


# On the Succinctness of Alternating Parity Good-for-Games Automata (Full Version\*)

**Udi Boker**


Interdisciplinary Center (IDC) Herzliya, Israel  
udiboker@gmail.com

**Denis Kuperberg** 

CNRS, LIP, École Normale Supérieure, Lyon, France  
denis.kuperberg@ens-lyon.fr

**Karoliina Lehtinen** 

University of Liverpool, United Kingdom  
k.lehtinen@liverpool.ac.uk

**Michał Skrzypczak** 

Institute of Informatics, University of Warsaw, Poland  
mskrzypczak@mimuw.edu.pl

---

## Abstract

We study alternating parity good-for-games (GFG) automata, i.e., alternating parity automata where both conjunctive and disjunctive choices can be resolved in an online manner, without knowledge of the suffix of the input word still to be read.

We show that they can be exponentially more succinct than both their nondeterministic and universal counterparts. Furthermore, we present a single exponential determinisation procedure and an EXPTIME upper bound to the problem of recognising whether an alternating automaton is GFG.

We also study the complexity of deciding “half-GFGness”, a property specific to alternating automata that only requires nondeterministic choices to be resolved in an online manner. We show that this problem is PSPACE-hard already for alternating automata on finite words.

**2012 ACM Subject Classification** Theory of computation → Logic and verification

**Keywords and phrases** Good for games, history-determinism, alternation

**Funding** *Udi Boker*: Israel Science Foundation grant 1373/16

*Karoliina Lehtinen*: This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 892704.

## 1 Introduction

*Good-for-games* (GFG) automata were first introduced in [11] as a tool for solving the synthesis problem. The equivalent notion of *history-determinism* was introduced independently in [7] in the context of regular cost functions. Intuitively, a nondeterministic automaton is GFG if nondeterminism can be resolved on the fly, only with knowledge of the input word read so far. GFG automata can be seen as an intermediate formalism between deterministic and nondeterministic ones, with advantages from both worlds. Indeed, like deterministic automata, GFG automata enjoy good compositional properties—useful for solving games and composing automata and trees—and easy inclusion checks [3]. Like nondeterministic automata, they can be exponentially more succinct than deterministic automata [16].

In recent years, much effort has gone into understanding various properties of nondeterministic GFG automata, for instance their relationship with deterministic automata [3, 16, 4, 15],

---

\* This is the full version of the paper of the same name published at FSTTCS 2020

applications in probabilistic model checking [14] and synthesis of LTL,  $\mu$ -calculus and context-free properties [12, 18], decision procedures for GFGness [19, 16, 2], minimisation [1], and links with recent advances in parity games [9].

Alternating GFG automata are a natural generalisation of nondeterministic GFG automata that enjoy the same compositional properties as nondeterministic GFG automata, while providing more flexibility. As we show in the present work, for some languages alternating GFG parity automata can also be exponentially more succinct, allowing for better synthesis procedures. Indeed, two-player games with winning conditions given by alternating GFG automata are solvable in quasipolynomial time, via a linear reduction to parity games, while for winning conditions given by arbitrary alternating automata, solving games requires determinisation and has therefore double-exponential complexity.

Alternating GFG automata were introduced independently by Colcombet [8] and Quirkl [22] while a form of alternating GFG automata with requirements specific to counters were also considered in [17], as a tool to study cost functions on infinite trees. Boker and Lehtinen studied the expressiveness and succinctness of alternating GFG automata in [5], showing that they

- are not more succinct than DFAs on finite words,
- are as expressive as deterministic ones of the same acceptance condition on infinite words,
- and can be determined with a  $2^{\theta(n)}$  size blowup for the Büchi and coBüchi conditions.

Many questions about GFG alternating automata were left open, in particular whether there exists a doubly exponential succinctness gap between alternating GFG and deterministic automata, and the complexity of deciding whether an alternating parity automaton is GFG.

**Succinctness of alternating GFG automata.** We show that there is a single exponential gap between alternating parity GFG automata and deterministic ones, thereby answering a question left open in [5]. This is in contrast to general alternating automata, for which determinisation incurs a double-exponential size increase. However, we also show that alternating GFG automata can present exponential succinctness compared to both nondeterministic and universal GFG automata. This means that alternating GFG automata can be used to reduce the complexity of solving some games with complex acceptance conditions.

**Recognising GFG automata.** We give an EXPTIME upper bound to the problem of deciding whether an alternating parity automaton is GFG, matching the known upper bound for recognising nondeterministic parity GFG automata.

We also study the complexity of deciding “half-GFGness”, i.e., whether the nondeterminism (or universality) of an automaton is GFG. This property guarantees that composition with games preserves the winner for one of the players. We show that already on finite words, this problem is PSPACE-hard, and it is in EXPTIME for alternating Büchi automata. This shows that a PTIME algorithm for deciding GFGness must exploit the subtle interplay between nondeterminism and universality, and cannot be reduced to checking independently whether each of them is GFG.

## Roadmap

We begin with some definitions, after which, in Section 3, we define alternating GFG automata, study their succinctness and the complexity of deciding half-GFGness, that is, whether the nondeterminism within an alternating automaton is GFG. Section 4 provides a single-exponential determinisation procedure for alternating GFG parity automata. Section 5

shows that GFGness of alternating parity automata is in EXPTIME, using the determinisation of the previous section. Throughout the paper, we provide high-level proof sketches, with detailed technical developments in the appendix.

## 2 Preliminaries

**Words and automata.** An *alphabet*  $\Sigma$  is a finite nonempty set of letters. A finite (resp. infinite) *word*  $u = u_0 \dots u_k \in \Sigma^*$  (resp.  $w = w_0 w_1 \dots \in \Sigma^\omega$ ) is a finite (resp. infinite) sequence of letters from  $\Sigma$ . A *language* is a set of words, and the empty word is written  $\epsilon$ . We denote a set  $\{i, i+1, \dots, j\}$  of integers by  $[i, j]$ .

An *alternating word automaton* is a tuple  $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$ , where:  $\Sigma$  is an alphabet;  $Q$  is a finite nonempty set of states;  $\iota \in Q$  is an initial state;  $\delta: Q \times \Sigma \rightarrow \mathbf{B}^+(Q)$  is a transition function where  $\mathbf{B}^+(Q)$  is the set of positive Boolean formulas (*transition conditions*) over  $Q$ ; and  $\alpha$ , on which we elaborate below, is either an acceptance condition or a transition labelling on top of which an acceptance condition is defined. For a state  $q \in Q$ , we denote by  $\mathcal{A}^q$  the automaton that is derived from  $\mathcal{A}$  by setting its initial state  $\iota$  to  $q$ .

An automaton  $\mathcal{A}$  is nondeterministic (resp. universal) if all its transition conditions are disjunctions (resp. conjunctions), and it is deterministic if all its transition conditions are just states. We represent the transition function of nondeterministic and universal automata as  $\delta: Q \times \Sigma \rightarrow 2^Q$ , and of a deterministic automaton as  $\delta: Q \times \Sigma \rightarrow Q$ . A *transition* of an automaton is a triple  $(q, a, q') \in Q \times \Sigma \times Q$ , sometimes also written  $q \xrightarrow{a} q'$ .

We denote by  $\hat{\delta} \subseteq \mathbf{B}^+(Q)$  the set of all subformulas of formulas in the image of  $\delta$ , i.e., all the Boolean formulas that “appear” somewhere in the transition function of  $\mathcal{A}$ .

**Acceptance conditions.** There are various acceptance (winning) conditions, defined with respect to the set of transitions<sup>1</sup> that a path of  $\mathcal{A}$  visits infinitely often. (Notice that a transition condition allows for many possible transitions.) We later formally define acceptance of a word  $w$  by  $\mathcal{A}$  in terms of games, and consider a path of  $\mathcal{A}$  on a word  $w$  as a play in that game. For nondeterministic automata, a “run” coincides with a “path”.

Some of the acceptance conditions are defined on top of a labelling of the transitions rather than directly on the transitions. In particular, in the parity condition, we have  $\alpha: Q \times \Sigma \times Q \rightarrow \Gamma$ , where  $\Gamma \subseteq \mathbb{N}$  is a finite set of priorities and a path is accepting if and only if the highest priority seen infinitely often on it is even.

The Büchi and coBüchi conditions are special cases of the parity condition with  $\Gamma = \{1, 2\}$  and  $\Gamma = \{0, 1\}$ , respectively. When speaking of Büchi and coBüchi automata, we often refer to  $\alpha$  as the set of “accepting transitions”, namely the transitions that are mapped to 2 in the Büchi case and to 0 in the coBüchi case. The weak condition is a special case of both the Büchi and coBüchi conditions, in which every path eventually remains in the same priority.

The Rabin and Streett conditions are more involved, yet defined directly on the set  $T$  of transitions. A Rabin condition is a set  $\{(B_1, G_1), (B_2, G_2), \dots, (B_k, G_k)\}$ , with  $B_i, G_i \subseteq T$ , and a path  $\rho$  is accepting iff for some  $i \in [1, k]$ , we have that the set  $\text{inf}(\rho)$  of transitions that are visited infinitely often in  $\rho$  satisfies  $(\text{inf}(\rho) \cap B_i = \emptyset \text{ and } \text{inf}(\rho) \cap G_i \neq \emptyset)$ . A Streett condition is dual: a set  $\{(B_1, G_1), (B_2, G_2), \dots, (B_k, G_k)\}$ , with  $B_i, G_i \subseteq Q$ , whereby a path  $\rho$  is accepting iff for all  $i \in [1, k]$ , we have  $(\text{inf}(\rho) \cap B_i = \emptyset \text{ or } \text{inf}(\rho) \cap G_i \neq \emptyset)$ .

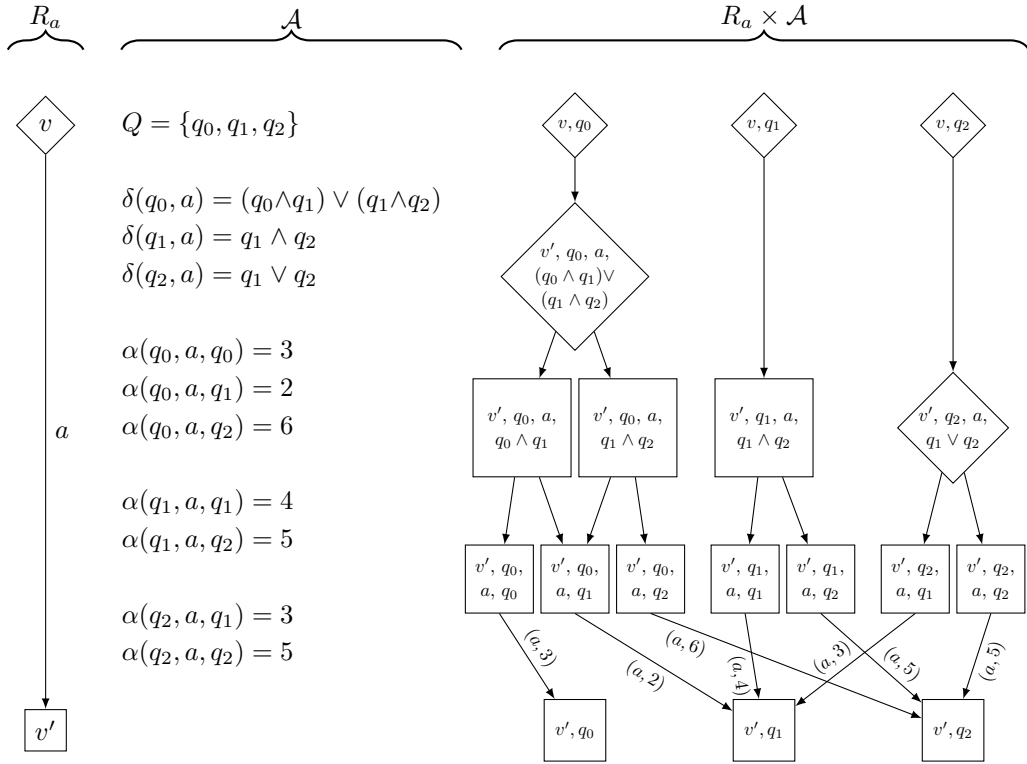
<sup>1</sup> Acceptance is defined in the literature with respect to either states or transitions; for technical reasons we prefer to work with acceptance on transitions.

**Sizes and types of automata.** The size of  $\mathcal{A}$  is the maximum of the alphabet size, the number of states, the transition function length, which is the sum of the transition condition lengths over all states and letters, and the acceptance condition's index, which is 1 for weak, Büchi and coBüchi,  $|\Gamma|$  for parity, and  $k$  for Rabin and Street.

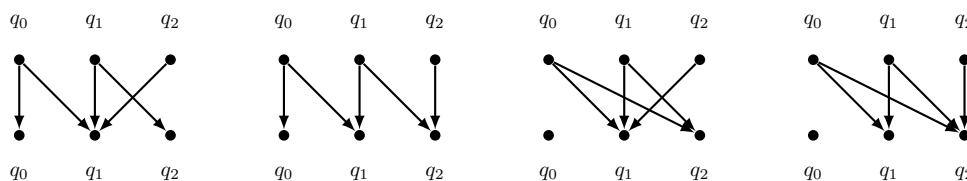
We sometimes abbreviate automata types by three-letter acronyms in  $\{D, N, U, A\} \times \{F, W, B, C, P, R, S\} \times \{A, W\}$ . The first letter stands for the transition mode, the second for the acceptance condition, and the third indicates that the automaton runs on finite or infinite words. For example, DPW stands for a deterministic parity automaton on infinite words.

**Games and strategies.** Some of our technical proofs use standard concepts of an arena, a game, a winning strategy, etc. For the sake of completeness, we provide precise mathematical definitions of these objects in Appendix A. Here we will just overview the involved concepts.

First, we work with two-player games of perfect information, where the players are Eve and Adam. These games are played on graphs (called arenas). Most of the considered games are of infinite duration and their winning condition is expressed in terms of the infinite sequences of edges taken during the play. We invoke results of determinacy (one of the players has a winning strategy), as well as of *positional determinacy* (one of the players has a strategy that depends only on the last position of the play).



■ **Figure 1** A one-step arena over a letter  $a \in \Sigma$ , obtained as a product of a simple arena  $R_a$  with the alternating parity automaton  $\mathcal{A}$ . In this example  $v$  is controlled by Eve and  $v'$  by Adam. The transitions with no label are labelled by  $\epsilon$ . Diamond-shaped positions belong to Eve and square-shaped positions belong to Adam.



■ **Figure 2** The four possible boxes corresponding to Eve's choices in the one-step arena of Figure 1. (All edges should be labelled with  $a$ , which we omit for better readability.)

**Model-checking games.** To represent the semantics of an alternating automaton  $\mathcal{A}$ , we treat the Boolean formulas that appear in the transition conditions of  $\mathcal{A}$  as games. More precisely, given a letter  $a \in \Sigma$  we represent the transition conditions  $q \mapsto \delta(q, a) \in \mathbf{B}^+(Q)$  as the *one-step arena* over  $a$  (see Figure 1). A play over this arena begins in a state  $q \in Q$ ; then the players go down the formula  $\delta(q, a)$  with Eve resolving disjunctions and Adam resolving conjunctions; and finally they reach an atom  $q' \in Q$  and the play stops. This means that a play over the one-step arena over  $a$  results in a transition of the form  $q \xrightarrow{a} q'$ .

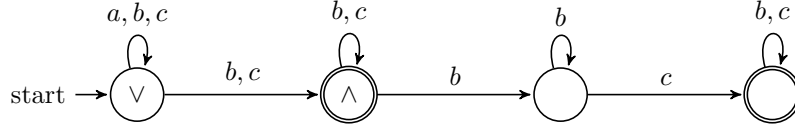
The language  $L(\mathcal{A})$  of an alternating automaton  $\mathcal{A}$  over an alphabet  $\Sigma$  is defined via the *model-checking game*, defined for an automaton  $\mathcal{A}$  and a word  $w = a_0 a_1 a_2 \dots \in \Sigma^\omega$ . A *configuration* of this game is a state  $q$  of  $\mathcal{A}$  and a position  $i \in \omega$  of  $w$ , starting at  $(\iota, 0)$ . In the  $i$ th round, starting at configuration  $(q_i, i)$ , the players play on the one-step arena from  $q_i$  over  $a_i$ , resulting in a transition  $q_i \xrightarrow{a_i} q_{i+1}$ . The next configuration is  $(q_{i+1}, i+1)$ . The acceptance condition of  $\mathcal{A}$  becomes the winning condition of this game.  $\mathcal{A}$  *accepts*  $w$  if Eve has a winning strategy in this game.

For technical convenience, we define (in Appendix A) the model-checking game in terms of a *synchronised product* of the word  $w$  (treated as an infinite graph) and the automaton  $\mathcal{A}$ . Synchronised products turn out to be useful in the analysis of various games presented in this paper and will be used throughout the technical versions of the proofs, in the appendix.

► **Definition 1.** Given an alternating automaton  $\mathcal{A}$ , we denote by  $\overline{\mathcal{A}}$  the dual automaton: it has the same alphabet, set of states, and initial state. Its transition conditions  $\delta_{\overline{\mathcal{A}}}(q, a)$  are obtained from those of  $\mathcal{A}$  by replacing each disjunction  $\vee$  with conjunction  $\wedge$  and vice versa. Its acceptance condition is the dual of  $\mathcal{A}$ 's condition. (In parity automata, all priorities are increased by 1.)  $\overline{\mathcal{A}}$  recognises the complement  $L(\mathcal{A})^c$  of  $L(\mathcal{A})$ .

**Boxes.** Another technical concept that we use is that of *boxes* (see Figure 2), which describe Eve's *local* strategies for resolving disjunctions within a transition condition. Consider an alternating automaton  $\mathcal{A}$  and a letter  $a \in \Sigma$ . Moreover, fix a strategy  $\sigma$  of Eve that resolves disjunctions in all the transition conditions  $\delta(q, a)$  for  $q \in Q$ . Now, the *box* of  $\mathcal{A}$ ,  $a$ , and  $\sigma$  is a subset of  $Q \times \Sigma \times Q$  and contains a triple  $(q, a, q')$  iff  $\sigma$  resolves disjunctions of  $\delta(q, a)$  in such a way that Adam (resolving conjunctions) can reach the atom  $q'$ . In other words, this box contains  $(q, a, q')$  if there is a play consistent with  $\sigma$  on  $\delta(q, a)$  that reaches the atom  $q'$ . We use  $\beta$  to denote single boxes and by  $\mathbf{B}_{\mathcal{A}, a}$  we denote the set of all boxes of  $\mathcal{A}$  and  $a$ , while  $\mathbf{B}_{\mathcal{A}}$  denotes the union  $\bigcup_{a \in \Sigma} \mathbf{B}_{\mathcal{A}, a}$ . We give a more formal definition based on synchronised products in the Appendix, see page 15.

► **Definition 2.** Given a sequence of boxes  $\pi = b_0, b_1, \dots$  of an automaton  $\mathcal{A}$  and a path  $\rho = (q_0, a_0, q_1), (q_1, a_1, q_2), \dots$ , we say that  $\rho$  is a path of  $\pi$  if for every  $i$  we have  $(q_i, a_i, q_{i+1}) \in b_i$ . The sequence  $\pi$  is said to be *universally accepting* if every path in  $\pi$  is accepting in  $\mathcal{A}$ .



■ **Figure 3** Alternating weak automaton accepting words over  $\{a, b, c\}$  in which  $a$  occurs finitely often and  $c$  occurs infinitely often. Omitted transitions lead to a rejecting sink.

Intuitively, a sequence of boxes  $\pi$  as above represents a particular positional strategy  $\sigma$  of Eve in the model-checking game over the word  $w = a_0 a_1 a_2 \dots$ . In that case, a path of  $\pi$  corresponds to a possible play of this game consistent with  $\sigma$ , and the sequence is universally accepting if and only if the strategy is winning.

### 3 Good-For-Games Alternating Automata

Good-for-games (GFG) nondeterministic automata are automata in which the nondeterministic choices can be resolved without looking at the future of the word. For example, consider an automaton that consists of a nondeterministic choice between a component that accepts words in which  $a$  occurs infinitely often and a component that accepts words in which  $a$  occurs finitely often. This automaton accepts all words but is not GFG since the nondeterministic choice of component cannot be resolved without knowing the whole word.

To extend this definition to alternating automata, we must look both at its nondeterminism and universality and require that both can be resolved without knowledge of the future. The following letter games capture this intuition.

► **Definition 3** (Letter games [5]). *Given an alternating automaton  $\mathcal{A}$ , Eve’s letter game proceeds at each turn from a state  $q$  of  $\mathcal{A}$ , starting from the initial state of  $\mathcal{A}$ , as follows:*

- Adam chooses a letter  $a$ ,
- Adam and Eve play on the one-step arena over  $a$  from  $q$  to a new state  $q'$ , where Eve resolves disjunctions and Adam conjunctions.

*A play of the letter game thus generates a word  $w$  and a path  $\rho$  of  $\mathcal{A}$  on  $w$ . Eve wins this play if either  $w \notin L(\mathcal{A})$  or  $\rho$  is accepting in  $\mathcal{A}$ .*

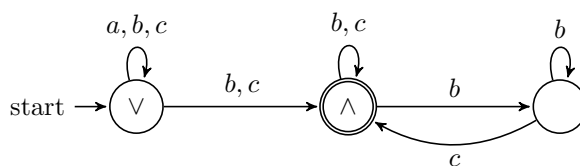
*Adam’s letter game is similar, except that Eve chooses letters and Adam wins if either  $w \in L(\mathcal{A})$  or the path  $\rho$  is rejecting.*

A more formal definition is given in Appendix B.

► **Definition 4** (GFG automata [5]). *An automaton  $\mathcal{A}$  is  $\exists$ -GFG if Eve wins her letter game; it is  $\forall$ -GFG if Adam wins his letter game. Finally,  $\mathcal{A}$  is GFG if it is both  $\exists$ -GFG and  $\forall$ -GFG.*

As shown in [5, Theorem 8], an automaton  $\mathcal{A}$  is GFG if and only if it is indeed “good for playing games”, in the sense that its product with every game whose winning condition is  $L(\mathcal{A})$  preserves the winner of the game.

► **Example 5.** The automaton in Figure 3 accepts the language  $L$  of words in which  $a$  occurs finitely often and  $c$  occurs infinitely often. Here Eve loses her letter game: Adam can play  $c$  until Eve takes the transition to the second state, and then play  $a$  followed by  $c^\omega$ . Conversely, Eve wins Adam’s letter game: her strategy is to play  $b$ , take the transition to the second state and keep playing  $b$  until Adam takes the transition into the third state, after which



■ **Figure 4** Alternating coBüchi  $\forall$ -GFG automaton accepting words over  $\{a, b, c\}$  in which  $a$  occurs finitely often and  $c$  occurs infinitely often.

she plays  $c$  once and then  $b^\omega$ . This automaton is neither  $\exists$ -GFG nor  $\forall$ -GFG, and taking its product with games with  $L$  as winning condition does not preserve the winner of the game.

In contrast, the automaton in Figure 4 is  $\forall$ -GFG but not  $\exists$ -GFG. Indeed, Adam's winning strategy in his letter game is to resolve the conjunction from the middle state by always moving to the right-hand state when Eve plays  $b$ . This forces Eve to choose between playing  $c$  infinitely many times (in which case, the word is in the language) or letting Adam build a rejecting run. Taking its product with *one-player games* with winning condition  $L$  preserves the winner whenever Eve is the player controlling all positions. However, this is not the case for one-player games where Adam is the sole player.

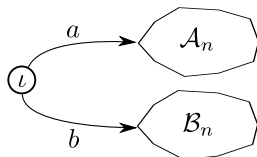
### 3.1 Alternating GFG vs. Nondeterministic and Universal Ones

We show in this section that alternating GFG automata can be more succinct than both nondeterministic and universal GFG automata.

► **Lemma 6.** *There is a family  $(C_n)_{n \in \mathbb{N}}$  of alternating GFG  $\{0, 1, 2\}$ -parity automata of size linear in  $n$  over a fixed alphabet, such that every nondeterministic GFG parity automaton and universal GFG parity automaton for  $L(C_n)$  is of size  $2^{\Omega(n)}$ .*

**Proof.** From [16], there is a family  $(\mathcal{A}_n)_{n \in \mathbb{N}}$  of GFG-NCWs with  $n$  states over a fixed alphabet  $\Sigma$ , such that every DPW for  $L_n = L(\mathcal{A}_n)$  is of size  $2^{\Omega(n)}$ . For every  $n \in \mathbb{N}$ , let  $\mathcal{B}_n$  be the dual of  $\mathcal{A}_n$ , so  $\mathcal{B}_n$  is a UBW accepting  $\overline{L_n}$ . We build an APW  $C_n$  over  $\Sigma$  of size linear in  $n$ , by setting its initial state to move to the initial state of  $\mathcal{A}_n$  when reading the letter  $a \in \Sigma$  and to the initial state of  $\mathcal{B}_n$  when reading the letter  $b \in \Sigma$ . The acceptance condition of  $C_n$  is a parity condition with priorities  $\{0, 1, 2\}$ : accepting transitions of  $\mathcal{A}_n$  are assigned priority 0, and accepting transitions of  $\mathcal{B}_n$  priority 2. Other transitions have priority 1.

The automaton  $C_n$  is represented below:



Observe that  $L(C_n) = aL_n \cup b\overline{L_n}$ , and that  $C_n$  is GFG: its initial state has only deterministic transitions, and over the  $\mathcal{A}_n$  and  $\mathcal{B}_n$  components, the strategy to resolve the nondeterminism and universality, respectively, follows the strategy to resolve the nondeterminism of  $\mathcal{A}_n$ , which is guaranteed due to  $\mathcal{A}_n$ 's GFGness.

Consider a GFG UPW  $\mathcal{E}_n$  for  $L(C_n)$ , and let  $q$  be a state to which  $\mathcal{E}_n$  moves when reading  $a$ , according to some strategy that witnesses  $\mathcal{E}_n$ 's GFGness. Then  $\mathcal{E}_n^q$  is a GFG UPW for  $L_n$ . Its dual is therefore a GFG NPW  $\mathcal{E}'_n$  for  $\overline{L_n}$ .

Since  $\mathcal{A}_n$  is a GFG NPW for  $L_n$ , by [3, Theorem 4] we obtain a DPW for  $L_n$  of size  $|\mathcal{A}_n||\mathcal{E}'_n|$ . By choice of  $L_n$ , this DPW must be of size  $2^{\Omega(n)}$ , and since  $\mathcal{A}_n$  is of size  $n$ , it follows that  $\mathcal{E}'_n$ , and hence  $\mathcal{E}_n$ , must be of size  $2^{\Omega(n)}$ . By a symmetric argument, every GFG NPW for  $L(\mathcal{C}_n)$  must also be of size  $2^{\Omega(n)}$ . ◀

Informally, the language  $L_n$  above describes a set of threads, of which at least one eventually satisfies a safety property. Then, the above construction can be understood as describing a property of reactive systems where, depending on the input, the system guarantees either that there is a thread that eventually satisfies a safety property, or that all threads satisfy a liveness (Büchi) property. The GFG alternating automaton can then be used to solve in polynomial time games with such languages as winning condition, for example in the context of synthesis: the product of the game arena and the alternating automaton for  $L_n$  is a parity game with 3 priorities with the same winner as the original game. In contrast, a DPW, GFG NPW and GFG UPW for the same language would all be exponentially larger.

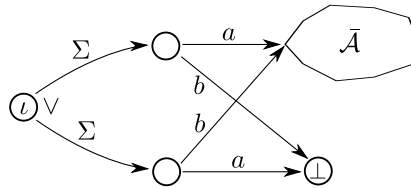
### 3.2 Deciding Half-GFGness

In order to decide GFGness, it is enough to be able to decide the  $\exists$ -GFG property on the automaton and its dual. A natural first approach is therefore to study the complexity of deciding whether an APW is  $\exists$ -GFG. Yet, we will show that already on finite words, this problem is PSPACE-hard, while we conjecture that deciding GFGness is in PTIME.

► **Lemma 7.** *Deciding whether an AFA is  $\exists$ -GFG is PSPACE-hard.*

**Proof.** We reduce from NFA universality: starting from an NFA  $\mathcal{A}$ , we build an AFA  $\mathcal{B}$  based on the dual of  $\mathcal{A}$ , with an additional non-GFG choice to be resolved by Eve. This AFA  $\mathcal{B}$  is  $\exists$ -GFG if and only if  $L(\mathcal{B}) = \emptyset$ , which happens if and only if  $L(\mathcal{A}) = \Sigma^*$ . We crucially use the fact that  $\mathcal{B}$  is not necessarily  $\forall$ -GFG.

Let  $\mathcal{A}$  be an NFA over an alphabet  $\Sigma = \{a, b\}$  and  $\bar{\mathcal{A}}$  its dual. We want to check whether  $L(\mathcal{A}) = \Sigma^*$ . We build an AFA  $\mathcal{B}$ , as depicted below, by first making Eve guess the second letter. If her guess is wrong, the automaton proceeds to a rejecting sink state  $\perp$ . Otherwise, it proceeds to the initial state of  $\bar{\mathcal{A}}$ . The size of  $\mathcal{B}$  is linear in the size of  $\mathcal{A}$ .



If  $L(\bar{\mathcal{A}}) = \emptyset$ , then  $L(\mathcal{B}) = \emptyset$ , so  $\mathcal{B}$  is trivially  $\exists$ -GFG. However, if there is some  $u \in L(\bar{\mathcal{A}})$ , then Adam has a winning strategy in Eve's letter game on  $\mathcal{B}$ . This strategy consists of playing  $a$ , then playing the letter that brings Eve to  $\perp$ , and finally playing  $u$ . The resulting word is in  $L(\mathcal{B}) = \Sigma^2 L(\mathcal{A})$ , so this witnesses that  $\mathcal{B}$  is not  $\exists$ -GFG. We obtain that  $L(\mathcal{A}) = \Sigma^* \Leftrightarrow L(\bar{\mathcal{A}}) = \emptyset \Leftrightarrow \mathcal{B}$  is  $\exists$ -GFG, which is the wanted reduction. ◀

For Büchi automata, and so in particular for finite words, we can give an EXPTIME algorithm for this problem.

► **Lemma 8.** *Deciding whether an ABW is  $\exists$ -GFG is in EXPTIME.*



**Proof.** It is shown in [5, Lemma 23] that removing alternation from an ABW  $\mathcal{A}$  using the breakpoint construction [20] yields an NBW  $\mathcal{B}$  such that if  $\mathcal{A}$  is  $\exists$ -GFG then  $\mathcal{B}$  is GFG. Moreover, the converse also holds: if  $\mathcal{B}$  is GFG then  $\mathcal{A}$  is  $\exists$ -GFG, since playing Eve's letter game in  $\mathcal{B}$  is more difficult for Eve than playing it in  $\mathcal{A}$ . This means that starting from an ABW  $\mathcal{A}$ , we can build an exponential size NBW  $\mathcal{B}$  via breakpoint construction, and test whether  $\mathcal{B}$  is GFG via the algorithm from [2], in time polynomial with respect to  $\mathcal{B}$ . Overall, this yields an EXPTIME algorithm deciding whether  $\mathcal{A}$  is  $\exists$ -GFG.  $\blacktriangleleft$

## 4 Determinisation of Alternating GFG Parity Automata

In this section we provide a procedure that, given an alternating GFG parity automaton, produces an equivalent deterministic parity automaton with single-exponentially many states. To do so, we first provide an alternation-removal procedure that preserves GFG status. Then, we apply this procedure to both the input automaton and its complement and use the GFG strategies in these two automata to determinise the input automaton. Our proofs, in Appendix C, rely on some analysis of when GFG strategies can use the history of the word, rather than the full history of the play (which also includes the choices of how to resolve the nondeterminism and universality), and on the memoryless determinacy of parity games.

Our method for going from alternating to nondeterministic automata is similar to that of Dax and Klaedtke [10]: they take a nondeterministic automaton that recognises the universally-accepting words in  $(B_A)^\omega$  and add nondeterminism that upon reading a letter  $a \in \Sigma$  chooses a box in  $B_A$  over  $a$ . Yet in our approach, in order to guarantee that the outcome preserves GFGness, the intermediate automaton is deterministic.

### 4.1 Alternation Removal in GFG Parity Automata

► **Theorem 9.** *Consider an alternating parity automaton  $\mathcal{A}$  with  $n$  states and index  $k$ . There exists a nondeterministic parity automaton  $\text{box}(\mathcal{A})$  with  $2^{O(nk \log nk)}$  states that is equivalent to  $\mathcal{A}$  such that if  $\mathcal{A}$  is GFG then  $\text{box}(\mathcal{A})$  is also GFG.*

In Section 5, where we discuss decision procedures, we will show that  $\text{box}(\mathcal{A})$  is GFG *exactly* when  $\mathcal{A}$  is  $\exists$ -GFG. For now, the rest of this section is devoted to the proof of Theorem 9, of which a detailed version can be found in Appendix C.2.

► **Lemma 10.** *Consider an alternating parity automaton  $\mathcal{A}$  with  $n$  states and index  $k$ . Then there exists a deterministic parity automaton  $\mathcal{B}$  with  $2^{O(nk \log nk)}$  states over the alphabet  $B_A$  that recognises the set of universally-accepting words for  $\mathcal{A}$ . If  $\mathcal{A}$  is a Büchi automaton, then  $\mathcal{B}$  can also be taken as Büchi, and in general the parity index of the automaton  $\mathcal{B}$  is linear in the number of transitions of  $\mathcal{A}$ .*

**Proof sketch.** We first construct a nondeterministic parity (resp. coBüchi) automaton over the alphabet  $B_A$  that recognises the complement of the set of universally-accepting words for  $\mathcal{A}$ . This automaton is easy to build: it guesses a path that is not accepting, and has the dual acceptance condition to  $\mathcal{A}$ . We then obtain the automaton  $\mathcal{B}$  by determinising and complementing this automaton.  $\blacktriangleleft$

We now build the automaton  $\text{box}(\mathcal{A})$  of Theorem 9. It is the same as the automaton  $\mathcal{B}$  of Lemma 10, except that the alphabet is  $\Sigma$  and the transition function is defined as follows: For every state  $p$  of  $\mathcal{B}$  and  $a \in \Sigma$ , we have  $\delta_{\text{box}(\mathcal{A})}(p, a) := \bigcup_{\beta \in B_{\mathcal{A},a}} \delta_{\mathcal{B}}(p, \beta)$ .

In other words, the automaton  $\text{box}(\mathcal{A})$  reads a letter  $a$ , nondeterministically guesses a box  $\beta \in B_{\mathcal{A},a}$ , and follows the transition of  $\mathcal{B}$  over  $\beta$ . Thus, the runs of  $\text{box}(\mathcal{A})$  over a word

$w = w_0w_1w_2\cdots \in \Sigma^\omega$  are in bijection with sequences of boxes  $(\beta_i)_{i \in \mathbb{N}}$  such that  $\beta_i \in \mathbb{B}_{\mathcal{A}, w_i}$  for all  $i \in \mathbb{N}$ .

Fix an infinite word  $w \in \Sigma^\omega$ . Our aim is to prove that  $w \in L(\mathcal{A}) \Leftrightarrow w \in L(\text{box}(\mathcal{A}))$ .

► **Lemma 11.** *There exists a bijection between positional strategies of Eve in the acceptance game of  $\mathcal{A}$  over  $w$  and runs of  $\text{box}(\mathcal{A})$  over  $w$ . Moreover, a strategy is winning if and only if the corresponding run is accepting. Thus  $L(\mathcal{A}) = L(\text{box}(\mathcal{A}))$ .*

► **Remark 12.** The above alternation-removal procedure also extends to alternating Rabin automata but fails for alternating Streett automata  $\mathcal{A}$ : since Streett games are not positionally determined for Eve, the acceptance game of  $\mathcal{A}$  over a word  $w$  is not positionally determined for Eve.

► **Lemma 13.** *For an alternating  $\exists$ -GFG parity automaton  $\mathcal{A}$ , the automaton  $\text{box}(\mathcal{A})$  is GFG.*

Intuitively, this is because the construction of  $\text{box}(\mathcal{A})$  preserves the nondeterminism of  $\mathcal{A}$ .

## 4.2 Single-Exponential Determinisation

The aim of this section is to prove the following determinisation theorem; see Appendix C.3 for a detailed proof.

► **Theorem 14.** *If  $\mathcal{A}$  is an alternating parity GFG automaton then there exists a deterministic parity automaton  $\mathcal{D}$  that recognises the same language and has size at most exponential in the size of  $\mathcal{A}$ . Moreover, the parity index of  $\mathcal{D}$  is the same as that of  $\mathcal{A}$ .*

► **Remark 15.** Theorem 9 and [3, Theorem 4], which uses an NRW-GFG and its complement NRW-GFG to obtain a DRW, together give an exponential deterministic parity automaton for  $L(\mathcal{A})$ . However, the index of  $\mathcal{A}$  might not be preserved. On the other hand, from [5, Theorem 19] we know that there exists a deterministic parity automaton equivalent to  $\mathcal{A}$  with the same index, but it might have more than exponentially many states. Here we are able to guarantee both the preservation of the index and an exponential upper bound on the size of the deterministic automaton.

Observe that Theorem 9 can be applied both to  $\mathcal{A}$  and its dual. Therefore, we can fix a pair of nondeterministic GFG parity automata  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  that recognise  $L(\mathcal{A})$  and  $L(\mathcal{A})^c$  respectively and are both of size exponential in  $\mathcal{A}$ . We use the automata  $\mathcal{A}$ ,  $\text{box}(\mathcal{A})$ , and  $\text{box}(\bar{\mathcal{A}})$  to construct two auxiliary games  $G(\mathcal{A})$  and  $G'(\mathcal{A})$ .

The game  $G(\mathcal{A})$  proceeds from a configuration consisting of a pair  $(p, q)$  of states from  $\text{box}(\bar{\mathcal{A}})$  and  $\mathcal{A}$  respectively, starting from their initial states, as follows:

- Adam chooses a letter  $a \in \Sigma$ ;
- Eve chooses a transition  $p \xrightarrow{a} p'$  in  $\text{box}(\bar{\mathcal{A}})$ ;
- Eve and Adam play on the one-step arena over  $a$  from  $q$  to a new state  $q'$ .

A play in  $G(\mathcal{A})$  consists of a run  $\rho$  in  $\text{box}(\bar{\mathcal{A}})$  and a path  $\rho'$  in  $\mathcal{A}$ . It is winning for Eve if either  $\rho$  is accepting in  $\text{box}(\bar{\mathcal{A}})$  (in which case  $w \notin L(\mathcal{A})$ ), or  $\rho'$  is accepting in  $\mathcal{A}$ .

If  $\mathcal{A}$  is  $\exists$ -GFG and  $\text{box}(\bar{\mathcal{A}})$  is GFG, Eve has a winning strategy in  $G(\mathcal{A})$  consisting of building a run in  $\text{box}(\bar{\mathcal{A}})$  using her GFG strategy in  $\text{box}(\bar{\mathcal{A}})$  and a path in  $\mathcal{A}$  using her  $\exists$ -GFG strategy in  $\mathcal{A}$ . This guarantees that if  $w \in L(\mathcal{A})$  then the path in  $\mathcal{A}$  is accepting, and otherwise the run in  $\text{box}(\bar{\mathcal{A}})$  is accepting.

We then argue that as the winning condition of  $G(\mathcal{A})$  is a Rabin condition, Eve also has a winning strategy that is positional in  $\mathcal{A}$ , that is, which only depends on the history of the word and the current position. (Interestingly, the question of whether Eve can resolve the nondeterminism in a class of alternating GFG automata with only the knowledge of the word read so far does not tightly correspond to whether the acceptance condition of this class is memoryless. For example, it does hold for the generalised-Büchi condition, though it is not memoryless.) See Appendix C for details.

► **Remark 16.** There is some magic here: both the GFG strategies of Eve in  $\mathcal{A}$  and in  $\text{box}(\bar{\mathcal{A}})$  may require exponential memory, yet, when she needs to satisfy the disjunction of the two conditions, no more memory is needed. In a sense, the states of  $\mathcal{A}$  provide the memory for  $\text{box}(\bar{\mathcal{A}})$  and the states of  $\text{box}(\bar{\mathcal{A}})$  provide the memory for  $\mathcal{A}$ .

The game  $G'(\mathcal{A})$  is similar, except that Adam is given control of  $\text{box}(\mathcal{A})$  and Eve is in charge of letters. This time Adam wins a play, consisting of a run of  $\text{box}(\mathcal{A})$  and a path in  $\mathcal{A}$ , if either the path of  $\mathcal{A}$  is rejecting or the run of  $\text{box}(\mathcal{A})$  is accepting.

Accordingly, if  $\mathcal{A}$  is GFG, then he can win by using the  $\exists$ -GFG strategy in  $\text{box}(\mathcal{A})$  and the  $\forall$ -GFG strategy in  $\mathcal{A}$ . Then if  $w \in L(\mathcal{A})$ , the run in  $\text{box}(\mathcal{A})$  is accepting, and otherwise the path of  $\mathcal{A}$  is rejecting. As before, he also has a positional winning strategy in  $G'(\mathcal{A})$ .

We are now ready to build the deterministic automaton from a GFG APW  $\mathcal{A}$ , using positional winning strategies  $\sigma$  and  $\tau$  for Eve and Adam in  $G(\mathcal{A})$  and  $G'(\mathcal{A})$ , respectively.

Let  $\mathcal{D}$  be the automaton with states of the form  $(q, p_1, p_2)$ , with  $q$  a state of  $\mathcal{A}$ ,  $p_1$  a state of  $\text{box}(\mathcal{A})$  and  $p_2$  a state of  $\text{box}(\bar{\mathcal{A}})$ . A transition of  $\mathcal{D}$  over  $a$  moves to  $(q', p'_1, p'_2)$  such that moving from  $(q, p_1)$  to  $(q', p_1)$  is consistent with  $\tau$ ; and moving from  $(q, p_2)$  to  $(q', p'_2)$  is consistent with  $\sigma$ . The acceptance condition of  $\mathcal{D}$  is inherited from  $\mathcal{A}$ .

► **Lemma 17.** *For a GFG APW  $\mathcal{A}$  and  $\mathcal{D}$  built as above,  $L(\mathcal{A}) = L(\mathcal{D})$ .*

► **Remark 18.** To extend this construction to an alternating GFG Rabin automaton  $\mathcal{A}$ , we would need to remove alternations from both  $\mathcal{A}$  and its dual while preserving GFGness. However, the dual is a Streett automaton, for which we cannot invoke positional determinacy.

## 5 Deciding GFGness of Alternating Automata

We use the development of the last section to show that deciding whether an APW is GFG is in EXPTIME. This matches the best known upper bound for the same problem on NPW.

The main result of this section is the following theorem; its proof is in Appendix D.1.

► **Theorem 19.** *There exists an EXPTIME algorithm that takes as input an alternating parity automaton  $\mathcal{A}$  and decides whether  $\mathcal{A}$  is GFG.*

The idea is to construct the (exponential size) NPWs  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  for  $L(\mathcal{A})$  and  $L(\mathcal{A})^c$  respectively, which are GFG if and only if  $\mathcal{A}$  is  $\exists$ -GFG and  $\forall$ -GFG respectively. Then, it remains to check whether both are indeed GFG. Since we don't have a polynomial procedure to check this, instead, we will build a game which Eve wins if and only if *both* are indeed GFG, and which we can solve in exponential time with respect to the size of  $\mathcal{A}$ .

First, we observe the following reciprocal of Lemma 13.

► **Lemma 20.** *If  $\text{box}(\mathcal{A})$  is GFG then  $\mathcal{A}$  is  $\exists$ -GFG.*

**Proof.** Assume that  $\text{box}(\mathcal{A})$  is GFG and consider a strategy witnessing this. Such a strategy can be easily turned into a function  $\sigma': \Sigma^+ \rightarrow \mathcal{B}_{\mathcal{A}}$  that, given a word  $w \in L(\mathcal{A})$  produces

a universally accepting word of boxes of  $\mathcal{A}$ . Now, due to the definition of a box, each such box defines a positional strategy of Eve in the respective one-step game. This allows us to construct a winning strategy of Eve in the letter game over  $\mathcal{A}$ . ◀

Thus,  $\mathcal{A}$  is GFG if and only if both  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  are GFG. To decide this, we consider a game  $G''$  where Adam plays letters and Eve produces runs of the automata  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  in parallel. The winning condition of  $G''$  requires that at least one of the constructed runs must be accepting.

Now, each sequence of letters given by Adam belongs either to the language of  $\text{box}(\mathcal{A})$  or to  $\text{box}(\bar{\mathcal{A}})$  and therefore, a winning strategy of Eve in  $G''$  must comprise of two strategies witnessing GFGness of both  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$ . Dually, if both  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  are GFG then Eve wins  $G''$  by playing the two strategies in parallel.

It remains to show that  $G''$  is solvable in EXPTIME. Its winning condition is a disjunction of parity conditions, with index linear in the number of transitions of  $\mathcal{A}$ . This winning condition is recognised by a deterministic parity automaton of exponential size with polynomial index. To solve  $G''$ , we take its product with this deterministic automaton that recognises its winning condition, and solve the resulting parity game with an algorithm that is polynomial in the size of the game whenever, like here, the number of priorities is logarithmic in the size of the game, for instance [6]. Details of this construction and its complexity are in Appendix D.1.

## 6 Conclusions

The results obtained in this work shed new light on where alternating GFG automata resemble nondeterministic ones, and where they differ. Overall, our results show that allowing GFG alternations add succinctness without significantly increasing the complexity of determinisation nor decision procedures.

In particular, we show that alternating parity GFG automata can be exponentially more succinct than any equivalent nondeterministic GFG automata, yet this succinctness does not become double exponential when compared to deterministic automata, answering a question from [5]. Some further succinctness problems are left open here, such as the possibility of a doubly exponential gap between alternating GFG automata of stronger acceptance conditions and deterministic ones, as well as between  $\exists$ -GFG parity automata and deterministic ones.

We also show that the interplay between the two players can be used to decide whether an automaton is GFG without deciding  $\exists$ -GFG and  $\forall$ -GFG separately, yielding an EXPTIME algorithm. This matches the current algorithms for deciding GFGness on non-deterministic automata. Bagnol and Kuperberg conjectured that GFGness is PTIME decidable for non-deterministic parity automata of fixed index [2]; we extend this conjecture to alternating automata.

It then becomes interesting to ask how to build an alternating automaton GFG. Indeed, Henzinger and Piterman [11] proposed a transformation of nondeterministic automata into GFG automata, which, despite in some cases leading to a deterministic automaton, is, conceptually, a much simpler procedure than determinisation. Indeed, in many examples of non-GFG automata, adding transitions suffices to obtain a GFG one. We leave finding such a procedure for alternating automata as future work.

---

## References

- 1 Bader Abu Radi and Orna Kupferman. Minimizing GFG transition-based automata. In *Proceedings of ICALP*, pages 100:1–100:16, 2019.

- 2 Marc Bagnol and Denis Kuperberg. Büchi good-for-games automata are efficiently recognizable. In *Proceedings of FSTTCS*, pages 16:1–16:14, 2018.
- 3 Udi Boker, Denis Kuperberg, Orna Kupferman, and Michał Skrzypczak. Nondeterminism in the presence of a diverse or unknown future. In *Proceedings of ICALP*, pages 89–100, 2013.
- 4 Udi Boker, Orna Kupferman, and Michał Skrzypczak. How deterministic are good-for-games automata? In *Proceedings of FSTTCS*, pages 18:1–18:14, 2017.
- 5 Udi Boker and Karoliina Lehtinen. Good for games automata: From nondeterminism to alternation. In *Proceedings of CONCUR*, 2019.
- 6 Cristian S Calude, Sanjay Jain, Bakhadyr Khossainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In *Proceedings of STOC*, pages 252–263, 2017.
- 7 Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *Proceedings of ICALP*, pages 139–150, 2009.
- 8 Thomas Colcombet. Fonctions régulières de coût. Habilitation thesis, 2013.
- 9 Thomas Colcombet and Nathanaël Fijalkow. Universal graphs and good for games automata: New tools for infinite duration games. In *Proceedings of FOSSACS*, pages 1–26, 2019.
- 10 Christian Dax and Felix Klaedtke. Alternation elimination by complementation. In *Proceedings of LPAR*, pages 214–229, 2008.
- 11 Thomas Henzinger and Nir Piterman. Solving games without determinization. In *Proceedings of CSL*, pages 395–410, 2006.
- 12 Simon Iosti and Denis Kuperberg. Eventually safe languages. In *Proceedings of DLT*, pages 192–205, 2019.
- 13 Nils Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. *Annals of Pure and Applied Logic*, 69(2):243–268, 1994.
- 14 Joachim Klein, David Müller, Christel Baier, and Sascha Klüppelholz. Are good-for-games automata good for probabilistic model checking? In *Proceedings of LATA*, pages 453–465, 2014.
- 15 Denis Kuperberg and Anirban Majumdar. Computing the width of non-deterministic automata. *Logical Methods in Computer Science*, 15(4), 2019.
- 16 Denis Kuperberg and Michał Skrzypczak. On determinisation of good-for-games automata. In *Proceedings of ICALP*, pages 299–310, 2015.
- 17 Denis Kuperberg and Michael Vanden Boom. Quasi-weak cost automata: A new variant of weakness. In *Proceedings of FSTTCS*, pages 66–77, 2011.
- 18 Karoliina Lehtinen and Martin Zimmermann. Good-for-games  $\omega$ -pushdown automata. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 689–702, 2020.
- 19 Christof Löding and Stefan Repke. Decidability Results on the Existence of Lookahead Delegates for NFA. In *Proceedings of FSTTCS*, pages 327–338, 2013.
- 20 Satoru Miyano and Takeshi Hayashi. Alternating finite automata on  $\omega$ -words. *Theoretical Computer Science*, 32:321–330, 1984.
- 21 Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Logical Methods in Computer Science*, 3(3):5, 2007.
- 22 Domenic Quirl. Bachelor Thesis, supervised by Christof Löding, RWTH Aachen, 2018.

# Appendix

## A Appendix of Section 2

In this section of the appendix we provide the remaining technical definitions from Section 2 that are used in the proofs.

**Games.** A  $\Sigma$ -arena is a directed (finite or infinite) graph with nodes (positions) split into  $E$ -labelled positions of Eve and  $A$ -labelled positions of Adam, where the edges (transitions) are labelled by elements of  $\Sigma \sqcup \{\epsilon\}$ . The role of  $\epsilon$  is to mark edges that have no influence on the winner of a play, e.g., edges allowing players to resolve some Boolean formula.

We represent such an arena as  $R = (V, X, V_E, V_A)$ , where  $V$  is its set of positions;  $X \subseteq V \times (\Sigma \sqcup \{\epsilon\}) \times V$  its transitions;  $V_E \subseteq V$  the  $E$ -positions; and  $V_A = V \setminus V_E$  the  $A$ -positions.

Notice that the definition allows more than one transition between a pair of positions (such transitions needs to have distinct labels). We will require that each infinite path contains infinitely many  $\Sigma$ -labelled transitions. An arena might be rooted at an initial position  $v_i \in V$ . We say that a position  $v$  is *terminal* if there is no outgoing transition from  $v$  (i.e. no element of  $X$  of the form  $(v, a, v')$ ). If we don't say that an arena is *partial* then it is assumed that there are no terminal positions.

If  $R$  is a (partial)  $\Sigma$ -arena and  $V' \subseteq V$  is a set of positions, then  $R|_{V'}$  is the *sub-arena* of  $R$  defined as the restriction of  $R$  to the positions in  $V'$ , namely for  $P \in \{E, A\}$ , the  $P$ -positions of  $R|_{V'}$  are  $V'_P := V_P \cap V'$ , and its transitions are  $X' := X \cap (V' \times (\Sigma \sqcup \{\epsilon\}) \times V')$ . We say that two (partial)  $\Sigma$ -arenas  $R = (V, X, V_E, V_A)$  and  $R' = (V', X', V'_E, V'_A)$  are *isomorphic* if there exists a bijection  $i: V \rightarrow V'$  that preserves the membership in  $V_P/V'_P$ , for  $P \in \{E, A\}$ , and sets of transitions  $X/X'$ .

A *partial play* in  $R$  is a path in  $R$ , i.e., an element  $\pi = v_0 e_0 v_1 e_1 \dots$  of  $V \cdot (X \cdot V)^* \cup (V \cdot X)^\omega$ , where for every  $i$  we have  $e_i = (v_i, a_i, v_{i+1})$ . Such a partial play is said to *begin* in  $v_0$ . A partial play is a *play* if either it is infinite or the last position  $v_i$  is terminal.

A *game* is a  $\Sigma$ -arena together with a winning condition  $W \subseteq \Sigma^\omega$ . An infinite play  $\pi$  is said to be winning for Eve in the game if the sequence of  $\Sigma$ -labels  $(a_i)_{i \in \mathbb{N}}$  of the transitions along  $\pi$  form a word in  $W$ . Else  $\pi$  is winning for Adam. Games with some class  $X$  of winning conditions (e.g., the parity condition) are called  $X$  games (e.g., parity games).

A *strategy* for Eve (resp. Adam) is a function  $\tau: V \cdot (X \cdot V)^* \rightarrow X$  that maps a *history*  $v_0 e_0 v_1 \dots e_{i-1} v_i$ , i.e. a finite prefix of a play in  $R$ , to a transition  $e_i$  whenever  $v_i$  belongs to  $V_E$  (resp. to  $V_A$ ). A partial play  $v_0 e_0 v_1 e_1 \dots$  agrees with a strategy  $\tau$  for Eve (Adam) if whenever  $v_i \in V_E$  (resp. in  $V_A$ ), we have  $e_i = \tau(v_0 e_0 v_1 \dots e_{i-1} v_i)$ . A strategy for Eve (Adam) is winning from a position  $v \in V$  if all plays beginning in  $v$  that agree with it are winning for Eve (Adam). We say that a player wins the game from a position  $v \in V$  if they have a winning strategy from  $v$ . If the game is rooted at  $v_i$ , we say that a player wins the game if they win from  $v_i$ .

A strategy is *positional* if its value depends only on the last position, i.e.,  $\tau(v_0 e_0 \dots e_{i-1} v_i)$  depends only on  $v_i$ . In that case the strategy of a player  $P$  can be represented as a function  $\tau: V_P \rightarrow X$ .

We also define the notion of *strategy with memory*  $M$  for player  $P$ . This is a tuple  $(\sigma, M, m_0, upd)$  where  $M$  is a set of *memory states*;  $m_0$  is an *initial memory state*;  $upd: M \times X \rightarrow M$  is an *update function*, and  $\sigma: M \times V_P \rightarrow X$  is a strategy deciding which move should be played, depending only on the current memory state and on the current position.

Along a play, the memory starts with  $m_0$ , and is updated along every transition according to  $upd$ . The general notion of strategy corresponds to  $M = (V \cdot X)^*$ , and positional strategies correspond to  $M$  being a singleton. A player has a *finite-memory winning strategy* if there exists a winning strategy using a finite memory set  $M$ .

► **Proposition 21.** *Let  $G$  and  $G'$  be two  $\Sigma$ -games with the same winning condition, such that the unfoldings of  $G$  and  $G'$  are isomorphic. Then Eve has a winning strategy in  $G$  if and only if she has a winning strategy in  $G'$ .*

► **Proposition 22** ([13]). *Rabin games are positionally determined for Eve. (If Eve has a winning strategy then she has a positional winning strategy.)*

► **Definition 23** (Synchronised product). *The synchronised product  $R \times \mathcal{A}$  of a (partial)  $\Sigma$ -arena  $R = (V, X, V_E, V_A)$  and an alternating automaton  $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$  with a set of transitions  $T$  and labelling  $\alpha : T \rightarrow \Gamma$  is a (partial)  $\Sigma \times \Gamma$ -arena defined as follows. Its set of positions is  $(V \times Q) \cup (V \times Q \times \Sigma \times \widehat{\delta})$ , and its transitions are defined by:*

$$\begin{aligned} \langle (v, q), \epsilon, (v', q, a, \delta(q, a)) \rangle & \quad \text{for } (v, q) \in V \times Q \text{ and } \langle v, a, v' \rangle \in X; \\ \langle (v, q), \epsilon, (v', q) \rangle & \quad \text{for } (v, q) \in V \times Q \text{ and } \langle v, \epsilon, v' \rangle \in X; \\ \langle (v, q, a, b), \epsilon, (v, q, a, b_i) \rangle & \quad \text{for } (v, q, a, b) \in V \times Q \times \Sigma \times \widehat{\delta} \\ & \quad \text{with } b = b_1 \vee b_2 \text{ or } b = b_1 \wedge b_2 \text{ and } i = 1, 2; \\ \langle (v, q, a, q'), (a, \alpha(q, a, q')), (v, q') \rangle & \quad \text{for } (v, q, a, q') \in V \times Q \times \Sigma \times \widehat{\delta} \text{ with } q' \in Q. \end{aligned}$$

*The positions belonging to Eve are of the form  $(v, q)$  where  $v \in V_E$  and of the form  $(v, q, a, b_1 \vee b_2)$ . The remaining ones belong to Adam. If  $R$  has an initial position  $v_\iota$  then the initial position of the product is  $(v_\iota, \iota)$ .*

We implicitly assume that the arena only contains vertices that are reachable from  $V \times Q$ . (They need not be reachable from an initial position of  $R$  and an initial state of  $\mathcal{A}$ , but from some position of  $R$  and state of  $\mathcal{A}$ .)

We will sometimes consider longer products, like  $(R \times \mathcal{A}_1) \times \mathcal{A}_2$ , where  $R$  is a  $\Sigma$ -arena and both automata  $\mathcal{A}_1$  and  $\mathcal{A}_2$  are over the alphabet  $\Sigma$ . Assume that  $\mathcal{A}_1$  and  $\mathcal{A}_2$  have transitions labelled in sets  $\Gamma_1$  and  $\Gamma_2$  respectively. Notice that in that case the arena  $R \times \mathcal{A}_1$  is formally a  $\Sigma \times \Gamma_1$ -arena. Thus, to make the above formula precise, we treat the automaton  $\mathcal{A}_2$  as an automaton over the alphabet  $\Sigma \times \Gamma_1$  and assume that it ignores the second component of the letters read.

**One-step arenas.** For a letter  $a \in \Sigma$ , we denote by  $R_a$  a partial  $\Sigma$ -arena consisting of two vertices  $v$  and  $v'$  (it does not matter which of the players controls them), and one transition  $(v, a, v')$ . Then, for an automaton  $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$ , the product  $R_a \times \mathcal{A}$  is a partial arena, in which the players should resolve their choices in the formulas  $\delta(q, a)$  for all the possible states  $q \in Q$ . We call it the *one-step arena* of  $\mathcal{A}$  over  $a$ . Such an arena contains one position of the form  $(v, q)$  for each state  $q \in Q$ ; a set of non-terminal positions of the form  $(v', q, a, \psi)$  for some  $q \in Q$  and  $\psi \in \widehat{\delta}$ ; and one terminal position of the form  $(v', q)$  for each state  $q \in Q$ . (See Figure 1.)

**Boxes.** In the later exposition, we will be interested in the combinatorial structure of possible strategies of Eve over one-step arenas  $R_a \times \mathcal{A}$ . For a positional strategy  $\sigma$  of Eve in a game of the form  $R_a \times \mathcal{A}$ , we define the *box* of  $\mathcal{A}$ ,  $a$ , and  $\sigma$ , denoted by  $\beta(\mathcal{A}, a, \sigma)$ , as the relation that is a subset of  $Q \times \Sigma \times Q$  and contains a triple  $(q, a, q')$  iff there exists a play in  $R_a \times \mathcal{A}$  that is consistent with  $\sigma$ , starting in  $(v, q)$  and ending in  $(v', q')$ . We further define for every  $a \in \Sigma$ , the set  $B_{(\mathcal{A}, a)} = \{\beta(\mathcal{A}, a, \sigma) \mid \sigma \text{ is a positional strategy of Eve}\}$ . Finally, let

$B_{\mathcal{A}} := \bigcup_{a \in \Sigma} B_{(\mathcal{A}, a)}$ . Notice that  $|B_{\mathcal{A}}| \leq 2^{|Q \times \Sigma \times Q|}$ . When speaking of an arbitrary box, we mean any non-empty relation  $\beta \subseteq Q \times \Sigma \times Q$  where all the letters  $a$  appearing on the middle component are equal.

Figure 2 represents  $B_{(\mathcal{A}, a)}$  for the automaton  $\mathcal{A}$  of Figure 1: Since there are two binary-choice positions of Eve in the corresponding one-step arena, there are four distinct positional strategies of Eve, which give the four possible boxes. They correspond to Eve choosing respectively LL, LR, RL, RR, where L stands for a left choice and R for a right choice in each of her two binary-choice positions.

► **Proposition 24.** *Consider a letter  $a$  and an automaton  $\mathcal{A}$  with states  $Q$  and transition function  $\delta$ . Then there is a bijection between  $B_{(\mathcal{A}, a)}$  and the positional strategies of Eve in the one-step arena of  $\mathcal{A}$  and  $a$ .*

**Acceptance of a word by an automaton.** We define the acceptance directly in terms of the model-checking (acceptance/membership) game, which happens to be exactly the product of the automaton with a path-like arena describing the input word. More precisely, given a word  $w \in \Sigma^\omega$ , the *model-checking game* is defined as the product  $R_w \times \mathcal{A}$ , where the arena  $R_w$  consists of an infinite path  $\omega$ , of which all positions belong to Eve (although it does not matter); the transitions are of the form  $\langle i, w_i, i+1 \rangle$ ; the initial position is 0; and the winning condition is based on the winning condition of  $\mathcal{A}$  (the  $\Sigma$ -component of the labels is ignored). We say that  $\mathcal{A}$  *accepts*  $w$  if Eve has a winning strategy in the model-checking game  $R_w \times \mathcal{A}$ . The language of an automaton  $\mathcal{A}$ , denoted by  $L(\mathcal{A})$ , is the set of words that it accepts (recognises).

Notice that for each  $i \in \mathbb{N}$ , the sub-arena of  $R_w \times \mathcal{A}$  with positions in

$$(\{i\} \times Q) \cup (\{i+1\} \times Q \times \Sigma \times B^+(Q)) \cup (\{i+1\} \times Q)$$

is isomorphic to the one-step arena of  $\mathcal{A}$  over  $w_i$ .

## B Appendix of Section 3

► **Definition 25** (A formalisation of Definition 3). *Let  $R_{A, \Sigma}$  be the  $\Sigma$ -arena consisting of a single position  $v$  that belongs to Adam and the set of transitions  $X$  of the form  $(v, a, v)$  for each letter  $a \in \Sigma$  (see Figure 5). The arena  $R_{E, \Sigma}$  is the same except that  $v$  belongs to Eve. Notice that the products  $R_{A, \Sigma} \times \mathcal{A}$  and  $R_{E, \Sigma} \times \mathcal{A}$  are both labelled by  $\Sigma \times \Gamma$ , where  $\Sigma$  is the alphabet of  $\mathcal{A}$  and  $\Gamma$  is  $\mathcal{A}$ 's labelling, on top of which its acceptance condition is defined. Thus, the winning condition of games defined on these arenas can depend on a sequence of labels of the form  $(a_i, \gamma_i)_{i \in \mathbb{N}}$ . Then, Eve's letter game is played over  $R_{A, \Sigma} \times \mathcal{A}$ , where Eve wins if:*

$$(a_i)_{i \in \mathbb{N}} \notin L(\mathcal{A}) \text{ or the sequence } (\gamma_i)_{i \in \mathbb{N}} \text{ satisfies the acceptance condition of } \mathcal{A}.$$

Dually, Adam's letter game is played over  $R_{E, \Sigma} \times \mathcal{A}$ , where Adam wins if:

$$(a_i)_{i \in \mathbb{N}} \in L(\mathcal{A}) \text{ or the sequence } (\gamma_i)_{i \in \mathbb{N}} \text{ violates the acceptance condition of } \mathcal{A}.$$

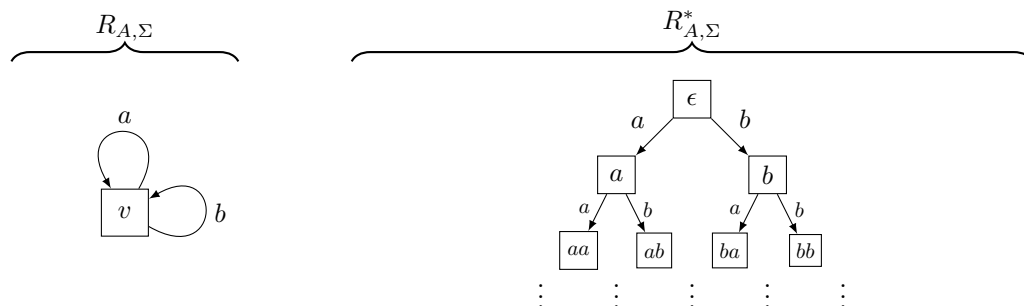
## C Appendix of Section 4

This section provides the technical details of the determinisation procedure in Section 4. We start with some technical analysis of the types of histories needed to win letter games.



### C.1 Good for Games Automata: Required Histories

We begin by considering an *expanded* letter game. This will allow us to use a form of positional determinacy in letter games.



■ **Figure 5** The arenas  $R_{A, \Sigma}$  and  $R_{A, \Sigma}^*$ , allowing Adam to choose an arbitrary sequence of letters. We define Eve's letter game for an automaton  $\mathcal{A}$  over the product of  $R_{A, \Sigma} \times \mathcal{A}$ , and her expanded letter game over  $R_{A, \Sigma}^* \times \mathcal{A}$ ; She wins a play if the word generated by Adam is not in  $L(\mathcal{A})$  or the path generated by her (resolving  $\mathcal{A}$ 's nondeterminism) and by Adam (resolving  $\mathcal{A}$ 's universality) satisfies  $\mathcal{A}$ 's acceptance condition.

**Expanded letter games.** The definition of the letter games (Definition 3) has the important advantage of being defined over a finite-arena. Yet, as a result, these games generally do not allow for positional determinacy.

We provide below an expanded variant of the letter game that will have same unfolding as the original one, while being defined over an infinite arena. This will allow Eve to have positional determinacy in these games for Rabin automata.

Let  $R_{A, \Sigma}^*$  be the  $\Sigma$ -arena with the set of positions  $V = \Sigma^*$ , all belonging to Adam, and the set of transitions  $X$  of the form  $\langle w, a, w \cdot a \rangle$  for each word  $w \in \Sigma^*$  and letter  $a \in \Sigma$  (see Figure 5). The initial position of this arena is  $\epsilon$ . The arena  $R_{E, \Sigma}^*$  is the same, except that all the positions belong to Eve. We define *Eve's expanded letter game* over  $R_{A, \Sigma}^* \times \mathcal{A}$  and *Adam's expanded letter game* over  $R_{E, \Sigma}^* \times \mathcal{A}$  with the same winning conditions as in their (non expanded) variants. The following follows directly from Proposition 21.

► **Proposition 26.** *For every automaton  $\mathcal{A}$ , the expanded letter games for  $\mathcal{A}$  have the same winners as the (standard) letter games for  $\mathcal{A}$ .*

**History Requirement.** Although  $R_{A, \Sigma}^*$  and  $R_{E, \Sigma}^*$  are trees, the arenas of the expanded letter games are directed acyclic graphs as there can exist two distinct paths from the initial position to a given position  $(w, q)$ . Thus, a priori, a winning strategy of a player of such a game might need some history of a play. However, as expressed by the following theorem, it is not the case.

► **Theorem 27.** *If  $\mathcal{A}$  is a Rabin (or parity) automaton then Eve's expanded letter game is positionally determined for Eve.*

**Proof.** We will show that the winning condition of this game can be represented as a Rabin condition and invoke Proposition 22. Let  $\mathcal{D}'$  be a deterministic parity automaton recognising the complement of the language  $L(\mathcal{A})$ , for a Rabin automaton  $\mathcal{A}$  over the alphabet  $\Sigma$ . Let

$\Omega: Q_{\mathcal{D}'} \rightarrow \mathbb{N}$  be the priority assignment of  $\mathcal{D}'$  (without loss of generality we can assume that the states of  $\mathcal{D}'$  bear priorities).

Consider the arena  $R_{A,\Sigma}^*$  that is derived from the arena  $R_{A,\Sigma}^*$  by adding to transitions priorities according to the deterministic of runs of  $\mathcal{D}'$ , that is, by changing the labelling of every transition  $\langle w, a, w \cdot a \rangle$  to  $\langle w, (\Omega_{\mathcal{D}'}(q), a), w \cdot a \rangle$ , where  $a \in \Sigma$  and  $q$  is the state of  $\mathcal{D}'$  reached after reading the word  $w$  from the initial state of  $\mathcal{D}'$ .

Consider the product  $R_{A,\Sigma}^* \times \mathcal{A}$ , in which for  $\mathcal{A}$ 's transitions we ignore these additional labels. The labels of that product are now of the form  $(\ell, a, \gamma)$ , where  $\ell \in \mathbb{N}$  is a priority of  $\mathcal{D}'$ ,  $a \in \Sigma$ , and  $\gamma \in \Gamma$  is a label of  $\mathcal{A}$ . Notice that when one forgets about the first coordinate of the label, the game is equal to  $R_{A,\Sigma}^* \times \mathcal{A}$ . Moreover, given a sequence of labels  $(\ell_i, a_i, \gamma_i)_{i \in \mathbb{N}}$ , by the choice of  $\mathcal{D}'$ , we know that  $(a_i)_{i \in \mathbb{N}} \notin L(\mathcal{A})$  if and only if the sequence  $(\ell_i)_{i \in \mathbb{N}}$  satisfies the parity condition.

Define the game  $\mathcal{G}$  over  $R_{A,\Sigma}^* \times \mathcal{A}$ , in which Eve wins a play labelled by  $(\ell_i, a_i, \gamma_i)_{i \in \mathbb{N}}$  if

$(\ell_i)_{i \in \mathbb{N}}$  satisfies the parity condition of  $\mathcal{D}'$  or  $(\gamma_i)_{i \in \mathbb{N}}$  satisfies the Rabin condition of  $\mathcal{A}$ .

Notice that both disjuncts above can be written as Rabin conditions and therefore  $\mathcal{G}$  is positionally determined for Eve. Moreover, the choice of  $\mathcal{D}'$  guarantees that the new winning condition is equivalent to Eve's condition in her expanded letter game on  $\mathcal{A}$ —the same plays are winning for Eve in  $\mathcal{G}$  and her expanded letter game on  $\mathcal{A}$ . Since the structure of the game is also preserved, it means that Eve's expanded letter game on  $\mathcal{A}$  is positionally determined for Eve.  $\blacktriangleleft$

► **Remark 28.** Dually, Adam's expanded letter game for a Streett automaton is positionally determined for Adam.

As a consequence of Theorem 27, for alternating  $\exists$ -GFG Rabin automata, a strategy for Eve to resolve the nondeterminism may ignore the history of the play, and only consider the history of the word read, as is the case for nondeterministic GFG automata.

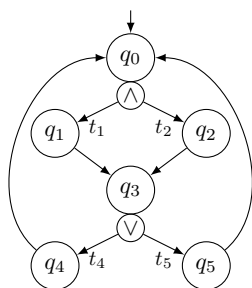
We will now argue that Eve's positional strategy  $\sigma$  in the expanded letter game on an alternating automaton  $\mathcal{A}$  can be represented as a function  $\sigma': \Sigma^+ \rightarrow \mathbf{B}_{\mathcal{A}}$  that assigns to each word  $wa$  a box  $\beta_{wa} \in \mathbf{B}_{(\mathcal{A},a)}$ . Indeed, let  $(w \cdot a) \in \Sigma^+$  and let  $V_{wa}$  be the set of positions of  $R_{A,\Sigma}^* \times \mathcal{A}$  of the form  $(w, q)$ ,  $(w \cdot a, q, a, b)$ , or  $(w \cdot a, q)$  for  $q \in Q$  and  $b \in \mathbf{B}^+(Q)$ . Observe that the partial arena of  $R_{A,\Sigma}^* \times \mathcal{A}$  restricted to  $V_{wa}$  is isomorphic to the one-step arena  $R_a \times \mathcal{A}$ . Thus,  $\sigma$  provides a positional strategy over this arena, which by Proposition 24 can be encoded as a box  $\beta_{wa}$ . More formally, let  $\beta_{wa}$  contain  $(q, a, q')$ , if there is a play consistent with  $\sigma$  that visits both the positions  $(w, q)$  and then  $(wa, q')$ .

Then, in the next lemma we show that if  $\sigma$  is also winning, then the sequences of boxes  $\beta_{wa}$  only has accepting paths.

► **Definition 29.** Consider an automaton  $\mathcal{A}$  with states  $Q$  and initial state  $\iota$ , and an infinite word  $u = \beta_0, \beta_1, \dots \in (\mathbf{B}_{\mathcal{A}})^\omega$ . We say that a sequence of transitions  $\rho = (q_i, a_i, q_{i+1})_{i \in \mathbb{N}}$  is a path of  $u$  if  $q_0 = \iota$  and for every  $i \in \mathbb{N}$ , we have  $(q_i, a_i, q_{i+1}) \in \beta_i$ . The word  $u$  is universally accepting for  $\mathcal{A}$  if each of its paths satisfies the acceptance condition of  $\mathcal{A}$ .

► **Lemma 30.** Given an alternating  $\exists$ -GFG Rabin automaton  $\mathcal{A}$ , there is a positional strategy  $\sigma$  in her expanded letter game on  $\mathcal{A}$  such that for every word  $w \in L(\mathcal{A})$  the sequence of boxes  $u = \beta_0, \beta_1, \dots \in (\mathbf{B}_{\mathcal{A}})^\omega$  defined as  $\beta_i = \sigma(w \upharpoonright_{i+1})$  is universally accepting for  $\mathcal{A}$ .

**Proof.** Consider words  $w$  and  $u$  as above. Let  $\rho = (q_i, a_i, q_{i+1})_{i \in \mathbb{N}}$  be a path of  $u$ . Since the strategy  $\sigma$  is positional, the definition of  $\sigma(w \upharpoonright_{i+1})$  implies that there exists a single play of



Acceptance condition:  
 (Finitely often  $t_1$  or Infinitely often  $t_4$ ) and  
 (Finitely often  $t_2$  or Infinitely often  $t_5$ )

■ **Figure 6** A GFG ASW over a singleton alphabet, for which Eve's  $\exists$ -GFG strategy cannot only remember the prefix of the word read so far, but also some history about the visited states.

the expanded letter game that visits all the positions of the form  $(w|_i, q_i)$  for  $i \in \mathcal{N}$ . Since  $w \in L(\mathcal{A})$ , the winning condition of the expanded letter game guarantees that the path  $\rho$  must be accepting. ◀

Observe that the above arguments do not hold for alternating GFG Streett automata: Since Streett games are not positionally determined for Eve, Eve's expanded letter game for a Streett automaton is not positionally determined for Eve (an analogous of Theorem 27 does not hold). Furthermore, we provide in Figure 6 an example of an alternating GFG Streett automaton, in which Eve cannot resolve her nondeterminism only according to the history of the word read.

► **Proposition 31.** *Consider an  $\exists$ -GFG alternating Streett automaton  $\mathcal{A}$  with transition conditions in DNF. Then Eve might not have a strategy  $\sigma: \Sigma^+ \rightarrow \mathcal{B}_{\mathcal{A}}$  satisfying Lemma 30.*

**Proof.** Consider the ASW depicted in Figure 6. It is  $\exists$ -GFG, as witnessed by the strategy that chooses the transition  $t_4$  in  $q_3$  if the last visited state was  $q_1$  and  $t_5$  otherwise. Yet, there is no strategy that only remembers the word read so far, as this only gives the length of the word, and cannot help in determining whether the path visited  $q_1$  or  $q_2$ . ◀

## C.2 Alternation Removal in GFG Parity Automata

This section presents the proof of the following theorem:

► **Theorem 9.** *Consider an alternating parity automaton  $\mathcal{A}$  with  $n$  states and index  $k$ . There exists a nondeterministic parity automaton  $\text{box}(\mathcal{A})$  with  $2^{O(nk \log nk)}$  states that is equivalent to  $\mathcal{A}$  such that if  $\mathcal{A}$  is GFG then  $\text{box}(\mathcal{A})$  is also GFG.*

► **Lemma 10.** *Consider an alternating parity automaton  $\mathcal{A}$  with  $n$  states and index  $k$ . Then there exists a deterministic parity automaton  $\mathcal{B}$  with  $2^{O(nk \log nk)}$  states over the alphabet  $\mathcal{B}_{\mathcal{A}}$  that recognises the set of universally-accepting words for  $\mathcal{A}$ . If  $\mathcal{A}$  is a Büchi automaton, then  $\mathcal{B}$  can also be taken as Büchi, and in general the parity index of the automaton  $\mathcal{B}$  is linear in the number of transitions of  $\mathcal{A}$ .*

**Proof.** Notice that it is easy to construct a nondeterministic parity automaton  $\mathcal{S}$  over the alphabet  $\mathcal{B}_{\mathcal{A}}$  that recognises the complement of the set of universally-accepting words for  $\mathcal{A}$ —it is enough to guess a path that is not accepting, and have the acceptance condition that is the dual of  $\mathcal{A}$ 's condition. If  $\mathcal{A}$  is a Büchi automaton, then  $\mathcal{S}$  is a coBüchi one. Formally, for an alternating parity (resp. Büchi) automaton  $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$ , we define

the nondeterministic parity (resp. coBüchi) automaton  $\mathcal{S} = (\mathbb{B}_{\mathcal{A}}, Q, \iota, \delta_{\mathcal{S}}, \bar{\alpha})$ , where  $\bar{\alpha}$  is the dual of  $\alpha$  and  $\delta_{\mathcal{S}}$  is defined as follows. For every states  $q, q' \in Q$  and box  $\beta \in \mathbb{B}_{\mathcal{A}}$ , we have  $q' \in \delta_{\mathcal{S}}(q, \beta)$  iff  $q \xrightarrow{a} q' \in \beta$  for some  $a$ .

Now, one can translate  $\mathcal{S}$  to an equivalent deterministic parity automaton  $\mathcal{B}'$  with  $2^{O(nk \log nk)}$  states [21] and then complement the acceptance condition of  $\mathcal{B}'$ , getting the required automaton  $\mathcal{B}$ .

Since nondeterministic coBüchi automata can be determinised into deterministic coBüchi automata, if  $\mathcal{A}$  is Büchi, so is  $\mathcal{B}$ . In general, the parity index of the automaton  $\mathcal{B}$  is linear in the number of transitions of  $\mathcal{A}$ . ◀

We now proceed to the construction of the automaton  $\text{box}(\mathcal{A})$  of Theorem 9. It is the same as the automaton  $\mathcal{B}$  of Lemma 10, except that the alphabet is  $\Sigma$  and the transition function is defined as follows: For every state  $p$  of  $\text{box}(\mathcal{A})$  and  $a \in \Sigma$ , we have  $\delta_{\text{box}(\mathcal{A})}(p, a) := \bigcup_{\beta \in \mathbb{B}_{(\mathcal{A}, a)}} \delta_{\mathcal{B}}(p, \beta)$ .

In other words, the automaton  $\text{box}(\mathcal{A})$  reads a letter  $a$ , nondeterministically guesses a box  $\beta \in \mathbb{B}_{\mathcal{A}, a}$ , and follows the transition of  $\mathcal{B}$  over  $\beta$ . Thus, the runs of  $\text{box}(\mathcal{A})$  over a word  $w \in \Sigma^\omega$  are in bijection between sequences of boxes  $(\beta_i)_{i \in \mathbb{N}}$  such that  $\beta_i \in \mathbb{B}_{\mathcal{A}, w_i}$  for  $i \in \mathbb{N}$ .

Fix an infinite word  $w \in \Sigma^\omega$ . Our aim is to prove that  $w \in L(\mathcal{A}) \Leftrightarrow w \in L(\text{box}(\mathcal{A}))$ .

► **Lemma 11.** *There exists a bijection between positional strategies of Eve in the acceptance game of  $\mathcal{A}$  over  $w$  and runs of  $\text{box}(\mathcal{A})$  over  $w$ . Moreover, a strategy is winning if and only if the corresponding run is accepting. Thus  $L(\mathcal{A}) = L(\text{box}(\mathcal{A}))$ .*

**Proof.** Consider a run of  $\text{box}(\mathcal{A})$  over  $w$ , and observe that it corresponds to a sequence of boxes  $\beta_0, \dots$ . Notice that each box  $\beta_i$  corresponds to Eve's choices in  $\mathcal{A}$  over  $w_i$ , and therefore provides a positional strategy for Eve in the one-step arena  $R_{w_i} \times \mathcal{A}$ . The sequence of these choices provides a positional strategy for Eve in  $R_w \times \mathcal{A}$ .

Dually, given a positional strategy for Eve in  $R_w \times \mathcal{A}$ , one can extract a sequence of strategies for Eve in the one-step arenas  $R_{w_i} \times \mathcal{A}$ , and each of them corresponds to a box  $\beta_i$ . Proposition 24 shows that each path in  $\beta_0, \dots$  corresponds to a play consistent with the constructed strategy and vice versa: each play gives rise to a path.

Now, a run is accepting if and only if the sequence of boxes is universally accepting, which means exactly that all the plays consistent with the corresponding strategy are winning. ◀

We now show that the automaton  $\text{box}(\mathcal{A})$  is also GFG.

► **Lemma 13.** *For an alternating  $\exists$ -GFG parity automaton  $\mathcal{A}$ , the automaton  $\text{box}(\mathcal{A})$  is GFG.*

**Proof.** Let  $\sigma$  be a positional winning strategy for Eve in her expanded letter game for  $\mathcal{A}$  (over the arena  $R_{\mathcal{A}, \Sigma}^* \times \mathcal{A}$ ). The proof is based on the construction of the function  $\sigma' : \Sigma^+ \rightarrow \mathbb{B}_{\mathcal{A}}$ , see the paragraph before Definition 29.

Consider the following way of resolving the nondeterminism of  $\text{box}(\mathcal{A})$ : after reading  $w \in \Sigma^*$ , when the next letter  $a \in \Sigma$  is provided, the automaton moves to the state  $\delta_{\mathcal{B}}(p, \beta_{wa})$  where  $\beta_{wa} = \sigma'(wa)$ . Consider an infinite word  $w \in L(\mathcal{A})$  and let  $\beta_0, \dots$  be the sequence of boxes used to construct the run of  $\text{box}(\mathcal{A})$  over  $w$ . Lemma 30 implies that this sequence is universally accepting and therefore, the constructed run of  $\mathcal{B}$  must also be accepting. ◀

### C.3 Single-Exponential Determinisation of Alternating Parity GFG Automata

The aim of this section is to prove the following determinisation theorem.

► **Theorem 14.** *If  $\mathcal{A}$  is an alternating parity GFG automaton then there exists a deterministic parity automaton  $\mathcal{D}$  that recognises the same language and has size at most exponential in the size of  $\mathcal{A}$ . Moreover, the parity index of  $\mathcal{D}$  is the same as that of  $\mathcal{A}$ .*

First consider the synchronised product  $R_{A,\Sigma} \times \text{box}(\bar{\mathcal{A}})$ , which is a game with labels of the form  $\Sigma \times \Gamma_{\text{box}(\bar{\mathcal{A}})}$ , where  $\Gamma_{\text{box}(\bar{\mathcal{A}})}$  is the parity condition of  $\text{box}(\bar{\mathcal{A}})$ . Now, we can treat the automaton  $\mathcal{A}$  as an automaton over the alphabet  $\Sigma \times \Gamma_{\text{box}(\bar{\mathcal{A}})}$  that just ignores the second component of the given letter. Thus, we can define a game  $G(\mathcal{A}) = (R_{A,\Sigma} \times \text{box}(\bar{\mathcal{A}})) \times \mathcal{A}$ .

Notice that  $G(\mathcal{A})$  is naturally divided into rounds, between two consecutive positions of the form  $(v, p, q)$ , where  $v$  is the unique position of  $R_{A,\Sigma}$ ,  $p$  is a state of  $\text{box}(\bar{\mathcal{A}})$ , and  $q$  is a state of  $\mathcal{A}$ . Such a round, starting in  $(v, p, q)$ , consists of first Adam choosing a letter  $a$ ; then Eve resolving nondeterminism of  $\text{box}(\bar{\mathcal{A}})$  from  $p$  over  $a$ ; and then both players playing the game corresponding to the transition condition  $\delta(q, a)$  of  $\mathcal{A}$ .

Let the winning condition of  $G(\mathcal{A})$  say that either the sequence of transitions of  $\text{box}(\bar{\mathcal{A}})$  is accepting or the sequence of transitions of  $\mathcal{A}$  is accepting. Since  $\mathcal{A}$  is  $\exists$ -GFG and  $\text{box}(\bar{\mathcal{A}})$  is GFG, we know that Eve has a winning strategy in  $G(\mathcal{A})$ : she just plays her GFG strategies in both automata and is guaranteed to win whether the word produced by Adam is in  $L(\mathcal{A})$  or  $L(\text{box}(\bar{\mathcal{A}}))$ .

As the winning condition of  $G(\mathcal{A})$  is a disjunction of two Rabin conditions, Eve has a positional winning strategy. Fix such a strategy  $\sigma$ .

Now do the same with  $\mathcal{A}$  and  $\text{box}(\mathcal{A})$  for Adam: define  $G'(\mathcal{A})$  as  $(R_{E,\Sigma} \times \overline{\text{box}(\mathcal{A})}) \times \mathcal{A}$ , where  $\overline{\text{box}(\mathcal{A})}$  is the automaton  $\text{box}(\mathcal{A})$  where the transitions are turned from nondeterministic to universal, i.e, we replace  $\vee$  with  $\wedge$ .

Again, in a round of  $G'(\mathcal{A})$  from a position  $(v, p, q)$ : Eve plays a letter  $a$ ; Adam resolves nondeterminism of  $\text{box}(\mathcal{A})$  (i.e., the universality in its dual); then they both resolve the choices in  $\mathcal{A}$ . Let Adam win  $G'$  if either the play of  $\mathcal{A}$  is rejecting or the run of  $\text{box}(\mathcal{A})$  is accepting. Again we can ensure that Adam has a winning strategy in  $G'(\mathcal{A})$ , because both automata are GFG: he uses the GFG strategy of  $\text{box}(\mathcal{A})$  and the  $\forall$ -GFG strategy over  $\mathcal{A}$ . If the word given by Eve belongs to  $L(\mathcal{A})$  then Adam wins by producing an accepting run of  $\text{box}(\mathcal{A})$ , otherwise he wins by refuting an accepting run of  $\mathcal{A}$ . Let  $\tau$  be his positional winning strategy in that game.

We are now ready to build the deterministic automaton from a GFG APW  $\mathcal{A}$ , using positional winning strategies  $\sigma$  and  $\tau$  for Eve and Adam in  $G'(\mathcal{A})$  and  $G(\mathcal{A})$ , respectively.

Let  $\mathcal{D}$  be the automaton with states of the form  $(q, p_1, p_2)$ , with  $q$  a state of  $\mathcal{A}$ ,  $p$  a state of  $\text{box}(\mathcal{A})$  and  $p'$  a state of  $\text{box}(\bar{\mathcal{A}})$ . A transition of  $\mathcal{D}$  over  $a$  moves to  $(q', p'_1, p'_2)$  such that  $((q, p_1), (q', p_1))$  is consistent with  $\tau$  and  $((q, p'_2), (q', p'_2))$  is consistent with  $\sigma$ . In other words, when reading a letter  $a$  in such a state, the following computations are performed:

1. We simulate the choices made by  $\sigma$  in  $G'(\mathcal{A})$  upon obtaining  $a$  from Adam. This way we know how to resolve nondeterminism of  $\text{box}(\bar{\mathcal{A}})$  and what to do with disjunctions inside  $\mathcal{A}$ .
2. We simulate the choices made by  $\tau$  in  $G(\mathcal{A})$  upon obtaining  $a$  from Eve. This way we know how to resolve nondeterminism in  $\text{box}(\mathcal{A})$  and what to do with conjunctions of  $\mathcal{A}$ .
3. In the end we proceed to a new state of  $\mathcal{A}$  and resolved nondeterminism of both  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$ .

The acceptance condition of  $\mathcal{D}$  is inherited from  $\mathcal{A}$ .

► **Lemma 17.** *For a GFG APW  $\mathcal{A}$  and  $\mathcal{D}$  built as above,  $L(\mathcal{A}) = L(\mathcal{D})$ .*

**Proof.** Take a word  $w \in \Sigma^\omega$ . First assume that  $w \in L(\mathcal{A})$ . Eve cannot win a play of the game  $G$  with the letters played in  $R_{A,\Sigma}$  coming from  $w$  using by the first disjunct of her

winning condition, since  $L(\text{box}(\bar{\mathcal{A}})) = L(\bar{\mathcal{A}})$ . Thus, all the plays over  $w$  consistent with her winning strategy  $\sigma$  in  $\mathcal{G}'$  must guarantee that the constructed path of  $\mathcal{A}$  is accepting. Thus, the run of the automaton  $\mathcal{D}$  over  $w$  is accepting.

Now assume that  $w \notin L(\mathcal{A})$ . Dually, no play of the game  $G'$  with the letters coming from  $w$  can produce an accepting run of  $\text{box}(\mathcal{A})$  over  $w$ . Thus, the strategy  $\tau$  guarantees that the sequence of visited states of  $\mathcal{A}$  is rejecting. Thus, the run of  $\mathcal{D}$  over  $w$  must be rejecting. ◀

## D Appendix of Section 5

### D.1 Proof of Theorem 19

Our aim is to provide an EXPTIME algorithm for deciding if a given alternating parity automaton is GFG.

Recall the construction of the two nondeterministic parity automata  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  for  $L(\mathcal{A})$  and  $L(\mathcal{A})^c$  respectively, as defined in Section 4.2. We will use these automata to design a game characterising the fact that  $\mathcal{A}$  is both  $\exists$ -GFG and  $\forall$ -GFG, i.e.  $\mathcal{A}$  is just GFG.

Recall that the automata  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  have exponential number of states in the number of states of  $\mathcal{A}$ . However, due to Lemma 10 their parity index is linear in the number of transitions of  $\mathcal{A}$ . Consider the game  $G'' = (R_{A,\Sigma} \times \text{box}(\mathcal{A})) \times \text{box}(\bar{\mathcal{A}})$ , i.e. the game where Adam plays a letter and Eve replies with two boxes, one of  $\mathcal{A}$  and the other of  $\bar{\mathcal{A}}$ . Let the winning condition of that game for Eve say that either of the runs of  $\text{box}(\mathcal{A})$  or  $\text{box}(\bar{\mathcal{A}})$  must be accepting.

► **Lemma 32.** *Eve has a winning strategy in  $G''$  if and only if  $\mathcal{A}$  is GFG.*

**Proof.** Clearly if  $\mathcal{A}$  is GFG then both  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  are GFG as nondeterministic automata. Therefore, one can use strategies witnessing their GFGness to construct a single strategy for Eve in  $G''$ . This strategy must be winning, because each word proposed by Adam either belongs to  $L(\text{box}(\mathcal{A})) = L(\mathcal{A})$  or to  $L(\text{box}(\bar{\mathcal{A}})) = L(\mathcal{A})^c$ .

Now assume that Eve has a winning strategy in  $G''$ . This strategy consists of two components: one is a strategy in  $R_{A,\Sigma} \times \text{box}(\mathcal{A})$  and the other in  $R_{A,\Sigma} \times \text{box}(\bar{\mathcal{A}})$ . By the fact that the languages of  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  are disjoint, the above components are in fact winning strategies in the letter games for  $\text{box}(\mathcal{A})$  and  $\text{box}(\bar{\mathcal{A}})$  respectively. Thus, by Lemma 20 we know that  $\mathcal{A}$  is both  $\exists$ -GFG and  $\forall$ -GFG. ◀

What remains is to show how to solve the game  $G''$  in EXPTIME. Let  $n$  be the size of the automaton  $\mathcal{A}$ . Our aim is to turn it into a parity game of size exponential in  $n$  but with a number of priorities polynomial in  $n$ . Then, by invoking for instance [6], we know that such a game can be solved in EXPTIME.

► **Lemma 33.** *Let  $\Gamma = \{0, \dots, N\}$  be a set of priorities. Then, there exists a deterministic parity automaton of size exponential in  $N$ , with a number of priorities polynomial in  $N$  that recognises the language  $L$  of words  $w \in (\Gamma \times \Gamma)^\omega$  that satisfy the parity condition on at least one coordinate.*

**Proof.** It is a rather standard construction. One possibility is to design a nondeterministic Büchi automaton for  $L$  with  $N^2$  states. Then, the standard determinisation procedure [21] applied to this automaton gives a deterministic parity automaton as in the statement. ◀

Therefore, we conclude the proof of Theorem 19 by taking a product of the game  $G''$  with the automaton from Lemma 33 and then solving the resulting parity game.