Eve-Positional Languages: Putting Order into Büchi Automata

Olivier Idir
Université Paris Cité, CNRS, IRIF, France

Abstract

An ω-regular language is Eve-positional if, in all games with this language as objective, the existential player can play optimally without keeping any information from the previous moves. This notion plays a crucial role in verification, automata theory and synthesis.

Casares and Ohlmann recently gave several characterisations of Eve-positionality of ω-regular languages. For this, they introduce the notion of ε-complete parity automaton and show (among other results) that an ω-regular language is Eve-positional if and only if it can be recognised by some ε-completion of a deterministic parity automaton. Colcombet and Idir built on their work, and obtained a more direct algebraic characterisation of Eve-positionality.

We introduce a new formalism that characterises the Eve-positional languages, consisting of a restriction of non-deterministic Büchi automata. This allows us to complete a missing implication in Casares and Ohlmann's work. We then use this formalism to describe a determinization procedure for non-deterministic Büchi automata recognising such languages, with size blow-up at most factorial. We also show that this construction is state-wise optimal for languages over sufficiently complete alphabets.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects

Keywords and phrases Büchi automata, parity automata, Eve-positional language, ε-complete automata, positional strategy, ordered Büchi automata

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.33

Related Version *Full Version*: `https://arxiv.org/abs/2602.09896` [13]

Acknowledgements I thank Thomas Colcombet for his many insights during the elaboration of this paper. I also thank Antonio Casares and Pierre Ohlmann for their enthusiastic answers concerning their article.

1 Introduction

1.1 Context

The problem of reactive synthesis is the following: given a system equipped with a formal specification and interacting with its environment for a possibly infinite amount of time, we want to design a controller that can react to the environment while ensuring that its specification is met.

A formalism frequently used for this is that of games on graphs. In these games, two players, Eve and Adam, move a token along the edges of an edge-coloured directed graph called an arena. Its vertices are partitioned between those belonging to Eve and those belonging to Adam, and when the token arrives in a vertex, its owner chooses the next edge, which the token will follow. This interaction continues infinitely, thus producing an infinite path through the arena and a corresponding infinite sequence of colours w (also called an ω-word). A play is won by Eve if the corresponding word belongs to some set W, called her objective. Else, it is won by Adam. Over the games we consider, it always stands that one of the players has a winning strategy, which describes a way to win the game even when this strategy is known in advance by the opponent.

The interaction between the system and its environment can be modeled by such a game, where the specification provides the winning objective W [4]. In such a context, the complexity of the player's winning strategy is crucial, as games with simple winning strategies are generally easier to solve algorithmically and result in controllers that can be represented more succinctly.

ω-regular languages. An ω-regular language is a set of ω-words that can be described by many equivalent formalisms (e.g., automata that can be either deterministic, non-deterministic, or alternating; one-way or two-way; with Muller, Parity, Rabin, or Streett acceptance conditions; or by monadic-second-order logic). The class of ω-regular languages plays a pivotal role in many decision procedures and mathematical arguments related to verification, automata theory, controller synthesis, etc.

Eve-positionality. One of the simplest cases, when attempting to solve a game, is when the winning strategy is positional. This means that the corresponding player does not need any memory of the past to play optimally: they can make their decision simply by looking at the current state of the game. Note that this memoryless property need not be the same for both players. In this work, we are considering the objectives W such that over all graphs, if Eve wins, she can do so with a positional strategy. They are called Eve-positional languages. Until recently, little was known about this class. Major progress was made in the last few years, but there was still lacking a convenient formalism to handle them, which we propose here.

1.2 Contributions

In this paper, we introduce the notion of ordered Büchi automaton: these are non-deterministic Büchi automata such that Büchi-ε-transitions form a total order on the states (these automata are very close cousins of the ε-complete Büchi automata introduced by Casares and Ohlmann [5]).

1. We establish that languages recognised by ordered Büchi automata are exactly the Eve-positional ω-regular languages (Theorem 9),
2. We propose a quadratic transformation from an ε-complete non-deterministic parity automaton towards an ordered Büchi automaton recognising the same language (Theorem 14).
3. We provide a generic determinization procedure from ordered Büchi automata to deterministic parity automata, with at most factorial blow-up (Theorem 20). This determinization furthermore proves state-wise optimal over some languages (in the full version only, [13, Lemma 24])

Our first contribution implies taking a more detailed look at the ε-complete Büchi automata. They consist of parity automata with an underlying tree-like structure, such that they can be saturated with ε-transitions following this structure without changing the recognised language. Intuitively, this structure allows us to keep track of how advantageous is the prefix read so far. In order to gain a clearer view of the action of each letter, we quotient the states mutually reachable by non-Büchi ε-transitions and follow the approach of transfer graphs (as in [1]). This allows us to view each letter as the set of transitions it induces in the automaton (called a tile), and observe that these tiles are upwards-closed for some order $\leqslant$ over the transitions. We obtain that the corresponding automata, called ordered Büchi automata, recognise exactly the Eve-positional languages. We believe that this formalism is useful both thanks to the ease with which it allows representing Eve-positional languages, and as it provides a better structural understanding of these languages.

We then propose a transformation from any ε-complete non-deterministic parity automaton with n states and priorities $[0, 2k-1]$ towards an ordered Büchi automaton recognising the same language, with nk states. According to Casares and Ohlmann, all non-deterministic parity automata recognising an Eve-positional language are ε-completable (Theorem 1); this transformation thus applies to all parity automata recognising such languages. This result allows, in conjunction with our first contribution, to complete a missing implication in their work: if a non-deterministic automaton is ε-complete, then it recognises an Eve-positional language. Note that in the general case, when going from parity to Büchi automata, the known lower bound for this transformation is also of $\Theta(kn)$ [21]; therefore, since being ε-completable is purely a property of the recognised language, it is possible to go from an ε-completable automaton to an ordered Büchi automaton without any additional state blow-up beyond what is already needed.

Our last contribution concerns the algorithmic investigation of ordered Büchi automaton, and more precisely the question of determinizing such automata. We obtain that a n-state ordered Büchi automaton can be effectively transformed into a deterministic parity automaton recognising the same language, with at most $1 + \sum_{i=0}^{n-1} i!$ states. We deduce from this that all Eve-positional languages can be determinized with at most factorial blow-up, compared to $O((1.64n)^n)$ in the general case [20]. It is even possible to exploit some conditions on the alphabet in order to further reduce this blow-up. Notably, using a technique introduced in [10], we can establish the minimality of the resulting number of states over some languages, such as the Rabin language with n pairs and all the possible letters with different behaviours (called the n-complete Rabin).

1.3 Related works

Bipositionality. When looking at positionality, one can consider the bipositional languages, which are both Eve-positional and Adam-positional. We know from [9] that the prefix-independent languages that are bipositional are exactly the parity languages, even when considering languages that are not necessarily ω-regular. There have been some attempts to generalize the latter result [3], but a more general characterisation was only established in [5, Theorem 7.1].

Eve-positionality. The Eve-positionality of some central classes was established as early as the 90s, such as parity languages [11] and Rabin languages [14]. However, the first thorough analysis of Eve-positionality was undertaken by Kopczyński [15]. He notably established that it is decidable whether a prefix-independent ω-regular language is Eve-positional [15, 16], albeit with a quite slow procedure (in $O(n^{O(n^2)})$) that is not very informative on the reason why the language is Eve-positional. More generally, despite these breakthroughs, he did not obtain any general characterisation of Eve-positional languages.

Some partial characterisations were established in the following decade on specific subclasses of ω-regular languages. Colcombet, Fijalkow, and Horn established a semantic characterisation of Eve-positionality for safety languages [7], that is, languages where it is known in finite time whether some word is out of the language. A characterisation for a larger class was actually obtained a few years later by Bouyer, Casares, Randour, and Vandenhove [2], as they obtained a semantic characterisation of Eve-positional languages recognised by deterministic Büchi automata. This, however, does not correspond to the general ω-regular case, and for instance fails to cover the deterministic coBüchi case.

Recently, using the well-monotonic universal graphs initially developed by Ohlmann [17], he and Casares gave several other characterisations for Eve-positionality [5]. They also proposed new decision procedures, in polynomial time over the size of a deterministic automaton recognising the target language. A key notion in their work is that of ε-completable automata, which they introduced in the latter article. Let us recall some of these characterisations here:

▶ **Theorem 1** (3.1 and 3.2 in [5]). *Let $L \subseteq \Sigma^\omega$ be an ω-regular language. The following conditions are equivalent:*

1. *L is Eve-positional over all arenas (potentially infinite and containing ε-moves).*
2. *L is Eve-positional over finite ε-free Eve-only arenas.*
3. *There exists a ε-completable deterministic parity automaton recognising L.*
4. *All (non-deterministic) parity automata for L are ε-completable.*

These articles notably provide the first characterisations achieved over all ω-regular languages. They however require one to reason with an automaton recognising the language, rather than directly with the language and inclusion properties.

Using their characterisation and notably the 1-to-2 player lift, Colcombet and Idir obtained an algebraic characterisation of these languages [8], where a language is Eve-positional if and only if it satisfies three algebraic conditions (see Theorem 3). Each condition is of the form "If certain word(s) belong to L, then at least one word obtained from their factors also belongs to L". This supports the idea that Eve-positionality can be intuitively understood as establishing a total preference order among factors. Furthermore, the first two of their conditions correspond exactly to the first two items of the characterisation obtained by Bouyer et al., and the third one is a generalisation of their third item that allows one to consider all Eve-positional ω-regular languages.

This last characterisation, however, does not provide a convenient formalism to handle Eve-positional languages. Building on these last two results, we propose such a formalism, exactly capturing Eve-positional ω-regular languages and that comes with a determinization procedure.

1.4 Layout of the article

We first recall a few standard definitions in Section 2, along with the formalism of tiles to describe automata. We then define the desired ordered Büchi automata in Section 3. This section will also give a broad overview of this article's results, along with examples and a key intuition as to why these ordered Büchi automata characterise Eve-positionality.

From Section 4 onward, we will focus on the more technical aspects of our contribution. Section 4 describes some semantic properties of these automata, and establishes the equivalence between ordered Büchi automata and Eve-positional languages. It notably details the transformation from a parity automaton recognising an Eve-positional language towards an ordered Büchi automaton of same language (up to letter renaming). Finally, Section 5 describes the determinization procedure from an ordered Büchi automaton to a deterministic parity automaton recognising the same language, along with a size-optimality result.

The missing proofs can be found in the full version of the article ([13]).

2 General definitions

2.1 Basic notations

For a set A, its powerset is denoted $\mathcal{P}(A)$.

An *alphabet* is a finite set Π, whose elements are called letters. The sets Π^* and Π^ω respectively denote the sets of finite *words* and infinite *words* of length ω over Π (also denoted *ω-words*). Subsets of Π^* and Π^ω are called languages. An ω-word is *ultimately-periodic* if it is of the form uv^ω, with u and v finite words, and v is non-empty. The set of non-empty finite words is denoted Π^+, and the *empty word* is denoted ι. A language L is *prefix-independent* if, for all $u \in \Pi^+$ and $w \in \Pi^\omega$, $w \in L$ if and only if $uw \in L$. A *residual* of a language L is a language $u^{-1}L := \{w \mid uw \in L\}$, where u is a finite word.

The length of a finite word u is written $|u|$. For a (possibly infinite) word u and $n \in \mathbb{N}$, u_n denotes its letter at index n.

An *index* $[i, j]$ is a non-empty finite interval of natural numbers $J = \{i, i+1, \dots, j\} \subseteq \mathbb{N}$. For $n \in \mathbb{N}$, we denote the index $[0, n-1]$ as $[n]$. Elements $c \in J$ are called *priorities*. We say that an infinite sequence of priorities $(c_n)_{n\in\mathbb{N}}$ is *parity accepting* (or simply *accepting*) if $\liminf_{n\to\infty} c_n \equiv 0 \mod 2$, else it is *parity rejecting* (or *rejecting*).

For a function $f : A \to B$ and $b \in B$, we denote by $f^{-1}(b)$ the set $\{a \in A \mid f(a) = b\}$.

2.2 Parity automata

A *J-automaton* (or *automaton*, or more generally a *non-deterministic J-parity automaton*), with J an index, is a tuple $A = (\Pi, Q_A, I_A, \Delta_A)$, where Π is an alphabet, Q_A the finite set of *states*, $I_A \subseteq Q_A$ a set of *initial states*, and $\Delta_A \subseteq Q_A \times \Pi \times J \times Q_A$ is the transition relation. A *transition* $(q, a, c, q') \in \Delta_A$ is said to be from the state q to the state q', over the letter a and with priority c. By default, all automata in consideration are complete[a], that is, for each state $q \in Q_A$ and letter $a \in \Pi$, there is at least one transition from q over a in Δ_A. We also consider automata to be restricted by default to the states accessible from the initial states by a sequence of transitions. The automaton A is said *deterministic* if I_A is a singleton and Δ_A can be described as a function from $Q_A \times \Pi$ to $J \times Q_A$. When an automaton A is known from the context, we skip the subscript and write just Q, Δ, etc.

We denote by $\xrightarrow{a:c}$ the relation induced by transitions of parity c over a letter a. That is, $p\xrightarrow{a:c}q$ denotes the existence of a transition $\delta = (p, a, c, q) \in \Delta$. We extend this notation to paths over words, where the label corresponds to the minimum priority encountered along this path.

For $w \in \Pi^*$ a word, the function $\Delta(q, w)$ is defined inductively over the length of w as the set of states accessible from q after a path over w, with $\Delta(q, \iota) := \{q\}$.

A *run* $\rho = (\delta_i)_{i\in\mathbb{N}} \in \Delta^\omega$ over a word $w \in \Pi^\omega$ is a sequence of transitions $(q_i, w_i, c_i, q_{i+1})_{i\in\mathbb{N}}$ such that $q_0 \in I$. A *partial run* is a finite prefix of a run. A run is called *accepting* if its priority sequence $(c_i)_{i\in\mathbb{N}}$ is parity accepting (recall that we consider the min-parity condition). A word $w \in \Pi^\omega$ is *recognised* by A if there exists an accepting run over w in A. The set of all recognised words of an automaton is called its *language*, and is denoted $\mathcal{L}(A)$. Note that $\mathcal{L}(A)$ admits a finite number of different residuals (at most one for each state in Q). A language $L \subseteq \Pi^\omega$ is called *ω-regular* if there exists an automaton recognising it.

We are particularly interested in the $[2]$-automata, which are also called *Büchi automata*. It is well-known that all ω-regular languages can be recognised by Büchi automata [18]. In such an automaton, a transition is called a *Büchi transition* if it has priority 0.

An *ε-automaton* over an alphabet Π (or automaton with ε-transitions) is an automaton over the alphabet $\Pi_\varepsilon := \Pi \uplus \{\varepsilon\}$, where $\varepsilon \notin \Pi$ is a distinguished letter. The *language* of an ε-automaton A is the set of words $w \in \Pi^\omega$ such that there exists $w' \in (\Pi \uplus \{\varepsilon\})^\omega$ accepted by A and such that w is obtained from w' by removing all occurrences of the letter ε.

[a] In the examples that follow, any missing transition is assumed to instead lead to a rejecting sink state.

Tile automata

Another equivalent formalism for automata is to consider, for each letter, the set of transitions it entails. That is, for a J-automaton $A = (\Pi, Q, I, \Delta)$ and a letter $a \in \Pi$, we can consider $t_a := \{(p, c, q) \mid (p, a, c, q) \in \Delta\} \subseteq Q \times J \times Q$. We call this subset a *tile*, and its elements are still called transitions. For t a tile, a transition of t is called *horizontal* if it is of the form (q, c, q) for some $q \in Q$.

Note that we can equip the set of all possible tiles $\Sigma := \mathcal{P}(Q \times J \times Q)$ with a semigroup structure by setting the *product* $t_1 \cdot t_2$ for tiles $t_1, t_2 \in \Sigma$ as

$$t_1 \cdot t_2 := \{(p, \min(c_1, c_2), r) \mid \exists q,\ (p, c_1, q) \in t_1,\ (q, c_2, r) \in t_2\}\ .$$

This semigroup is actually a monoid, as $1_\Sigma := \{(q, \max(J), q) \mid q \in Q\}$ is a neutral element for $\cdot$. Intuitively, the product allows us to consider directly, as a single tile, the behaviour of a non-empty word. This notably allows to observe more directly the ω-semigroup for a given language.

An infinite sequence of tiles $w = (w_j)_{j\in\mathbb{N}}$ thus directly corresponds to a word $w_\Pi \in \Pi^\omega$ (or to many words, in the case where different letters map to the same tile, but they then all have the same behaviour in A), and a *run* over w is an infinite sequence of transitions $(\delta_j = (q_j, r_j, q_{j+1}))_{j\in\mathbb{N}}$ with $\delta_j \in w_j$ for all j and $q_0 \in I$.

An automaton can be directly described by its alphabet Γ of tiles, instead of via its set Δ of transitions. In that case, we can abstract away the corresponding letters, and only remember the induced tiles. We then call this automaton a *tile automaton*, described as a tuple $\mathcal{A} = (Q, I, \Gamma)$, where Q is a finite set of states, $I \subseteq Q$ is the set of initial states, and $\Gamma \subseteq \Sigma$ is a set of tiles[b]. Its *language* is still denoted $\mathcal{L}(\mathcal{A})$.

▶ Remark 2. Note that for $\mathcal{A} = (Q, I, \Gamma)$ an automaton, for $\mathcal{A}' := (Q, I', \Gamma')$ such that $I' \subseteq I$ and $\Gamma' \subseteq \Gamma$, we have that $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$. Indeed, any accepting run in $\mathcal{A}'$ still exists in $\mathcal{A}$.

2.3 Eve-positional languages

In this article, we focus on a specific class of ω-regular languages.

A language L is called *Eve-positional* if, whenever Eve wins a game with objective L, she can win with a positional strategy. In most of this article, however, we will not discuss games, and describe these languages via the characterisation established by Colcombet and Idir.

▶ **Theorem 3** ([8]). *An ω-regular language $L \subseteq \Pi^\omega$ is Eve-positional if and only if, for all u, u' finite words (possibly empty), v, v' non-empty finite words, and w, w' infinite words, the following local preference properties hold:*

1. *if $uw \in L$ and $u'w' \in L$, then $uw' \in L$ or $u'w \in L$,*
2. *if $uvw \in L$, then $uv^\omega \in L$ or $uw \in L$, and*
3. *if $u(vv')^\omega \in L$, then $uv^\omega \in L$ or $uv'^\omega \in L$.*

When talking of Eve-positional languages, a notion of particular relevance is the one of ε-complete automata, introduced by Casares and Ohlmann in [5].

[b] We can omit the letter corresponding to the tile in this representation, as two letters with the same associated tile are functionally equivalent.

▶ **Definition 4** (Casares, Ohlmann). *A parity ε-automaton $A = (\Pi, Q, I, \Delta)$ is* ε-complete *if its index J is of the form $[2d+2]$, and*

1. *the relations $\xrightarrow{\varepsilon:1}, \xrightarrow{\varepsilon:3}, \dots, \xrightarrow{\varepsilon:d+1}$ define total preorders, each refining the previous one;*
2. *for each even $c \in J$, the relation $\xrightarrow{\varepsilon:c}$ is the strict variant of $\xrightarrow{\varepsilon:c+1}$: for any q, q', it holds that $q \xrightarrow{\varepsilon:c} q'$ if and only if $q' \xrightarrow{\varepsilon:c+1} q$ does not hold.*

Note that in the case of an ε-complete automaton, I can be assumed downwards-closed for the total preorder $\xrightarrow{\varepsilon:2d+1}$.

An automaton is *ε-completable* if it is possible to add ε-transitions to it to make it ε-complete, without enlarging the recognised language.

They notably established (Theorem 1) that if an automaton A recognises an Eve-positional language, it is ε-completable. However, they only proved the converse implication for deterministic automata: if some non-deterministic automaton A is ε-completable, their result does not state that $\mathcal{L}(A)$ is Eve-positional.

3 Ordered Büchi tiles and overview of the contributions

In this section, we introduce the ordered Büchi automata. They are Büchi automata with a total order $\sqsubseteq$ over their set of states, such that at any point during a run it is always possible to decrease along $\sqsubseteq$ whilst seeing an additional Büchi transition. We then give a few examples, before giving an intuition of their link with Eve-positional languages. We end this section with an overview of our contributions, and a discussion as to the interest of ordered Büchi automata.

3.1 Ordered Büchi languages

Let Q be a finite set of states, and $\Sigma := \mathcal{P}(Q \times \{0,1\} \times Q)$ its set of Büchi tiles. We fix a total order $\sqsubseteq$ over Q.

Let $\delta = (p, c, q), \delta' = (p', c', q')$ be two transitions in $Q \times \{0,1\} \times Q$. We say that $\delta \leqslant \delta'$ if $p \sqsubseteq p'$, $q' \sqsubseteq q$ (note the swap of order), and, if $(p, q) = (p', q')$, $c \leqslant c'$. We observe that $\leqslant$ is a partial order. This captures the idea of downward Büchi ε-transitions : intuitively $\delta \leqslant \delta'$ if the transition δ' can be realized with ε-transitions decreasing along $\sqsubseteq$ and the transition δ[c].

We define $\Sigma_\sqsubseteq$ as the set of upwards-closed tiles under the order $\leqslant$, and the tile $1_\sqsubseteq$ as the upward closure of 1_Σ.

▶ **Lemma 5.** *$(\Sigma_\sqsubseteq, \cdot)$ is a sub-semigroup of $(\Sigma, \cdot)$, and $(\Sigma_\sqsubseteq, \cdot, 1_\sqsubseteq)$ is a monoid*[d].

When considering an automaton (Q, I, Γ) with $\Gamma \subseteq \Sigma_\sqsubseteq$, we may always assume that I is downwards-closed, as the first transition taken can always be taken from some $q' \sqsubseteq q$ for some $q \in I$. More generally, due to this upward closure by $\leqslant$, if a partial run over a word $w \in {\Sigma_\sqsubseteq}^+$ ends in some state q, then for all $q' \sqsubset q$, there exists a run over w ending in q' that sees a Büchi transition (provided by upward closure in the last tile).

An *ordered Büchi automaton* consists of an automaton $\mathcal{A} := (Q, \sqsubseteq, I, \Gamma)$ such that (Q, I, Γ) is a Büchi tile automaton, and $\sqsubseteq$ is a total order over Q such that I is downwards-closed for $\sqsubseteq$ and $\Gamma \subseteq \Sigma_\sqsubseteq$.

---

[c] The case where $c' = 1$ yet $c = 0$ cannot be obtained in this manner, but is still included so that $\leqslant$ is compatible with upward closure under $\cdot$, as shown in Lemma 5.
[d] It is not a submonoid of $(\Sigma, \cdot, 1_\Sigma)$ since the units do not coincide.

An ω-regular language L over some alphabet Π is said to be an *ordered Büchi language* if it is the language of some ordered Büchi automaton, up to renaming the letters in Π. That is, there exists an ordered Büchi automaton $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ and a morphism $f : \Pi \to \Gamma$ (extended to words) such that for all $w \in \Pi^\omega$, $w \in L$ if and only if $f(w) \in \mathcal{L}(\mathcal{A})$.

We finally introduce the notion of skeleton of a tile, that allows for a more succinct description. We can define a unique minimal generator for each tile of $\Sigma_\sqsubseteq$, which we shall call skeleton.

▶ **Lemma 6** (existence of the skeleton). *For all tiles t in $\Sigma_\sqsubseteq$, there exists a unique minimal (for inclusion) tile s such that t is the upward closure of s. This tile s furthermore satisfies that for all distinct transitions $p \xrightarrow{s} p'$ and $q \xrightarrow{s} q'$, both $p \neq q$ and $p' \neq q'$.*

Let t be a tile in $\Sigma_\sqsubseteq$, the minimal tile s as described in Lemma 6 is called the *skeleton* of t and is denoted $\mathrm{skel}(t)$.

3.2 Examples

We now provide a few examples of such ordered Büchi automata, in order to clarify the objects and how they are handled.

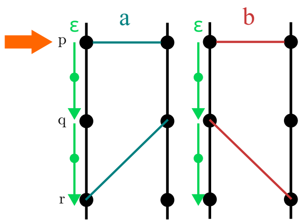


Figure 1 An ordered Büchi automaton recognising the words with infinitely often a factor aa and only finitely often bb.

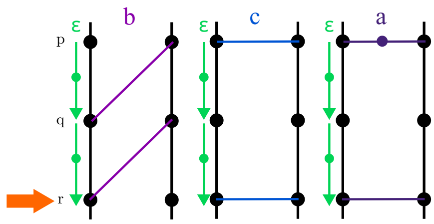


Figure 2 An ordered Büchi automaton recognising the words with infinitely many b's, or a factor bb and infinitely many a's.

In these figures, ordered Büchi automata are represented by the skeletons of the letters in their corresponding alphabet, with the order $\sqsubseteq$ represented vertically (the highest element being on top). The highest element of the starting set I is designated by the orange arrow, and Büchi transitions are represented as lines with a circle in the middle. The ε-labelled arrows on the left are there to recall that it is always possible, using the upward closure for $\leqslant$, to pass to a smaller vertex while seeing a Büchi transition. As this move can take place between any transitions, they are thus equivalent to ε-transitions.

The language recognised by the ordered Büchi automaton in Figure 1 is the set of words with only a finite number of times the factor bb, and infinitely often the factor aa. Indeed, an accepting run in this automaton cannot remain infinitely often in the state p, and thus at some point goes in q or r. Past this point, it can no longer see a factor bb, which would end the run, and can only see a Büchi transition when seeing an a whilst the run is in q – hence an infinity of factor aa.

We similarly observe that the ordered Büchi automaton described in Figure 2 is accepting in the case of a word with infinitely many b's, as it can then always see a Büchi transition coming back to r, or in the case of some factor bb, allowing it to reach the state p, followed by a word with an infinity of a's.

Rabin languages are ordered Büchi. We also observe that ordered Büchi automata can recognise Rabin languages. A language L over an alphabet Π is a *Rabin language* if there exists a finite set of pairs $(G_i, R_i)_{i<n}$, with $G_i, R_i \subseteq \Pi$, such that a word w belongs to L if and only if, for $\inf(w)$ the set of its infinitely recurring letters, there exists some $i < n$ such that $\inf(w) \cap G_i \neq \emptyset$ and $\inf(w) \cap R_i = \emptyset$.

Let L be such a Rabin language over an alphabet Π, with Rabin pairs $(G_i, R_i)_{i<n}$. We define $\mathcal{A} = ([n+1], \leqslant, [n+1], \Gamma)$ an ordered Büchi automaton, along with a morphism $f : \Pi \to \Gamma$, and will prove that it recognises exactly L, up to renaming the letters with f. For a a letter in Π, we define the following tile:

$$t_a := \{(n, 1, n)\} \uplus \{(i, 0, i) \mid i < n, a \in G_i \setminus R_i\} \uplus \{(i, 1, i) \mid i < n, a \notin R_i\} .$$

Note that t_a is not necessarily a skeleton, as it can have two transitions starting from the same state i. We then define $f(a)$ as the upwards closure of t_a, and Γ as $f(\Pi)$. An example of such a construction can be observed in Figure 3.

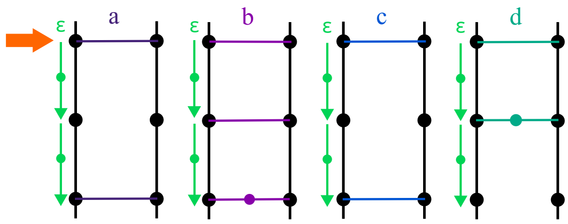


Figure 3 The following tiles describe the Rabin language of pairs $(\{d\}, \{a, c\})$ and $(\{b\}, \{d\})$.

We claim that for $w = (w_j)_{j\in\mathbb{N}} \in \Pi^\omega$, $w \in L$ if and only if $f(w) \in \mathcal{L}(\mathcal{A})$. Indeed, if $w \in L$, there exists i such that infinitely often, $w_j \in G_i$, and past some index j_0, $w_j \notin R_i$. One easily checks that $(n, 1, n)^{j_0-1}\ (n, 0, i)\ (i, r_{j_0}, i)\ (i, r_{j_0+1}, i) \ldots$ (in which for all $j \geqslant j_0$, r_j is 0 if $w_j \in G_i$, and 1 otherwise), is an accepting run over w.

Conversely, suppose that $f(w) \in \mathcal{L}(\mathcal{A})$, with corresponding run $((q_j, r_j, q_{j+1}))_{j\in\mathbb{N}}$. Note that by construction of f, for all $a \in \Pi$, if $i \xrightarrow{f(a)} i'$, then $i' \leqslant i$. Therefore, the sequence $(q_j)_{j\in\mathbb{N}}$ eventually stabilizes in some i_∞ at some index j_0 (with $i_\infty < n$, as there is no horizontal Büchi transition over the state n). Thus, for all $j \geqslant j_0$, $w_j \notin R_{i_\infty}$, and as the run is accepting, infinitely often, $(i_\infty, 0, i_\infty) \in f(w_j)$, thus infinitely often $w_j \in G_{i_\infty}$. Therefore i_∞ is an accepting Rabin pair, and $w \in L$.

3.3 Ordered Büchi languages are the Eve-positional languages

The main interest of these ordered Büchi automata lies in their ability to exactly recognise Eve-positional ω-regular languages:

▶ **Theorem 9.** *Let L be an ω-regular language. It is Eve-positional if and only if it is ordered Büchi.*

This theorem will be proved in Section 4, but we already provide the intuition of why, for $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ an ordered Büchi automaton recognising a prefix-independent language $L := \mathcal{L}(\mathcal{A})$, this language is Eve-positional.

It thus suffices to establish that L satisfies the local preference properties (stated in Theorem 3). By prefix-independence, we only need to prove that it satisfies the third one. That is, for all $v, v' \in \Gamma^+$, if $(vv')^\omega \in L$, either $v^\omega \in L$ or $v'^\omega \in L$.

Without loss of generality, we can assume that all the states of Q are reachable at any point along a run, as L is prefix-independent. We first claim that for $u \in \Pi^+$, $u^\omega \in L$ if and only if there exists a horizontal Büchi transition along u. The converse implication is immediate. For the forward direction, we observe that $1_{\sqsubseteq}{}^\omega \notin \mathcal{L}(\mathcal{A})$, as there is no infinite path in $1_{\sqsubseteq}{}^\omega$ witnessing an infinity of Büchi transitions. Therefore, up to considering t_u as the tile obtained by product of the tiles of u, necessarily $t_u \not\subseteq 1_{\sqsubseteq}$. We then observe that any transition in $t_u \setminus 1_{\sqsubseteq}$ induces a horizontal Büchi transition by upward closure, which establishes the desired claim.

In consequence, for $v, v' \in \Sigma_{\sqsubseteq}{}^+$ such that $(vv')^\omega \in L$, we are in one of the four cases described in Figure 4. In the first and second cases, immediately, either $v^\omega \in L$ or $v'^\omega \in L$. In the third case, by upward closure, we observe that $q \xrightarrow{v:0} q$, and similarly in the fourth case $q \xrightarrow{v':0} q$, which concludes the proof.

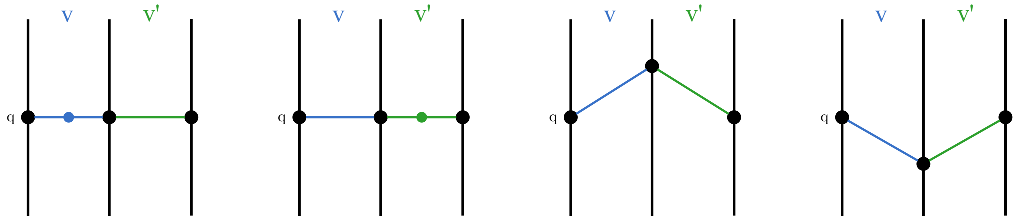


Figure 4 If $(vv')^\omega \in L$, necessarily one of the following holds for some state $q \in Q$. In the third and fourth cases, one of the two transitions depicted (over v or v') is necessarily Büchi, but we do not use this property for the proof.

We can prove that this result also holds in the non-prefix-independent case, using again the same kind of tile manipulations.

▶ **Proposition 10.** *Let $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ be an ordered Büchi automaton. Then $\mathcal{L}(\mathcal{A})$ is Eve-positional.*

The converse inclusion is shown by exhibiting a transformation from any ε-complete Büchi automaton to an ordered Büchi automaton recognising the same language (up to letter renaming): as all Eve-positional ω-regular languages can be recognised by an ε-complete Büchi automaton [5], this suffices.

▶ **Proposition 13.** *For L a language recognised by an ε-completable Büchi automaton, L is ordered Büchi.*

We even provide a transformation from a general ε-complete parity automaton to an ordered Büchi automaton recognising the same language, that is asymptotically optimal.

▶ **Theorem 14.** *For A an ε-completable $[2k]$-automaton with n states, recognising a language L, there effectively exists an ordered Büchi automaton with kn states recognising L.*

A corollary of these results is that we can complete the missing implication in Casares and Ohlmann's work. Indeed, chaining Theorem 14 and Proposition 10, we obtain that:

▶ **Corollary 15.** *If A is an ε-completable automaton, then $\mathcal{L}(A)$ is Eve-positional.*

3.4 Determinization of ordered Büchi automata

We finally provide a determinization procedure for ordered Büchi automata, with size blow-up at most factorial. This determinization implies maintaining a kind of Last Appearance Record (initially introduced in [12]), where we keep track, after some prefix, of the best run candidate ending in each state, ordered by comparative longevity. The states of this automaton, denoted records, thus correspond to permutations of downwards-closed subsets of Q. Section 5 is dedicated to this determinization and the corresponding subresults.

▶ **Theorem 20.** *For $\mathcal{S} = (Q, \sqsubseteq, I, \Gamma)$ an ordered Büchi automaton of language L, there effectively exists a deterministic parity automaton recognising L, with at most*

$$1 + \sum_{i=0}^{|Q|-1} i!$$

states.

This result notably implies that all Büchi automata recognising Eve-positional languages can be determinized in size at most factorial, as opposed to the lower bound in $O((1.64n)^n)$ in the general case [20].

We finally show that over some sufficiently complete tile alphabets, such as for the n-complete Rabin (which is the Rabin language with n pairs and all the letters with different behaviours), this determinization procedure yields automata optimal as to their number of states.

3.5 Discussion

We introduce in this article ordered Büchi automata as an alternative formalism to characterise Eve-positional languages. They allow a succinct yet easy-to-handle representation of these languages. Using this formalism, we propose a determinization procedure, optimal over some languages (as soon as the tile alphabet contains an expressive enough sub-alphabet), which establishes a new upper bound on the determinization of Eve-positional languages. In particular, this determinization makes explicit a structural simplicity of these languages, as it only requires to store permutations of the states instead of the Safra trees used in the general case ([19]).

A continuation of this work that looks promising is to study the same idea over larger classes than ω-regular languages, as in [6]. This would amount to considering an ordered Büchi automaton with ω states.
Unfortunately, we did not manage, using these ordered Büchi automata, to obtain a faster procedure to decide Eve-positionality than the one proposed by Casares and Ohlmann [5].

4 Tile monoids and Eve-positionality of ordered Büchi languages

In this section, we give some results over ordered Büchi automata, before establishing that the ordered Büchi automata recognise exactly the Eve-positional ω-regular languages.

4.1 Properties of ordered Büchi tiles

We observed that when we set a total order $\sqsubseteq$ over Q, and consider the set $\Sigma_{\sqsubseteq}$ of upwards-closed tiles for the corresponding order $\leqslant$, the set $\Sigma_{\sqsubseteq}$ still has a monoid structure, by Lemma 5.

We can then observe that for all ordered Büchi automaton $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$, a tile $t \in \Gamma$ is such that $t^\omega \in \mathcal{L}(\mathcal{A})$ if and only if it admits a horizontal Büchi transition over a state in I. This result will notably be useful when reasoning on ultimately-periodic words, typically for the local preference properties.

▶ **Lemma 7.** *Let $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ be an ordered Büchi automaton. For all $t \in \Gamma$, $t^\omega \in \mathcal{L}(\mathcal{A})$ if and only if there exists $q \in I$ such that $q \xrightarrow{t:0} q$.*

Proof. If there is some $q \in I$ such that $q \xrightarrow{t:0} q$, immediately $t^\omega \in \mathcal{L}(\mathcal{A})$, as shown by the run $((q, 0, q))_{i \in \mathbb{N}}$.

Conversely, we assume that there is no such horizontal Büchi transition. Notably, there is no transition δ in t of the shape $p_I \xrightarrow{t} q$, with $p_I \in I$ and $q \notin I$. Indeed, if it were the case, by the downward closure of I, $p_I \sqsubset q$, and thus, denoting $\delta' := (p_I, 0, p_I)$, we have $\delta < \delta'$. This would imply, by upward closure, that $\delta' \in t$: contradiction. Similarly, we get that for all $(q_I, c, q') \in t$ with $q_I \in I$, q' is such that $q' \sqsubseteq q_I$. Let $((q_i, r_i, q_{i+1}))_{i \in \mathbb{N}}$ be a run over t^ω, with $q_0 \in I$. By immediate recurrence using the previous result, for all $i \in \mathbb{N}$, $q_i \in I$ and $q_{i+1} \sqsubseteq q_i$. This means that the sequence $(q_i)_{i \in \mathbb{N}}$ is ultimately constant, equal to some q_∞. Since $q_\infty \xrightarrow{1_\sqsubseteq} q_\infty$ is not a Büchi transition, the run is rejecting. ◀

We also easily observe that the different states in Q directly define the residuals of the corresponding language, and that due to the structure imposed by $\sqsubseteq$, these residuals are totally ordered.

▶ **Lemma 8.** *Let $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ be an ordered Büchi automaton. Then the residuals of $\mathcal{L}(\mathcal{A})$ are exactly the languages $\mathcal{L}(\mathcal{A}_U)$, where $\mathcal{A}_U = (Q, \sqsubseteq, U, \Gamma)$ for U a downwards-closed subset of Q. Furthermore, these residuals are totally ordered for the inclusion.*

4.2 Ordered Büchi languages are Eve-positional

The goal of this section is to prove the following theorem:

▶ **Theorem 9.** *Let L be an ω-regular language. It is Eve-positional if and only if it is ordered Büchi.*

We do so by double inclusion, making use of the previous structural results on ordered Büchi automata. We first exhibit that the ordered Büchi languages are Eve-positional, as they satisfy the local preference properties (using Lemma 8 and Lemma 7).

▶ **Proposition 10.** *Let $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ be an ordered Büchi automaton. Then $\mathcal{L}(\mathcal{A})$ is Eve-positional.*

We can then easily observe the following, by unfolding the definition of an ordered Büchi language.

▶ **Corollary 11.** *All ordered Büchi languages are Eve-positional.*

We now show the converse implication, that is, all Eve-positional ω-regular languages are ordered Büchi languages. To do so, we first establish that the words admitting an accepting run in an ε-complete automaton can be chosen of a specific shape, before building an ordered Büchi automaton mirroring the behaviour of such an automaton.

▶ **Lemma 12.** *Let A be an ε-complete automaton. For every word $w = (w_i)_{i \in \mathbb{N}} \in \mathcal{L}(A)$, there exists an accepting run in A over the word $(\varepsilon w_i \varepsilon)_{i \in \mathbb{N}}$.*

Then, for L an Eve-positional ω-regular language, according to Theorem 1, any Büchi automaton recognising L is ε-completable. It therefore suffices to design an ordered Büchi automaton that mimics the behaviour of some such automaton (which exists, as L is ω-regular).

▶ **Proposition 13.** *For L a language recognised by an ε-completable Büchi automaton, L is ordered Büchi.*

The corresponding ordered Büchi automaton $\mathcal{A}$ is obtained by ε-completing the initial automaton, then taking the quotient by the equivalence relation induced by $\xrightarrow{\varepsilon:1}$. We can then observe that the natural morphism from a to the induced tile $\mathcal{A}$ satisfies the desired properties.

It is even possible to obtain a stronger version of this result, with a transformation from an ε-complete non-deterministic parity automaton to an ordered Büchi automaton recognising the same language (once again, up to a renaming of alphabet). This transformation incurs no additional state blow-up beyond what is already needed when transforming non-deterministic parity automata into non-deterministic Büchi automata. This result, however, comes with a more elaborate proof, detailed in the appendix:

▶ **Theorem 14.** *For A an ε-completable $[2k]$-automaton with n states, recognising a language L, there effectively exists an ordered Büchi automaton with kn states recognising L.*

We thus obtain the missing implication in Theorem 1: for L an ω-regular language recognised by an ε-completable non-deterministic automaton, by Theorem 14, L is ordered Büchi. Therefore, by Corollary 11, L is indeed Eve-positional.

▶ **Corollary 15.** *If A is an ε-completable automaton, then $\mathcal{L}(A)$ is Eve-positional.*

5 Determinization of Eve-positional languages

In this section, we present a transformation from an ordered Büchi automaton to a deterministic parity automaton recognising the same language. This transformation follows ideas from the last appearance record technique [12] and has at most a factorial blow-up. We observe that it produces an automaton of minimal size, provided that the tile alphabet contains some expressive enough subset.

5.1 Determinization from ordered Büchi to parity

For a tile $t \in \Sigma_{\sqsubseteq}$ and a subset $X \subseteq Q$, we denote $t(X) := \{q \mid \exists p \in X, p \xrightarrow{t} q\}$. That is, $t(X)$ represents all the states reachable from X after reading t. Note that by upward closure of t for $\leqslant$, $t(X)$ is a downwards closed set. We additionally define, for a tile $t \in \Sigma_{\sqsubseteq}$, $\delta_t : Q \to Q \uplus \bot$ as the function such that $\delta_t(q) = \max_{\sqsubseteq}\{q' \mid q \xrightarrow{t} q'\}$, with $\delta_t(q) = \bot$ if no such q' exists (by convention $\bot$ is inferior to all the elements in Q). We immediately observe that δ_t is monotone with respect to $\sqsubseteq$, once again by upward closure of t.

5.1.1 Definition of the deterministic automaton

Let $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ be an ordered Büchi automaton. The first goal of this section is to describe a deterministic parity automaton

$$A_D = (S, \Gamma, s_I, \Delta)$$

that recognises the language $\mathcal{L}(\mathcal{A})$ (see Lemma 19 and Lemma 18). Its behaviour is quite complex, but intuitively it aims to keep track, for each reachable state q in Q, of the best feasible run ending in q. We refer to the Example 16 for a better understanding.

The set of *records* S is the set of tuples $(s_0, \dots, s_k)$ such that

1. $\{s_0, \dots, s_k\}$ is a downward closed subset of Q for $\sqsubseteq$,
2. the s_i's are pairwise distinct, and
3. s_0 is maximal for $\sqsubseteq$ among the s_i's.

Equivalently, a record is an injective map $s : [k] \to Q$ (for some $k \leqslant n$) with a downward closed image.

Intuitively, the current record remembers what the different accessible vertices in Q are and the relative age of the corresponding subpaths, with older paths occupying smaller indices. These paths will then be extended along δ_t, for t the tile read. The exact rules for the fusion of paths and creation of new paths are detailed below.

The initial state s_I is the record whose image contains exactly I, such that $s(0) = \max_{\sqsubseteq}(I)$ and the following positions contain the states of I in decreasing order over $\sqsubseteq$.

The transition function $\Delta : S \times \Sigma_{\sqsubseteq} \to [-1, 2n-1] \times S$ is defined for all $s = (s(0), \dots, s(k)) \in S$ and $t \in \Sigma_{\sqsubseteq}$ to be

$$\Delta(s, t) = (c, s')$$

in which c and s' are defined as follows. For all $i \in \mathrm{dom}(s)$, we define

$$\mathrm{best}_t(i) := \delta_t(s(i)), \qquad \mathrm{leader}_t(i) := \min_{\leqslant}\{j \in \mathrm{dom}(s) \mid \mathrm{best}_t(i) = \mathrm{best}_t(j)\}$$

$$P := \mathrm{img}(\mathrm{leader}_t) \subseteq [k], \qquad R := t(\mathrm{img}(s)) \setminus \mathrm{best}_t(P).$$

For i an index in $\mathrm{dom}(s)$, $\mathrm{leader}_t(i)$ thus corresponds to the smallest index of $\mathrm{dom}(s)$ that has same image as i by $\delta_t(s) = \mathrm{best}_t$. As the $s(i)$'s correspond to the end vertices of different concurrent paths, $\mathrm{leader}_t(i)$ thus corresponds to the oldest path that has for image $\mathrm{best}_t(i)$. We denote P the set of leaders, as they will correspond to "preserved" paths. The indices in R, conversely, correspond to "reset" paths in s'. We say that an index $i \in \mathrm{dom}(s)$ is *forgotten* if $i \neq \mathrm{leader}_t(i)$ (that is, $i \in [k] \setminus P$).

Let b be the unique increasing map from $[\,|P|\,]$ to P, and let f be the unique decreasing map from $[\,|R|\,]$ to R [e]. We define the record s' to have domain $[\,|P| + |R|\,] = [\,|t(\mathrm{img}(s))|\,]$, image $t(\mathrm{img}(s))$ and to be such that

$$\text{for all } i \in [\,|P| + |R|\,], \quad s'(i) := \begin{cases} \mathrm{best}_t(b(i)) & \text{for } 0 \leqslant i < |P| \\ f(i - |P|) & \text{otherwise.} \end{cases}$$

Note that if $P \neq \emptyset$, $s'(0) = \mathrm{best}_t(b(0)) = \mathrm{best}_t(0) = \delta_t(s(0)) = \max_{\sqsubseteq} t(\mathrm{img}(s))$, and this state is well-defined (if $P = \emptyset$, s' is the empty record). Intuitively, all the paths from indices in P are preserved, in their initial order. The indices after $|P|$ then correspond to brand new paths (in R), assigned in decreasing order.

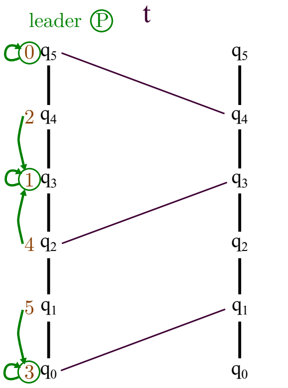
The priority of this transition is defined as $c := \min(2\mathit{green}, 2\mathit{red} - 1)$ where

$$\mathit{green} := \min_{\leqslant}\{i \mid s(i) \xrightarrow{t:0} s'(i)\} \qquad \text{and} \qquad \mathit{red} := \min_{\leqslant}(\mathrm{dom}(s) \setminus P)\ ,$$

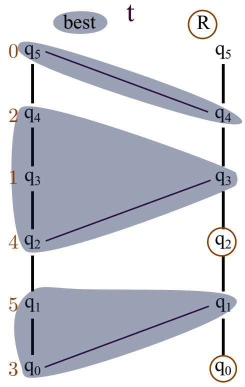
with, in each case, the convention that the minimum over the empty set is $|t(\mathrm{img}(s))|$. That is, intuitively, we want to accept with priority $2i$ if eventually the path at index i is never reset, and witnesses infinitely many Büchi transitions. Conversely, if we are rejecting with priority $2i+1$, then all the paths at indices $< i$ keep going infinitely but eventually no longer see any Büchi transition, and all the other paths are eventually forgotten.

[e] f could be any bijection from $[|R|]$ to R, but choosing consistently an order ensures a smaller resulting automaton.

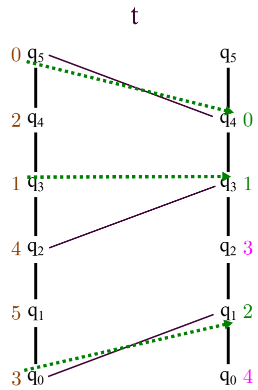
▶ **Example 16.** Let $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, with $\sqsubseteq$ corresponding to the standard order over the states' index. We observe here a transition in A_D over some tile $t \in \Sigma_{\sqsubseteq}$ whose skeleton is described in Figure 5. More precisely, we are looking at a transition from a given record, here $(q_5, q_3, q_4, q_0, q_2, q_1)$, over the tile t. The first step is then to establish, for each index, its leader. Here, the set P is thus $\{0, 1, 3\}$, as shown in Figure 5. We then



■ **Figure 5** We compute the function leader_t from the given record s in brown. This defines the set P.



■ **Figure 6** We regroup the indices that have the same image by best_t, and thus by δ_t. This defines the set R.



■ **Figure 7** (c) We now compute s', with the help of the functions b and f.

consider the image $\text{best}_t(P) = \{q_1, q_3, q_4\}$, which allows us to define the set $R = \{q_0, q_2\}$ as its complement in $t(\text{img}(s))$, as shown in Figure 6. We finally look at the unique increasing function b from $[|P|] = [3]$ to $\{0, 1, 3\}$. We can thus describe $s'([2])$ as the different $\delta_t(b(i))$, as shown in Figure 7 with the green arrows. The remaining positions in the record are assigned the remaining states in the downwards closed subset, in decreasing order, as visible in magenta in Figure 7.

Whenever two paths would fuse (due to having the same image by δ_t), we keep the oldest one – the one of smallest index in s. Therefore, all the indices $i \notin P$ are forgotten, as they describe paths which are not as relevant as the ones they fuse into, and the states in R are now considered as the starting points of brand new paths.
The priority choice mirrors this intuition: it outputs an odd priority corresponding to the oldest path forgotten, or an even priority corresponding to the oldest path witnessing a Büchi transition. Thus, an even priority $2i$ ultimately dominating implies that the index i is never forgotten past some point (nor all the smaller indices), and the infinite path it describes witnesses an infinity of Büchi transitions.

Note that this automaton can have at most $1 + \sum_{i=0}^{n-1} i!$ states : 1 for the empty record, then, for each non-empty record s, $s(0)$ is fixed and there are thus $(|\text{img}(s)| - 1)!$ possible such records. In many cases, however, they are not all accessible. As we consider our automata as trimmed to their accessible states by default, we then consider A_D to be smaller. We can even prove the optimality of the construction when the alphabet is complete enough. This, however, is only detailed in the full version of the article ([13]).

5.1.2 Correction of the recognised language

Let us now prove that A_D recognises $\mathcal{L}(\mathcal{A})$. By determinism, for $w \in {\Sigma_{\sqsubseteq}}^*$ and $s \in S$, we denote as $\Delta(s, w)$ the record s' reached after reading w from s.

▶ **Lemma 17.** *Let $w \in {\Sigma_{\sqsubseteq}}^*$ be a finite word and $s := \Delta(s_I, w)$ be the state reached in A_D after reading w. Then $\mathrm{img}(s)$ is the set of states of Q reachable by partial runs over w starting in I.*

We can now prove the double inclusion stating that A_D recognises exactly the accepting tile sequences.

▶ **Lemma 18.** *For $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ an ordered Büchi automaton, and A_D obtained by the previous determinization procedure applied on $\mathcal{A}$, $L(A_D) \subseteq \mathcal{L}(\mathcal{A})$.*

This lemma is proven by looking at the vertex sequence defined by the $(s_j(g))_{j\in\mathbb{N}}$, for $2g$ the accepting priority of an accepting run in A_D. We observe that this vertex sequence is Büchi accepting.

▶ **Lemma 19.** *For $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$ an ordered Büchi automaton, and A_D obtained by the previous determinization procedure applied on $\mathcal{A}$, $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(A_D)$.*

The proof of this direction is more intricate, and requires to look at sequences of states $(q_j)_{j\in\mathbb{N}}$ such that q_{j+1} is the maximal reachable state from q_j in t_j. We can then observe that one of these sequences $(p_j)_{j\in\mathbb{N}}$ is accepting whenever there exists an accepting Büchi path through $(t_j)_{j\in\mathbb{N}}$. We then conclude stating that there exists an index i such that past some point, in all the records s_j, the first $i-1$ values are never forgotten, and $s_j(i) = p_j$. This establishes that the run of A_D over $(t_j)_{j\in\mathbb{N}}$ is accepting with priority $2i$.

We obtain as a consequence of the two preceding lemmas that

▶ **Theorem 20.** *For $\mathcal{S} = (Q, \sqsubseteq, I, \Gamma)$ an ordered Büchi automaton of language L, there effectively exists a deterministic parity automaton recognising L, with at most*

$$1 + \sum_{i=0}^{|Q|-1} i!$$

states.

As A_D recognises $\mathcal{L}(\mathcal{A})$, Eve-positional, by Theorem 1, it is ε-completable. Actually, this ε-completion is quite easy to describe and corresponds to the lexicographical order over records (seen as tuples).

▶ **Lemma 21.** *Let A_D be the automaton obtained by the determinization of some ordered Büchi automaton $\mathcal{A} = (Q, \sqsubseteq, I, \Gamma)$. Then A_D can be ε-completed by adding the ε-transitions in $\Delta_\varepsilon := \{(s, \varepsilon, 2i, s') \mid s >^i_{\mathrm{lex}} s'\} \cup \{(s', \varepsilon, 2i+1, s) \mid s \leqslant^i_{\mathrm{lex}} s'\}$, where $\leqslant^i_{\mathrm{lex}}$ corresponds to the lexicographical order over the first i elements of a record.*

References

1 Nathalie Bertrand, Miheer Dewaskar, Blaise Genest, Hugo Gimbert, and Adwait Amit Godbole. Controlling a population. *Logical Methods in Computer Science*, Volume 15, Issue 3, July 2019. doi:10.23638/lmcs-15(3:6)2019.

2 Patricia Bouyer, Antonio Casares, Mickael Randour, and Pierre Vandenhove. Half-Positional Objectives Recognized by Deterministic Büchi Automata. In Bartek Klin, Sławomir Lasota, and Anca Muscholl, editors, *33rd International Conference on Concurrency Theory (CONCUR 2022)*, volume 243 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:18, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2022.20.

3 Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs. *TheoretiCS*, Volume 2, January 2023. doi:10.46298/theoretics.23.1.

4 Julius Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969. doi:10.2307/1994916.

5 Antonio Casares and Pierre Ohlmann. Positional ω-regular languages. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 21:1–21:14. ACM, 2024. doi:10.1145/3661814.3662087.

6 Antonio Casares and Pierre Ohlmann. Characterising memory in infinite games. *Log. Methods Comput. Sci.*, 21(1), March 2025. doi:10.46298/lmcs-21(1:28)2025.

7 Thomas Colcombet, Nathanaël Fijalkow, and Florian Horn. Playing Safe. In Venkatesh Raman and S. P. Suresh, editors, *34th International Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS 2014)*, volume 29 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 379–390, Dagstuhl, Germany, 2014. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2014.379.

8 Thomas Colcombet and Olivier Idir. An algebraic characterisation of eve-positional languages. *CoRR*, abs/2604.22648, 2026. doi:10.48550/arXiv.2604.22648.

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

10 Thomas Colcombet and Konrad Zdanowski. A tight lower bound for determinization of transition labeled büchi automata. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming*, pages 151–162, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. doi:10.1007/978-3-642-02930-1_13.

11 E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, October 1-4, 1991*, pages 368–377. IEEE Computer Society, 1991. doi:10.1109/SFCS.1991.185392.

12 Yuri Gurevich and Leo Harrington. Trees, automata, and games. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, STOC '82, pages 60–65, New York, NY, USA, 1982. ACM. doi:10.1145/800070.802177.

13 Olivier Idir. Eve-positional languages: putting order into büchi automata, 2026. doi:10.48550/arXiv.2602.09896.

14 Nils Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. In *[1992] Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 382–393, 1992. doi:10.1016/0168-0072(94)90086-8.

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

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

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

18 Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Bulletin of the American Mathematical Society*, 74:1025–1029, 1968. URL: https://api.semanticscholar.org/CorpusID:6015948.

19 Shmuel Safra. On the complexity of omega-automata. In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, SFCS '88, pages 319–327, USA, 1988. IEEE Computer Society. doi:10.1109/SFCS.1988.21948.

20 Sven Schewe and Thomas Varghese. Determinising parity automata. *CoRR*, abs/1401.5394:486–498, 2014. doi:10.48550/arXiv.1401.5394.

21 Helmut Seidl and Damian Niwinski. On distributive fixed-point expressions. *RAIRO Theor. Informatics Appl.*, 33(4/5):427–446, 1999. doi:10.1051/ita:1999101.