} 0_{>x+2}}_{s^0}) \xrightarrow{c} (q', m', \underbrace{s_{<x} s^{(b-1)}_{x+1} 0_{>x+2}}_{s^{b-1}}) \text{ in } \overline{G}.$$

  Indeed, by definition of S we have $s^0 \xrightarrow{x} s^{b-1}$ in S therefore $(q, s^0) \xrightarrow{c} (q', s^{b-1})$ in $\mathcal{A} \ltimes S$ and thus we get the wanted result since G is a k-blowup of $\mathcal{A} \ltimes S$.
  Now by the observation above, we also get that (q, m) strongly c-dominates (q', m') at $(s_{<x}, i')$ for any $i' \leq b-1$, and thus (q, m) strongly c-dominates (q', m') at $s_{<x}$.
- If $y = x + 1$ is odd. As above, we first prove that for every i, there is m'_i such that (q, m) weakly c-dominates (q', m'_i) at $(s_{<x}, i)$, which means

  $$(q, m, \underbrace{s_{<x} s^{(i+1)}_{x+1} 0_{>x+2}}_{s^{i+1}}) \xrightarrow{c} (q', m'_i, \underbrace{s_{<x} s^{(i)}_{x+1} 0_{>x+2}}_{s^i}) \text{ in } \overline{G}.$$

  Indeed, by definition of S we have $s^{i+1} \xrightarrow{x+1} s^i$ in S therefore $(q, s^{i+1}) \xrightarrow{c} (q', s^i)$ in $\mathcal{A} \ltimes S$ and thus we get the wanted result since G is a k-blowup of $\mathcal{A} \ltimes S$.
  Now, by the pigeonhole principle, take m' such that $m' = m'_i$ for at least b/k values $i_0 < i_1 < \cdots < i_{b/k-1}$ of i. By the observation above, we get that for every $j' < j < b/k$,

  $$(q, m, s^{i_j}) \xrightarrow{c} (q', m', s^{i_{j'}{}^{i'}}) \text{ in } \overline{G}.$$

  Therefore in any tree satisfying that the children of $s_{<x}$ are a subset of the $s^{i_j}_x$'s, we get that (q, m) weakly c-dominates (q', m') at $s_{<x}$. ◀

Therefore, given a node $s_{<x}$ at level x, and starting with some tree in which $s_{<x}$ has branching κ, we apply the claim to all transitions $q \xrightarrow{c:x+1} q'$ in $\mathcal{A}$ successively, each time removing all nodes which are not (descendants of) the children given by the lemma. Since there are $\leq |Q|^2|\Sigma \cup \{\varepsilon\}|$ such transitions, the node $s_{<x}$ has branching $\geq \kappa/k^{|Q|^2(|\Sigma|+1)}$ in the resulting tree. If this inequality is strict, we may remove more children arbitrarily so that it becomes an equality. Note that the branchings of other nodes that remain in the tree are not affected.

Applying this first to $s_{<0}$ being the root, and then in a top-down fashion, gives the wanted result. ◀

Note that the conclusion of the lemma still holds when taking a subset of T; this will allow us to perform further extractions to guarantee more properties.

3.2.3 Guaranteeing that $\mathcal{B}$ is ε-complete: Regular words and embeddings

We now introduce some more notions which will allow us to ensure item 2 above.

Regular words. A word $w \in C^*$ is called an n-word if every letter from C occurs exactly n times in w (later in the proof, we will take C to be Q). Recall that we denote $n = \{0, 1, \dots, n-1\}$. Given an n-word w and two letters a, b, we say that

- a is strongly after b (or b is strongly before a), if all occurrences of a are after all occurrences of b in w;
- a is weakly after b (or b is weakly before a), if for every $i < j$, the j-th occurrence of a is after the i-th occurrence of b; and
- a and b are interleaved, if a is both weakly after and weakly before b.

For instance, in

$$\underset{(0)}{a}\ \underset{(1)}{a}\ \underset{(2)}{a}\ \underset{(0)}{c}\ \underset{(1)}{c}\ \underset{(3)}{a}\ \underset{(2)}{c}\ \underset{(4)}{a}\ \underset{(0)}{b}\ \underset{(1)}{b}\ \underset{(2)}{b}\ \underset{(3)}{b}\ \underset{(3)}{c}\ \underset{(4)}{c}\ \underset{(4)}{b},$$

we have that a is strongly before b, and c is weakly before b, and in

$$\underset{(0)}{a}\ \underset{(0)}{b}\ \underset{(1)}{b}\ \underset{(1)}{a}\ \underset{(2)}{a}\ \underset{(2)}{b},$$

letters a and b are interleaved.

An n-word is called regular if for every pair of letters (a, b), either a is strongly before b or b is weakly after a. This amounts to saying that for every unordered pair of letters $\{a, b\}$, either one is strongly before the other, or they are interleaved (applying the previous property to both (a, b) and (b, a)). This also amounts to saying that the alphabet can be partitioned into p parts, and the word can be broken into p corresponding subsequent blocs, where each block is an interleaving of the letters of the corresponding part. For instance,

$$abbabacccdedeedfff$$

is a regular 3-word, where the partition is $\{a, b\} \sqcup \{c\} \sqcup \{d, e\} \sqcup \{f\}$.

Word-embeddings and extraction lemma. A word-embedding from w to w' is a set R of positions in w', such that restricting w' to positions in R gives the word w. We display in bold which letters of w' belong to R, for instance $\boldsymbol{a}ba\boldsymbol{b}$ represents the embedding given by $R = \{0, 3\}$ (recall that we number positions from 0) and therefore $w = ab$. In this case, we say that w is a subword of w'.

Given a subset P of n' and an n'-word w', we may define the embedding given by keeping the i-th occurrence of each letter, for each $i \in P$. For instance, if $P = \{1, 4, 5\}$, we get the following embedding

$$\underset{(0)}{a}\ \underset{(0)}{b}\ \underset{(1)}{\boldsymbol{a}}\ \underset{(1)}{\boldsymbol{b}}\ \underset{(2)}{a}\ \underset{(3)}{a}\ \underset{(2)}{b}\ \underset{(3)}{b}\ \underset{(4)}{\boldsymbol{a}}\ \underset{(5)}{\boldsymbol{a}}\ \underset{(6)}{a}\ \underset{(4)}{\boldsymbol{b}}\ \underset{(5)}{\boldsymbol{b}}\ \underset{(6)}{b}.$$

In this case, we say that the obtained word w is the P-subword of w', and we say that w is a synchronised subword of w' if there is such a P. Note that in this case w is an n-word for $n = |P|$. The following easy lemma will often be used below.

▶ **Lemma 23.** *A synchronised subword of a regular word is regular.*

We are now ready to state our main extraction lemma.

▶ **Lemma 24.** *Let n, k be positive integers, fix a finite alphabet C, let $w'_0, \dots, w'_{k-1}$ be n'-words with $n' \geq n^{k2^{|C|^2}}$. There is a set $P \subseteq n'$ of size n such that the P-subwords $w_0, \dots, w_{k-1}$ of $w'_0, \dots, w'_{k-1}$ are regular.*

Proof. In this proof, we write $a^{(i)} <_w b^{(j)}$ to say that the i-th a occurs before the j-th b in the word w.

- **One word, two letters.** First assume that there is a single word w' and $|C| = 2$. In this case, we will show that it suffices to assume that $n' \geq n^2$.
  - If there is i such that for some letter $a \in C$, it holds that $a^{(i+n-1)} <_{w'} b^{(i)}$, where b is the other letter. Then for $P = \{i, i+1, \dots, i+n-1\}$, we get that a is strongly before b in the P-subword w of w', and thus w is regular (and has size n).
  - Otherwise, we let $P = \{0, n, 2n, \dots, (n-1)n\}$, and claim that both letters are interleaved in the P-subword w of w'. Indeed, we should prove that for any letter a, it holds that $a^{((i+1)n)} >_w b^{(in)}$, and this is because we are not in the first case.
- **One word, many letters.** We now assume that there is still just one word w, but C is any finite set and $n' = n^{2^{|C|^2}}$. Say that an unordered pair of different letters is good in a given word if either one is strongly before the other, or they are interleaved. Just as for Lemma 23, observe that if a pair of letters is good in a given word, then it is also good in its synchronised subwords.

  We proceed as follows. Choose any (unordered) pair of different letters a, b, and apply the previous case to get P_0 of size $(n')^{1/2}$ such that in the P_0-subword w_0, the pair a, b is good. Then take another pair a', b' and apply the same reasoning to w_0: we get $P_1 \subseteq P_0$ such that both a', b' and a, b (by the above observation) are good in the P_1-subword of w. Then repeat this process for each of the $|C|(|C|-1)/2 \leq |C|^2$ unordered pairs of letters. In each step, we get that $|P_i| \geq |P_{i-1}|^{1/2}$, by the previous case.
- **Many words, many letters.** We are now ready to prove the lemma. First we apply the previous case to w'_0, which yields P_0 such that the P_0 subword of w'_0 is regular. Then we apply the previous case to the P_0-subword of w'_1: this gives $P_1 \subseteq P_0$ such that the P_1-subword of w'_1 is regular. By the observation above, since $P_1 \subseteq P_0$, the P_1-subword of w'_0 is also regular. We proceed in this fashion for $i = 2, 3, \dots, k-1$, which proves the lemma. ◀

Guaranteeing ε-completability. Recall that the definitions of strong and weak domination (see beginning of Section 3.2.1) are implicitely parameterised by the choice of a tree T. We are now ready to prove the requirement over T, which will ensure ε-completability of the obtained automaton $\mathcal{B}_T$.

▶ **Lemma 25.** *There exists a tree $T \subseteq \kappa^{d_{\mathrm{odd}}}$ with branching*

$$b_2 = (b_1)^{1/(k2^{|Q|^2})} = 3^{k^2|Q|^2(|\Sigma|+1)^{d/2+1}} 2|Q|^2$$

such that the conclusion of Lemma 21 holds and moreover for every memory state $m \in k$, every node $s_{<x}$ at level x and every pair of states (q, q'), either (q, m) strongly ε-dominates (q', m) or (q', m) weakly ε-dominates (q, m).

Proof. First let T' be a tree of branching b_1 obtained from Lemma 21. Fix a node $s_{<x}$ at level x in T'. Then for each memory state $m \in k$, consider the word $w'_m \in Q^*$ indexed by vertices of $\overline{G}$ of the form $(q, m, s_{<x}s_{x+1}0_{>x+2})$ such that $s_{<x}s_{x+1}$ is a child of $s_{<x}$ in T'. These vertices, which correspond to the positions in w'_m, are linearly ordered by $\xrightarrow{\varepsilon}$ over $\overline{G}$,

and the letters correspond to the projections on the first coordinate. Note that these are b_1-words: each state has exactly b_1 occurrences.

We apply Lemma 24 to the words $w'_0, \dots, w'_{k-1}$, which yields a set $P \subseteq b_1$ of size b_2 such that the corresponding P-subwords are regular. This means that if we set the b_2 children of $s_{<x}$ to be the $s^{(p)}_{x+1}$ for $p \in P$, then for each pair of states (q, q') (i.e. letters), either q is strongly before q', in which case q ε-strongly dominates q' at $s_{<x}$, or q' is weakly after q, in which case q' ε-weakly dominates q at $s_{<x}$.

Let $P^{s_{<x}}$ be the set P as above obtained with respect to the node $s_{<x}$ of T'. We produce a tree satisfying the conclusion of the lemma by succesively restricting the subtree at $s_{<x}$ to the children in $P^{s_{<x}}$ ◀

Note also that thanks to Lemma 23, the conclusion of Lemma 25 still holds for trees that are subsets of T, so we may still extract trees with more properties.

3.2.4 Making T homogeneous

We now show how to ensure item 3 from the overview (Section 3.2.1).

Extracting highly branching trees. We need another extraction lemma, which is a straightforward adaptation of Lemma 12 to the finitary setting. Recall that an inner labelling is a map from nodes of T to some set L, and that it is constant per level if for every $x \in d_{\text{odd}}$, λ is constant over nodes of level x in T.

▶ **Lemma 26.** *Let b be a positive integer and consider a tree T' with branching $b' = b|L|^{d/2+1}$ and an inner labelling λ of T' by L. There is a tree $T \subseteq T'$ with branching b such that $\lambda|_T$ is constant per level.*

Proof. The proof is exactly the same as for Lemma 12, by induction on d. We still reproduce it for clarity.

For $d = 0$ there is nothing to prove since d_{odd} is empty; let $d \geq 1$ and assume the result known for $d - 2$. For each node $s_{<2}$ at level 2, apply the induction hypothesis on the subtree of $\kappa^{d_{\text{odd}}}$ rooted at $s_{<2}$, which gives a tree $T_{s_{<2}}$ with branching b such that λ is constant per level over $T_{s_{<2}}$. Let $\ell_2^{s_{<2}}, \dots, \ell_d^{s_{<2}}$ denote the constant values of λ on the corresponding levels of $T_{s_{<2}}$, and define a new auxiliary labelling of the nodes s at level 2 of S by the tuple $\ell = (\lambda(s), \ell_2^{s_{<2}}, \dots, \ell_d^{s_{<2}})$.

Now since there are at most $|L|^{d/2+1}$ new auxiliary labels, and there are b' nodes at level 2, there is a new labelling such that $\geq b$ nodes have this new label. We conclude by taking T to be the union of $\{s\} \times T'_s$, where s ranges over nodes at level 2 with the new labelling ℓ, and T'_s are the corresponding everywhere cofinal trees. ◀

We are now ready to prove the wanted statement.

▶ **Lemma 27.** *There exists a tree $T \subseteq \kappa^{d_{\text{odd}}}$ with branching*

$$b_2 / 3^{k^2|Q|^2(|\Sigma|+1)^{d/2+1}} = 2|Q|^2$$

such that the conclusion of Lemmas 21 and 25 hold for T and moreover, the fact that (q, m) strongly (or weakly) c-dominates (q', m') at $s_{<x}$ depends only on the level x but not on the choice of the node $s_{<x}$.

Proof. Let T' be a tree of branching b_2 obtained from Lemma 25. Consider the inner labelling which assigns to each node $s_{<x}$ of level x in T', the information, for each (ordered) pair of nodes (q,m), (q',m') and each $c \in \Sigma \cup \{\varepsilon\}$, whether (q,m) strongly, or weakly, or not at all, c-dominates (q',m'). This defines an inner labelling of T' by L where $|L| = 3^{k^2|Q|^2(|\Sigma|+1)}$. Therefore applying Lemma 26 concludes. ◀

We are now ready to define $\mathcal{B}$.

3.2.5 Correctness of $\mathcal{B}$

We are finally ready to prove Proposition 19. Fix a tree T with branching $2|Q|^2$ as given by Lemma 27, and let $\mathcal{B} = \mathcal{B}_T$ (see definition in Section 3.2.1).

The facts that $\mathcal{B}$ is a k-blowup of $\mathcal{A}$ and that it is ε-complete are rephrasings of Lemmas 21 and 25 respectively. There remains to prove that $\mathcal{B}$ recognises W (and that $\mathcal{B}$ is indeed an automaton in the sense that it does not have an accepting run over $\Sigma^*\varepsilon^\omega$, which, as in Section 3.1, will be proved at the same time). This is done by adapting Lemmas 16, 17 and 18 from Section 3.1 to the current construction.

▶ **Lemma 28.** *Let $x \in d$ be even, assume that $(q,m) \xrightarrow{c:y} (q',m')$ in $\mathcal{B}$ for some $y > x$ (which can be even or odd), let $i \in 2|Q|^2$ and let $s_{<x}$ be a node at level x in T. Then*

$$\underbrace{(q,m,s_{<x}s_{x+1}^{(i+1)}0_{>x+2})}_{u} \xrightarrow{c} \underbrace{(q',m',s_{<x}s_{x+1}^{(i)}0_{>x+2})}_{u'} \text{ in } \overline{G}.$$

Proof. Let x' be even such that $y \in \{x', x'+1\}$; note that $x \leq x'$. By definition of $\mathcal{B}$ (independently of the parity of y), (q,m) weakly c-dominates (q',m') at level x', so

$$\underbrace{(q,m,s_{<x'}s_{x'+1}^{(i+1)}0_{>x'+2})}_{v} \xrightarrow{c} \underbrace{(q',m',s_{<x'}s_{x'+1}^{(i)}0_{>x'+2})}_{v'}, \tag{1}$$

for every i and $s_{<x'} = s_{<x}s_{x+1}^{(i)}0_{(x+2,x')}$, where $0_{(x+2,x')}$ denotes the all-zero tuple indexed by odd numbers between $x+2$ and x' (if any). Since moreover we have

$$s_{<x}s_{x+1}^{(i+1)}0_{>x+2} > s_{<x'}s_{x'+1}^{(i+1)}0_{>x'+2} \tag{2}$$

and

$$s_{<x'}s_{x'+1}^{(i)}0_{>x'+2} > s_{<x'}0_{>x+2} \tag{3}$$

which respectively give edges

$$\underbrace{(q,m,s_{<x}s_{x+1}^{(i+1)}0_{>x+2})}_{u} \xrightarrow{\varepsilon} \underbrace{(q,m,s_{<x'}s_{x'+1}^{(i+1)}0_{>x'+2})}_{v} \text{ in } \overline{G}$$

and

$$\underbrace{(q',m',s_{<x'}s_{x'+1}^{(i)}0_{>x'+2})}_{v'} \xrightarrow{\varepsilon} \underbrace{(q',m',s_{<x'}0_{>x+2})}_{u'} \text{ in } \overline{G},$$

we get the wanted edge $u \xrightarrow{c} u'$ in $\overline{G}$ by transitivity. ◀

▶ **Lemma 29.** *Assume that $(q,m) \xrightarrow{c:x} (q',m')$ in $\mathcal{B}$ for some even x, and let $s_{<x}$ be a node at level x in T. Then for every $i \in 2|Q|^2$, it holds that*

$$(q,m,s_{<x}s_{x+1}^{(0)}0_{>x+2}) \xrightarrow{c} (q',m',s_{<x}s_{x+1}^{(i)}0_{>x+2}).$$

Proof. This is exactly the definition of $(q,m) \xrightarrow{c:x} (q',m')$ in $\mathcal{B}$ for even x. ◀

We are now ready to prove the final result.

▶ **Lemma 30.** *The language of $\mathcal{B}$ is contained in W. Moreover, there is no accepting run labelled by a word in $\Sigma^*\varepsilon^\omega$.*

The proof is just like for Lemma 18 in Section 3.1.

Proof. Thanks to Lemma 20, it suffices to prove that accepting $2|Q|^2$-simple runs are labelled by words whose projection on Σ is infinite and belongs to W. Take such an accepting run

$$(q_0,m_0) \xrightarrow{c_0:y_0} (q_1,m_1) \xrightarrow{c_1:y_1} \dots$$

in $\mathcal{B}$. Let $x = \liminf_i y_i$ (it is even since the run is accepting), and let i_0 be such that $y_i \geq x$ for $i \geq i_0$.

As previously our goal is to endow each (q_i,m_i) with some $s_i \in T$ such that for all i, $(q_i,m_i,s_i) \xrightarrow{c_i} (q_{i+1},m_{i+1},s_{i+1})$ in $\overline{G}$. This implies the result since $\overline{G}$ satisfies W, and since it does not have paths labelled by words in $\Sigma^*\varepsilon^\omega$ by well-foundedness.

To define the s_i's, we proceed as follows. Let $s_{<x}$ denote an arbitrarily node at level x in T, and let $s_{<x}s^0_{x+1},\dots,s_{<x}s^{2|Q|^2-1}_{x+1}$ denote its $2|Q|^2$ children in T. Then for each i, we let j_i be the least integer ≥ 0 such that $y_{i+j_i} = x$, and observe that $j_i < 2|Q|^2$ since the run is $2|Q|^2$-simple. Then for each i, we set $s_i = s_{<x}s^{j_i}_{x+1}0_{>x+2}$.

For i's such that $j_i > 0$, we get that indeed $(q_i,m_i,s_i) \xrightarrow{c_i} (q_{i+1},m_{i+1},s_{i+1})$ thanks to Lemma 28. For i's such that $j_i = 0$, the same is true thanks to Lemma 29. This concludes the proof. ◀

As before, the proof adapts directly to the chromatic case: if G is chromatic with update χ then so is $\mathcal{B}$.

3.3 Existence of deterministic ε-completable automata

We now prove the implication from (ii) to (iii) in Theorem 7. Take $\mathcal{A}$ to be a deterministic automaton recognising W, and let $\mathcal{B}^\varepsilon$ be the obtained k-blowup of $\mathcal{A}$ which is (chromatically) k-wise ε-complete and recognises W. Then let $\mathcal{B}$ be obtained from $\mathcal{B}^\varepsilon$ by only keeping, for each state $(q,m) \in V(\mathcal{B})$ and each transition $q \xrightarrow{a:y} q'$ in $\mathcal{A}$, a single transition of the form $(q,m) \xrightarrow{a:y} (q',m')$ chosen arbitrarily. Note that $\mathcal{B}$ is a k-blowup of $\mathcal{A}$, so we have: $L(\mathcal{A}) \subseteq L(\mathcal{B}) \subseteq L(\mathcal{B}^\varepsilon)$. We conclude that $\mathcal{B}$ is a deterministic (chromatically) k-wise ε-completable automaton recognising W, as required.

3.4 From deterministic ε-completable automata to universal graphs

We now prove the implication from (iii) to (iv) in Theorem 7. This result was already proved in [CO24b, Prop. 5.30] for the case of $k = 1$; extending to greater values for k presents no difficulty.

Let $\mathcal{B}$ be a deterministic[8] (chromatic) k-wise ε-completable automaton recognising W and $\mathcal{B}^\varepsilon$ be an ε-completion. Fix a cardinal κ and let S denote S^κ_d. Consider the cascade product $U = \mathcal{B}^\varepsilon \ltimes S$. We claim that U is (chromatic) k-wise ε-complete and (κ,W)-universal. Universality follows from the facts that $\mathcal{B}$ is deterministic and S is $(\kappa,\mathrm{Parity}_d)$-universal.

[8] For the purpose of this proof, history-determinism would be sufficient (we refer to [BL23] for the definition and context on history-determinism).

▷ Claim 31. The graph $U = \mathcal{B}^\varepsilon \ltimes S$ is (κ, W)-universal.

Proof. Take a tree T of size $< \kappa$ which satisfies W. We should show that $T \to U$. Since $\mathcal{B}$ is deterministic, we can define a labelling $\rho\colon V(T) \to V(\mathcal{B})$ by mapping t_0 to q_0 and $t \mapsto q$ if the run of $\mathcal{B}$ on the finite word labelling the path $t_0 \to t$ ends in q. Then, any infinite path from t_0 in T is mapped to a run in $\mathcal{B}$ that is accepting (since T satisfies W). Therefore the tree T' obtained by taking T and replacing edge-labels by the priorities appearing in their ρ-images satisfies Parity_d and has size $< \kappa$, so there is a morphism $\mu : T \to S$. It is a direct check that $(\rho, \mu)\colon V(T) \to V(\mathcal{B}) \times V(S) = V(U)$ indeed defines a morphism. ◁

Showing that U is k-wise ε-complete is slightly trickier.

▷ Claim 32. The graph $U = \mathcal{B}^\varepsilon \ltimes S$ is well-founded and (chromatic) k-wise ε-complete.

Proof. Well-foundedness of U follows directly from Lemma 2. Let us write $B_1, \dots, B_k$ for the k parts of $\mathcal{B}^\varepsilon$; the k parts of U will be $B_1 \times V(S), \dots, B_k \times V(S)$. If applicable, chromaticity is a direct check. Let $(b, s), (b', s') \in V(U)$ be in the same part, i.e. $b, b' \in B_i$ for some i. Let x_0 be the minimal even priority such that $b \xrightarrow{\varepsilon:x_0} b'$ in $\mathcal{B}^\varepsilon$ (if such an x does not exist then $x_0 = d + 2$). Then let y_0 be the minimal odd priority such that $s'_{\leq y_0} > s_{\leq y_0}$ (as previously, if $s = s'$ then we let $y_0 = d + 1$). We distinguish two cases.

(1) If $x_0 < y_0$. Then we have $b \xrightarrow{\varepsilon:x_0} b'$ in $\mathcal{B}^\varepsilon$ and $s_{<x_0} = s'_{<x_0}$ which gives $s \xrightarrow{x_0} s'$ in S. Thus we get $(b, s) \xrightarrow{\varepsilon} (b', s')$ in U.

(2) If $y_0 < x_0$. Then $s'_{\leq y_0} > s_{\leq y_0}$, which gives $s' \xrightarrow{y_0} s$ in S. Since $\mathcal{B}^\varepsilon$ is k-wise ε-complete and by definition of x_0, we also have $b' \xrightarrow{\varepsilon:y_0} b$ in $\mathcal{B}^\varepsilon$, therefore $(b', s') \xrightarrow{\varepsilon} (b, s)$ in U.

We conclude that either $(b, s) \xrightarrow{\varepsilon} (b', s')$ or $(b', s') \xrightarrow{\varepsilon} (b, s)$, as required. ◁

4 Union of objectives

In this section, we establish a strong form of the generalised Kopczyński conjecture for $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ objectives. Recall that an objective $W \subseteq \Sigma^\omega$ is prefix-increasing[9] if for all $a \in \Sigma$ and $w \in \Sigma^\omega$, it holds that if $w \in W$ then $aw \in W$. In words, one remains in W when adding a finite prefix to a word of w. Examples of prefix-increasing objectives include prefix-independent and closed objectives.

▶ **Theorem 10** (Union has bounded memory). *Let $W_1, W_2 \subseteq \Sigma^\omega$ be two $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ objectives over the same alphabet, such that W_2 is prefix-increasing. Assume that W_1 has memory $\leq k_1$ and W_2 has memory $\leq k_2$. Then $W_1 \cup W_2$ has memory $\leq k_1 k_2$.*

▶ Remark 33. The assumption that one of the two objectives is prefix-increasing is indeed required: for instance if $W_1 = aa(a+b)^\omega$ and $W_2 = bb(a+b)^\omega$, which are positional but not prefix-increasing, the union $(aa + bb)(a+b)^\omega$ is not positional (it has memory 2).

▶ Remark 34. The bound $k_1 k_2$ in Theorem 10 is tight: For every k_1, k_2, there are objectives W_1, W_2 with memories k_1, k_2 respectively, such that $W_1 \cup W_2$ has memory exactly $k_1 k_2$. One such example is as follows: let $\Sigma = \{a_1, \dots, a_{k_1}, b_1, \dots, b_{k_2}\}$ and $W_1 = \{w \mid w$ contains at least two different a_i infinitely often$\}$ and $W_2 = \{w \mid w$ contains at least two b_i infinitely often$\}$. We can see that W_1, W_2 and $W_1 \cup W_2$ have memory, respectively, k_1, k_2, and $k_1 \cdot k_2$ by building the Zielonka tree of these objectives and applying [DJW97, Thms. 6, 14].

[9] In other papers [Ohl21, Ohl23, CO25] this notion is called prefix-decreasing, as Eve is seen as a "minimiser" player who aims to minimise some quantity.

The rest of the section is devoted to the proof of Theorem 10. We explicit a construction for the union of two parity automata, inspired from the Zielonka tree of the union of two parity conditions, and show that it is (k_1k_2)-wise ε-completable.

Union of parity languages. We give an explicit construction of a deterministic parity automaton $\mathcal{T}$ recognising the union of two parity languages, which may be of independent interest. This corresponds to the automaton given by the Zielonka tree of the union (a reduced and structured version of the LAR).

We let $[d_1] = \{0, 1, \dots, d_1\}$ and $[d_2]^* = \{1^*, \dots, d_2^*\}$. Let $[d_1]_{\text{odd}}$ and $[d_2]^*_{\text{odd}}$ denote the restrictions of $[d_1]$ and $[d_2]^*$ to odd elements. By a slight abuse of notation, we sometimes treat elements in $[d_2]^*$ as natural numbers (e.g. when comparing them).

Alphabet. The input alphabet is $[d_1] \times [d_2]^*$, and we write letters as (y, z). The index of $\mathcal{T}$ is at most $d_1 + d_2 + 2$, we use the letter t for its output priorities.

States. States are given by interleavings of the sequences $\langle 1, 3, \dots, d_1\rangle$ and $\langle 1^*, 3^*, \dots, d_2^*\rangle$ of the elements in $[d_1]_{\text{odd}}$ and $[d_2]^*_{\text{odd}}$ (assuming d_1, d_2 odd). Any state can be taken as initial. For instance, for $d_1 = d_2 = 5$, an example of a state is:

$$\tau = \langle 1, 3, 1^*, 5, 3^*, 5^*\rangle.$$
$$ \;\; 1 \;\; 2 \;\; 3 \;\;\; 4 \;\; 5 \;\;\; 6$$

We use τ to denote such a sequence, which we index from 1 to $(d_1 + d_2)/2$, and write $\tau[i]$ for its i-th element. For $y \in [d_1]$, y odd, we let $\mathsf{ind}_\tau(y) = i$ for the index such that $\tau[i] = y$. For $y \geq 2$ even, we let $\mathsf{ind}_\tau(y)$ be the index i such that $\tau[i] = y - 1$. We let $\mathsf{ind}_\tau(0) = 0$. We use the same notation for $z \in [d_2]^*$, $z \geq 1$. For example, in the state above, $\mathsf{ind}_\tau(2^*) = 3$.

The intuition is that a state stores a local order of importance between input priorities.

Transitions. Let τ be a state and (y, z) an input letter. We define the transition $\tau \xrightarrow{(y,z):t} \tau'$ as follows:

Let $i = \min\{\mathsf{ind}_\tau(y), \mathsf{ind}_\tau(z)\}$. In the following, we assume $i = \mathsf{ind}_\tau(y)$ (the definition for $i = \mathsf{ind}_\tau(z)$ is symmetric). We let $t = 2i$, if y even, and $t = 2i - 1$ if y is odd. If y is even, we let $\tau' = \tau$. If y is odd, let i' be the smallest index $i < i'$ such that $\tau[i'] \in [d_2]^*$, and let τ' be the sequence obtained by inserting $\tau[i']$ on the left of $\tau[i]$ (or $\tau' = \tau$ if no such index i' exists). Formally,

$$\tau'[j] = \tau[j] \text{ for } j < i \text{ and } i' < j, \quad \tau'[i] = \tau[i'] \quad \text{and} \quad \tau'[j] = \tau[j-1] \text{ for } i < j \leq i'.$$

For example, for the state above, we have: $\langle 1, 3, 1^*, 5, 3^*, 5^*\rangle \xrightarrow{(3,2^*):3} \langle 1, 1^*, 3, 5, 3^*, 5^*\rangle$.

For $w = (y_1, z_1)(y_2, z_2) \cdots \in ([d_1] \times [d_2]^*)^\omega$, we let $\pi_1(w) = y_1y_2 \dots$ and $\pi_2(w) = z_1z_2 \dots$.

▶ **Lemma 35.** *The automaton $\mathcal{T}$ recognises the language*

$$L = \{w \in ([d_1] \times [d_2]^*)^\omega \mid \pi_1(w) \in \text{Parity}_{d_1} \text{ or } \pi_2(w) \in \text{Parity}_{d_2}\}.$$

Proof. We show that $L \subseteq L(\mathcal{T})$, the other inclusion is similar (and implied by Lemma 39 below). Let $(y_1, z_1)(y_2, z_2) \cdots \in L$, and assume w.l.o.g. that $y_1y_2 \cdots \in \text{Parity}_{d_1}$. Let $y_{\min} = \liminf y_i$, which is even, and let n_0 be so that for all $n \geq n_0$ we have $y_{\min} \leq y_n$. Let

$$\tau_0 \xrightarrow{(y_1,z_1):t_1} \tau_1 \xrightarrow{(y_2,z_2):t_2} \dots$$

denote the corresponding run in $\mathcal{T}$. Let $i_n = \mathsf{ind}_{\tau_n}(y_{\min})$ be the index where $y_{\min} - 1$ appears in τ_n. Note that the sequence $(i_n)_{n \geq n_0}$ is decreasing. Let n_1 be the moment where this sequence stabilises, i.e., $i_n = i_{n_1}$ for $n \geq n_1$. By definition of the transitions of $\mathcal{T}$, for $n \geq n_1$ all output priorities are $\geq 2i_{n_1}$, and priority $2i_{n_1}$ is produced every time that a letter $(y_{\min}, z_n)$ is read. We conclude that $\mathcal{T}$ accepts w. ◀

0-freeness of automata for prefix-increasing objectives. The fact that W_2 is prefix-increasing will be used via the following lemma. It recasts the fact that we can add $\xrightarrow{\varepsilon:1}$ transitions everywhere to automata recognising prefix-increasing objectives.

▶ **Lemma 36.** *Let W a be prefix-increasing objective with memory $\leq k$. There exists a deterministic k-wise ε-completable automaton $\mathcal{A}$ recognising W and an ε-completion $\mathcal{A}^\varepsilon$ of $\mathcal{A}$ such that $\mathcal{A}^\varepsilon$ does not have any transition with priority 0.*

Proof. First, take any automaton $\mathcal{A}_0$ recognising W. Then let $\mathcal{A}_1$ be obtained by shifting every priority from $\mathcal{A}_0$ by 2. Clearly $\mathcal{A}_1$ also recognises W and does not have any transition with priority 0. Then apply Proposition 11 to get a k-blowup $\mathcal{A}$ of $\mathcal{A}_1$ which is k-wise ε-completable, and let $\mathcal{A}^\varepsilon$ denote the corresponding ε-completion. Since $\mathcal{A}$ has no transition with priority 0, the only possible such transitions in $\mathcal{A}^\varepsilon$ are ε-transition. Then remove all ε-transitions with priority 0 in $\mathcal{A}^\varepsilon$, and add $\xrightarrow{\varepsilon:1}$ transitions between all pairs of states in $\mathcal{A}^\varepsilon$ (in both directions).

Clearly, the obtained automaton $\tilde{\mathcal{A}}^\varepsilon$ is ε-complete and has no transition with priority 0. There remains to prove that it recognises W. Take an accepting run in $\tilde{\mathcal{A}}^\varepsilon$ and observe that the priority 1 is only seen finitely often. Hence from some moment on, the run coincides with a run in $\mathcal{A}^\varepsilon$. We conclude since W is prefix-increasing. ◀

Main proof: ε-completion of the product. We now proceed with the proof of Theorem 10. Using Theorem 7, for $l = 1, 2$, we take a deterministic k_l-wise ε-completable automata $\mathcal{A}_l$ of index d_l recognising W_l, and its ε-completion $\mathcal{A}_l^\varepsilon$. For $l = 2$, we assume thanks to Lemma 36 that $\mathcal{A}_2^\varepsilon$ does not have any transition with priority 0.

We consider the product $\mathcal{A} = (\mathcal{A}_1 \times \mathcal{A}_2) \ltimes \mathcal{T}$, with states $V(\mathcal{A}_1) \times V(\mathcal{A}_2) \times V(\mathcal{T})$ and transitions $(q_1, q_2, \tau) \xrightarrow{a:t} (q_1', q_2', \tau')$ if $q_1 \xrightarrow{a:y} q_1'$ in $\mathcal{A}_1$, $q_2 \xrightarrow{a:z} q_2'$ in $\mathcal{A}_2$, and $\tau \xrightarrow{(y,z):t} \tau'$ in $\mathcal{T}$. The correctness of such a construction is folklore.[10]

▷ Claim 37. The automaton $\mathcal{A}$ is deterministic and recognises $W = W_1 \cup W_2$.

Therefore, there only remains to show the following lemma.

▶ **Lemma 38.** *The automaton $\mathcal{A} = (\mathcal{A}_1 \times \mathcal{A}_2) \ltimes \mathcal{T}$ is $(k_1 k_2)$-wise ε-completable.*

The ε-completion of $\mathcal{A}$ will be a variant of a product of the form $\mathcal{A}_1^\varepsilon \times \mathcal{A}_2^\varepsilon \ltimes \overline{\mathcal{T}}$, where $\overline{\mathcal{T}}$ is a non-deterministic extension of $\mathcal{T}$ with more transitions, but which still recognises the same language.

The automaton $\overline{\mathcal{T}}$. Intuitively, we obtain $\overline{\mathcal{T}}$ by allowing to reconfigure the elements of index $> i$ in a state τ by paying an odd priority $2i - 1$, as well as allowing to move elements of $[d_1]$ to the left. We precise this idea next.

[10] $\mathcal{A}_1 \times \mathcal{A}_2$ can be seen as a Muller automaton with acceptance condition the union of two parity languages. The composition with $\mathcal{T}$ yields a correct parity automaton, as $\mathcal{T}$ recognises the acceptance condition.

We order the states of $\mathcal{T}$ lexicographically, where we assume that $y < z$ for $y \in [d_1]$ and $z \in [d_2]^*$. Formally, we let $\tau < \tau'$ if for the first position j where τ and τ' differ, $\tau[j] \in [d_1]$ (and therefore necessarily $\tau'[j] \in [d_2]^*$). We let $\tau[..i]$ be the prefix of τ up to (and including) $\tau[i]$. We write $\tau <_i \tau'$ if $\tau[..i] < \tau'[..i]$.

Let $\tau \xrightarrow{(y,z):t}$ be a transition in $\mathcal{T}$ as above, and $i_0 = \min\{\mathsf{ind}_\tau(y), \mathsf{ind}_\tau(z)\}$ be the index determining t (i.e. $t \in \{2i_0 - 1, 2i\}$). The automaton $\overline{\mathcal{T}}$ contains a transition $\tau \xrightarrow{(y,z):t'} \tau'$ if:

1. $t' = t$ is odd, and $\tau' \leq_{i_0-1} \tau$; or
2. $t' = t$ is even, and $\tau' \leq_{i_0} \tau$; or
3. $t' \in \{2i' - 1, 2i'\}$ for some $i' \leq i_0$ and $\tau' <_{i'} \tau$. (Note that, if t is odd, this includes all (possibly even) $t' \leq t + 1$.)

In words, we are allowed to output a small (i.e. important) priority when following a strict decrease on sufficiently small components in τ. Note that transitions in $\mathcal{T}$ also belong to $\overline{\mathcal{T}}$ thanks to the rules (1) and (2).

▶ **Lemma 39.** *The automaton $\overline{\mathcal{T}}$ recognises the same language as $\mathcal{T}$.*

Proof. It is clear that $L(\mathcal{T}) \subseteq L(\overline{\mathcal{T}})$. We show the other inclusion.

Consider

$$\tau_0 \xrightarrow{(y_1,z_1):t_1} \tau_1 \xrightarrow{(y_2,z_2):t_2} \dots, \quad \text{an accepting run in } \overline{\mathcal{T}}.$$

Let $t_{\min} = \liminf t_1 t_2 \dots$, which is even. Let $i_{\min} = t_{\min}/2$ be the index responsible for producing priority $t_{\min}$. Let n_0 be such that $t_n \geq t_{\min}$ for all $n \geq n_0$. Observe that for all $n \geq n_0$, we have $\tau_n \geq_{(i_{\min}-1)} \tau_{n+1}$, and therefore there is $n_1 \geq n_0$ such that the prefix $\tau_n[..i_{\min} - 1]$ is the same for all $n \geq n_1$. In fact, the prefix $\tau_n[..i_{\min}]$ must be constant too, as we can only modify $\tau_n[i_{\min}]$ using rule (1.) and that would output priority $t_{\min} - 1$. Assume w.l.o.g. that $\tau_n[i_{\min}] = y \in [d_1]_{\text{odd}}$. Now, for each $n \geq n_1$ it must be that $y_i \geq y + 1$ and it must be that $y_i = y + 1$ each time that priority $t_{\min}$ is produced. Therefore, $\liminf y_1 y_2 \cdots = y + 1$ is even. ◀

The ε-completion. We define $\mathcal{A}^\varepsilon$ as a version of the cascade product of $\mathcal{A}_1^\varepsilon \times \mathcal{A}_2^\varepsilon$ with $\mathcal{T}$, in which ε-transitions are also allowed to use the transitions of $\overline{\mathcal{T}}$. We let $\preccurlyeq$ denote the preference ordering over priorities, given by $1 \preccurlyeq 3 \preccurlyeq \dots \preccurlyeq d - 1 \preccurlyeq d \preccurlyeq \dots 2 \preccurlyeq 0$ (d even). The transitions in $\mathcal{A}^\varepsilon$ are defined as follows:

- For $a \in \Sigma$: $(q_1, q_2, \tau) \xrightarrow{a:t} (q_1', q_2', \tau')$ if this transition appears in $(\mathcal{A}_1 \times \mathcal{A}_2) \ltimes \mathcal{T}$.
- $(q_1, q_2, \tau) \xrightarrow{\varepsilon:t} (q_1', q_2', \tau')$ if $q_1 \xrightarrow{\varepsilon:y} q_1'$ in $\mathcal{A}_1^\varepsilon$, $q_2 \xrightarrow{\varepsilon:z} q_2'$ in $\mathcal{A}_2^\varepsilon$, and $\tau \xrightarrow{(y,z):t'} \tau'$ in $\overline{\mathcal{T}}$ with $t' \preccurlyeq t$.

Note that the condition $t' \preccurlyeq t$ simply allows to output a less favorable priority, so it does not create extra accepting runs. By definition, $\mathcal{A}^\varepsilon$ has been obtained by adding ε-transitions to $\mathcal{A}$. It is a folklore result that composition of non-deterministic automata also preserves the language recognised, so this construction is correct.

▷ Claim 40. The automaton $\mathcal{A}^\varepsilon$ recognises W.

Therefore, we there only remains to prove the following lemma.

▶ **Lemma 41.** *The automaton $\mathcal{A}^\varepsilon$ is $(k_1 k_2)$-wise ε-complete.*

First, we need a few remarks on the structure of ε-complete automata. We write $q \xleftrightarrow{\varepsilon:x+1} q'$ to denote the conjunction of $q \xrightarrow{\varepsilon:x+1} q'$ and $q' \xrightarrow{\varepsilon:x+1} q$. Given two states q, q' in the same part of a k-wise ε-complete automaton, we call breakpoint priority of q and q' the least even $x^{(0)}$ such that $q \xleftrightarrow{\varepsilon:x^{(0)}+1} q'$ does not hold. Note that this is a property of the unordered pair $\{q, q'\}$. Observe also that by the definition of ε-completeness, we have either $q \xrightarrow{\varepsilon:x^{(0)}} q'$ or $q' \xrightarrow{\varepsilon:x^{(0)}} q$. Moreover, assuming that $q \xrightarrow{\varepsilon:x^{(0)}} q'$, we also get that there can be no $q' \xrightarrow{\varepsilon:x} q$, for even x, otherwise we would accept some run labelled by $\Sigma^*\varepsilon^\omega$; therefore for even $x \geq x^{(0)}$ we also have $q \xrightarrow{\varepsilon:x} q'$. To sum up, if $x^{(0)}$ is the breakpoint priority of $\{q, q'\}$ and $q \xrightarrow{\varepsilon:x^{(0)}} q'$, then:

- $q \xleftrightarrow{\varepsilon:y} q'$ for all odd $y < x^{(0)}$;
- $q \xrightarrow{\varepsilon:x} q'$ for all $x \geq x^{(0)}$; and
- there is no transition $q' \xrightarrow{\varepsilon:x} q$ for even x.

Finally, observe that in $\mathcal{A}_2^\varepsilon$, since there are no transitions with priority 0 (and therefore $\xrightarrow{\varepsilon:1}$ connects every ordered pair of states), breakpoint priorities are always ≥ 2.

We are now ready to prove Lemma 41

Proof of Lemma 41. First observe that the k_1 parts of $\mathcal{A}_1^\varepsilon$ and the k_2 parts of $\mathcal{A}_2^\varepsilon$ naturally induce a partition of the states of $\mathcal{A}^\varepsilon$ into k_1k_2 parts. Let $r = (q_1, q_2, \tau)$ and $r' = (q_1', q_2', \tau')$ be two states in the same part of $\mathcal{A}^\varepsilon$, that is, q_l and q_l' are in the same part in $\mathcal{A}_l^\varepsilon$, for $l = 1, 2$. We will show that for some even output priority $x^{(0)}$, it holds that:

1. $r \xleftrightarrow{\varepsilon:x^{(0)}-1} r'$; and

2. either $r \xrightarrow{\varepsilon:x^{(0)}} r'$ or $r' \xrightarrow{\varepsilon:x^{(0)}} r$.

Note that since $r \xrightarrow{\varepsilon:t} r'$ in $\mathcal{A}^\varepsilon$ implies $r \xrightarrow{\varepsilon:t'} r'$ for all $t' \preccurlyeq t$, the two points above imply that for every even $x < x^{(0)}$ we have $r \xleftrightarrow{\varepsilon:x+1} r'$ and for every even $x \geq x^{(0)}$, either $r \xrightarrow{\varepsilon:x^{(0)}} r'$ or $r' \xrightarrow{\varepsilon:x^{(0)}} r$. Therefore, this will prove that $\mathcal{A}^\varepsilon$ is (k_1k_2)-wise ε-complete.

Let $x_1^{(0)}$ and $x_2^{(0)}$ denote the breakpoint priorities of $\{q_1, q_1'\}$, $\{q_2, q_2'\}$ in $\mathcal{A}_1^\varepsilon$ and $\mathcal{A}_2^\varepsilon$, respectively (even and ≥ 2). Let i_T be the largest index such that $\tau[..i_T] = \tau'[..i_T]$, with $i_T = 0$ if $\tau[1] \neq \tau'[1]$. We distinguish two cases, depending on whether some $x_l^{(0)} - 1$ appears in $\tau[..i_T]$.

a) $i_T < \mathsf{ind}_\tau(x_1^{(0)}), \mathsf{ind}_\tau(x_2^{(0)})$. Note that, in particular, $0 < x_1^{(0)}, x_2^{(0)}$.

We show that in this case, we can set $x^{(0)} = 2i_T + 2$. We prove the two points above:

1. We will find odd priorities y_1 and y_2 such that

$$\underbrace{q_1 \xrightarrow{\varepsilon:y_1} q_1'}_{\text{in } \mathcal{A}_1} \text{ and } \underbrace{q_2 \xrightarrow{\varepsilon:y_2} q_2'}_{\text{in } \mathcal{A}_2} \text{ and } \underbrace{\tau \xrightarrow{(y_1,y_2):x^{(0)}-1} \tau'}_{\text{in } \overline{\mathcal{T}}},$$

which gives the wanted result when applied symmetrically in the other direction. Consider the element $\tau[i_T + 1]$ (odd), and assume w.l.o.g. that it belongs to $[d_1]$. We let $y_1 = \tau[i_T + 1]$. Note that $1 \leq y_1 < x_1^{(0)}$, as $i_T + 1 \leq \mathsf{ind}_\tau(x_1^{(0)})$, and therefore we have $q_1 \xleftrightarrow{\varepsilon:y_1} q_1'$. We let $y_2 = x_2^{(0)} - 1$; by definition of $x_2^{(0)}$ we have $q_2 \xleftrightarrow{\varepsilon:y_2} q_2'$. As $\mathsf{ind}_\tau(y_2) = \mathsf{ind}_\tau(x_2^{(0)}) > \mathsf{ind}_\tau(x_1^{(0)})$ we have the third wanted transition $\tau \xrightarrow{(y_1,y_2):2i_T+1} \tau'$ in $\overline{\mathcal{T}}$, as wanted. By flipping τ and τ' and applying the same reasoning, we get the transition $r' \xrightarrow{\varepsilon:x^{(0)}-1} r$, as required (note that we also have $i_T < \mathsf{ind}_{\tau'}(x_1^{(0)}), \mathsf{ind}_{\tau'}(x_2^{(0)})$ by definition of i_T).

2. We assume w.l.o.g. that $\tau' < \tau$. By definition of i_T, we have that $\tau' <_{i_T+1} \tau$. Let $z = \tau[i_T + 1]$ (which belong to $[d_2]^*_{\text{odd}}$ if $\tau' <_{i_T+1} \tau$). As $z \leq x_2^{(0)}$, we have $q_2 \xrightarrow{\varepsilon:z} q_2'$. Also, $q_1 \xrightarrow{\varepsilon:x_1^{(0)}-1} q_1'$. Using point (3) of the definition of $\overline{\mathcal{T}}$, we have $\tau \xrightarrow{(x_1^{(0)}-1,z):2i_T+2} \tau'$. We conclude that $r \xrightarrow{\varepsilon:x^{(0)}} r'$.

b) Either $\mathsf{ind}_\tau(x_1^{(0)}) \leq i_T$ or $\mathsf{ind}_\tau(x_2^{(0)}) \leq i_T$. We assume w.l.o.g. that $\mathsf{ind}_\tau(x_1^{(0)}) < \mathsf{ind}_\tau(x_2^{(0)})$. We let $x^{(0)} = 2\mathsf{ind}_\tau(x_1^{(0)})$; note that $\tau \xrightarrow{(x_1^{(0)},x_2^{(0)}-1):x^{(0)}}$ and $\tau \xrightarrow{(x_1^{(0)}-1,x_2^{(0)}-1):x^{(0)}-1}$ in $\mathcal{T}$. We verify the two cases highlighted above.

1. We have $q_l \xleftrightarrow{\varepsilon:x_l^{(0)}-1} q_l$ for $l = 1, 2$. As $\tau[..i_T] = \tau'[..i_T]$, thanks to rule (1) in the definition of $\overline{\mathcal{T}}$, we have $\tau \xleftrightarrow{(x_1^{(0)}-1,x_2^{(0)}-1):x^{(0)}-1} \tau'$, and so we get $r \xleftrightarrow{\varepsilon:x^{(0)}-1} r'$, as required.
2. We have one of the transitions $q_1 \xrightarrow{\varepsilon:x_1^{(0)}} q_1'$ or $q_1' \xrightarrow{\varepsilon:x_1^{(0)}} q_1$; assume we are in the first case. Since $q_2 \xleftrightarrow{\varepsilon:x_2^{(0)}-1} q_2'$, we also have $\tau \xrightarrow{(x_1^{(0)},x_2^{(0)}-1):x} \tau'$, so we conclude that $r \xrightarrow{\varepsilon:x} r'$. ◀

This concludes the proof of Theorem 10.

5 Conclusions and open questions

We characterised objectives in $\mathrm{BC}(\mathbf{\Sigma}_2^0)$ with memory (or chromatic memory) $\leq k$ as those recognised by a well-identified class of automata. In particular, this gives the first known characterisation of (chromatic) memory for ω-regular objectives, and proves that it is decidable (in fact even in **NP**) to compute it. We also showed that for ω-regular objectives, memory (and also chromatic memory) coincides over finite or arbitrary game graphs. Finally, we settled (a strengthening of) Kopczyński's conjecture for $\mathrm{BC}(\mathbf{\Sigma}_2^0)$ objectives. We now discuss some directions for future work.

Exact complexity of computation of memory. We established that computing the (chromatic) memory of an ω-regular objective is in **NP**. In fact, computing the chromatic memory is **NP**-hard already for simple classes of objectives, such as Muller [Cas22] or safety ones [BFRV23]. However, no such hardness results are known for non-chromatic memory.

▶ **Question 1.** *Given a deterministic parity automaton $\mathcal{A}$ and a number k, can we decide whether the memory of $L(\mathcal{A})$ is $\leq k$ in polynomial time?*

This question is open already for the simpler case of regular open objectives (that is, those recognised by reachability automata). A related question is whether one can improve on the triply exponential bound that we establish for the size of games witnessing memory $> k$ (see Proposition 19).

Assymetric 1-to-2-player lifts. A celebrated result of Gimbert and Zielonka [GZ05] states that if for an objective W both players can play optimally using positional strategies in finite games where all vertices belong to one player, then W is bipositonal over finite games. This result has been extended in two orthogonal directions: to objectives where both players require finite chromatic memory [BRO$^+$22, BRV23] (symmetric lift for memory), and to ω-regular objectives where Eve can play positionally in 1-player games [CO24a] (asymmetric lift for positionality). In this work, we have not provided an asymmetric lift for memory, as

in most cases no such result can hold. For $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ objectives, it is known to fail already for positional objectives [GK22, Section 7]. For non-chromatic memory, it cannot hold for ω-regular objectives neither, because of the example described below.

▶ **Proposition 42.** *Let $\Sigma_n = \{1, \dots, n\}$. For every n, the objective*

$$W_n = \{w \in \Sigma_n^\omega \mid w \text{ contains two different letters infinitely often}\}$$

has memory 2 over games where Eve controls all vertices and memory n over arbitrary games.

Proof. The fact that W_n has memory n follows from [DJW97]. We show that in every game with objective W_n and all vertices controlled by Eve, if she can win, she has a winning strategy with memory 2. Let G be such a game. By prefix-independence of W_n, we can assume that Eve wins no matter what is the initial vertex of G. For each vertex $v \in V(G)$, let $\chi_1(v)$ be the smallest element in Σ_n such that there is a path starting in v that produces a colour c, and fix one such finite path $\pi_v^1 = v \xrightarrow{uc} v'$ of minimal length. Let $\chi_2(v)$ be the second smallest such element (which exists, as Eve wins the game), and fix a finite path π_v^2 of minimal length producing it. Note that if $v \xrightarrow{u} v'$ is a path in G, then $\chi_1(v) \leq \chi_1(v')$.

We define a 2-memory strategy for Eve as follows: when in a vertex v and memory state 1, she will take the first edge from π_v^1. If this edge has colour $\chi_1(v)$, we update the memory state to 2, and keep it 1 on the contrary. When in the memory state 2, she will take the first edge from π_v^1, and update the memory state to 1 if and only if this edge has colour $\chi_2(v)$.

We show that this strategy ensures the objective W. Let $v_0 \xrightarrow{c_1} v_1 \xrightarrow{c_2} \dots$ be an infinite play consistent with this strategy. Let $a = \limsup \chi_1(v_i)$. We claim that we produce both a and a colour $> a$ infinitely often. Let i_0 be large enough so that $\chi_1(v_i) > a$ for $i \geq i_0$. Note that in this case, if $v_i \xrightarrow{u} v_{i'}$ is a path that does not contain colour $\chi_2(v_i)$, then $\chi_2(v_{i'}) = \chi_2(v_i)$. If in step i we are in memory state 1, in exactly $|\pi^1_{v_i}|$ steps we will produce output $\chi_1(v_i)$ and change the memory state to 2. Likewise, by the remark above, if we are in the memory state 2, in $|\pi^2_{v_i}|$ steps we will produce $\chi_2(v_i) > a$ and change to the memory state 1. This concludes the proof. ◀

In his PhD thesis, Vandenhove conjectures that an asymmetric lift for chromatic memory holds for ω-regular objectives [Van23, Conjecture 9.1.2]. This question remains open.

▶ **Question 2.** *Is there an ω-regular objective with chromatic memory k over games where Eve controls all vertices and chromatic memory $k' > k$ over arbitrary games?*

Further decidability results for memory. As mentioned in the introduction, many extensions of ω-automata (including deterministic ω-Turing machines and unambiguous ω-petri nets [FSJ+22]) compute languages that are in $\mathrm{BC}(\mathbf{\Sigma}^0_2)$. We believe that our characterisation may lead to decidability results regarding the memory of objectives represented by these models.

Objectives in Δ^3_0. Some of the questions answered in this work in the case of $\mathrm{BC}(\mathbf{\Sigma}^0_2)$ objectives are open in full generality, for instance, the generalised Kopczyński's conjecture. A reasonable next step would be to consider the class $\mathbf{\Delta}^0_3 = \mathbf{\Sigma}^0_3 \cap \mathbf{\Pi}^0_3$. Objectives in $\mathbf{\Delta}^0_3$ are those recognised by max-parity automata using infinitely many priorities [Skr13]. Our methods seem appropriate to tackle this class, however, we have been unable to extend the extraction lemma (Lemma 12) used in the proof of Section 3.1.

References

BCRV24 Patricia Bouyer, Antonio Casares, Mickael Randour, and Pierre Vandenhove. Half-positional objectives recognized by deterministic Büchi automata. *Log. Methods Comput. Sci.*, volume 20(3), 2024. doi: http://doi.org/10.46298/LMCS-20(3:19)2024.

BFRV23 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: http://doi.org/10.4230/LIPIcs.ICALP.2023.118.

BL69 J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, volume 138:295–311, 1969. doi: http://www.jstor.org/stable/1994916.

BL23 Udi Boker and Karoliina Lehtinen. When a little nondeterminism goes a long way: An introduction to history-determinism. *ACM SIGLOG News*, volume 10(1):24–51, 2023. doi: http://doi.org/10.1145/3584676.3584682.

BRO+22 Patricia Bouyer, Stéphane Le Roux, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. Games where you can play optimally with arena-independent finite memory. *Log. Methods Comput. Sci.*, volume 18(1), 2022. doi: http://doi.org/10.46298/LMCS-18(1:11)2022.

BRV22 Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. The true colors of memory: A tour of chromatic-memory strategies in zero-sum games on graphs (invited talk). In *FSTTCS*, volume 250 of *LIPIcs*, pages 3:1–3:18. 2022. doi: http://doi.org/10.4230/LIPICS.FSTTCS.2022.3.

BRV23 Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs. *TheoretiCS*, volume 2, 2023. doi: http://doi.org/10.46298/THEORETICS.23.1.

Cas22 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: http://doi.org/10.4230/LIPIcs.CSL.2022.12.

CCL22 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: http://doi.org/10.4230/LIPIcs.ICALP.2022.117.

CDK93 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.*, volume 46(6):301–308, 1993. doi: http://doi.org/10.1016/0020-0190(93)90069-L.

CFH14 Thomas Colcombet, Nathanaël Fijalkow, and Florian Horn. Playing safe. In *FSTTCS*, volume 29 of *LIPIcs*, pages 379–390. 2014. doi: http://doi.org/10.4230/LIPIcs.FSTTCS.2014.379.

CFH22 Thomas Colcombet, Nathanaël Fijalkow, and Florian Horn. Playing safe, ten years later. *CoRR*, volume abs/2212.12024, 2022. doi: http://doi.org/10.48550/arXiv.2212.12024.

CHVB18 Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors. *Handbook of Model Checking*. Springer, Cham, 2018. doi: http://doi.org/10.1007/978-3-319-10575-8_2.

CN06 Thomas Colcombet and Damian Niwiński. On the positional determinacy of edge-labeled games. *Theor. Comput. Sci.*, volume 352(1-3):190–196, 2006. doi: http://doi.org/10.1016/j.tcs.2005.10.046.

CO24a Antonio Casares and Pierre Ohlmann. Positional ω-regular languages. In *LICS*, pages 21:1–21:14. ACM, 2024. doi: http://doi.org/10.1145/3661814.3662087.

CO24b Antonio Casares and Pierre Ohlmann. Positional ω-regular languages. *CoRR*, volume abs/2401.15384, 2024. doi: http://doi.org/10.48550/ARXIV.2401.15384.

CO25 Antonio Casares and Pierre Ohlmann. Characterising memory in infinite games. *Logical Methods in Computer Science*, volume Volume 21, Issue 1:28, 2025. doi: http://doi.org/10.46298/lmcs-21(1:28)2025.

DJW97 Stefan Dziembowski, Marcin Jurdziński, and Igor Walukiewicz. How much memory is needed to win infinite games? In *LICS*, pages 99–110. 1997. doi: http://doi.org/10.1109/LICS.1997.614939.

FSJ+22 Olivier Finkel, Michał Skrzypczak, Ryszard Janicki, Slawomir Lasota, and Natalia Sidorova. On the expressive power of non-deterministic and unambiguous petri nets over infinite words. *Fundamenta Informaticae*, volume 183(3-4):243–291, 2022. doi: http://doi.org/10.3233/FI-2021-2088.

GH82 Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *STOC*, pages 60–65. 1982. doi: http://doi.org/10.1145/800070.802177.

GK22 Hugo Gimbert and Edon Kelmendi. Submixing and shift-invariant stochastic games, 2022. doi: https://arxiv.org/abs/1401.6575v2. Version 2.

GZ05 Hugo Gimbert and Wieslaw Zielonka. Games where you can play optimally without any memory. In *CONCUR*, volume 3653, pages 428–442. 2005. doi: http://doi.org/10.1007/11539452_33.

Kop08 Eryk Kopczyński. *Half-positional Determinacy of Infinite Games*. Ph.D. thesis, University of Warsaw, 2008.

Koz24 Alexander Kozachinskiy. Energy games over totally ordered groups. In *CSL*, volume 288 of *LIPIcs*, pages 34:1–34:12. 2024. doi: http://doi.org/10.4230/LIPICS.CSL.2024.34.

Ohl21 Pierre Ohlmann. *Monotonic graphs for parity and mean-payoff games. (Graphes monotones pour jeux de parité et à paiement moyen)*. Ph.D. thesis, Université Paris Cité, France, 2021. doi: https://tel.archives-ouvertes.fr/tel-03771185.

Ohl23 Pierre Ohlmann. Characterizing positionality in games of infinite duration over infinite graphs. *TheoretiCS*, volume Volume 2, 2023. doi: http://doi.org/10.46298/theoretics.23.3.

OS24 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: http://doi.org/10.4230/LIPICS.STACS.2024.54.

SE89 Robert S. Streett and E. Allen Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Inf. Comput.*, volume 81(3):249–264, 1989. doi: http://doi.org/10.1016/0890-5401(89)90031-X.

Skr13 Michal Skrzypczak. Topological extension of parity automata. *Inf. Comput.*, volume 228:16–27, 2013. doi: http://doi.org/10.1016/J.IC.2013.06.004.

Van23 Pierre Vandenhove. *Strategy complexity of zero-sum games on graphs. (Complexité des stratégies des jeux sur graphes à somme nulle)*. Ph.D. thesis, University of Mons, Belgium, 2023. doi: https://tel.archives-ouvertes.fr/tel-04095220.

Wal96 Igor Walukiewicz. Pushdown processes: Games and model checking. In *CAV*, volume 1102 of *Lecture Notes in Computer Science*, pages 62–74. 1996. doi: http://doi.org/10.1007/3-540-61474-5_58.

A Proof of the structuration theorem

We give a proof of Theorem 4.

▶ **Theorem 4** (Adapted from Lemma 3.4 in [CO25]). *Let G be a well-founded pointed graph satisfying an objective W which is assumed to have (chromatic) memory $\leq k$ over games of size $\leq 2^{|G|}$. There is a (chromatic) k-blowup G' of G which is well-founded, k-wise ε-complete, and satisfies W.*

The idea is to use choice arenas, which were introduced in Ohlmann's PhD thesis [Ohl21, Section 3.2 in Chapter 3] for positionality, and then adapted to memory in [CO24a].

Proof. Let H be the game defined as follows. The set of vertices is $V(G) \sqcup \mathcal{P}_{\neq\varnothing}(V(G))$, where $\mathcal{P}_{\neq\varnothing}(V(G))$ is the set of non-empty subsets X of $V(G)$, partitioned into $V_{\text{Adam}} = V(G)$ and $V_{\text{Eve}} = \mathcal{P}_{\neq\varnothing}(V(G))$. The initial vertex is v_0, the one of G. Then the edges are given by taking those of G, and then adding $v \xleftrightarrow{\varepsilon} X$ whenever $v \in X$. The objective is W.

In words, when playing in H, Adam follows a path of his choice in G, except that at any point, he may choose a set X containing the current vertex v, and allow Eve to continue the game from any vertex of her choice in X. In some sense, Adam can hide the current vertex; this is especially true if Eve is required to play with finite memory.

Since G satisfies W, Eve wins, simply by going back to the previous vertex v every time Adam picks an edge $v \xrightarrow{\varepsilon} X$. (Formally, the corresponding winning strategy has vertices $V(G) \sqcup \{(v, X) \mid v \in X\}$, projection $\pi(v) = v$ and $\pi(v, X) = X$, edges $E(G) \cup \{v \xleftrightarrow{\varepsilon} (v, X) \mid v \in X\}$, and initial vertex v_0.) Therefore by our assumption on W, there is a (chromatic) winning strategy S with memory k, i.e. $V(S) = V(H) \times k$.

Now we define G' by $V(G') = V(G) \times k$, initial vertex (v_0, m_0), the initial vertex of S, and with the edges from $E(S) \cap (V(G') \times (\Sigma \cup \{\varepsilon\}) \times V(G'))$, together with edges $(v, m) \xrightarrow{\varepsilon} (v', m)$ whenever there is $X \ni v, v'$ such that $(X, m) \xrightarrow{\varepsilon} (v', m)$ in S.

We prove that G' satisfies the conclusion of the theorem, except for well-foundedness which is dealt with below.

G' is a k-blowup of G. This is because S is a strategy, and vertices in $V(G)$ belong to Adam in H. Therefore, for each $(v, m) \in V(G')$, and each edge $v \xrightarrow{c} v'$ in G, $v \xrightarrow{c} v'$ is also an edge in H, thus there is m' such that $(v, m) \xrightarrow{c} (v', m')$ is an edge in G'.

In the chromatic case, G' is chromatic. This is because S is chromatic, therefore by definition of G', it is chromatic with the same chromatic update function.

G' is k-wise ε-complete. Since it is a graph, we should prove that for each v, v' and each m, either $(v, m) \xrightarrow{\varepsilon} (v', m)$ or $(v', m) \xrightarrow{\varepsilon} (v, m)$ in G. This follows from applying the definition of G' to $X = \{v, v'\}$, since either $(X, m) \xrightarrow{\varepsilon} v$ or $(X, m) \xrightarrow{\varepsilon} v'$ is an edge in S (because S is without dead-ends).

G' satisfies W. This is because S is a winning strategy, and every path in G' corresponds to a path in S, by replacing each edge $(v, m) \xrightarrow{\varepsilon} (v', m)$ by $(v, m) \xrightarrow{\varepsilon} X \xrightarrow{\varepsilon} (v', m)$.

There remains a slight technical difficulty, which is that G' may have $\xrightarrow{\varepsilon}$-cycles (in fact, it even has all ε-self-loops, by applying the definition to X being a singleton). However for each memory state m, the relation $\xrightarrow{\varepsilon}$ has the property that for every subset of states X, there is $v \in X$ (which should be seen as a minimal element) such that every $v' \in X$ satisfies $(v', m) \xrightarrow{\varepsilon} (v, m)$ in G'.

Therefore for each m, and each $X \subseteq V(G)$ such that $X \times \{m\}$ is a strongly connected component for $\xrightarrow{\varepsilon}$ in G', we pick an arbitrary strict well-order $\to$ over X which extends the ε-edges already present in G over X. This is possible because G is assumed G to be

well-founded (and it is necessary so that the obtained graph remains a k-blowup of G). Finally, we rewire ε-edges over $X \times \{m\}$ so that they correspond to $\to$; it is not hard to see that the above points are not broken by this construction. ◀