

# On Determinisation of Good-for-Games Automata<sup>\*</sup>

Denis Kuperberg<sup>1,2,3</sup> and Michał Skrzypczak<sup>3,4</sup>

<sup>1</sup> Onera/DTIM, Toulouse

<sup>2</sup> IRIT, University of Toulouse

<sup>3</sup> University of Warsaw

<sup>4</sup> LIAFA, University of Paris 7

**Abstract.** In this work we study Good-For-Games (GFG) automata over  $\omega$ -words: non-deterministic automata where the non-determinism can be resolved by a strategy depending only on the prefix of the  $\omega$ -word read so far. These automata retain some advantages of determinism: they can be composed with games and trees in a sound way, and inclusion  $L(\mathcal{A}) \supseteq L(\mathcal{B})$  can be reduced to a parity game over  $\mathcal{A} \times \mathcal{B}$  if  $\mathcal{A}$  is GFG. Therefore, they could be used to some advantage in verification, for instance as solutions to the synthesis problem.

The main results of this work answer the question whether parity GFG automata actually present an improvement in terms of state-complexity (the number of states) compared to the deterministic ones. We show that a frontier lies between the Büchi condition, where GFG automata can be determinised with only quadratic blow-up in state-complexity; and the co-Büchi condition, where GFG automata can be exponentially smaller than any deterministic automaton for the same language. We also study the complexity of deciding whether a given automaton is GFG.

## 1 Introduction

One of the classical problems of automata theory is synthesis — given a specification, decide if there exists a system that fulfils it and if there is, automatically construct one. The problem was solved positively by Büchi and Landweber [BL69] for the case of  $\omega$ -regular specifications. There are two standard approaches to the problem: either by deterministic automata [McN66] or by tree automata [Rab72]. Henzinger and Piterman [HP06] have proposed a model of Good-For-Games (shortly GFG) automata that enjoy a weak form of non-determinism while still preserving soundness and completeness when solving the synthesis problem.

An automaton is Good-For-Games if there exists a strategy that resolves the non-deterministic choices, by taking into account only the prefix of the input  $\omega$ -word read so far. The strategy is supposed to construct an accepting run of

---

<sup>\*</sup> Research funded by ANR/DGA project Cx (ref. ANR-13-ASTR-0006); and by *foundation STAE* project BRIefcaSE. The second author has been supported by Poland's National Science Centre grant (decision DEC-2014-13/B/ST6/03595).

the automaton whenever an  $\omega$ -word from the language is given. The motivation for this model in [HP06] was to simplify the transition structure of automata as solutions of the synthesis problem for Linear Temporal Logic. Experimental evaluation of GFG automata and their applications to stochastic problems were discussed in [KMBK14].

The notion of GFG automata was independently discovered in [Col09] under the name history-determinism, in the more general framework of regular cost functions. It turns out that deterministic cost automata have strictly smaller expressive power than non-deterministic ones and therefore history-determinism is used whenever a *sequential* model is needed.

In the survey [Col12] two important results about GFG automata over finite words are mentioned: first that every GFG automaton over finite words contains an equivalent deterministic subautomaton, second that it is decidable in PTIME if a given automaton over finite words is GFG. Additionally, a conjecture stating that every parity GFG automaton over  $\omega$ -words contains an equivalent deterministic subautomaton is posed.

In [BKKS13], examples were given of Büchi and co-Büchi GFG automata which do not contain any equivalent deterministic subautomaton. Moreover, a link between GFG and tree automata was established: an automaton for a language  $L$  of  $\omega$ -words is GFG if and only if its infinite tree version accepts the language of trees that have all their branches in  $L$ . However, the problem of the gap in the number of states between deterministic and GFG automata over  $\omega$ -words was left open. Indeed, for all the available examples of GFG automata, there was an equivalent deterministic automaton of the same size.

We settle this question in the present paper. We show that for Büchi automata determinisation can be done with only a quadratic state-space blow-up. The picture is very different for co-Büchi automata (and all higher parity conditions), for which for every  $n$  we give an example of a GFG automaton with  $2n + 1$  states that does not admit any equivalent deterministic automaton with less than  $\frac{2^n}{2n+1}$  states.

The lower bound for determinising co-Büchi GFG automata shows that these automata can be exponentially more succinct than deterministic ones. Therefore, it indicates possibility of avoiding exponential blow-up by using GFG automata instead of deterministic automata in the problems of containment or synthesis. On the other hand, the quadratic determinisation construction for Büchi GFG automata shows that in this case GFG automata are close to deterministic ones. Therefore, the GFG model may be considered less relevant (with respect to succinctness) for Büchi condition than for general parity condition.

We emphasize the fact that although the model of GFG automata requires the existence of a strategy resolving the non-determinism, this strategy is not used in algorithms but only in proofs. Therefore, it is not a part of the size of the input in computations based on GFG automata. This is what allows an improvement on deterministic automata: we just rely on the existence of this strategy without having to explicit it.

In the present paper we additionally consider the problem of deciding whether a given parity automaton is GFG. The problem is decidable in EXPTIME (see [HP06]) but no efficient algorithm is known. In the special case where the automaton accepts all  $\omega$ -words, we show that this is equivalent to solving a parity game, so it is in PTIME for any fixed parity condition, and in  $\text{NP} \cap \text{co-NP}$  if the parity condition is a part of the input. The general case of deciding GFGness of parity automata is a priori more complicated. We show that it is in PTIME for co-Büchi automata, moreover the procedure involves building another automaton that could be GFG even if the input automaton is not. Therefore, this procedure could be used as a tool to produce co-Büchi GFG automata in some cases. The PTIME complexity in this case is surprising — although the required strategy can be of exponential size in the co-Büchi case, we can decide in polynomial time whether it exists. In the Büchi case we show that it is in NP to decide whether a given automaton is GFG. The problem of efficiently deciding GFGness of automata of higher parity indices remains open.

**Structure of the paper.** In Section 2 we briefly introduce the basic notions used in our constructions. In Section 3 we provide the lower bound on the state-complexity of determinising co-Büchi GFG automata. Section 4 is devoted to the determinisation construction for Büchi GFG automata. In Section 5 we study the problem of deciding GFGness of a given automaton and in Section 6 we conclude. The technical details of the presented results are given in Appendix.

## 2 Definitions

By  $A$  we denote a finite *alphabet*, elements  $a \in A$  are called *letters*.  $A^*$  is the set of finite words over  $A$  and  $A^\omega$  is the set of  $\omega$ -words over  $A$ .  $\epsilon$  stands for the empty word. The successive letters of a word  $\alpha$  are  $\alpha(0), \alpha(1), \dots$ . The length of a finite word  $w$  is  $|w|$ . We use the standard notions of prefix and suffix of a word. By  $u\alpha$  we denote the concatenation of a finite word  $u$  with a finite word or an  $\omega$ -word  $\alpha$ . If  $K \subseteq A^\omega$  and  $w \in A^*$  then we define  $w^{-1}K \stackrel{\text{def}}{=} \{\alpha \in A^\omega \mid w\alpha \in K\}$ .

In our constructions it is easier to work with an acceptance condition over transitions instead of states. Clearly, the translation from the state-based acceptance to the transition-based acceptance does not influence the number of states of a parity automaton. The opposite translation may increase the number of states by the factor corresponding to the acceptance condition but this translation is still polynomial (even linear for a fixed condition). Except that, the proposed definitions are standard.

### 2.1 Automata over $\omega$ -words

A non-deterministic parity automaton over  $\omega$ -words (shortly *parity automaton*) is a tuple  $\mathcal{A} = \langle A^\mathcal{A}, Q^\mathcal{A}, q_I^\mathcal{A}, \Delta^\mathcal{A}, \Omega^\mathcal{A} \rangle$  that consists of: a finite set  $A^\mathcal{A}$  called the *input alphabet*; a finite set  $Q^\mathcal{A}$  of *states*; an *initial state*  $q_I^\mathcal{A} \in Q^\mathcal{A}$ ; a *transition*

relation  $\Delta^A \subseteq Q^A \times A^A \times Q^A$ ; and a *priority function*  $\Omega^A: \Delta^A \rightarrow \mathbb{N}$ . If the automaton  $\mathcal{A}$  is known from the context then we skip the superscript  $\mathcal{A}$ .

Transitions  $(q, a, q') \in \Delta$  are usually noted  $q \xrightarrow{a} q'$ . Similarly, if  $w = a_0 a_1 \dots a_n$  and  $q_i \xrightarrow{a_i} q_{i+1}$  is a transition of  $\mathcal{A}$  for all  $i \leq n$  then we write  $q_0 \xrightarrow{w} q_{n+1}$  and call it a *path* in  $\mathcal{A}$ . We additionally require that for every  $q \in Q$ ,  $a \in A$  there is at least one transition in  $\Delta$  of the form  $q \xrightarrow{a} q'$  for some  $q' \in Q$ .

If  $\Omega: \Delta \rightarrow \{i, i+1, \dots, j\}$  then we say that the *parity index* of  $\mathcal{A}$  is  $(i, j)$ . An automaton of parity index  $(1, 2)$  is called a *Büchi* automaton and an automaton of parity index  $(0, 1)$  is called a *co-Büchi* automaton. If  $\mathcal{A}$  is a Büchi automaton then we additionally define  $F \subseteq \Delta$  as  $\Omega^{-1}(2)$  and call it the set of *accepting transitions*. Similarly, if  $\mathcal{A}$  is a co-Büchi automaton then we define  $R \subseteq \Delta$  as  $\Omega^{-1}(1)$  and call it the set of *rejecting transitions*.

If  $\Delta$  is such that for every  $q \in Q$  and  $a \in A$ , there is a unique state  $q' \in Q$  such that  $q \xrightarrow{a} q'$  then  $\mathcal{A}$  is a *deterministic* automaton. In this case, we might denote its transition relation by a function  $\delta: Q \times A \rightarrow Q$  instead of  $\Delta$ .

For an  $\omega$ -word  $\alpha \in A^\omega$ , a *run* of  $\mathcal{A}$  over  $\alpha$  from a state  $q \in Q$  is a function  $\rho: \omega \rightarrow Q$  where for every  $n \geq 0$ , we have a transition of  $\mathcal{A}$   $\rho(n) \xrightarrow{\alpha(n)} \rho(n+1)$  and  $\rho(0) = q$ .  $\rho$  is *accepting* over  $\alpha$  if<sup>5</sup>  $\limsup_{n \rightarrow \infty} \Omega((\rho(n), \alpha(n), \rho(n+1)))$  is even. In other words, the condition requires the highest priority that occurs infinitely often to be even. The priorities can be seen as *positive* (even) and *negative* (odd) events, ordered by their importance. The formula says that the most important event happening infinitely often has to be *positive*.

By the definition, if  $\mathcal{A}$  is Büchi it means that the above sequence of transitions should contain infinitely many *accepting transitions*. Similarly, if  $\mathcal{A}$  is co-Büchi then it should contain only finitely many *rejecting transitions*.

An automaton  $\mathcal{A}$  *accepts* an  $\omega$ -word  $\alpha$  from  $q \in Q$  if there exists an accepting run  $\rho$  of  $\mathcal{A}$  from  $q$  over  $\alpha$ . By  $L(\mathcal{A}, q)$  we denote the set of all  $\omega$ -words that are accepted by  $\mathcal{A}$  from  $q$ . The *language* of an automaton  $\mathcal{A}$  is  $L(\mathcal{A}) \stackrel{\text{def}}{=} L(\mathcal{A}, q_1)$ .

An automaton  $\mathcal{A}$  is *Good-For-Games* (GFG, for short) if there exists a function  $\sigma: A^* \rightarrow Q$  that resolves the non-determinism of  $\mathcal{A}$  depending only of the prefix of the input  $\omega$ -word read so far: over every  $\omega$ -word  $\alpha$ , the function  $n \mapsto \sigma(\alpha(0)\alpha(1)\dots\alpha(n-1))$  is a run of  $\mathcal{A}$  from  $q_1$  over  $\alpha$ , and it is accepting over  $\alpha$  whenever  $\alpha \in L(\mathcal{A})$ . Clearly, every deterministic automaton is GFG.

### 3 Co-Büchi case

In this section we provide the following result about the state-complexity of determinising co-Büchi GFG automata.

**Theorem 1.** *For every  $n$  there exists a co-Büchi GFG automaton  $\mathcal{C}_n$  with  $2n+1$  states such that any equivalent deterministic automaton has at least  $\frac{2^n}{2n+1}$  states.*

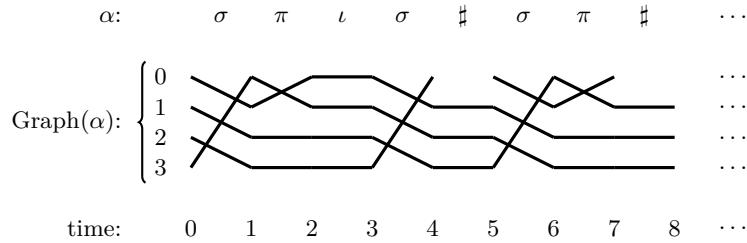
All the automata  $\mathcal{C}_n$  for  $n \geq 1$  share the same alphabet consisting of four symbols  $A \stackrel{\text{def}}{=} \{\iota, \sigma, \pi, \#\}$ . The letters of the alphabet enable to manipulate on

---

<sup>5</sup> Note that whether a run  $\rho$  is accepting over  $\alpha$  depends on the  $\omega$ -word  $\alpha$ .

the set  $\{0, 1, 2, \dots, 2n - 1\}$ :  $\iota, \sigma, \pi$  are three permutations of this set such that every permutation of this set can be obtained as a composition of these three (in fact  $\iota$  is the identity permutation used for padding). The symbol  $\sharp$  corresponds to the identity permutation on  $\{1, \dots, 2n - 1\}$  but it is undefined on 0.

This way a finite word or an  $\omega$ -word  $\alpha$  over the alphabet  $A$  can be seen as a sequence of relations on the set  $\{0, \dots, 2n - 1\}$  as depicted on Figure 1. We will represent these relations as a graph (denoted  $\text{Graph}(\alpha)$ ). If  $\alpha$  is finite let  $D = \{0, 1, \dots, |\alpha|\}$ , otherwise  $D = \omega$ . The graph is a *plait* of width  $2n$ : the domain of  $\text{Graph}(\alpha)$  is  $\{0, 1, \dots, 2n - 1\} \times D$  and all the edges are of the form  $(i, k) \rightarrow (\alpha(k)(i), k + 1)$  for  $i \in \{0, \dots, 2n - 1\}$  and  $k, k + 1 \in D$ .



**Fig. 1.** The infinite sequence of relations on the set  $\{0, \dots, 3\}$  (i.e.  $n = 2$ ) represented by an  $\omega$ -word  $\alpha \in A^\omega$ .

The language  $L_n$  contains an  $\omega$ -word  $\alpha \in A^\omega$  if and only if  $\text{Graph}(\alpha)$  contains at least one infinite path.

The set of states of the automaton  $\mathcal{C}_n$  is  $Q = \{\perp, 0, 1, 2, \dots, 2n - 1\}$ . The states  $\{0, \dots, 2n - 1\}$  are deterministic: reading  $a \in A$  in such a state  $q$  the automaton moves to the successive state according to the relation represented by  $a$  (or to  $\perp$  if  $a = \sharp$  and  $q = 0$ ). The state  $\perp$  is non-deterministic — the automaton can move from  $\perp$  over any letter  $a \in A$  to any state  $q' \in \{0, \dots, 2n - 1\}$ . Let the initial state of  $\mathcal{C}_n$  be  $\perp$  and the rejecting transitions be those of the form  $\perp \xrightarrow{a} q'$ .

Note that every accepting run of  $\mathcal{C}_n$  over an  $\omega$ -word  $\alpha$  indicates an infinite path in  $\text{Graph}(\alpha)$ . Therefore, we obtain the following fact.

**Fact 2.**  $L(\mathcal{C}_n) \subseteq L_n$ .

**Lemma 3.**  $\mathcal{C}_n$  is a GFG automaton recognising the language  $L_n$ .

*Proof.* It is enough to construct a function  $\sigma: A^* \rightarrow Q$  that for every  $\omega$ -word  $\alpha \in L_n$  produces an accepting run of  $\mathcal{C}_n$  over  $\alpha$  — it will prove that  $L_n \subseteq L(\mathcal{C}_n)$  and that  $\mathcal{C}_n$  is GFG. We will do it inductively with  $\sigma(\epsilon) = \perp = q_{\perp}^{\mathcal{C}_n}$ .

Let  $\sigma$  follow deterministically the transitions of  $\mathcal{C}_n$  for all the states  $q \neq \perp$ . It remains to define  $\sigma(wa)$  if  $\sigma(w) = \perp$  and a successive letter  $a$  is given. Assume that  $|wa| = k$ .

For every  $i \in \{0, 1, \dots, 2n - 1\}$  let  $p_i$  be the unique maximal path containing the node  $(i, k)$  in  $\text{Graph}(wa)$ . Note that each of these paths  $p_i$  has a *starting*

*position* — a node  $(\bar{i}, k_i)$  on the path  $p_i$  with a minimal moment of time  $k_i$ . Clearly  $k_i \leq k$ . We say that  $p_i$  is *older* than  $p_{i'}$  if  $k_i < k_{i'}$  — in other words,  $p_i$  reaches further *to the left* than  $p_{i'}$ .

Let  $\sigma(wa) = i$  such that  $p_i$  is the oldest among these paths (if there are two paths equally old, we move to that with smaller  $i$ ).

Assume that  $\alpha \in L_n$ . We need to prove that  $\sigma$  produces an accepting run of  $\mathcal{C}_n$  over  $\alpha$ . Let  $p_1, p_2, \dots, p_m$  be the set of infinite paths in  $\text{Graph}(\alpha)$  (we know that  $1 \leq m \leq 2n$ ). Assume that  $p_1$  is an oldest among them and that it starts in a moment of time  $k_1$ . For every node  $(i, k_1)$  for  $i = 0, \dots, 2n - 1$  that does not belong to any of these infinite paths, the unique maximal path containing  $(i, k_1)$  is finite. Therefore, for some  $k' > k_1$ , one of the paths  $p_1, \dots, p_m$  is the oldest among the paths intersecting the  $(k')$ th moment of time. So the function  $\sigma$  will use at most once a rejecting transition of  $\mathcal{C}_n$  after reading the  $(k')$ th symbol of  $\alpha$  and then it will follow one of the paths  $p_1, \dots, p_m$  and accept.  $\square$

We now assume for the sake of contradiction that there exists a deterministic automaton  $\mathcal{D}$  recognising  $L_n$  that has strictly less than  $\frac{2^n}{2n+1}$  states. By Theorem 4 from [BKKS13] it means that we can use  $\mathcal{D}$  as a memory structure for the automaton  $\mathcal{C}_n$  to recognise  $L_n$ . Therefore, we focus on the product  $\mathcal{C}_n \times \mathcal{D}$  with the acceptance condition taken from  $\mathcal{C}_n$ . What is important is that  $\mathcal{C}_n \times \mathcal{D}$  has to follow the transitions of  $\mathcal{C}_n$ . We know that  $\mathcal{C}_n \times \mathcal{D}$  is a deterministic co-Büchi automaton with strictly less than  $2^n$  states and  $\text{L}(\mathcal{C}_n \times \mathcal{D}) = L_n$ .

We will use the symbol  $\rho$  to denote finite and infinite runs of  $\mathcal{C}_n \times \mathcal{D}$ . For a given run  $\rho$  there are possibly many  $\omega$ -words  $\alpha$  that induce this run, since only the sequence of states is considered in  $\rho$ .

The rest of the argument aims at providing an  $\omega$ -word  $\alpha$  that belongs to  $L_n$  but is rejected by the product automaton  $\mathcal{C}_n \times \mathcal{D}$ . Intuitively, the construction of  $\alpha$  requires to balance between the two aims: we need to infinitely often force the product automaton  $\mathcal{C}_n \times \mathcal{D}$  to take a rejecting transition of  $\mathcal{C}_n$  but at the same time to ensure that there is at least one infinite path in  $\text{Graph}(\alpha)$ . The  $\omega$ -word  $\alpha$ , an infinite path in  $\text{Graph}(\alpha)$ , and the rejecting run of  $\mathcal{C}_n \times \mathcal{D}$  over  $\alpha$  will be constructed as a limit of inductively constructed finite approximations. We will not control exactly the way  $\mathcal{C}_n \times \mathcal{D}$  works in every position of our approximation, we will be interested only in some *checkpoints* controlled by partial runs.

**Definition 4.** A partial run is a finite partial mapping  $\tau: \omega \rightarrow Q^{\mathcal{C}_n} \times Q^{\mathcal{D}}$  such that  $\tau(0)$  is defined and equal to  $(\perp, q_P^{\mathcal{D}})$ .

A partial run  $\tau$  is *rejecting* if all its states are of the form  $(\perp, m)$ .

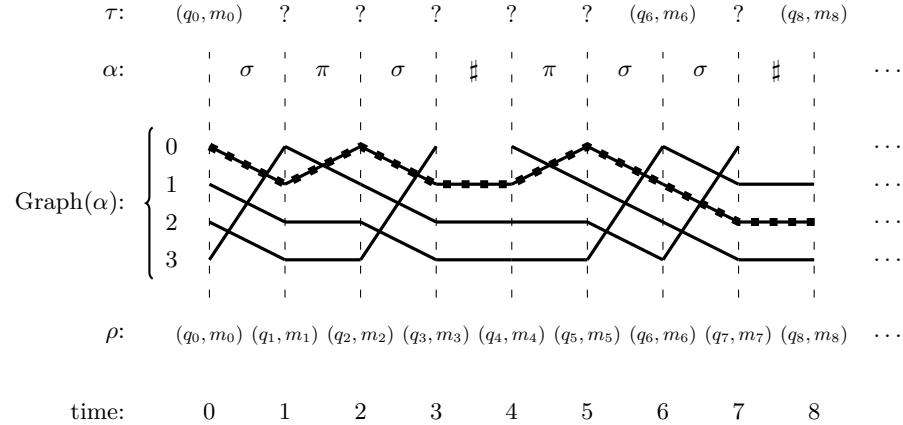
By  $\tau \subseteq \rho$  we denote the fact that a run  $\rho$  agrees with  $\tau$  wherever  $\tau$  is defined. The length of  $\tau$  is the maximal moment of time  $k$  such that  $\tau(k)$  is defined.

Note that the domain of a partial run  $\tau$  does not have to be an initial segment of  $\omega$ . The following definition is crucial.

**Definition 5.** Let  $\tau$  be a partial run of length  $k$ . We say that a value  $i \in \{0, \dots, 2n - 1\}$  is *alive* in  $\tau$  if there exists an  $\omega$ -word  $\alpha$  such that for the run  $\rho$  of  $\mathcal{C}_n \times \mathcal{D}$  over  $\alpha$  we have  $\tau \subseteq \rho$  and there exists a path  $p: \{0, 1, \dots, k\} \rightarrow$

$\{0, 1, \dots, 2n - 1\}$  in  $\text{Graph}(\alpha)$  that starts in the moment of time 0 and ends in the moment of time  $k$  with the value  $i$  (i.e.  $p(k) = i$ ).

Note that in the above definition we actually care only about the first  $k$  letters of  $\alpha$ . However, it is cleaner to consider  $\omega$ -words  $\alpha$  here.



**Fig. 2.** An example of a partial run  $\tau$  and an  $\omega$ -word  $\alpha$  that witnesses the fact that 2 is alive in  $\tau$ .  $\rho$  is the run of  $\mathcal{C}_n \times \mathcal{D}$  over  $\alpha$  and the states of  $\rho$  and  $\tau$  agree wherever defined. The dashed path is the path witnessing 2 is alive in  $\tau$ .

Figure 2 depicts a partial run and a witness that the value  $i = 2$  is alive.

Our aim is to construct a sequence of partial rejecting runs of increasing lengths  $\tau_0 \subset \tau_1 \subset \dots$  such that for all  $\ell \in \mathbb{N}$  there are at least  $n$  alive values in  $\tau_\ell$ . It will give a contradiction with our assumptions by the following lemma.

**Lemma 6.** *Assume that there exists a sequence of partial rejecting runs  $\tau_0 \subset \tau_1 \subset \dots$  of increasing lengths such that for all  $\ell \in \mathbb{N}$  there exists an alive value in  $\tau_\ell$ . Then there exists an  $\omega$ -word  $\alpha \in L_n$  such that the run  $\rho$  of  $\mathcal{C}_n \times \mathcal{D}$  over  $\alpha$  is rejecting.*

*Proof.* Let  $k_\ell$  be the length of  $\tau_\ell$ . Take any  $\ell$  and assume that  $i_\ell$  is a value that is alive in  $\tau_\ell$ . Observe that it is witnessed by:

- an  $\omega$ -word  $\alpha_\ell$ ,
- a run  $\rho_\ell$  of  $\mathcal{C}_n \times \mathcal{D}$  over  $\alpha_\ell$ , such that  $\tau_\ell \subset \rho_\ell$ ,
- a path  $p_\ell: \{0, \dots, k_\ell\} \rightarrow \{0, \dots, 2n - 1\}$  in  $\text{Graph}(\alpha_\ell)$  with  $p_\ell(k_\ell) = i_\ell$ .

Now we take a subsequence of  $(\alpha_\ell, \rho_\ell, p_\ell)_{\ell \in \mathbb{N}}$  that is point-wise convergent to a triple

$$(\alpha, \rho, p) \in \left( A \times (Q^{\mathcal{C}_n} \times Q^{\mathcal{D}}) \times \{0, \dots, 2n - 1\} \right)^\omega,$$

such that:

- $\rho$  is the run of  $\mathcal{C}_n \times \mathcal{D}$  over  $\alpha$ ,
- for infinitely many  $\ell$  we have  $\tau_\ell \subseteq \rho$ ,
- $p$  encodes an infinite path in  $\text{Graph}(\alpha)$ .

To formally construct  $(\alpha, \rho, p)$  we can proceed similarly as in the proof of König's lemma. We fix  $(\alpha(i), \rho(i), p(i))$  inductively for  $i = 0, 1, \dots$ . At each moment we require that infinitely many  $(\alpha_\ell, \rho_\ell, p_\ell)$  agree with  $(\alpha, \rho, p)$  on the first  $i$  positions. Since for each  $i$  there are only finitely many choices of  $(\alpha(i), \rho(i), p(i))$  so we can fix these values in such a way that still infinitely many  $(\alpha_\ell, \rho_\ell, p_\ell)$  agree with them.

By the properties of  $(\alpha, \rho, p)$  we know that  $\rho$  is rejecting as it contains infinitely many times a state of the form  $(\perp, m)$ . On the other hand,  $\alpha \in L_n$  because  $p$  is a witness that  $\text{Graph}(\alpha)$  contains an infinite path.  $\square$

What remains is to construct the sequence  $\tau_\ell$  inductively. Our inductive assumption is that  $\tau_\ell$  is a partial rejecting run and the values  $1, 3, 5, \dots, 2n-1$  are alive in  $\tau_\ell$  (note that there are  $n$  such values). We put  $\tau_0 = [0 \mapsto (\perp, q_1^D)]$ . Clearly  $\tau_0$  satisfies the inductive assumption (in fact all the values  $i = 0, \dots, 2n-1$  are alive in  $\tau_0$ ).

Let  $k_\ell$  be the length of  $\tau_\ell$ . We construct  $\tau_{\ell+1}$  from  $\tau_\ell$  by applying some words to the last state  $(\perp, m_\ell) = \tau_\ell(k_\ell)$  of  $\tau_\ell$  and observing the behaviour of  $\mathcal{C}_n \times \mathcal{D}$ .

Observe that there are  $N = 2^n$  words  $u_1, \dots, u_N \in \{\iota, \sigma, \pi\}^*$  that encode distinct permutations  $P$  of  $\{0, \dots, 2n-1\}$  such that for all  $i \in \{0, \dots, 2n-1\}$ , we have  $[i/2] = [P(i)/2]$  i.e. such a permutation maps  $\{2i, 2i+1\}$  to itself.

We can assume that all the words  $u_1, \dots, u_N$  are of equal length by padding them with  $\iota$ . Since there are strictly less than  $N = 2^n$  states of  $\mathcal{C}_n \times \mathcal{D}$ , there are two distinct such words  $u, u'$  leading from  $(\perp, m_\ell)$  to the same state  $(q'_\ell, m'_\ell)$  of  $\mathcal{C}_n \times \mathcal{D}$ . By the construction of  $\mathcal{C}_n \times \mathcal{D}$  we know that  $q'_\ell \in \{0, \dots, 2n-1\}$ .

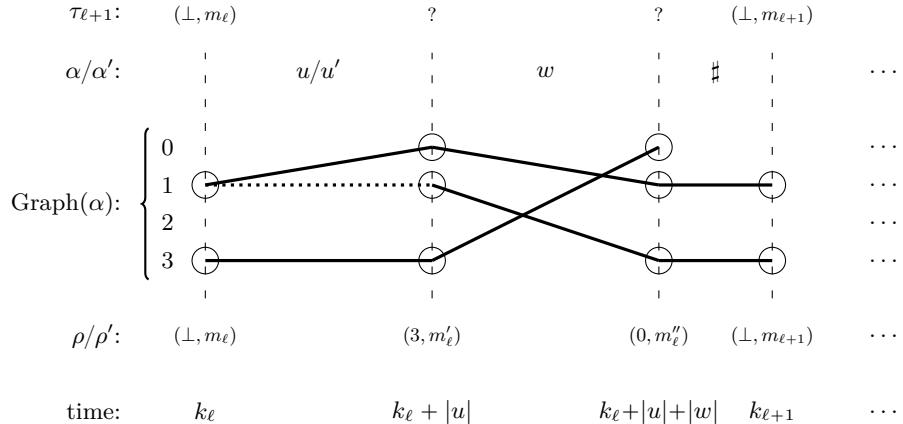
Assume that the permutations corresponding to  $u$  and  $u'$  differ on  $2i+1$ , i.e. one of them maps  $2i+1$  to  $2i$  and the other to  $2i+1$ . Let  $X$  be the set of the values  $\{u(1), u(3), \dots, u(2n-3), u(2n-1), u'(2i+1)\}$  (we write here  $u(i')$  for the value assigned to  $i'$  by the permutation corresponding to  $u$ , the same for  $u'$ ). By the above observations  $X$  contains exactly  $n+1$  elements.

Consider  $w \in \{\iota, \sigma, \pi\}^*$  encoding a permutation that maps:

- $q'_\ell$  to 0,
- $X \setminus \{q'_\ell\}$  to  $1, 3, 5, \dots, 2n-1$  if  $q'_\ell \in X$ ,
- $X$  to  $1, 3, \dots, 2n-1$ , and 2 if  $q'_\ell \notin X$ .

Since  $w$  as a permutation maps  $q'_\ell$  to 0, we know that after reading  $w\sharp$  from the state  $(q'_\ell, m'_\ell)$  the automaton  $\mathcal{C}_n \times \mathcal{D}$  reaches a state of the form  $(\perp, m_{\ell+1})$ . For an illustration of these permutations, see Figure 3.

**Fact 7.** Consider  $\tau_{\ell+1}$  defined as  $\tau_\ell \cup [k_\ell + |u| + |w| + 1 \mapsto (\perp, m_{\ell+1})]$ . By the definition  $\tau_\ell \subset \tau_{\ell+1}$ ,  $\tau_{\ell+1}$  is rejecting, and all the values  $1, 3, \dots, 2n-1$  are alive in  $\tau_{\ell+1}$  (it is witnessed by the fact that these values were alive in  $\tau_\ell$  and by the words  $uw\sharp$  and  $u'w\sharp$ ).



**Fig. 3.** The behaviour of  $\mathcal{C}_n \times \mathcal{D}$  over  $uw\#$  and  $u'w\#$ . The alive values are in circles, only edges between the alive values are drawn. The dashed edge corresponds to the action of the word  $u'$  on the value 1 ( $u$  and  $u'$  differ on this value).  $X$  is the set of values in circles at the moment of time  $k_\ell + |u|$ .  $q'_\ell = 3$  is mapped to 0 by the permutation corresponding to  $w$ , the other elements of  $X$  are mapped to 1 and 3.

Therefore, we have constructed  $\tau_{\ell+1}$  that satisfies the inductive invariant. This concludes the inductive construction of the sequence  $(\tau_\ell)_{\ell \in \mathbb{N}}$ . By Lemma 6 it finishes the proof of Theorem 1.

#### 4 Büchi case

In this section we discuss the quadratic upper bound for the state-complexity of determinising Büchi GFG automata, as expressed by the following theorem.

**Theorem 8.** *For every Büchi GFG automaton there exists an equivalent deterministic Büchi automaton with quadratic number of states.*

Here we provide some high level overview of the construction. A detailed description of it can be found in Appendix B.

The main part of the construction is an inductive normalisation of a given Büchi automaton  $\mathcal{A}$ . The normalisation is guided by the powerset automaton  $\mathcal{D}$  having sets of states of  $\mathcal{A}$  as its states. It turns out that if  $\mathcal{A}$  is GFG then  $L(\mathcal{A}) = L(\mathcal{D})$ . During the normalisation we remove some irrelevant transitions of  $\mathcal{A}$  and mark some existing transitions as accepting (while ensuring that we preserve the language  $L(\mathcal{A})$  and the fact that  $\mathcal{A}$  is GFG).

When reaching a fixed-point of the normalisation, we know that  $\mathcal{A}$  is in certain formal sense *optimal*. This optimal  $\mathcal{A}$  needs not be deterministic. However, we can prove that there is a function  $\sigma$  witnessing that  $\mathcal{A}$  is GFG that uses  $\mathcal{A}$  as a memory structure. Therefore, by combining  $\mathcal{A}$  with  $\sigma$ , we can define a structure of a deterministic Büchi automaton for  $L(\mathcal{A})$  over  $\mathcal{A} \times \mathcal{A}$ .

## 5 Recognising GFG automata

We now investigate the algorithmic complexity of recognising whether a given automaton is GFG. We provide three results about general GFG-automata, Büchi GFG automata, and co-Büchi GFG-automata. Let us recall that in general, the problem of deciding if a given parity automaton is GFG was shown in [HP06] to belong to EXPTIME.

**Equivalence with parity games.** The following theorem shows that in general, the problem of GFGness of a given parity automaton is at least as hard as solving parity games. The later is known to be  $NP \cap co\text{-}NP$  but there is no PTIME algorithm known.

**Theorem 9.** *Finding the winner of a parity game of index  $(i, j)$  is polynomially equivalent to deciding whether a given parity automaton of index  $(i, j)$  that accepts all  $\omega$ -words is GFG.*

Indeed, we show that given a parity game  $\mathcal{G}$  between the players  $\exists$  and  $\forall$ , it is possible to build an automaton  $\mathcal{A}$  accepting all  $\omega$ -words, with the same parity index as  $\mathcal{G}$ , such that  $\mathcal{A}$  is GFG if and only if  $\exists$  wins  $\mathcal{G}$ . In the initial state of  $\mathcal{A}$  the automaton is supposed to non-deterministically guess the next letter. If the guess is correct, we move to an accepting sink state, otherwise we move to a subautomaton mimicking the game  $\mathcal{G}$ , where moves of  $\forall$  are represented by letters and moves of  $\exists$  are represented by a choice of transition. This way  $\mathcal{A}$  accepts all  $\omega$ -words but no GFG strategy can guarantee to reach the accepting sink state. Therefore,  $\mathcal{A}$  is GFG only if  $\exists$  has a strategy to win the original game  $\mathcal{G}$ . A polynomial reduction from the problem of GFGness of an automaton accepting all  $\omega$ -words to a parity game of the same index is an easy consequence of [HP06].

A detailed construction of  $\mathcal{A}$  and the proof of equivalence are presented in Appendix C.

**Recognising Büchi GFG automata.** The upper bounds given in Section 4 allow us to state the following theorem, see Appendix D for a detailed proof .

**Theorem 10.** *It is in  $NP$  to decide whether a given non-deterministic Büchi automaton  $\mathcal{A}$  is GFG. Moreover, if  $\mathcal{A}$  is GFG then we can construct an equivalent deterministic Büchi automaton in  $NP$ .*

**Recognising co-Büchi GFG automata.**

**Theorem 11.** *Given a non-deterministic co-Büchi automaton, we can decide whether it is GFG in polynomial time.*

We will only sketch the proof here, the detailed proof is presented in Appendix E.

The cornerstone of the construction is a game called the *Joker Game*, defined relatively to a co-Büchi automaton  $\mathcal{A}$ . This is a perfect information two players

game played between  $\exists$  and  $\forall$ . The set of positions is  $Q^{\mathcal{A}} \times Q^{\mathcal{A}}$ , the initial position is  $(q_{\exists}^{\mathcal{A}}, q_{\forall}^{\mathcal{A}})$ , and at a round  $n$  starting in  $(p_n, q_n)$  the following choices are made by the players:

- $\forall$  chooses a letter  $a_n \in A$ ,
- $\exists$  chooses a transition  $p_n \xrightarrow{a_n} p_{n+1}$  of  $\mathcal{A}$ ,
- $\forall$  chooses a transition  $q_n \xrightarrow{a_n} q_{n+1}$  of  $\mathcal{A}$  or plays JOKER and chooses a transition  $p_n \xrightarrow{a_n} q_{n+1}$  of  $\mathcal{A}$ .

After that the game moves to the position  $(p_{n+1}, q_{n+1})$ . Player  $\exists$  wins an infinite play if either:

- the run  $(p_n)_n$  of  $\mathcal{A}$  is accepting over  $(a_n)_n$ ,
- $\forall$  played infinitely many times JOKER,
- or the run  $(q_n)_n$  of  $\mathcal{A}$  is not accepting over  $(a_n)_n$ <sup>6</sup>.

Intuitively, the Joker Game forces  $\exists$  to produce an accepting run of  $\mathcal{A}$  over  $(a_n)_n$  sequentially, whenever possible. However, since we cannot put the fact that  $(a_n)_n \in L(\mathcal{A})$  into the acceptance condition (it would hide an exponential blow-up in the acceptance condition). Therefore, we ask  $\forall$  to concurrently produce a run of  $\mathcal{A}$  over  $(a_n)_n$ . If  $\forall$  manages to produce an accepting run while  $\exists$  fails to do so, it shows that  $\mathcal{A}$  is not GFG. The other implication is problematic: the automaton  $\mathcal{A}$  may not be GFG but  $\exists$  may win the Joker Game by relying on the choices made by  $\forall$ .

We start by computing in polynomial time the winner of the Joker Game (a parity game of index  $(0, 2)$ ) on  $\mathcal{A}$ . We show that if  $\forall$  wins the Joker Game then  $\mathcal{A}$  is not GFG. In the opposite case we are able to build a GFG automaton  $\mathcal{B}$  of the same number of states as  $\mathcal{A}$  that recognises the same language. Then, using again an appropriate game over  $\mathcal{A} \times \mathcal{B}$  we can decide GFGness of  $\mathcal{A}$  in polynomial time.

To build the automaton  $\mathcal{B}$ , we first compute a binary relation  $\rightarrow$  on the states of  $\mathcal{A}$ . This relation is the winning region of yet another game, the *safety game*, which is the Joker Game where seeing a rejecting transition means immediate loss. By referring to the Joker Game we prove that for all  $q$  there is  $p$  such that  $p \rightarrow p$  and  $p \rightarrow q$ .

This means that we can construct a deterministic safety automaton  $\mathcal{D}$  with states  $p$  such that  $p \rightarrow p$ . Every  $\omega$ -word that is accepted by  $\mathcal{A}$  has a suffix accepted by  $\mathcal{D}$  from some state  $p$ . It remains to add non-deterministic rejecting transitions to  $\mathcal{D}$  in order to allow it to guess such a state  $p$ . For this, we compute an equivalence relation  $E$  on the states of  $\mathcal{A}$  reflecting simultaneous reachability. We then use this relation to build  $\mathcal{B}$  by connecting  $E$ -equivalent states of  $\mathcal{D}$  using rejecting transitions. We finally show that the automaton  $\mathcal{B}$  is GFG and recognises  $L(\mathcal{A})$ . The strategy witnessing GFGness of  $\mathcal{B}$  uses the same intuition as the one in Lemma 3

---

<sup>6</sup> Formally, only the suffix of  $(q_n)_n$  after the last JOKER played by  $\forall$  is a run of  $\mathcal{A}$  over the suffix of  $(a_n)_n$ .

## 6 Conclusion

The main result of this paper is a solution of the open problem asking what is the state-complexity of determinising parity GFG automata over  $\omega$ -words. We prove that for co-Büchi GFG automata (and therefore all higher parity indices) the exponential blow-up cannot be avoided. For the remaining case of Büchi GFG automata we provide a construction of an equivalent deterministic automaton with quadratic number of states.

Using the tools developed to prove the above results, we are additionally able to study the complexity of the decision problem of verifying if a given parity automaton is GFG. We prove that for general parity automata the problem is at least as hard as solving parity games (for which no PTIME algorithm is known). Then we focus on the two subcases of Büchi and co-Büchi automata. In the case of Büchi automata we provide a very simple NP algorithm based on our determinisation construction. In the case of co-Büchi automata we have a bit more involved PTIME decision procedure. One of the advantages of the procedure is that, even if the automaton itself is not GFG, there could be cases when the procedure builds an equivalent GFG automaton with the same number of states. The possibilities of exploiting this fact are still to be studied.

Hopefully, the results presented in this paper will shed some light on possible efficient applications of GFG automata in the classical problems of verification.

For future research, in the Büchi case, both the exact time-complexity (between PTIME and NP) and state-complexity (between linear and quadratic) of the determinisation algorithm are still to be clarified.

The complexity of deciding GFGness for general parity automata is still open, with a lower bound of solving parity games and an EXPTIME upper bound.

# Appendix

## A Game approach

Games on finite arenas will play a central role in our study of the parity GFG automata. We will be mainly interested in two types of games, as defined in the following subsections.

### A.1 Game on a single automaton

In this section we recall the construction of a game  $\mathcal{G}(\mathcal{A})$  from [HP06,BKKS13] for resolving GFGness of parity automata.

Let  $\mathcal{A} = \langle A, Q, q_I, \Delta, \Omega \rangle$  be a parity automaton over an alphabet  $A$ . The set of positions of the game  $\mathcal{G}(\mathcal{A})$  is  $Q$  and the initial position  $v_I$  is the initial state  $q_I$ . In the  $n$ th round for  $n = 0, 1, \dots$  starting in a position  $q_n$  first  $\forall$  plays a letter  $a_n \in A$  and then  $\exists$  plays a state  $q_{n+1}$  such that  $q_n \xrightarrow{a_n} q_{n+1}$  is a transition of  $\mathcal{A}$ . Consider an infinite play in which  $\forall$  played an  $\omega$ -word  $\alpha$  and  $\exists$  played a run  $\rho$  of  $\mathcal{A}$  over  $\alpha$ . Such a play is won by  $\exists$  if either the run  $\rho$  is accepting or the  $\omega$ -word  $\alpha$  does not belong to  $L(\mathcal{A})$ .

Observe that the arena of the game  $\mathcal{G}(\mathcal{A})$  is finite and the winning condition is  $\omega$ -regular. Therefore, one of the players has a finite-memory winning strategy. It is shown in [BKKS13] that  $\exists$  wins  $\mathcal{G}(\mathcal{A})$  if and only if  $\mathcal{A}$  is GFG<sup>7</sup>.

We will also need a reformulation of Theorem 4 from [BKKS13].

**Proposition 12.** *Assume that  $\mathcal{A}$  is a parity GFG automaton.*

1. *If  $\exists$  has a finite-memory winning strategy in  $\mathcal{G}(\mathcal{A})$  with  $M$  memory states then there exists a deterministic parity automaton recognising  $L(\mathcal{A})$  with  $|Q^{\mathcal{A}}| \cdot M$  states.*
2. *If  $\mathcal{D}$  is a deterministic parity automaton recognising  $L(\mathcal{A})$  then  $\exists$  has a finite memory winning strategy in  $\mathcal{G}(\mathcal{A})$  with  $|Q^{\mathcal{D}}|$  memory states.*

### A.2 Game on two automata

Let  $\mathcal{A}, \mathcal{B}$  be two non-deterministic parity automata, both over an alphabet  $A$ .

We define the game  $\mathcal{G}(\mathcal{A}, \mathcal{B})$  similarly to  $\mathcal{G}(\mathcal{A})$  from the previous section, except that we require Player  $\forall$  to explicitly build a run of the other automaton  $\mathcal{B}$ . In more detail, the set of positions is  $Q^{\mathcal{A}} \times Q^{\mathcal{B}}$ , the initial position is  $(q_I^{\mathcal{A}}, q_I^{\mathcal{B}})$ , and at a round  $n$  starting in  $(p_n, q_n)$ :

- $\forall$  chooses a letter  $a_n \in A$ ,
- $\exists$  chooses a transition  $p_n \xrightarrow{a_n} p_{n+1}$  of  $\mathcal{A}$ ,

---

<sup>7</sup> Originally, in [HP06] the definition of GFG automata was given by the game  $\mathcal{G}(\mathcal{A})$ .

- $\forall$  chooses a transition  $q_n \xrightarrow{a_n} q_{n+1}$  of  $\mathcal{B}$ .

After that the game moves to the position  $(p_{n+1}, q_{n+1})$ . Player  $\exists$  wins an infinite play if the run  $(p_n)_n$  of  $\mathcal{A}$  is accepting over  $(a_n)_n$  or the run  $(q_n)_n$  of  $\mathcal{B}$  is not accepting over  $(a_n)_n$ .

The following remark follows from the existing algorithms for solving Rabin games in polynomial time for every fixed number of Rabin pairs [PP06].

*Remark 1.* For each fixed index  $(i, j)$  the winner of  $\mathcal{G}(\mathcal{A}, \mathcal{B})$  can be decided in polynomial time in the number of states of automata  $\mathcal{A}, \mathcal{B}$  of index  $(i, j)$ .

**Theorem 13.** *Let  $\mathcal{A}$  be a GFG automaton and  $\mathcal{B}$  be a non-deterministic automaton. Then  $\exists$  wins  $\mathcal{G}(\mathcal{A}, \mathcal{B})$  if and only if  $L(\mathcal{A}) \supseteq L(\mathcal{B})$ .*

*Proof.* If  $L(\mathcal{B}) \not\subseteq L(\mathcal{A})$  then playing  $\alpha \in L(\mathcal{B}) \setminus L(\mathcal{A})$  together with an accepting run of  $\mathcal{B}$  over  $\alpha$  is a winning strategy for  $\forall$  in  $\mathcal{G}(\mathcal{A}, \mathcal{B})$ .

On the other hand, if  $L(\mathcal{B}) \subseteq L(\mathcal{A})$ , playing the GFG strategy of  $\mathcal{A}$  is winning for  $\exists$ , since every  $\omega$ -word  $\alpha \in L(\mathcal{B})$  will yield an accepting run of  $\mathcal{A}$  over  $\alpha$  via this strategy.  $\square$

The following result from [HP06] provides the EXPTIME upper bound for the general GFGness problem.

**Theorem 14 ([HP06]).** *Let  $\mathcal{A}, \mathcal{B}$  be two non-deterministic parity automata such that  $L(\mathcal{A}) = L(\mathcal{B})$  and  $\mathcal{B}$  is GFG. Then  $\exists$  wins  $\mathcal{G}(\mathcal{A}, \mathcal{B})$  if and only if  $\mathcal{A}$  is GFG.*

*Proof.* By Theorem 13, if  $\mathcal{A}$  is GFG and  $L(\mathcal{B}) \subseteq L(\mathcal{A})$ , Player  $\exists$  wins  $\mathcal{G}(\mathcal{A}, \mathcal{B})$ .

Now, assume  $\exists$  wins  $\mathcal{G}(\mathcal{A}, \mathcal{B})$  using a strategy  $\sigma$ . We want to provide a GFG strategy

$$\sigma^{\text{GFG}}: A^* \rightarrow Q^{\mathcal{A}}.$$

The principle is simply to use the GFG strategy  $\sigma_{\mathcal{B}}$  of  $\mathcal{B}$  to play moves of  $\forall$ , and answer them with a winning strategy  $\sigma$  to find a valid run of  $\mathcal{A}$ . This way, if the input  $\omega$ -word is in  $L(\mathcal{A}) = L(\mathcal{B})$ , the correctness of  $\sigma_{\mathcal{B}}$  ensures that the run of  $\mathcal{B}$  is accepting, and in turn the correctness of  $\sigma$  ensures that the run of  $\mathcal{A}$  is accepting.

Let us describe the function  $\sigma^{\text{GFG}}$  more formally. Assume that  $u \in A^*$  and the sequence of states of  $\mathcal{B}$  produced by  $\sigma_{\mathcal{B}}$  over  $u$  is  $\rho'$ . Consider the play of  $\mathcal{G}(\mathcal{A}, \mathcal{B})$  in which  $\forall$  plays successive letters of  $u$  and successive states of  $\rho'$ . Let  $\sigma^{\text{GFG}}(u)$  be the state played by  $\exists$  according to her winning strategy after  $\forall$  played  $u$  and  $\rho'$ .

We need to prove that if  $\alpha \in L(\mathcal{A}) = L(\mathcal{B})$  then the sequence of states  $\rho$  defined by  $\sigma^{\text{GFG}}$  is an accepting run of  $\mathcal{A}$  over  $\alpha$ . Let  $\rho'$  be the sequence of states of  $\mathcal{B}$  given by  $\sigma_{\mathcal{B}}$  over  $\alpha$ . Since  $\alpha \in L(\mathcal{B})$  we know that  $\rho'$  is an accepting run of  $\mathcal{B}$  over  $\alpha$ . Since  $\rho$  is the run constructed by the winning strategy  $\sigma$  against  $\forall$  playing  $\alpha$  and  $\rho'$ , we know that  $\rho$  has to be an accepting run of  $\mathcal{A}$  over  $\alpha$ .  $\square$

## B Büchi case

In this section, we prove the following theorem:

**Theorem 8.** *For every Büchi GFG automaton there exists an equivalent deterministic Büchi automaton with quadratic number of states.*

Let  $\mathcal{A} = \langle A, Q, q_I, \Delta, F \rangle$  with  $F \subseteq \Delta$  be a Büchi GFG automaton recognising a language  $L \subseteq A^\omega$ .

The crucial phase of our construction will consist of inductively modifying  $\mathcal{A}$  while preserving  $L$ . The modifications will not influence the alphabet  $A$  nor the initial state  $q_I$ . Let us put  $Q_0 = Q$ ,  $\Delta_0 = \Delta$ ,  $F_0 = F$ , and  $\mathcal{A}_0 = \langle A, Q_0, q_I, \Delta_0, F_0 \rangle$ . During the construction we will keep the following *invariants*:

$$\begin{aligned} q_I &\in Q_{i+1} \subseteq Q_i \subseteq Q, \\ \Delta_{i+1} &\subseteq \Delta_i \subseteq \Delta, \\ F_i &\subseteq \Delta_i, \\ \Delta_{i+1} \setminus F_{i+1} &\subseteq \Delta_i \setminus F_i, \\ L(\mathcal{A}_{i+1}) &= L(\mathcal{A}_i) = L, \end{aligned}$$

and additionally we will ensure that the automaton  $\mathcal{A}_i$  is GFG.

### B.1 Residual languages

We will start by observing that the fact that a given automaton  $\mathcal{A}_i$  is GFG implies that the *residual languages*  $L(\mathcal{A}_i, q)$  of  $\mathcal{A}_i$  are in a sense simple.

Let us fix a function  $\sigma_i^{\text{GFG}} : A^* \rightarrow Q_i$  witnessing that  $\mathcal{A}_i$  is GFG.

**Definition 15.** *We say that  $\mathcal{A}_i$  is minimal with respect to  $\sigma_i^{\text{GFG}}$  if:*

- for every state  $q \in Q_i$  there is a word  $w$  such that  $\sigma_i^{\text{GFG}}(w) = q$ ,
- for every transition  $q \xrightarrow{a} q' \in \Delta_i$  there is a word  $w$  such that  $\sigma_i^{\text{GFG}}(w) = q$  and  $\sigma_i^{\text{GFG}}(wa) = q'$ , i.e. the transition is used somewhere in  $\sigma_i^{\text{GFG}}$ .

**Lemma 16.** *We can assure that the automaton  $\mathcal{A}_i$  is minimal with respect to  $\sigma_i^{\text{GFG}}$  while preserving the invariants.*

*Proof.* Let us define  $\mathcal{A}'_i$  as the copy of  $\mathcal{A}_i$  with all the states and transitions of  $\mathcal{A}_i$  not accessible via  $\sigma_i^{\text{GFG}}$  removed. Clearly after these modifications we still have  $L(\mathcal{A}'_i) = L$ : the  $\subseteq$  inclusion follows from monotonicity of the modifications and the  $\supseteq$  is witnessed by the strategy  $\sigma_i^{\text{GFG}}$ . Since  $\sigma_i^{\text{GFG}}$  is a complete function, the remaining  $\mathcal{A}'_i$  satisfies the condition that from every state there is at least one transition over every letter. Also the monotonicity constraints are satisfied. From that point on we assume that  $\mathcal{A}_i = \mathcal{A}'_i$ .  $\square$

We will now study residuals of  $\mathcal{A}_i$  — the languages  $L(\mathcal{A}_i, q)$  for states  $q$  of  $\mathcal{A}_i$ . The following lemma shows that the residuals of  $\mathcal{A}_i$  cannot *split* — for every pair of transitions  $q \xrightarrow{a} q'$  and  $q \xrightarrow{a} q''$  of  $\mathcal{A}_i$  the residuals  $L(\mathcal{A}_i, q')$  and  $L(\mathcal{A}_i, q'')$  are the same.

**Lemma 17.** *If  $\sigma_i^{\text{GFG}}(w) = q$  and  $\sigma_i^{\text{GFG}}(wa) = q'$  then*

$$L(\mathcal{A}_i, q') = a^{-1}L(\mathcal{A}_i, q).$$

*Proof.* Clearly the  $\subseteq$  containment holds — if  $\mathcal{A}_i$  has an accepting run over  $\alpha$  from  $q'$  then it has an accepting run over  $a\alpha$  from  $q$ .

For the  $\supseteq$  containment take  $\alpha \in A^\omega$  such that  $a\alpha$  belongs to  $L(\mathcal{A}_i, q)$ . Since  $\sigma_i^{\text{GFG}}(w) = q$  we know that  $q_1 \xrightarrow{w} q$ . Therefore, there exists an accepting run of  $\mathcal{A}_i$  over  $wa\alpha$  from  $q_1^A$  (i.e.  $wa\alpha \in L(\mathcal{A}_i)$ ). It means that the run constructed by  $\sigma_i^{\text{GFG}}$  over  $wa\alpha$  is also accepting. But  $\sigma_i^{\text{GFG}}(wa) = q'$ , therefore this run witnesses that there is an accepting run of  $\mathcal{A}_i$  over  $\alpha$  from  $q'$ . Therefore,  $\alpha \in L(\mathcal{A}_i, q')$ .  $\square$

**Corollary 18.** *If  $q \xrightarrow{a} q'$  is a transition of  $\mathcal{A}_i$  then  $L(\mathcal{A}_i, q') = a^{-1}L(\mathcal{A}_i, q)$ .*

*If  $q \xrightarrow{u} q'$  is a path in  $\mathcal{A}_i$  then  $L(\mathcal{A}_i, q') = u^{-1}L(\mathcal{A}_i, q)$ .*

*If  $q \xrightarrow{u} q'$  and  $q \xrightarrow{u} q''$  are paths in  $\mathcal{A}_i$  then  $L(\mathcal{A}_i, q') = L(\mathcal{A}_i, q'')$ .*

*Proof.* The first claim follows from Lemma 17 and minimality of  $\mathcal{A}_i$  with respect to  $\sigma_i^{\text{GFG}}$ .

The second claim follows from the first one by induction over  $u$ .

The third claim follows directly from the second one.  $\square$

Intuitively, this corollary guarantees that we can always make finitely many *bad* non-deterministic choices over an  $\omega$ -word  $\alpha \in L(\mathcal{A}_i)$  and still accept it.

## B.2 A simple deterministic automaton $\mathcal{D}_i$ for $L$

Now we will construct an exponential but simple deterministic Büchi automaton for  $L$ . Let us fix an arbitrary total order  $<_Q$  on the set of states  $Q$  of the original automaton  $\mathcal{A}$ .

Let us define

$$M_i = \{m \subseteq Q_i \mid m \neq \emptyset \wedge \forall q, q' \in m. L(\mathcal{A}_i, q) = L(\mathcal{A}_i, q')\}. \quad (1)$$

The set of states of  $\mathcal{D}_i$  is  $M_i$  and the initial state is  $\{q_1\}$ . Let the transition function of  $\mathcal{D}_i$  for  $m \in M_i$ ,  $a \in A$  be defined as:

$$\delta^{\mathcal{D}_i}(m, a) = \begin{cases} \{ \min_{<_Q} \{q \in Q_i \mid \exists p \in m. (p, a, q) \in F_i\} \} & \text{if such } q \text{ exists} \\ \{q \in Q_i \mid \exists p \in m. (p, a, q) \in \Delta_i\} & \text{otherwise} \end{cases}$$

In other words, if any of the transitions from  $m$  over  $a$  is accepting (i.e. in  $F_i$ ), we move to the singleton of the minimal state accessible by such a transition. Otherwise, we just proceed as in the standard powerset construction. Note that

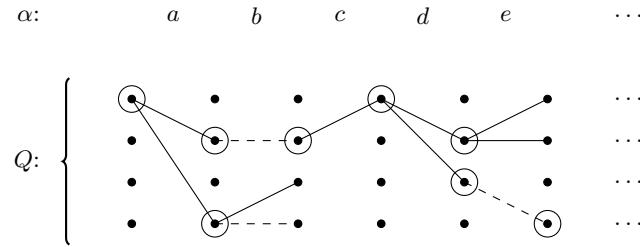
by Corollary 18 we know that the successive set of states of  $\mathcal{A}_i$  is also an element of  $M_i$ .

The accepting transitions  $F^{\mathcal{D}_i}$  of  $\mathcal{D}_i$  are the transitions of the first type (notice that their target is always a singleton). Summing up,

$$\mathcal{D}_i = \langle A, M_i, \{q_1\}, \delta^{\mathcal{D}_i}, F^{\mathcal{D}_i} \rangle$$

is a deterministic Büchi automaton.

See Figure 4 for a depiction of a run of  $\mathcal{D}_i$  over an  $\omega$ -word. In this section we follow the convention that accepting transitions are dashed.



**Fig. 4.** The behaviour of  $\mathcal{D}_i$  over an  $\omega$ -word  $\alpha$ . Dots correspond to states of  $\mathcal{A}_i$ . Nodes in circles belong to the current state of  $\mathcal{D}_i$ . Dashed edges are accepting transitions of  $\mathcal{A}_i$ , normal edges are normal transitions of  $\mathcal{A}_i$ . Transitions from states of  $\mathcal{A}_i$  outside the current state of  $\mathcal{D}_i$  are omitted. The order  $<_Q$  on the states of  $\mathcal{A}$  is from top to the bottom.

The following lemma implies that the automaton  $\mathcal{D}_i$  is equivalent to  $\mathcal{A}_i$  in a strong sense.

**Lemma 19.** *If  $m \in M_i$  and  $q \in m$  then*

$$L(\mathcal{D}_i, m) = L(\mathcal{A}_i, q).$$

*In particular,  $L(\mathcal{D}_i) = L(\mathcal{A}_i)$  and if  $m \subseteq m'$  for  $m, m' \in M_i$  then*

$$L(\mathcal{D}_i, m) = L(\mathcal{D}_i, m').$$

*Proof.* Clearly an accepting run of  $\mathcal{D}_i$  over  $\alpha$  from  $m$  encodes an accepting run of  $\mathcal{A}_i$  over  $\alpha$  from a state  $q' \in m$ , i.e.  $\alpha \in L(\mathcal{A}_i, q')$ . By (1) it means that also  $\alpha \in L(\mathcal{A}_i, q)$ . Therefore  $L(\mathcal{D}_i, m) \subseteq L(\mathcal{A}_i, q)$ .

Consider an  $\omega$ -word  $\alpha \in L(\mathcal{A}_i, q)$ . We need to prove that  $\mathcal{D}_i$  accepts  $\alpha$  from  $m$ . We will proceed by induction showing that  $\mathcal{D}_i$  uses infinitely many accepting transitions in its run over  $\alpha$ .

It will be achieved by the following inductive claim.

*Claim.* If  $q \in m \in M_i$  and  $\alpha \in L(\mathcal{A}_i, q)$  then  $\mathcal{D}_i$  uses an accepting transition when reading  $\alpha$  from  $m$ .

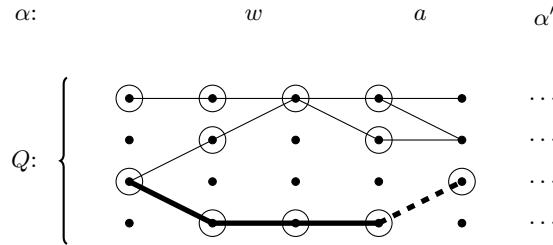
We start by proving why this claim finishes the proof of the lemma. Take  $q \in m \in M_i$  and  $\alpha \in L(\mathcal{A}_i, q)$ . Let  $m \xrightarrow{w} m'$  be the execution of  $\mathcal{D}_i$  from  $m$  over a prefix  $w$  of  $\alpha$  that contains an accepting transition given by the above claim. Let  $\alpha = wa'$ . For every  $q' \in m'$  there is a path  $q'' \xrightarrow{w} q'$  in  $\mathcal{A}_i$  with  $q'' \in m$ . By the definition of  $M_i$  we know that  $\alpha \in L(\mathcal{A}_i, q'')$  and by Corollary 18 we know that  $a' \in L(\mathcal{A}_i, q')$ . Therefore, we can apply the above claim inductively for  $q' \in m' \in M_i$  and  $a' \in L(\mathcal{A}_i, q')$ . This way we prove that the run of  $\mathcal{D}_i$  over  $\alpha$  from  $m$  contains infinitely many accepting transitions.

For the proof of the claim assume contrarily that the run of  $\mathcal{D}_i$  over  $\alpha$  from  $m$  does not contain any accepting transition. In that case for every  $n$  the state  $m_n$  of  $\mathcal{D}_i$  after reading  $\alpha(0) \dots \alpha(n-1)$  from  $m$  is

$$\{q' \mid q \xrightarrow{\alpha(0) \dots \alpha(n-1)} q' \text{ in } \mathcal{A}_i\}.$$

Consider an accepting run  $\rho = (q_0, q_1, \dots)$  of  $\mathcal{A}_i$  over  $\alpha$  from  $q$ . Let the first accepting transition in this run be  $q_n \xrightarrow{\alpha(n)} q_{n+1}$ . Since  $q \xrightarrow{\alpha(0) \dots \alpha(n-1)} q_n$  in  $\mathcal{A}_i$  we know that  $q_n \in m_n$  and therefore the transition of  $\mathcal{D}_i$  when reading  $\alpha(n)$  from  $m_n$  is accepting.

Figures 5 and 6 present the two possibilities for the run of  $\mathcal{D}_i$  from  $m$  to have an accepting transition if  $\rho$  is an accepting run of  $\mathcal{A}_i$  over  $\alpha$  from  $q \in m$ .



**Fig. 5.** The execution of  $\mathcal{D}_i$  over  $\alpha$  — the states in circles belong to the current state  $m$  of  $\mathcal{D}$ , the run  $\rho$  of  $\mathcal{A}_i$  is boldfaced, the dashed boldfaced edge is the first accepting transition in the run  $\rho$  over  $\alpha$ . This transition is also the source of the first accepting transition in the run of  $\mathcal{D}_i$  over  $\alpha$ .

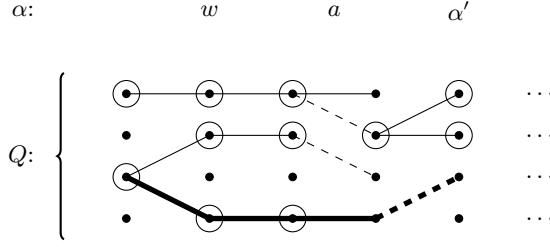
□

### B.3 Combinement of $\mathcal{A}_i$ and $\mathcal{D}_i$

We can now define formally the parity game  $\mathcal{G}_i$  that combines the automata  $\mathcal{A}_i$  and  $\mathcal{D}_i$ . Let  $W_i \subseteq Q_i \times M_i$  contain all pairs  $(q, m)$  such that

$$L(\mathcal{A}_i, q) = L(\mathcal{D}_i, m). \quad (2)$$

The set of positions of  $\mathcal{G}_i$  is of the form  $W_i \sqcup W_i \times A$ . The positions from  $W_i$  belong to  $\forall$  and the remaining ones to  $\exists$ . The edges are of the following two kinds:



**Fig. 6.** The execution of  $\mathcal{D}_i$  over  $\alpha$  — the states in circles belong to the current state  $m$  of  $\mathcal{D}$ , the run  $\rho$  of  $\mathcal{A}_i$  is boldfaced, the dashed boldfaced edge is the first accepting transition in  $\rho$ . The automaton  $\mathcal{D}_i$  performs an accepting transition before the first accepting transition is taken in the run  $\rho$  over  $\alpha$ . The state  $m' = \{q_1\}$  of  $\mathcal{D}_i$  after its first accepting transition does not contain the respective state in  $\rho$ . However, by the equivalence of residuals we know that  $\alpha' \in L(\mathcal{A}_i, q_1)$ .

- $(q, m) \rightarrow (q, m, a)$  for all  $a \in A$ ,
- $(q, m, a) \rightarrow (q', m')$  for all transitions  $q \xrightarrow{a} q'$  of  $\mathcal{A}_i$  and  $m' = \delta^{\mathcal{D}_i}(m, a)$ .

The priorities of the transitions of the first kind are 0. The priority of a transition of the second kind is either:

- 0 if both transitions  $q \xrightarrow{a} q'$  of  $\mathcal{A}_i$  and  $m \xrightarrow{a} m'$  of  $\mathcal{D}_i$  are not accepting,
- 2 if the transition  $q \xrightarrow{a} q'$  of  $\mathcal{A}_i$  is accepting,
- 1 otherwise (i.e. if the transition of  $\mathcal{A}_i$  is not accepting but the transition of  $\mathcal{D}_i$  is).

**Fact 20.** Consider a play of  $\mathcal{G}_i$  starting from a position  $(q, m) \in W_i$  in which  $\forall$  proposed a sequence of letters  $\alpha$  and  $\exists$  proposed a sequence of states  $\rho$  of  $\mathcal{A}_i$ . Let  $k$  be the lim sup of priorities of edges during this play. The following cases can occur (recall (2)):

- $k = 0$  and  $\alpha \notin L(\mathcal{A}_i, q)$ ,
- $k = 1$  and  $\alpha \in L(\mathcal{A}_i, q)$ , but the run  $\rho$  is not accepting over  $\alpha$ ,
- $k = 2$  and the run  $\rho$  is accepting over  $\alpha$  (it witnesses the fact that  $\alpha \in L(\mathcal{A}_i, q)$ ).

Summing up,  $\exists$  wins such a play if either  $\alpha \notin L(\mathcal{A}_i, q)$  or she managed to produce an accepting run  $\rho$  of  $\mathcal{A}_i$  over  $\alpha$  from  $q$ .

The following lemma follows directly from the fact that  $\mathcal{A}_i$  is GFG.

**Lemma 21.** For every pair  $(q, m) \in W_i$   $\exists$  has a winning strategy in  $\mathcal{G}_i$  from  $(q, m)$ .

*Proof.* By our assumptions on minimality of  $\mathcal{A}_i$  there is a word  $w \in A^*$  such that  $\sigma_i^{\text{GFG}}(w) = q$ . Consider the following strategy of  $\exists$  in  $\mathcal{G}_i$ : after  $\forall$  playing a sequence of letters  $v \in A^*$ ,  $\exists$  moves to the state  $\sigma_i^{\text{GFG}}(wv)$ . By the assumptions

on  $\sigma_i^{\text{GFG}}$  and Lemma 19 this is a correct play in  $\mathcal{G}_i$ . Consider a play in which  $\forall$  played  $\alpha \in A^\omega$ . Assume for contradiction that the limsup of the priorities in this play is 1. By Fact 20 it means that  $\alpha \in L(\mathcal{A}_i, q)$  but the run constructed by  $\sigma_i^{\text{GFG}}$  is not accepting over  $\alpha$ . But in that case  $w\alpha \in L$  but the run constructed by  $\sigma_i^{\text{GFG}}$  over  $w\alpha$  is not accepting. A contradiction with the assumptions on  $\sigma_i^{\text{GFG}}$ .  $\square$

In particular  $(q_1, \{q_1\}) \in W_i$  is a winning position of  $\exists$ . Note that  $W_i$  may contain some pairs  $(q, m)$  where  $q \notin m$ , such pairs will be essential in our construction (see e.g. Definition 29).

#### B.4 Optimal strategy

We now recall some simple variant of the theory of ranks (or signatures) in parity games. The definitions will be specialised for the game  $\mathcal{G}_i$  with priorities  $\{0, 1, 2\}$ .

Let us recall that a positional strategy is uniform, if it wins from all the winning positions in the game. By [EJ91, Mos91] parity games are uniformly positionally determined. Since  $\exists$  wins from all the positions  $W_i$ , a uniform positional strategy of  $\exists$  in  $\mathcal{G}_i$  is a function  $\sigma: W_i \times A \rightarrow W_i$ . Note that if  $\sigma(q, m, a) = (q', m')$  then by the definition of  $\mathcal{G}_i$  we have

$$\delta^{\mathcal{D}_i}(m, a) = m'.$$

We will additionally require our strategy to be optimal with respect to ranks defined as follows.

**Definition 22.** *Let  $\sigma$  be a winning strategy of  $\exists$  in  $\mathcal{G}_i$  from a position  $(q, m)$ . We say that  $\text{rank}(\sigma, q, m)$  is  $k$  if  $k$  is the maximal number of edges of priority 1 taken before the first<sup>8</sup> edge of priority 2 is taken in plays consistent with  $\sigma$ .*

Now we can ask for optimal ranks of given positions.

**Definition 23.** *For  $(q, m) \in W_i$  let  $\text{rank}_i(q, m)$  be the minimal  $\text{rank}(\sigma, q, m)$  ranging over winning strategies  $\sigma$  of  $\exists$  from  $(q, m)$ .*

*Let  $\text{opt}_i(q)$  be the minimal  $\text{rank}_i(q, m)$  ranging over  $m$  such that  $(q, m) \in W_i$ .*

The following proposition states that there exists a winning strategy optimising the values  $\text{rank}_i$  in all positions, see [SE89, Wal02].

**Proposition 24.** *There exists a uniform positional winning strategy  $\tau_i$  of  $\exists$  in  $\mathcal{G}_i$  such that for every position  $(q, m) \in W_i$  we have*

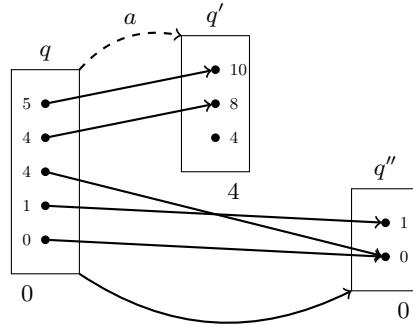
$$\text{rank}(\tau_i, q, m) = \text{rank}_i(q, m) \tag{3}$$

---

<sup>8</sup> If there is no such edge then we count all the edges of priority 1 in a given play.

*Sketch of a proof.* Having computed the values  $\text{rank}_i$  we can let the strategy  $\tau_i$  move from  $(q, m, a)$  to the position  $(q', m')$  of minimal  $\text{rank}_i$  among the available successive positions. This way  $\tau_i$  is a winning strategy because whenever it takes an edge of priority 1 the current value of  $\text{rank}_i$  decreases. Consider (3). First  $\tau_i$  witnesses the  $\geq$  inequality. But since  $\tau_i$  follows the optimal values of  $\text{rank}_i$ , the  $\leq$  inequality also holds.  $\square$

From that point on we work with a fixed optimal uniform positional strategy  $\tau_i: W_i \times A \rightarrow W_i$ . Let us additionally assume that all the ambiguous choices in the construction of  $\tau_i$  are resolved using the order  $<_Q$ . This way we guarantee that if the automaton  $\mathcal{A}_{i'}$  is the same as  $\mathcal{A}_i$  then the strategy  $\tau_{i'}$  is the same as  $\tau_i$ . Figure 7 presents an automaton  $\mathcal{A}_i$  and a strategy  $\tau_i$  in a schematic way.



**Fig. 7.** An illustration of  $\mathcal{A}_i$  and  $\tau_i$ . The rectangles are states  $q$  of  $\mathcal{A}_i$ , the dots inside them are respective values  $m$  such that  $(q, m) \in W_i$ . The straight edges are mappings by the function  $\tau_i$  under  $a$ , the bent edges are transitions of  $\mathcal{A}_i$  over  $a$ . The accepting transitions of  $\mathcal{A}_i$  are dashed. The number next to each dot is the value of  $\text{rank}_i(q, m)$  and the number at the bottom of the rectangle is the value of  $\text{opt}_i(q)$ .

The following lemma summarises the monotonicity properties of  $\text{rank}_i$ .

**Lemma 25.** Consider  $(q, m) \in W_i$  and  $a \in A$ . Assume that  $\tau_i(q, m, a) = (q', m')$  and  $q \xrightarrow{a} q'$  is not an accepting transition of  $\mathcal{A}_i$ . Then:

- $\text{rank}_i(q, m) \geq \text{rank}_i(q', m')$ ,
- if  $m \xrightarrow{a} m'$  is a non-accepting transition of  $\mathcal{D}_i$  then we have

$$\text{rank}_i(q, m) > \text{rank}_i(q', m').$$

## B.5 Inductive normalisation

We will now perform a sequence of modifications on  $\mathcal{A}_i$  to obtain  $\mathcal{A}_{i+1}$ . During these modifications we will preserve certain properties witnessing that our current automaton is still GFG. Let us introduce these concepts more formally.

Consider an automaton  $\mathcal{A}' = \langle A, Q_i, q_1, \Delta', F' \rangle$  with set of states  $Q_i$  and set of transitions  $\Delta' \subseteq \Delta_i$ . The set of accepting transitions  $F'$  can be any subset of transitions  $\Delta'$ .

Assume that  $\tau: W_i \times A \rightarrow W_i$  is a function such that for all  $(q, m) \in W_i$  and  $a \in A$  we have  $\tau(q, m, a) = (q', m')$  with:

- $q \xrightarrow{a} q'$  is a transition of  $\mathcal{A}'$  (i.e. also a transition of  $\mathcal{A}_i$ ),
- if the above transition is non-accepting in  $\mathcal{A}_i$  then

$$\text{rank}_i(q, m) \geq \text{rank}_i(q', m').$$

For each finite or infinite word  $\alpha \in A^* \sqcup A^\omega$  and  $(q, m) \in W_i$  such a function  $\tau$  induces a sequence (finite or infinite) of pairs

$$\tau[q, m] \cdot \alpha \stackrel{\text{def}}{=} (q_0, m_0), (q_1, m_1), \dots$$

defined inductively:  $(q_0, m_0) = (q, m)$  and  $(q_{n+1}, m_{n+1}) = \tau(q_n, m_n, \alpha(n))$ . Note that in that case the sequence  $q_0, q_1, \dots$  is a (finite or infinite) run of  $\mathcal{A}'$  over  $\alpha$  from  $q$ . This sequence is called the  $\tau$ -run of  $\mathcal{A}'$  over  $\alpha$  from  $(q, m)$ .

**Definition 26.** For  $\mathcal{A}'$  and  $\tau$  as above we say that  $\tau$  is a GFG-witness for  $\mathcal{A}'$  if:

1. for every  $q \in Q_i$  we have

$$L(\mathcal{A}', q) \subseteq L(\mathcal{A}_i, q), \quad (4)$$

2. for every  $(q, m) \in W_i$  and  $\alpha \in L(\mathcal{A}_i, q)$ , the  $\tau$ -run of  $\mathcal{A}'$  over  $\alpha$  from  $(q, m)$  contains at least one accepting transition of  $\mathcal{A}'$ .

**Lemma 27.** Assume that  $\mathcal{A}'$  is as above and  $\tau$  is a GFG-witness for  $\mathcal{A}'$ . Then:

- for every  $(q, m) \in W_i$  and  $\alpha \in L(\mathcal{A}_i, q)$ , the  $\tau$ -run of  $\mathcal{A}'$  over  $\alpha$  from  $(q, m)$  is accepting (with respect to the accepting transitions of  $\mathcal{A}'$ ),
- for every  $q \in Q_i$  we have  $L(\mathcal{A}', q) = L(\mathcal{A}_i, q)$ ,
- the automaton  $\mathcal{A}'$  is a GFG automaton for our language  $L$ .

*Proof.* The first claim follows from the inductive application of Item 2 from Definition 26: we start by finding one accepting transition  $q_n \xrightarrow{\alpha(n)} q_{n+1}$  in the run. Then we observe that  $(\alpha(n+1), \alpha(n+2), \dots) \in L(\mathcal{A}_i, q_{n+1})$  by Corollary 18 and we can proceed inductively.

The second claim follows from the first one and (4). For the last it is enough to observe that  $\tau[q_1, \{q_1\}] \cdot \alpha$  constructs letter-by-letter an accepting  $\tau$ -run of  $\mathcal{A}'$  over  $\alpha$ , for every  $\alpha \in L$ .  $\square$

Now we can prove the following lemma.

**Lemma 28.**  $\tau_i$  is a GFG-witness for  $\mathcal{A}_i$ .

*Proof.* It is enough to prove Item 2 from Definition 26. We will in fact prove that the run contains infinitely many accepting transitions.

Consider  $(q, m) \in W_i$  and  $\alpha \in L(\mathcal{A}_i, q)$ . By (2) we know that  $\alpha \in L(\mathcal{D}_i, m)$ . Consider the play of  $\mathcal{G}_i$  starting in  $(q, m)$  in which  $\forall$  proposes successive letters of  $\alpha$  and  $\exists$  plays according to  $\tau_i$ . The sequence of positions from  $W_i$  visited in this play is exactly the sequence  $\tau_i[q, m] \cdot \alpha$ .

Since  $\alpha \in L(\mathcal{D}_i, m)$ , the priority at least 1 is visited infinitely often in this play. Since  $\tau_i$  is a winning strategy, also priority 2 has to be seen infinitely often. But it means that  $\tau_i$  constructs an accepting run of  $\mathcal{A}_i$  over  $\alpha$ . This run is the  $\tau_i$ -run of  $\mathcal{A}_i$  over  $\alpha$  from  $(q, m)$ .  $\square$

**Overview of the steps** The construction of  $\mathcal{A}_{i+1}$  from  $\mathcal{A}_i$  will be done in four steps: we will define  $\mathcal{A}_i^{(j)}$  and  $\tau_i^{(j)}$  for  $j = 1, 2, 3, 4$ . The steps of the construction are illustrated on Figures 8, 9, 10, and 11. The convention for these figures is as explained in Figure 7. Note that during the following four steps of the construction we consider the original values  $\text{rank}_i(q, m)$  and  $\text{opt}_i(q)$  (they are not recomputed).

Let us overview the four steps that we will perform:

1.  $\mathcal{A}_i$  to  $\mathcal{A}_i^{(1)}$ : determinise transitions  $q \xrightarrow{a} q'$  if  $\text{rank}_i(q, \{q\}) = 0$ , see Figure 8,
2.  $\mathcal{A}_i^{(1)}$  to  $\mathcal{A}_i^{(2)}$ : determinise accepting transitions  $q \xrightarrow{a} q'$ , see Figure 9,
3.  $\mathcal{A}_i^{(2)}$  to  $\mathcal{A}_i^{(3)}$ : ensure that if  $q \xrightarrow{a} q'$  is non-accepting then  $\text{opt}_i(q) \geq \text{opt}_i(q')$ , see Figure 10,
4.  $\mathcal{A}_i^{(3)}$  to  $\mathcal{A}_i^{(4)}$ : make all transitions  $q \xrightarrow{a} q'$  with  $\text{opt}_i(q) > \text{opt}_i(q')$  accepting, see Figure 11.

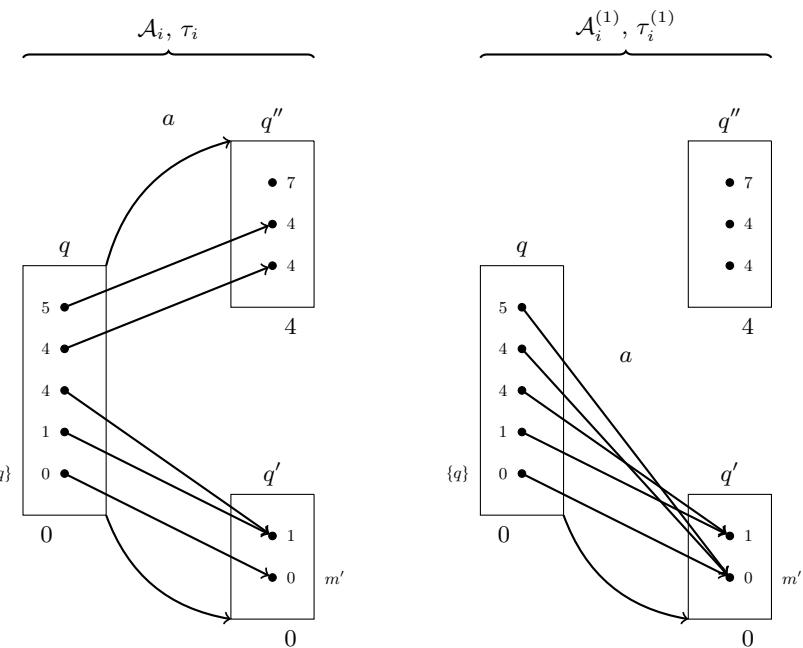
**Determinising self-dependent states** The first step is focused on the so-called *self-dependent* states. The dependency relation is defined as follows. It will play crucial role in defining the polynomial deterministic automaton equivalent to  $\mathcal{A}$ .

**Definition 29.** If  $(q, \{q'\}) \in W_i$  and  $\text{rank}_i(q, \{q'\}) = 0$  we say that  $q$  depends on  $q'$  and denote it  $q \hookrightarrow_i q'$  (or  $q \hookrightarrow q'$  if  $i$  is known from the context).

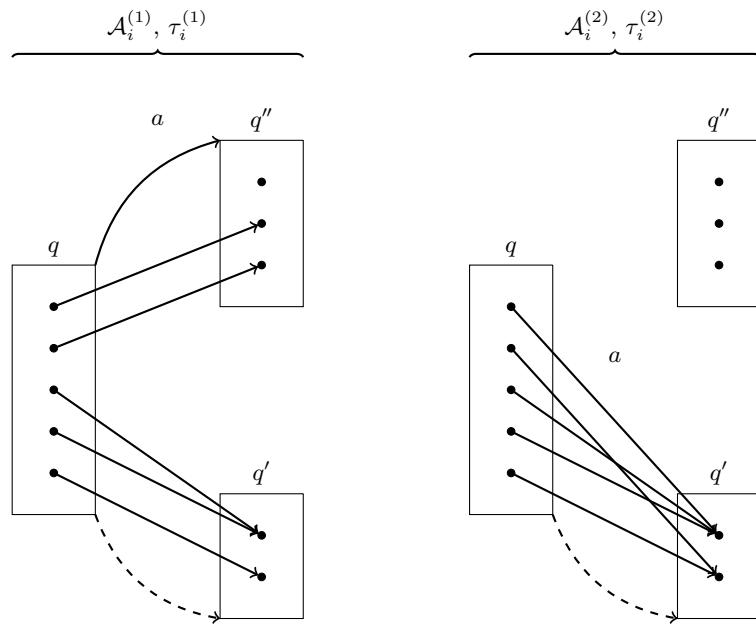
At this point of the construction we will turn deterministic all non-accepting transitions from a state  $q$  whenever  $\text{rank}_i(q, \{q\}) = 0$  (i.e.  $q \hookrightarrow_i q$  is *self-dependent*). Note that by the definition we always have  $(q, \{q\}) \in W_i$ .

Let  $\mathcal{A}_i^{(1)}$  be the automaton  $\mathcal{A}_i$  with the following modification: if  $q \xrightarrow{a} q'$  is a non-accepting transition of  $\mathcal{A}_i$ ,  $\text{rank}_i(q, \{q\}) = 0$ , and  $\tau_i(q, \{q\}, a) = (q', m')$  then remove all the transitions  $q \xrightarrow{a} q''$  with  $q'' \neq q'$ . Let  $\tau_i^{(1)} = \tau_i$  except for the values  $(q, m) \in W_i$ ,  $a \in A$  when  $\tau_i(q, m, a) = (q'', m'')$  and the transition  $q \xrightarrow{a} q''$  has been removed. Then let  $\tau_i^{(1)}(q, m, a) = \tau_i(q, \{q\}, a)$ . See Figure 8 for an illustration of the performed modifications.

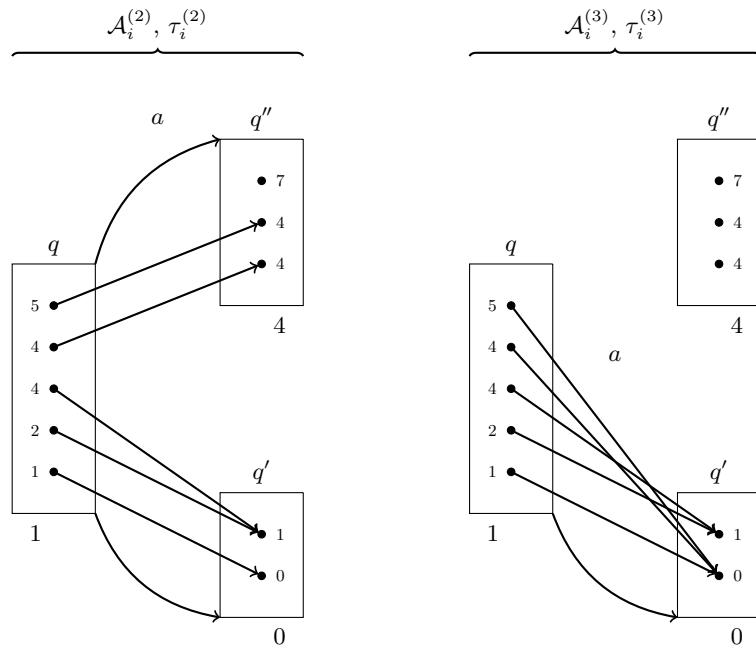
To prove that the above operation guarantees that  $\tau_i^{(1)}$  is a GFG-witness for  $\mathcal{A}_i^{(1)}$  we will use the following notion.



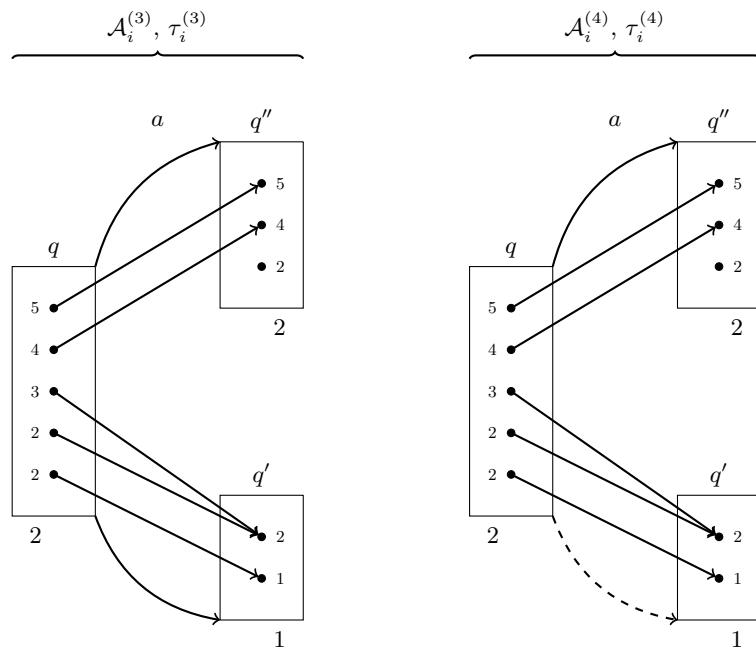
**Fig. 8.** The step from  $\mathcal{A}, \tau_i$  to  $\mathcal{A}_i^{(1)}, \tau_i^{(1)}$ . The pair  $(q, \{q\})$  has  $\text{rank}_i$  equal 0 and therefore triggers removal of all other transitions  $q \xrightarrow{a} q''$ .  $\tau_i^{(1)}$  maps all the values  $(q, m'')$  from the removed transition to  $(q', m') = \tau_i(q, \{q\}, a)$ .



**Fig. 9.** The step from from  $\mathcal{A}_i^{(1)}, \tau_i^{(1)}$  to  $\mathcal{A}_i^{(2)}$  to  $\tau_i^{(2)}$ . The transition  $q \xrightarrow{a} q'$  is accepting, it triggers removal of the transition  $q \xrightarrow{a} q''$  and modification of  $\tau_i^{(1)}$  on the two upper pairs of the form  $(q, m)$ . For these pairs any value  $m'$  such that  $(q', m') \in W_i$  is used.



**Fig. 10.** The step from  $\mathcal{A}_i^{(2)}, \tau_i^{(2)}$  to  $\mathcal{A}_i^{(3)}, \tau_i^{(3)}$ . Both transitions  $q \xrightarrow{a} q'$  and  $q \xrightarrow{a} q''$  are non-accepting. The transition  $q \xrightarrow{a} q''$  increases the value of  $\text{opt}_i$  and therefore is removed. The mapping by  $\tau_i^{(2)}$  is modified appropriately.



**Fig. 11.** The step from  $\mathcal{A}_i^{(3)}, \tau_i^{(3)}$  to  $\mathcal{A}_i^{(4)}$  to  $\tau_i^{(4)}$ . Both transitions  $q \xrightarrow{a} q'$  and  $q \xrightarrow{a} q''$  were non-accepting. The transition  $q \xrightarrow{a} q'$  decreases the value of  $\text{opt}_i$  and after the modification the it is made accepting.

**Definition 30.** For  $q \in Q_i$  let  $L^{\text{fin}}(\mathcal{A}_i, q)$  be the set of finite words  $w \in A^*$  such that there exists a finite run of  $\mathcal{A}_i$  over  $w$  from  $q$  that contains at least one accepting transition of  $\mathcal{A}_i$ .

Note that except the accepting transitions of  $\mathcal{A}_i$ , the automaton  $\mathcal{D}_i$  operates as the standard powerset construction. Therefore, we obtain the following fact. The second part of the fact follows from the definition of the ranks.

**Fact 31.** If  $(q, m) \in W_i$ ,  $q \in m$ , and  $w \in L^{\text{fin}}(\mathcal{A}_i, q)$  then the finite run of  $\mathcal{D}_i$  over  $w$  from  $m$  contains at least one accepting transition of  $\mathcal{D}_i$ .

If  $\text{rank}_i(q, m) = 0$ ,  $q \in m$ , and  $w \in L^{\text{fin}}(\mathcal{A}_i, q)$  then the  $\tau_i$ -run of  $\mathcal{A}_i$  over  $w$  from  $(q, m)$  contains an accepting transition of  $\mathcal{A}_i$ .

**Lemma 32.**  $\tau_i^{(1)}$  is a GFG-witness for  $\mathcal{A}_i^{(1)}$ .

*Proof.* First observe that by monotonicity we know that  $L(\mathcal{A}_i^{(1)}, q) \subseteq L(\mathcal{A}_i, q)$  for every  $q \in Q_i$ , since we just removed transitions. Therefore, it is enough to prove Item 2 of Definition 26.

For the sake of the contradiction assume that for  $(q, m) \in W_i$  and  $\alpha \in L(\mathcal{A}_i, q)$  the  $\tau_i^{(1)}$ -run of  $\mathcal{A}_i^{(1)}$  over  $\alpha$  from  $(q, m)$  does not contain any accepting transition. Since  $\tau_i$  is a GFG-witness for  $\mathcal{A}_i$ , it means that in the above run over  $\alpha$  infinitely many times a triple  $(q, m, a)$  appears with

$$\tau_i(q, m, a) \neq \tau_i^{(1)}(q, m, a) = \tau_i(q, \{q\}, a).$$

Moreover, notice that if no accepting transition is witnessed in  $\mathcal{A}_i^{(1)}$ , then the invariant that  $q \in m$  is preserved by all other transitions (of two types: powerset transitions and new transitions). Therefore, we can start our considerations from such a triple with  $\text{rank}_i(q, m) = 0$  and  $q \in m$ .

Since  $\alpha \in L(\mathcal{A}_i, q)$ , there exists a finite prefix  $w$  of  $\alpha$  such that  $w \in L^{\text{fin}}(\mathcal{A}_i, q)$ .

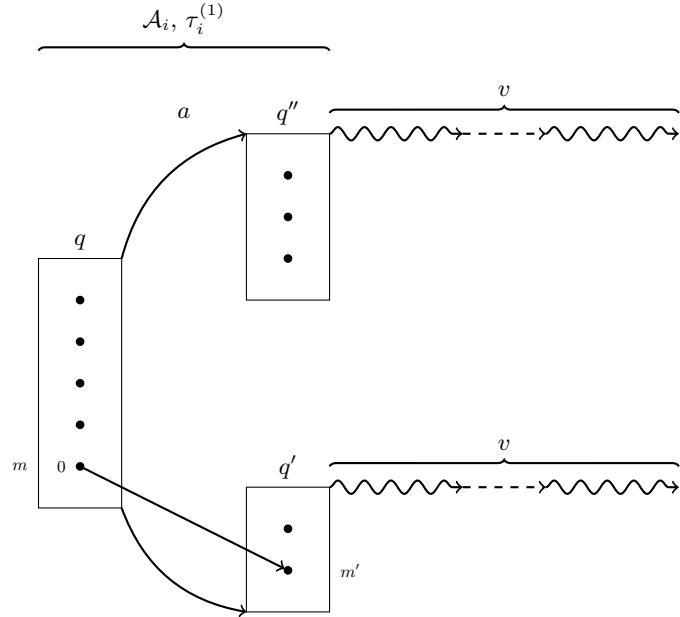
The contradiction follows from the following claim.

*Claim.* If  $(q, m) \in W_i$ ,  $\text{rank}_i(q, m) = 0$ ,  $q \in m$ , and  $w \in L^{\text{fin}}(\mathcal{A}_i, q)$  then the  $\tau_i^{(1)}$ -run of  $\mathcal{A}_i^{(1)}$  over  $w$  from  $(q, m)$  contains an accepting transition of  $\mathcal{A}_i^{(1)}$ .

Note that the claim holds for  $w = \epsilon$  as  $\epsilon \notin L^{\text{fin}}(\mathcal{A}_i, q)$  for every  $q$ . The proof of the claim is inductive in the length of  $w$ . Assume that  $w = av$  and the claim holds for all words of length at most  $|v|$ . Let  $(q', m') = \tau_i^{(1)}(q, m, a)$ . Note that either  $(q', m') = \tau_i(q, m, a)$  or  $(q', m') = \tau_i(q, \{q\}, a)$  with  $\text{rank}_i(q, \{q\}) = 0$ . In the latter case we can assume without loss of generality that  $m = \{q\}$  and thus  $\tau_i^{(1)}(q, m, a) = \tau_i(q, m, a)$  (the  $\tau_i^{(1)}$ -runs over  $w$  from  $(q, m)$  and from  $(q, \{q\})$  are the same except the first state).

If the transition  $q \xrightarrow{a} q'$  is accepting in  $\mathcal{A}_i$  then the claim clearly holds. Assume the opposite. Since  $\text{rank}_i(q, m) = 0$ , the transition  $m \xrightarrow{a} m'$  is a non-accepting transition of  $\mathcal{D}_i$ . Therefore,  $q' \in m'$  and  $\text{rank}_i(q', m') = 0$ . Thus, it remains to prove that  $v \in L^{\text{fin}}(\mathcal{A}_i, q')$  and use the inductive assumption.

By Fact 31 we know that the  $\tau_i$ -run of  $\mathcal{A}_i$  over  $av$  from  $(q, m)$  contains an accepting transition. But this run starts with the non-accepting transition  $q \xrightarrow{a} q'$ , therefore it witnesses the fact that  $v \in L^{\text{fin}}(\mathcal{A}_i, q')$ . See Figure 12 for an illustration of this proof.  $\square$



**Fig. 12.** The inductive proof of Lemma 32. By the assumption that  $w = av \in L^{\text{fin}}(\mathcal{A}_i, q)$  we know that the upper path exists in  $\mathcal{A}_i$  and contains an accepting transition. The value  $m$  is fixed such that  $q \in m$ ,  $\text{rank}_i(q, m) = 0$ , and  $\tau_i^{(1)}(q, m, a) = \tau_i(q, m, a) = (q', m')$ . If the transition  $q \xrightarrow{a} q'$  is not accepting then  $\tau_i$  constructs a run from  $(q', m')$  over  $v$  that contains an accepting transition (lower path). This path witnesses the fact that  $v \in L^{\text{fin}}(\mathcal{A}_i, q')$ .

**Deterministic accepting transitions** Now we will enforce that if  $\mathcal{A}_i^{(1)}$  can perform from a state  $q$  an accepting transition over  $a$ :  $q \xrightarrow{a} q'$  then this is the only transition from  $q$  over  $a$ . Let  $\mathcal{A}_i^{(2)}$  be obtained from  $\mathcal{A}_i^{(1)}$  by consequent picking an accepting transition  $q \xrightarrow{a} q'$  and removing all other transitions (both accepting and non-accepting)  $q \xrightarrow{a} q''$  with  $q'' \neq q'$ . The order in which we pick the accepting transitions is not relevant.

Now we need to define a new function  $\tau_i^{(2)}$ . Let  $\tau_i^{(2)}$  equal  $\tau_i^{(1)}$  except for the values  $(q, m) \in W_i$ ,  $a \in A$  such that  $\tau_i^{(2)}(q, m, a) = (q'', m'')$  with the transition  $q \xrightarrow{a} q''$  removed from  $\mathcal{A}_i^{(1)}$ . In such a case let  $q \xrightarrow{a} q'$  be the accepting transition

that triggered the removal and let  $m'$  be any value such that  $(q', m') \in W_i$ . Let us put  $\tau_i^{(1)}(q, m, a) = (q', m')$  instead of  $(q'', m'')$ . See Figure 9 for an illustration of the construction of  $\tau_i^{(2)}$  from  $\tau_i^{(1)}$ .

**Lemma 33.**  $\tau_i^{(2)}$  is a GFG-witness for  $\mathcal{A}_i^{(2)}$ .

*Proof.* First observe that by monotonicity we know that  $L(\mathcal{A}_i^{(2)}, q) \subseteq L(\mathcal{A}_i, q)$  for every  $q \in Q_i$ , since we only removed transitions. Therefore, it is enough to prove Item 2 of Definition 26.

Consider a pair  $(q, m) \in W_i$  and an  $\omega$ -word  $\alpha \in L(\mathcal{A}_i, q)$ . Let

$$\begin{aligned}\tau_i^{(1)}[q, m] \cdot \alpha &= (q_0, m_0), \dots \\ \tau_i^{(2)}[q, m] \cdot \alpha &= (q'_0, m'_0), \dots\end{aligned}$$

We will prove that the run  $q'_0, q'_1, \dots$  contains at least one accepting transition. Indeed, the first place where the runs  $(q'_n)_n$  and  $(q_n)_n$  can differ is, by the definition of  $\tau_i^{(2)}$ , an accepting transition in  $(q'_n)_n$ . If the runs do not differ then  $(q'_n)_n$  is accepting because  $(q_n)_n$  was accepting ( $\tau_i^{(1)}$  was a GFG-witness for  $\mathcal{A}_i^{(1)}$ ).  $\square$

**Decreasing  $\text{opt}_i(q)$**  We now want to modify  $\mathcal{A}_i^{(2)}$  and  $\tau_i^{(2)}$  in such a way to guarantee that if  $q \xrightarrow{a} q'$  is a non-accepting transition of  $\mathcal{A}_i^{(2)}$  then  $\text{opt}_i(q) \geq \text{opt}_i(q')$ . What we know is that the values of  $\text{rank}_i$  decrease along such transitions, see Lemma 25. It does not imply that the values of  $\text{opt}_i$  decrease, see the left-hand part of Figure 10.

Let  $\tau_i^{(3)} = \tau_i^{(2)}$  except for  $(q, m) \in W_i$  and  $a \in A$  such that  $\tau_i^{(2)}(q, m, a) = (q'', m'')$  and  $\text{opt}_i(q) < \text{opt}_i(q'')$ . For such values, let  $m_q$  be a value realising the minimal  $\text{rank}_i$  in  $q$ , i.e.  $(q, m_q) \in W_i$  and  $\text{rank}_i(q, m_q) = \text{opt}_i(q)$ . In that case put  $\tau_i^{(3)}(q, m, a) = \tau_i^{(2)}(q, m_q, a) = (q', m')$ , see Figure 10. Also, remove from  $\mathcal{A}_i^{(2)}$  all the non-accepting transitions  $q \xrightarrow{a} q''$  such that  $\text{opt}_i(q) < \text{opt}_i(q'')$  and obtain  $\mathcal{A}_i^{(3)}$ .

Note that for the values  $(q, m) \in W_i$  and  $a \in A$  where  $\tau_i^{(2)}(q, m, a) \neq \tau_i^{(3)}(q, m, a) = (q', m')$  as above, we have:

$$\begin{aligned}\text{rank}_i(\tau_i^{(3)}(q, m, a)) &= \text{rank}_i(q', m') && \text{by the definition of } \tau_i^{(3)} \\ &\leq \text{rank}_i(q, m_q) && \text{because } \tau_i^{(2)}(q, m_q, a) = (q', m') \\ &= \text{opt}_i(q) && \text{by the choice of } m_q \\ &< \text{opt}_i(q'') && \text{by the assumption} \\ &\leq \text{rank}_i(q'', m'') && \text{by the definition of } \text{opt}_i \\ &\leq \text{rank}_i(q, m) && \text{because } \tau_i^{(2)}(q, m, a) = (q'', m'').\end{aligned}$$

Summing up, if  $\tau_i^{(2)}(q, m, a) \neq \tau_i^{(3)}(q, m, a)$  then

$$\text{rank}_i(\tau_i^{(3)}(q, m, a)) < \text{rank}_i(q, m). \quad (5)$$

**Lemma 34.**  $\tau_i^{(3)}$  is a GFG-witness for  $\mathcal{A}_i^{(3)}$ .

*Proof.* As before, by monotonicity we know that  $L(\mathcal{A}_i^{(3)}, q) \subseteq L(\mathcal{A}_i, q)$  for every  $q \in Q_i$ , since we only removed transitions. Therefore, it is enough to prove Item 2 of Definition 26.

Assume contrarily, that there exists  $(q, m) \in W_i$  and  $\alpha \in L(\mathcal{A}_i, q)$  such that the  $\tau_i^{(3)}$ -run over  $\alpha$  from  $(q, m)$  does not contain any accepting transition. Let

$$\tau_i^{(3)}[q, m] \cdot \alpha = (q'_0, m'_0), (q'_1, m'_1), \dots$$

For every  $n \in \mathbb{N}$  there are two possibilities:

- $\tau_i^{(2)}(q'_n, m'_n, \alpha(n)) = (q'_{n+1}, m'_{n+1})$  (i.e.  $\tau_i^{(2)}$  and  $\tau_i^{(3)}$  agree in that case)
- $\tau_i^{(2)}(q'_n, m'_n, \alpha(n)) \neq (q'_{n+1}, m'_{n+1}) = \tau_i^{(3)}(q'_n, m'_n, \alpha(n))$  because of the above modification.

In the first case  $\text{rank}_i(q'_n, m'_n) \geq \text{rank}_i(q'_{n+1}, m'_{n+1})$  because the transition  $q'_n \xrightarrow{\alpha(n)} q'_{n+1}$  is non-accepting in  $\mathcal{A}_i^{(3)}$  and therefore also in  $\mathcal{A}_i$  and  $\tau_i^{(2)}$  is a GFG-witness. In the second case we know that  $\text{rank}_i(q'_n, m'_n) > \text{rank}_i(q'_{n+1}, m'_{n+1})$  by (5). Therefore, the second case can happen only finitely many times (ranks are non-negative). It means that from some point on, the  $\tau_i^{(3)}$ -run over  $\alpha$  agrees with the  $\tau_i^{(2)}$ -run over some suffix of  $\alpha$  and thus is accepting (see the first item in Lemma 27).

More formally, let  $N$  be a number such that for all  $n \geq N$  only the first case above happens. Let

$$\tau_i^{(2)}[q'_N, m'_N] \cdot (\alpha(N), \alpha(N+1), \dots) = (q_N, m_N), (q_{N+1}, m_{N+1}), \dots$$

It means that for  $n \geq N$  we have:

$$(q'_n, m'_n) = (q_n, m_n).$$

Since  $\tau_i^{(2)}$  is GFG-witness and  $(\alpha(N), \alpha(N+1), \dots) \in L(\mathcal{A}_i, q_N)$  we know that the run  $(q_N, q_{N+1}, \dots)$  is an accepting run of  $\mathcal{A}_i^{(3)}$  over  $(\alpha(N), \alpha(N+1), \dots)$  and so is the original run  $(q'_0, q'_1, \dots)$ .  $\square$

Let us note that the construction of  $\mathcal{A}_i^{(3)}$  guarantees the following fact.

**Fact 35.** If  $q \xrightarrow{a} q'$  in  $\mathcal{A}_i^{(3)}$  is a non-accepting transition of  $\mathcal{A}_i^{(3)}$  then  $\text{opt}_i(q) \geq \text{opt}_i(q')$ .

**Adding Büchi transitions** We can now proceed to the crucial step of the modifications of  $\mathcal{A}_i$  — we will add some new Büchi transitions to it. Because of a special care that will be taken, we will ensure that the added transitions do not enlarge the language recognised by the automaton. Intuitively, the values of  $\text{rank}_{i+1}$  computed with respect to this enriched automaton will be smaller — it will be easier to use an accepting transition of  $\mathcal{A}_{i+1}$  and thus take an edge of priority 2 in  $\mathcal{G}_{i+1}$ .

Let  $\tau_i^{(4)} = \tau_i^{(3)}$  and  $\mathcal{A}_i^{(4)}$  be the automaton  $\mathcal{A}_i^{(3)}$  with all the transitions  $q \xrightarrow{a} q'$  such that  $\text{opt}_i(q) > \text{opt}_i(q')$  made accepting. Figure 11 illustrates the modifications.

**Lemma 36.**  $\tau_i^{(4)}$  is a GFG-witness for  $\mathcal{A}_i^{(4)}$ .

*Proof.* It is enough to prove that for  $q \in Q_i$  we have  $L(\mathcal{A}_i^{(4)}, q) \subseteq L(\mathcal{A}_i^{(3)}, q)$ .

Let  $\rho$  be an accepting run of  $\mathcal{A}_i^{(4)}$  over an  $\omega$ -word  $\alpha$  from  $q \in Q_i$ . Of course  $\rho$  is also a run of  $\mathcal{A}_i^{(3)}$  over  $\alpha$  from  $q$ . We want to prove that  $\rho$  is also accepting with respect to  $\mathcal{A}_i^{(3)}$ . Assume contrarily that  $\rho$  does not contain any accepting transition of  $\mathcal{A}_i^{(3)}$  (as before, we inductively focus on this case). Let  $\rho = (q_0, q_1, \dots)$ . Observe that by Fact 35 we know that  $\text{opt}_i(q_0) \geq \text{opt}_i(q_1) \geq \dots$ . By the construction of  $\mathcal{A}_i^{(4)}$  if the transition  $q_n \xrightarrow{\alpha(n)} q_{n+1}$  is accepting in  $\mathcal{A}_i^{(4)}$  but not in  $\mathcal{A}_i^{(3)}$  then  $\text{opt}_i(q_n) > \text{opt}_i(q_{n+1})$ . Therefore, there may be only finitely many such transitions in  $\rho$  and therefore  $\rho$  is rejecting with respect to  $\mathcal{A}_i^{(4)}$  as well.  $\square$

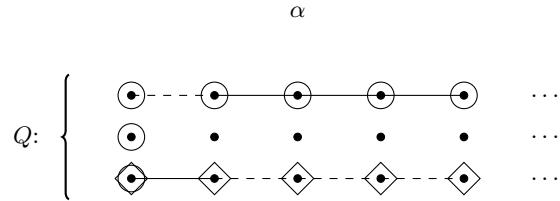
**Defining  $\mathcal{A}_{i+1}$**  We now define  $\mathcal{A}_{i+1} = \mathcal{A}_i^{(4)}$  and since  $\tau_i^{(4)}$  is a GFG-witness for  $\mathcal{A}_i^{(4)}$  we know that it is a GFG automaton recognising our language  $L$ . Clearly the invariants stated at the beginning of Section 4 are preserved by all the steps of our construction. This way we have completed the definition of  $\mathcal{A}_{i+1}$  from  $\mathcal{A}_i$ .

## B.6 Monotonicity

We will now show that there is some form of monotonicity of the values of  $\text{rank}_i(q, m)$  with respect to the set of states  $m$ , as expressed by the following lemma. Figure 13 shows that if we skip the assumption that  $\text{rank}_i(q, m) = 0$  then the monotonicity does not hold any more.

**Lemma 37.** Assume that  $(q, m) \in W_i$  and  $\text{rank}_i(q, m) = 0$ . If  $\emptyset \neq m' \subseteq m$  then  $(q, m') \in W_i$  and  $\text{rank}_i(q, m') = 0$ .

*Proof.* The fact that  $(q, m') \in W_i$  follows from the definition of  $W_i$ . To prove that also  $\text{rank}_i(q, m') = 0$  we will use Proposition 24 — it is enough to provide a winning strategy  $\sigma$  of  $\exists$  in  $\mathcal{G}_i$  from  $(q, m')$  that stays within the winning region (i.e. the whole set  $W_i$ ) and does not visit any edge of priority 1 before the first edge of priority 2. Recall that the only choice of  $\exists$  in  $\mathcal{G}_i$  is which state of  $\mathcal{A}_i$  to choose (the successive state of  $\mathcal{D}_i$  is taken deterministically).



**Fig. 13.** An  $\omega$ -word  $\alpha$  witnessing that  $\text{rank}_i(q_0, m'_0) \geq 3$  and  $\text{rank}_i(q_0, m_0) \geq 1$  with  $m'_0 \subseteq m_0$ . The states in circles are elements of  $m_0$ ,  $m_1, \dots$ , the states in squares are elements of  $m'_0, m'_1, \dots$ . We assume that  $q_0 = q'_0$ ,  $\tau_i(q_n, m_n, \alpha(n)) = (q_{n+1}, m_{n+1})$ ,  $\tau_i(q'_n, m'_n, \alpha(n)) = (q'_{n+1}, m'_{n+1})$ , and the runs  $(q_n)$ ,  $(q'_n)$  over  $\alpha$  do not contain any accepting transitions of  $\mathcal{A}_i$ . Then, the run  $(m_n)$  over  $\alpha$  contains one accepting transition of  $\mathcal{D}_i$  while the run  $(m'_n)$  over  $\alpha$  contains three accepting transitions of  $\mathcal{D}_i$ .

Let  $\sigma$  play from  $(q, m')$  as would play  $\tau_i$  from  $(q, m)$ , i.e. after  $\forall$  played  $w \in A^*$  let  $(q_w, m_w)$  be the successive state according to  $\tau_i$  and let  $\exists$  play  $q_w$  (the successive state will in that case be  $(q_w, m'_w)$  for  $m'_w$  possibly different than  $m_w$ ). When the first accepting transition of  $\mathcal{A}_i$  is taken let  $\sigma$  follow some winning strategy from a given position.

Clearly, since  $\tau_i$  does not visit any accepting transition of  $\mathcal{D}_i$  before an accepting transition of  $\mathcal{A}_i$  is taken, during this simulation we always preserve that  $m'_w \subseteq m_w$ . Therefore, we do not visit any accepting transition of  $\mathcal{D}_i$  before the first accepting transition of  $\mathcal{A}_i$  in all the plays consistent with  $\sigma$  from  $(q, m')$ .  $\square$

**Lemma 38.** Assume that  $(q, m) \in W_i$  and  $(q', m') \in W_i$  such that  $q' \in m$ . Then  $(q, m') \in W_i$  and if  $\text{rank}_i(q, m) = \text{rank}_i(q', m') = 0$  then also  $\text{rank}_i(q, m') = 0$ .

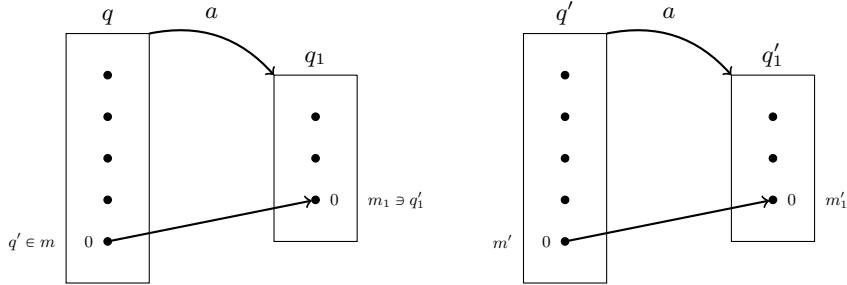
*Proof.* The fact that  $(q, m') \in W_i$  follows directly from the definition of  $W_i$ . Similarly as above we will provide a winning strategy  $\sigma$  of  $\exists$  from  $(q, m')$  that guarantees not visiting any accepting transition of  $\mathcal{D}_i$  before an accepting transition of  $\mathcal{A}_i$  is visited.

This claim can be proved inductively: it is enough to provide a response  $q_1$  for one letter  $a$  played by  $\forall$  from the position  $(q, m')$ . We need to guarantee that we do not pass through an accepting transition of  $\mathcal{D}_i$  before we take an accepting transition of  $\mathcal{A}_i$ .

First assume that there is an accepting transition of  $\mathcal{A}_i$  of the form  $q \xrightarrow{a} q_1$ . In that case just take it and stay within the winning region  $W_i$ .

Assume that there is no such transition and  $\tau_i$  moves over  $a$ : from  $(q, m)$  to  $(q_1, m_1)$  and from  $(q', m')$  to  $(q'_1, m'_1)$ , see Figure 14. Since  $\text{rank}_i(q, m) = 0$ , the transition  $m \xrightarrow{a} m_1$  of  $\mathcal{D}_i$  is not accepting, in particular the transition  $q' \xrightarrow{a} q'_1$  of  $\mathcal{A}_i$  is not accepting (recall that  $q' \in m$ ).

By monotonicity of  $\text{rank}_i$  over non-accepting transitions of  $\mathcal{A}_i$  we know that  $\text{rank}_i(q_1, m_1) = \text{rank}_i(q'_1, m'_1) = 0$ . By the definition of  $m_1$  we know that  $q'_1 \in m_1$ . Let  $\sigma$  move from  $(q, m')$  over  $a$  to  $q_1$ . By repeating the above construction we preserve the invariant that  $\text{rank}_i(q_n, m_n) = \text{rank}_i(q'_n, m'_n) = 0$  and  $q'_n \in m_n$ , therefore the strategy  $\sigma$  is a witness that  $\text{rank}_i(q, m') = 0$ .  $\square$



**Fig. 14.** An illustration of the proof of Lemma 38. We know that  $q' \in m$  and  $\text{rank}_i(q, m) = \text{rank}_i(q', m') = 0$ . The arrows are actions of  $\tau_i$ . The implications go as follows: the transition  $q \xrightarrow{a} q_1$  is not accepting so the transition  $m \xrightarrow{a} m_1$  is not accepting so the transition  $q' \xrightarrow{a} q'_1$  is not accepting so the transition  $m' \xrightarrow{a} m'_1$  is not accepting. Therefore,  $\text{rank}_i(q_1, m_1) = \text{rank}_i(q'_1, m'_1) = 0$  and  $q'_1 \in m_1$ . Note that  $m'$  is one of the dots in the state  $q$  but we do not care what is the value  $\tau_i(q, m', a)$ .

### B.7 Stabilisation point

The above inductive construction of  $\mathcal{A}_{i+1}$  from  $\mathcal{A}_i$  is monotone with respect to the set of states  $Q_i$ , set of transitions  $\Delta_i$ , and set of non-accepting transitions  $\Delta_i \setminus F_i$  (see the invariants at the beginning of Section 4). Therefore, there exists  $I$  such that  $\mathcal{A}_{I+1} = \mathcal{A}_I$ . We will prove the following lemma. It says that in the limit we succeed with diminishing the ranks to 0 (at least the  $\text{opt}_I$  ones).

**Lemma 39.** *If  $\mathcal{A}_{I+1} = \mathcal{A}_I$  and  $q \in Q^{\mathcal{A}_I}$  then  $\text{opt}_I(q) = 0$ .*

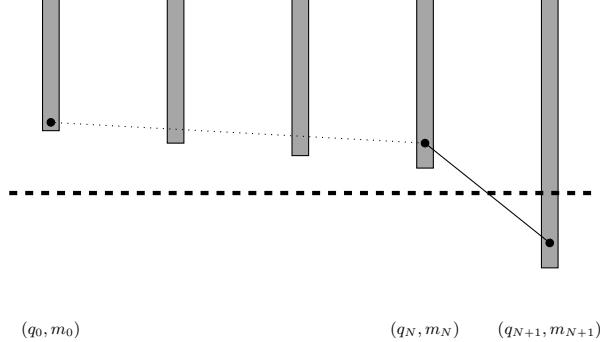
*Proof.* Assume contrarily that there exists a state  $q \in Q_I$  such that  $\text{opt}_I(q) > 0$ . Let  $m$  be a memory value such that  $(q, m) \in W_I$  and  $\text{rank}_I(q, m) = \text{opt}_I(q) > 0$ . Consider a play consistent with  $\tau_I$  in  $\mathcal{G}_I$  that witnesses this fact: an  $\omega$ -word  $\alpha$  such that there is at least one edge of priority 1 taken before the first edge of priority 2 is taken. Let  $\tau_I[q, m] \cdot \alpha = (q_n, m_n)_{n \in \mathbb{N}}$  be the sequence of positions visited during the play. Assume that  $m_N \xrightarrow{\alpha(N)} m_{N+1}$  is the first accepting transition of  $\mathcal{D}_I$  in this play and all the transitions  $q_n \xrightarrow{\alpha(n)} q_{n+1}$  for  $n \leq N$  are non-accepting in  $\mathcal{A}_I$ .

Our aim is to prove that for some transition  $q_n \xrightarrow{\alpha(n)} q_{n+1}$  with  $n \leq N$  we have  $\text{opt}_I(q_n) > \text{opt}_I(q_{n+1})$ . In that case this transition should be made accepting by the step performed in Section B.5. Observe that  $\text{opt}_I(q_n) \geq \text{opt}_I(q_{n+1})$  for  $n \leq N$ , otherwise some transition would be removed by the step performed in Section B.5.

By Lemma 25 we know that  $\text{rank}_I(q_N, m_N) > \text{rank}_I(q_{N+1}, m_{N+1})$ . Therefore we obtain

$$\begin{aligned} \text{opt}_I(q_0) &= \text{rank}_I(q_0, m_0) \geq \text{rank}_I(q_N, m_N) = \\ &= \text{rank}_I(q_N, m_N) > \text{rank}_I(q_{N+1}, m_{N+1}) \geq \text{opt}_I(q_{N+1}). \end{aligned}$$

It means that on the traversed path from  $q_0$  to  $q_{N+1}$  the value  $\text{opt}_I(q_n)$  has to strictly decrease, see Figure 15.  $\square$



**Fig. 15.** An illustration of the proof of Lemma 39. The path is the plot of values of  $\text{rank}_I(q_n, m_n)$  for  $n \leq N + 1$ . The gray rectangles denote the range of values  $\text{rank}_I(q_n, m')$  for possible values  $m'$ . At the beginning  $\text{opt}_I(q_0) = \text{rank}_I(q_0, m_0)$  and then  $\text{rank}_I(q_N, m_N) > \text{rank}_I(q_{N+1}, m_{N+1})$ . The horizontal dashed line marks the difference between  $\text{rank}_I(q_N, m_N)$  and  $\text{rank}_I(q_{N+1}, m_{N+1})$ . Therefore, somewhere on the path there is the first rectangle crossing the dashed line (i.e. a transition  $q_n \xrightarrow{\alpha(n)} q_{N+1}$  with  $\text{opt}_I(q_n) > \text{opt}_I(q_{N+1})$ ).

Let us recall the dependency relation from Definition 29:  $q$  depends on  $q'$  (denoted  $q \hookrightarrow q'$ ) if  $(q, \{q'\}) \in W_I$  and  $\text{rank}_I(q, \{q'\}) = 0$ . The following lemma summarises the properties of the dependency relation using the results of Section B.6.

**Lemma 40.** *The following conditions hold:*

1. *For every  $q \in Q_I$  there exists  $q' \in Q_I$  such that  $q \hookrightarrow q'$ .*
2. *If  $q \hookrightarrow q'$  and  $q' \hookrightarrow q''$  then  $q \hookrightarrow q''$ .*
3. *For every  $q \in Q_I$  there exists  $\bar{q} \in Q_I$  such that  $q \hookrightarrow \bar{q}$  and  $\bar{q} \hookrightarrow \bar{q}$ .*
4. *If  $q \hookrightarrow q$ ,  $a \in A$ , and  $\tau_I(q, \{q\}, a) = (q', m')$  such that  $q \xrightarrow{a} q'$  is a non-accepting transition of  $\mathcal{A}_I$  then  $q' \hookrightarrow q'$ .*

*Proof.* For the first claim observe that by Lemma 39 we know that  $\text{opt}_I(q) = 0$ . Let  $m$  be the value such that  $(q, m) \in W_I$  and  $\text{rank}_I(q, m) = 0$ . Since  $m \neq \emptyset$ , we can choose  $q'$  to be any element of  $m$ . By Lemma 37 we know that  $\text{rank}_I(q, \{q'\}) \leq \text{rank}_I(q, m) = 0$ .

Now take  $q \hookrightarrow q'$  and  $q' \hookrightarrow q''$ . In that case  $q' \in \{q'\}$  so Lemma 38 applies and  $\text{rank}_I(q, \{q''\}) = 0$  (i.e.  $q \hookrightarrow q''$ ).

For the third claim it is enough to apply the previous two inductively and use the fact that  $Q_I$  is finite.

Consider  $q \in Q_I$  such that  $q \hookrightarrow q$  and let  $\tau_i(q, \{q\}, a) = (q', m')$ . In that case we know that  $\text{rank}_I(q', m') = 0$  and  $q' \in m'$  therefore by Lemma 37 we know that also  $\text{rank}_I(q', \{q'\}) = 0$  and  $q' \hookrightarrow q'$ .  $\square$

## B.8 Construction of $\mathcal{B}$

Now we move to the construction of a small deterministic Büchi automaton  $\mathcal{B}$  recognising  $L$ . We start by defining a GFG-witness for  $\mathcal{A}_I$  that will involve only polynomially many pairs  $(q, m) \in W_I$ . Let  $\bar{\tau}_I = \tau_I$  for all values  $(q, m) \in W_I$ ,  $a \in A$  such that  $\tau_I(q, m, a) = (q', m')$  with the transition  $q \xrightarrow{a} q'$  non-accepting in  $\mathcal{A}_I$ . If the above transition is accepting in  $\mathcal{A}_I$ , let  $\bar{\tau}_I(q, m, a) = (q', \{\bar{q}'\})$  where  $\bar{q}'$  is the state given by Item 3 of Lemma 40.

**Lemma 41.** *The function  $\bar{\tau}_I$  is a GFG-witness for  $\mathcal{A}_I$ .*

*Proof.* It is enough to verify Item 2 of Definition 26. Since the action of  $\bar{\tau}_I$  differs from  $\tau_I$  only on accepting transitions of  $\mathcal{A}_I$ , this modification cannot lead to a non-accepting  $\bar{\tau}_I$ -run over  $\alpha \in A^\omega$ .  $\square$

We will now define the automaton  $\mathcal{B}$ . Let the set of states of  $\mathcal{B}$  be the subset of elements  $W_I$  accessible from  $(q_1, \{\bar{q}_1\})$  via  $\bar{\tau}_I$ . The transition function of  $\mathcal{B}$  is given by  $\bar{\tau}_I$ , the accepting transitions are those of the form  $(q, m) \xrightarrow{a} (q', m') = \bar{\tau}_I(q, m, a)$  with  $q \xrightarrow{a} q'$  an accepting transition of  $\mathcal{A}_I$ . This way  $\mathcal{B}$  is a deterministic Büchi automaton.

**Lemma 42.** *If  $(q, m)$  is a state of  $\mathcal{B}$  then  $m$  is a singleton.*

*Proof.* We will in fact prove that if  $(q, m)$  is a state of  $\mathcal{B}$  then  $m = \{q_1\}$  with  $q_1 \hookrightarrow q_1$ . It is enough to prove that  $\bar{\tau}_I$  preserves this property. Clearly the initial state of  $\mathcal{B}$  is of this form and the states obtained via accepting transitions of  $\mathcal{B}$  are of this form. Consider a transition  $(q, \{q_1\}) \xrightarrow{a} (q', m') = \bar{\tau}_I(q, \{q_1\}, a)$  of  $\mathcal{B}$  such that  $q \xrightarrow{a} q'$  is not an accepting transition of  $\mathcal{A}_I$ . By the assumption that  $q_1 \hookrightarrow q_1$  we know that  $\text{rank}_I(q_1, \{q_1\}) = 0$ . In the step performed in Section B.5 we have determinised all the transitions of the form  $q_1 \xrightarrow{a} q'_1$  with  $q_1 \hookrightarrow q_1$ . It means that  $m' = \delta^{\mathcal{D}_i}(\{q_1\}, a)$  is a singleton. Since  $q'_1 \in m'$ ,  $m' = \{q'_1\}$ . By Item 4 of Lemma 40 we know that in that case also  $q'_1 \hookrightarrow q'_1$  so the invariant is preserved.  $\square$

**Corollary 43.** *The number of states of  $\mathcal{B}$  is at most  $|Q^A|^2$ .*

**Lemma 44.** *The language  $L(\mathcal{B})$  is equal to the language  $L$ .*

*Proof.* Clearly  $L(\mathcal{B}) \subseteq L(\mathcal{A}_I) = L$  because of the definition of  $\mathcal{B}$ . It remains to prove that if  $\alpha \in L(\mathcal{A}_I)$  then  $\mathcal{B}$  accepts  $\alpha$ . But this follows directly from the fact that  $\bar{\tau}_I$  is a GFG-witness for  $\mathcal{A}_I$ .  $\square$

Therefore, we have completed the proof of Theorem 8 by constructing a deterministic Büchi automaton  $\mathcal{B}$  recognising  $L$  that has at most  $|Q^A|^2$  states. The construction presented in this section is effective but not efficient. In Section 5 we discuss how to determinise a Büchi GFG automaton efficiently.

### B.9 Example of the determinisation procedure

In this section we provide an example of the application of our determinisation procedure to a particular automaton  $\mathcal{A}$ . The automaton comes from [BKKS13] and it is GFG but not DBP (determinisable by pruning), i.e. it does not contain any equivalent deterministic subautomaton. The automaton  $\mathcal{A}$  is depicted on Figure 16. It recognises the language

$$L = [(xa + xb)^*(xaxa + xbxb)]^\omega.$$

The accepting transitions  $A'' \xrightarrow{a} I$  and  $B'' \xrightarrow{a} I$  are marked by dashed edges. For the sake of readability the sink state  $\perp$  and some of the transitions to it (i.e.  $I \xrightarrow{a} \perp$ ) are not presented.

The only non-determinism is in the state  $I$  when the letter  $x$  is read. In that case the automaton has to guess whether the successive letter will be  $a$  or  $b$ . One of the strategies for resolving this non-determinism is to move to  $A$  or  $B$  depending on the previous letter ( $a$  or  $b$ ). Correctness of this strategy relies on the fact that the language recognised by the automaton requires that infinitely many times the previous letter  $a$  or  $b$  reappears.

Figure 17 presents the automaton  $\mathcal{A}_0 = \mathcal{A}$  together with the data values  $M_0$  and a GFG-witness  $\tau_0$ . The red edges are transitions of  $\mathcal{A}_0$  while the black edges are transitions of  $\tau_0$ . Dashed red edges are accepting transitions of  $\mathcal{A}_0$  and dashed black edges correspond to the accepting transitions of  $\mathcal{D}_0$ . The numbers denote  $\text{rank}_0(q, m)$  and  $\text{opt}_0(q)$  respectively. Only the relevant pairs  $(q, m)$  are presented.

The only freedom in the choice of the strategy  $\tau_0$  is in the position  $(q, m) = (I, \{I\})$  when the letter  $x$  is played. The depicted strategy moves then to  $A$ . The choice that from the position  $(I, \{I, A'\})$  the strategy moves to  $A$  over  $x$  (and dually for  $b$ ) follows from the fact that the strategy is optimal with respect to the ranks (see Proposition 24).

In the first step of the construction the following steps are performed. The only modification of  $\mathcal{A}_0$  is done in the last step.

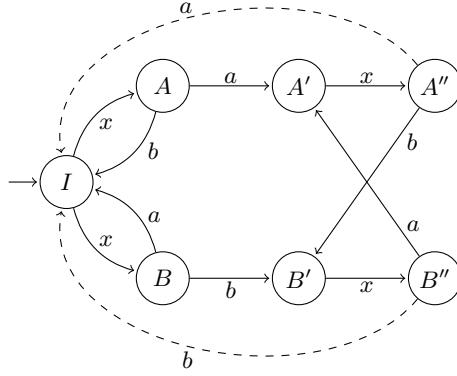
- all the transitions from the self-dependant states are already deterministic,
- all the accepting transitions are already deterministic,
- no non-accepting transition increases  $\text{opt}_0(q)$ ,
- the transitions  $A \xrightarrow{a} A'$  and  $B \xrightarrow{b} B'$  are made deterministic because they decrease  $\text{opt}_0(q)$ .

This way the automaton  $\mathcal{A}_1$  is obtained, see Figure 18. In this automaton all the states  $q$  satisfy  $\text{opt}_1(q) = 0$ . Further steps of the construction do not modify the automaton  $\mathcal{A}_1$  and therefore  $I = 1$ .

The following dependencies are important when constructing the deterministic automaton  $\mathcal{B}$ :

$$I \hookrightarrow A' \quad A' \hookrightarrow A' \quad I \hookrightarrow B' \quad B' \hookrightarrow B'.$$

The states  $A', B', A'', B''$  are self-dependant. There is a freedom in choosing  $\bar{I}$ : it can either be  $A'$  or  $B'$ , we assume that it is  $A'$ . Figure 19 presents the automaton  $\mathcal{B}$  obtained via our construction.



**Fig. 16.** The automaton  $\mathcal{A}$  — an example of a Büchi GFG automaton that is not DBP.

## C Reduction to parity games

In this section we prove Theorem 9.

**Theorem 9.** *Finding the winner of a parity game of index  $(i, j)$  is polynomially equivalent to deciding whether a given parity automaton of index  $(i, j)$  that accepts all  $\omega$ -words is GFG.*

*Proof.* First, if we are given a parity automaton of index  $(i, j)$  that accepts all  $\omega$ -words then deciding whether it is GFG is a parity game of index  $(i, j)$ :  $\forall$  chooses letters,  $\exists$  chooses transitions, and  $\exists$  wins if the resulting run is accepting over the given  $\omega$ -word. Therefore, solving this parity game allows us to decide whether the input automaton is GFG, provided this automaton accepts all  $\omega$ -words.

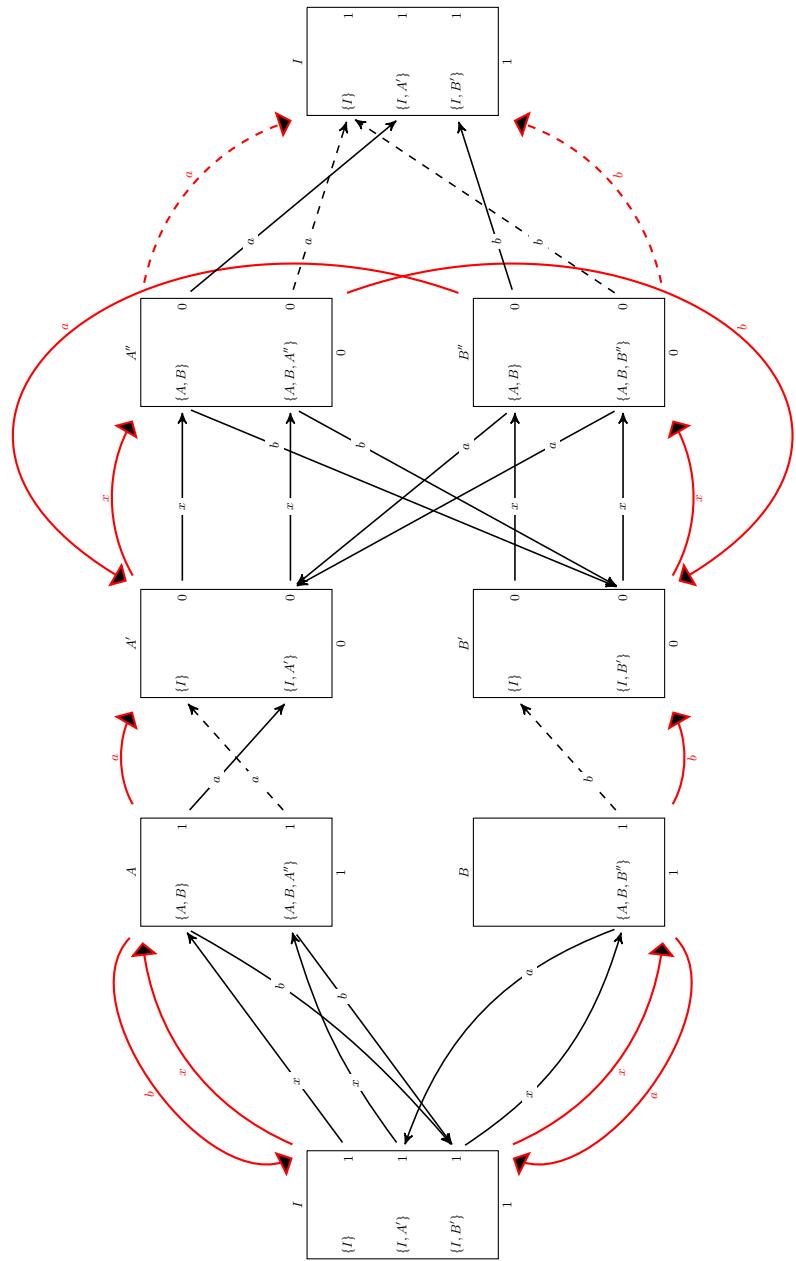
Conversely, let  $\mathcal{G} = \langle V_\exists, V_\forall, v_\mathbf{I}, \Gamma, \Omega^\mathcal{G} \rangle$  be a parity game of index  $(i_0, j_0)$ .

We build an automaton  $\mathcal{A} = \langle A, Q, q_\mathbf{I}, \Delta, \Omega^\mathcal{A} \rangle$  where the non-determinism corresponds to the choices of  $\exists$ , while the choices of  $\forall$  lie in the input alphabet. The set of states  $Q$  is  $V \sqcup \{\top\}$  and  $q_\mathbf{I} = v_\mathbf{I}$ . We define the alphabet  $A \stackrel{\text{def}}{=} \Gamma \cap V_\forall \times V$  — the edges that can be taken by  $\forall$  (we assume that there are at least two edges that can be taken by  $\forall$  in  $\mathcal{G}$ ). The state  $\top$  is a sink accepting state with only self-loops  $\top \xrightarrow{a} \top$  of even priority.

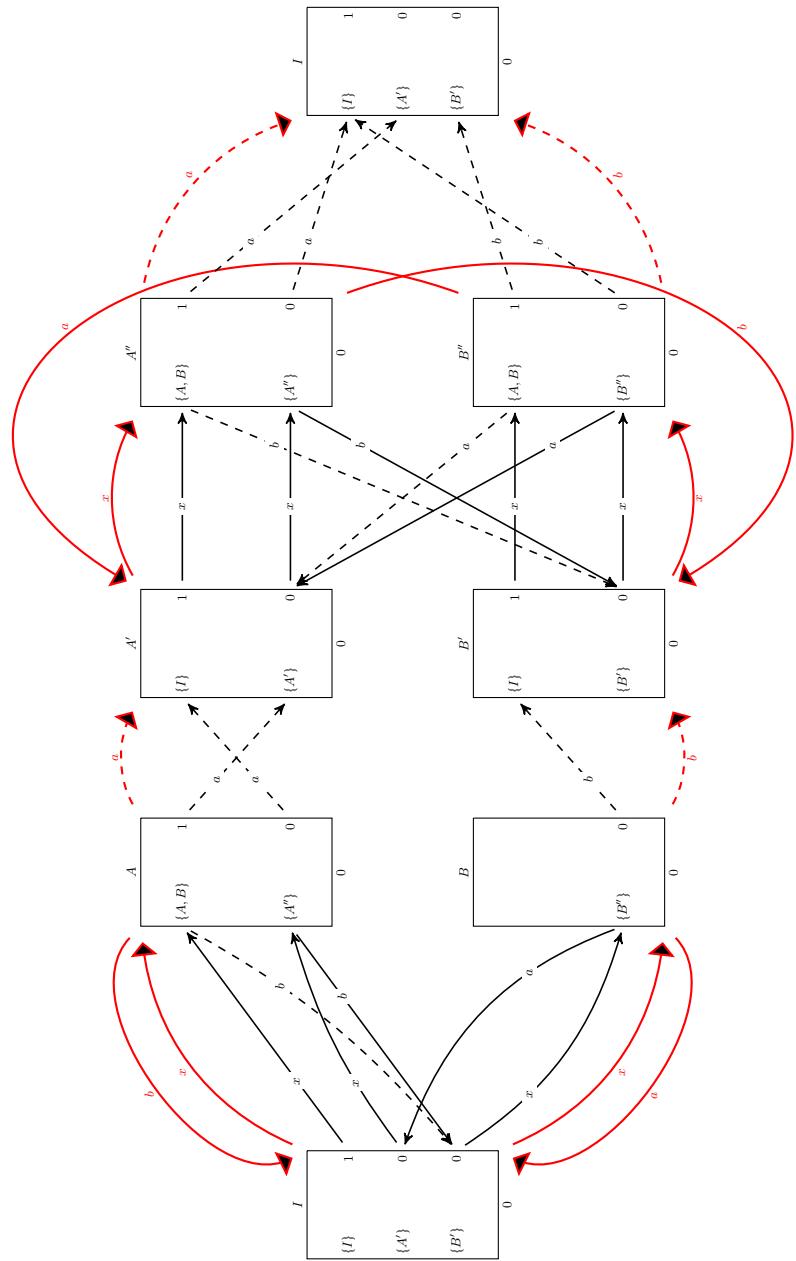
Finally, we define the set of transitions  $\Delta$  of  $\mathcal{A}$  for states other than  $\top$ :

- for  $v \in V_\forall$  and  $a = (v, v') \in A$  let  $\Delta$  contain the transition  $v \xrightarrow{a} v'$ ,
- for  $v \in V_\forall$  and  $a = (v', v'') \in A$  with  $v' \neq v$  let  $\Delta$  contain the transition  $v \xrightarrow{a} \top$ ,
- for  $v \in V_\exists$  and any  $a \in A$  let  $\Delta$  contain all the transitions  $v \xrightarrow{a} v'$  for  $(v, v') \in \Gamma$  (in that case  $a$  does not play any role).

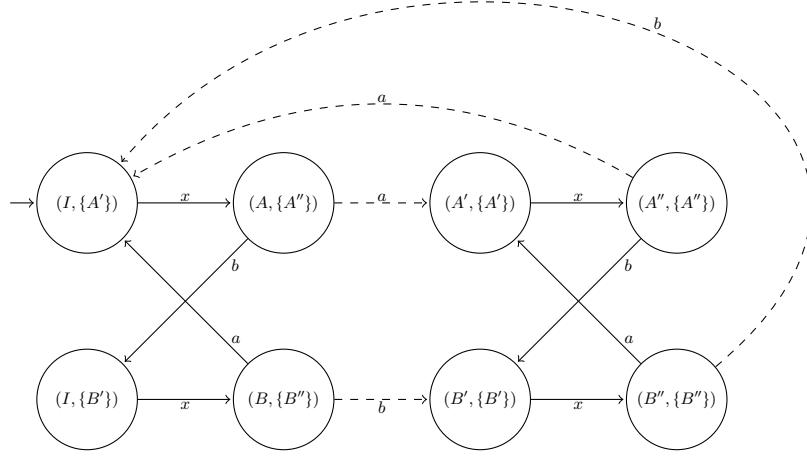
The priority  $\Omega^\mathcal{A}$  of a transition  $v \xrightarrow{a} v'$  with both  $v, v'$  different than  $\top$  is  $\Omega^\mathcal{G}(v, v')$ .



**Fig. 17.** The automaton  $\mathcal{A}_0$  together with  $\tau_0$ .



**Fig. 18.** The automaton  $\mathcal{A}_1$  together with  $\tau_1$ .



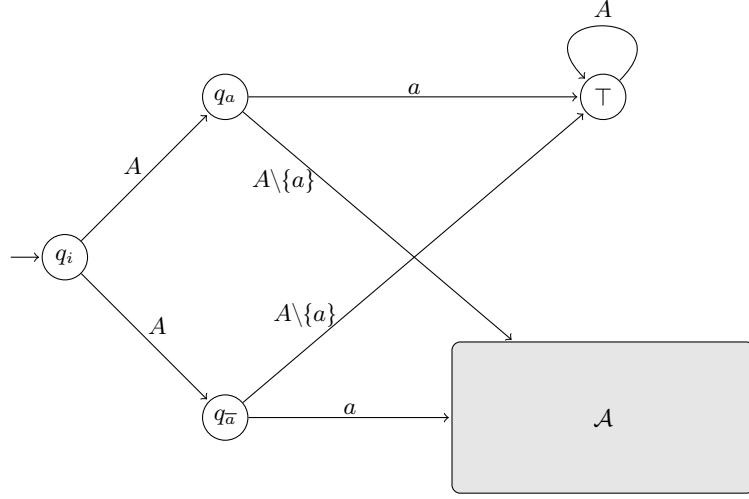
**Fig. 19.** The automaton  $\mathcal{B}$  — the polynomial determinisation of  $\mathcal{A}$ .

This way the games  $\mathcal{G}$  and  $\mathcal{G}(\mathcal{A})$  are equivalent — in positions  $v \in V_{\forall} \exists$  has no choice in  $\mathcal{G}(\mathcal{A})$  because there is a unique transition to take and in positions  $v \in V_{\exists}$  the letters chosen by  $\forall$  in  $\mathcal{G}(\mathcal{A})$  does not play any role. If  $\mathcal{A}$  accepts all  $\omega$ -words then the winning condition of  $\mathcal{G}(\mathcal{A})$  requires that the run proposed by  $\exists$  is accepting and therefore is equivalent to the winning condition of  $\mathcal{G}$ .

However, if  $\mathcal{A}$  does not accept all  $\omega$ -words then  $\exists$  may lose the game  $\mathcal{G}$  but the automaton  $\mathcal{A}$  might still be GFG for a smaller language. To avoid this, we need to build another automaton  $\mathcal{B}$  which accepts all  $\omega$ -words in any case but is GFG if and only if  $\exists$  wins  $\mathcal{G}$ .

We fix an arbitrary letter  $a \in A$  and build an automaton  $\mathcal{B} = \langle A, Q \sqcup \{q_i, q_a, q_{\bar{a}}\}, q_i, \Delta', \Omega' \rangle$ . We build  $\Delta'$  from  $\Delta$  by adding the following transitions:  $q_i \xrightarrow{A} q_a$ ,  $q_i \xrightarrow{A} q_{\bar{a}}$ ,  $q_a \xrightarrow{a} \top$ ,  $q_{\bar{a}} \xrightarrow{a} q_{\bar{A}}$ , and for all  $b \neq a$ ,  $q_{\bar{a}} \xrightarrow{b} \top$  and  $q_a \xrightarrow{b} q_{\bar{A}}$ .

First, it is clear that  $L(\mathcal{B}) = A^\omega$ : every  $\omega$ -word can be accepted by guessing if the second letter is  $a$ , and reaching the accepting sink state  $\top$ . We claim that  $\mathcal{B}$  is GFG if and only if  $\exists$  wins  $\mathcal{G}$ . Assume  $\mathcal{B}$  is GFG with a function  $\sigma: A^* \rightarrow Q^{\mathcal{B}}$  witnessing that. Then, there is a letter  $b \in A$  such that  $\sigma(ab) = q_{\bar{A}}$  (take  $b \neq a$  if  $\sigma(a) = q_a$  and  $b = a$  otherwise). Let  $\sigma': abw \mapsto \sigma(w)$  be the GFG strategy starting in  $q_{\bar{A}}$ . Since all  $\omega$ -words are in  $L(\mathcal{B})$ ,  $\sigma'$  must accept every  $\omega$ -word  $\alpha$ , and therefore corresponds to a winning strategy in  $\mathcal{G}$ . Conversely, every winning strategy in  $\mathcal{G}$  can be turned into a GFG strategy in  $\mathcal{B}$ , by adding an arbitrary choice at the beginning between  $q_a$  and  $q_{\bar{a}}$ .



**Fig. 20.** The construction of the automaton  $\mathcal{B}$  simulating a parity game  $\mathcal{G}$ . The sub-automaton  $\mathcal{A}$  corresponds to  $\mathcal{G}$  while the rest guarantees that  $L(\mathcal{B}) = A^\omega$ .

Finally, we showed that from a parity game  $\mathcal{G}$ , we can build an automaton  $\mathcal{B}$  with same parity index, such that  $\mathcal{G}$  is winning for  $\exists$  if and only if  $\mathcal{B}$  is GFG. Moreover, the language of  $\mathcal{B}$  is  $A^\omega$ .  $\square$

## D Recognising Büchi GFG automata

We give the detailed proof of the following theorem:

**Theorem 10.** *It is in NP to decide whether a given non-deterministic Büchi automaton  $\mathcal{A}$  is GFG. Moreover, if  $\mathcal{A}$  is GFG then we can construct an equivalent deterministic Büchi automaton in NP.*

*Proof.* Let  $\mathcal{A}$  be an input Büchi automaton and let  $Q$  be its set of states. By Theorem 8, if  $\mathcal{A}$  is GFG, it is witnessed by a strategy  $\bar{\tau}_I$  with memory  $M$  of size  $|Q^{\mathcal{A}}|$ . Therefore, we can guess such a structure  $M$ , and build the deterministic automaton  $\mathcal{B} = \mathcal{A} \times M$ , where runs of  $\mathcal{A}$  are guided by the current memory state in  $M$ . The acceptance component is inherited from  $\mathcal{A}$ , and therefore we have  $L(\mathcal{B}) \subseteq L(\mathcal{A})$ , since every accepting run of  $\mathcal{B}$  contains in particular an accepting run of  $\mathcal{A}$ . It remains for the algorithm to check  $L(\mathcal{A}) \subseteq L(\mathcal{B})$ , which can be done in polynomial time because  $\mathcal{B}$  is deterministic. More precisely, we can complement  $\mathcal{B}$  in  $\mathcal{B}^c$  and check for emptiness of  $L(\mathcal{A}) \cap L(\mathcal{B}^c)$ .

Conversely, if such a memory structure  $M$  is guessed by the non-deterministic algorithm, it is a witness that  $\mathcal{A}$  is GFG. Therefore, this non-deterministic polynomial algorithm is correct. Clearly if  $\mathcal{A}$  is GFG then a side-effect of a successful run of the algorithm is an equivalent deterministic automaton  $\mathcal{B}$ .  $\square$

## E Recognising co-Büchi GFG automata

In this section, we show that for the particular case of co-Büchi automata, the problem of deciding GFGness is in PTIME.

**Theorem 11.** *Given a non-deterministic co-Büchi automaton, we can decide whether it is GFG in polynomial time.*

Let  $\mathcal{A} = \langle A, Q, q_I, \Delta, R \rangle$  be a non-deterministic co-Büchi automaton.

### E.1 Normalised automata

The following notion will be important when we will consider the safety game  $\mathcal{G}^{\text{safe}}$ .

**Definition 45.** *A co-Büchi automaton is normalised if for any path  $p \xrightarrow{w} q$  without rejecting transition there is a path  $q \xrightarrow{w'} p$  without rejecting transition.*

**Lemma 46.** *Any co-Büchi automaton  $\mathcal{A}$  can be turned into an equivalent normalised automaton  $N(\mathcal{A})$  by changing some transitions to rejecting. This can be done in polynomial time.*

*For every  $\alpha$ , the sets of accepting runs of  $\mathcal{A}$  and  $N(\mathcal{A})$  over  $\alpha$  coincide. In particular  $L(\mathcal{A}, p) = L(N(\mathcal{A}), p)$  for every  $p \in Q$ .*

*Proof.* We start by computing the strongly-connected components (SCC) of the graph of the automaton without rejecting transitions. Then, all transitions that change SCC are switched to rejecting.

The accepting runs are preserved, because if an infinite path contains infinitely many new rejecting transitions, it also contains infinitely many original rejecting transitions. We get a normalised automaton, because the graph of the automaton restricted to non-rejecting transitions is a disjoint union of strongly-connected components.  $\square$

### E.2 Joker game

The crucial ingredient of the construction is the Joker Game, as defined below.

**Definition 47.** *The Joker Game on  $\mathcal{A}$  (denoted  $\mathcal{G}^{\text{Joker}}$ ,  $\mathcal{A}$  is fixed in this section) is defined on the set of positions  $Q \times Q$ . The initial position is  $(q_I, q_I)$ . The game is played in rounds  $n = 0, 1, \dots$ , in a round  $n$  starting in a position  $(p_n, q_n)$  the following actions are performed:*

- $\forall$  chooses a letter  $a_n \in A$ ,
- $\exists$  chooses a transition  $p_n \xrightarrow{a_n} p_{n+1}$  of  $\mathcal{A}$ ,
- $\forall$  either:
  - chooses a transition  $q_n \xrightarrow{a_n} q_{n+1}$  of  $\mathcal{A}$ ,
  - or plays JOKER and chooses a transition  $p_n \xrightarrow{a_n} q_{n+1}$  of  $\mathcal{A}$ .

After such a round the game moves to the position  $(p_{n+1}, q_{n+1})$ .

Now, the priority of an edge corresponding to a round as above is either:

- 2 if  $\forall$  played JOKER,
- otherwise 2 if the transition  $q_n \xrightarrow{a_n} q_{n+1}$  is rejecting in  $\mathcal{A}$ ,
- otherwise 1 if the transition  $p_n \xrightarrow{a_n} p_{n+1}$  is rejecting in  $\mathcal{A}$ ,
- otherwise 0.

An infinite play of the above game produces: an  $\omega$ -word  $\alpha = a_0 a_1 \dots$ , a run  $\rho = p_0 p_1 \dots$  of  $\mathcal{A}$ , and a *pseudo-run*  $\tau = q_0 q_1 \dots$  — each time  $\forall$  plays JOKER, the successive state  $q_{n+1}$  may not be accessible from  $q_n$  via a transition of  $\mathcal{A}$ . However, since the acceptance condition is prefix-independent, if  $\forall$  played only finitely many times JOKER then it makes sense to ask whether the pseudo-run  $\tau$  is accepting over  $\alpha$ .

Note that there are the following possibilities for the limes superior of the priorities of edges during this play:

- 0 and both  $\rho$  and  $\tau$  are accepting over  $\alpha$ ,
- 1 and the pseudo-run  $\tau$  is accepting over  $\alpha$  but  $\rho$  is not,
- 2 and either  $\forall$  played infinitely many times JOKER or  $\tau$  is not accepting over  $\alpha$ .

Therefore, we obtain the following fact.

**Fact 48.**  $\exists$  wins a play as above if and only if either:

- $\forall$  played JOKER infinitely many times,
- $\tau$  is not accepting over  $\alpha$ , or
- $\rho$  is accepting over  $\alpha$ .

A variant of the Joker Game with a bounded number of times  $\forall$  could play JOKER would be enough for our purposes. However, the unbounded variant presented above allows us to provide more elegant proofs.

**Lemma 49.** If the automaton  $\mathcal{A}$  is GFG then  $\exists$  wins  $\mathcal{G}^{\text{Joker}}$ .

*Proof.* Assume that  $\mathcal{A}$  is GFG and let use the function witnessing that  $\mathcal{A}$  is GFG as her strategy in the game  $\mathcal{G}^{\text{Joker}}$ , regardless of the current value  $q_n$ . We will prove that this strategy is winning. Assume contrarily and consider a play  $\alpha, \rho, \tau$  in which  $\forall$  won. Therefore, there exists  $n$  such that in the  $n$ th round  $\forall$  played JOKER for the last time (or he did not play JOKER at all and  $n = 0$ ). After that he managed to propose an accepting run of  $\mathcal{A}$  over  $\alpha' = (\alpha(n), \alpha(n+1), \dots)$  from the state  $\rho(n)$ . It means that  $\alpha' \in L(\mathcal{A}, \rho(n))$ . Since  $\rho(0) \xrightarrow{\alpha(0) \dots \alpha(n-1)} \rho(n)$ , we know that also  $\alpha \in L(\mathcal{A}, \rho(0)) = L(\mathcal{A})$ . Therefore, the run constructed by the function witnessing that  $\mathcal{A}$  is GFG has to be accepting in  $\mathcal{A}$ . So  $\exists$  won this play.  $\square$

Therefore, we can start our procedure by verifying (in polynomial time) if  $\exists$  wins  $\mathcal{G}^{\text{Joker}}$ . If she loses then  $\mathcal{A}$  is not GFG. From that point on we assume that  $\exists$  wins this game and  $W^J \subseteq Q \times Q$  is her winning region. Let  $Q^J$  be the projection of  $W^J$  onto the first coordinate — the set of states  $p$  such that  $\exists$  can win from some position of the form  $(p, q)$ . The winning condition of  $\mathcal{G}^{\text{Joker}}$  is a parity condition so we can fix a uniform positional winning strategy of  $\exists$  of the form

$$\sigma_{\exists}^J: W^J \times A \rightarrow Q^J \quad (6)$$

We say that a transition  $p \xrightarrow{a} p'$  is *used by*  $\sigma_{\exists}^J$  if there exists a position  $(p, q) \in W^J$  such that  $\sigma_{\exists}^J(p, q, a) = p'$ . Let  $\mathcal{A}^J$  be the non-deterministic co-Büchi automaton obtained by restricting  $\mathcal{A}$  to the set of states  $Q^J$  and transitions used by  $\sigma_{\exists}^J$ , and by normalising the resulting automaton using Lemma 46. Notice that normalising does not change the winning region, as the accepting runs are unchanged. The strategy  $\sigma_{\exists}^J$  witnesses that if  $p \in Q^J$  and  $\alpha \in L(\mathcal{A}, p)$  then also  $\alpha \in L(\mathcal{A}^J, p)$  (in particular,  $L(\mathcal{A}^J) = L(\mathcal{A})$ ).

### E.3 Structure of $\mathcal{A}^J$

Now we will study the structure of the automaton  $\mathcal{A}^J$ .

**Lemma 50.** *If  $p \in Q^J$  then  $(p, p) \in W^J$ .*

*Proof.* Let  $(p, q) \in W^J$  be a witness that  $p$  is used in  $\sigma_{\exists}^J$ . It is enough to observe that every strategy of  $\forall$  from  $(p, p)$  can be also used by  $\forall$  from  $(p, q)$  at the cost of playing JOKER at the first round of the play. Therefore, if  $\exists$  wins from  $(p, q)$  then she wins from  $(p, p)$  as well.  $\square$

**Lemma 51.** *If  $p \xrightarrow{a} p'$  is a transition of  $\mathcal{A}^J$  then  $L(\mathcal{A}, p') = a^{-1}L(\mathcal{A}, p)$ .*

*Proof.* The  $\subseteq$  containment is trivial. For the  $\supseteq$  consider an  $\omega$ -word  $a\alpha \in L(\mathcal{A}, p)$ . We need to prove that  $\alpha \in L(\mathcal{A}, p')$ . Let  $\tau$  be an accepting run of  $\mathcal{A}$  over  $a\alpha$  from  $p$ . Assume that the transition  $p \xrightarrow{a} p'$  is used by  $\sigma_{\exists}^J$  from a position  $(p, q) \in W^J$ .

Consider the play of  $\mathcal{G}^{\text{Joker}}$  from  $(p, q)$  in which  $\exists$  plays according to  $\sigma_{\exists}^J$  and  $\forall$  plays: in the first round the letter  $a$  and then JOKER to use the transition  $p = \tau(0) \xrightarrow{a} \tau(1)$ ; and in further rounds successive letters of  $\alpha$  and successive transitions of  $\tau$ . Since  $\sigma_{\exists}^J$  is winning and the run  $\tau$  proposed by  $\forall$  is accepting over  $a\alpha$ , also the run  $\rho$  constructed by  $\exists$  has to be accepting over  $a\alpha$ . But the first transition of this run is  $p \xrightarrow{a} p'$ , therefore the run witnesses that  $\alpha \in L(\mathcal{A}, p')$ .  $\square$

Similarly as in Section B.1 we can obtain a variant of Corollary 18 for  $\mathcal{A}^J$ .

### E.4 Equivalence relation

We define an equivalence relation  $E \subseteq Q^J \times Q^J$  that will keep track of states that are accessible in  $\mathcal{A}^J$  via the same word from a common state.

**Definition 52.** Let  $E \subseteq Q^J \times Q^J$  be the smallest equivalence relation on  $Q^J$  such that for any  $(p, q) \in E$  and  $a \in A$ , if there are transitions  $p \xrightarrow{a} p'$  and  $q \xrightarrow{a} q'$  in  $\mathcal{A}^J$  then  $(p', q') \in E$ .

The relation  $E$  can be computed in polynomial time via a standard saturation algorithm.

*Remark 2.* By Lemma 51, for all  $(p, q) \in E$  we have  $L(\mathcal{A}, p) = L(\mathcal{A}, q)$ .

### E.5 Safety game

We will now consider a variant  $\mathcal{G}^{\text{safe}}$  of the game  $\mathcal{G}(\mathcal{A}^J, \mathcal{A}^J)$  where the first rejecting transition is losing for the respective player. Let the set of positions of  $\mathcal{G}^{\text{safe}}$  be  $E \subseteq Q^J \times Q^J$ . In such a position  $(p, q)$  the following choices are done:

- first  $\forall$  proposes a letter  $a \in A$ ,
- then  $\exists$  proposes a transition  $p \xrightarrow{a} p'$  of  $\mathcal{A}^J$ ,
- then  $\forall$  proposes a transition  $q \xrightarrow{a} q'$  of  $\mathcal{A}^J$ .

If the transition  $p \xrightarrow{a} p'$  (resp.  $q \xrightarrow{a} q'$ ) is rejecting in  $\mathcal{A}^J$  then  $\exists$  (resp.  $\forall$ ) immediately loses (if both transitions are rejecting then  $\forall$  immediately loses). All the infinite plays are won by  $\exists$ .

Let  $W^S \subseteq E$  be the winning region of  $\exists$  and let  $\sigma_{\exists}^S: W^S \times A \rightarrow Q^J$  be a uniform positional winning strategy of  $\exists$  in her winning region of  $\mathcal{G}^{\text{safe}}$ .

For each  $p \in Q$ , we now define the language  $L^{\text{safe}}(\mathcal{A}^J, p)$  as the set of  $\omega$ -words  $\alpha$  such that there exists a run of  $\mathcal{A}^J$  from  $p$  over  $\alpha$  that does not contain any rejecting transition of  $\mathcal{A}^J$ .

The winning region  $W^S$  provides us a dependency relation on  $Q^J$ . Let us write  $p \rightarrow q$  if  $(p, q) \in W^S$ . This relation will be essential in the rest of the construction.

**Lemma 53.** For every  $q \in Q^J$  there exists  $p \in Q^J$  such that  $p \rightarrow q$ .

*Proof.* Assume contrarily that there exists  $q \in Q^J$  such that for no  $p \in Q^J$  we have  $p \rightarrow q$ .

We will inductively construct a play of  $\mathcal{G}^{\text{Joker}}$  from  $(q, q)$ . Let us start with  $p_0 = q$ . The invariant is that  $(p_n, q) \in Q^J \times Q^J$  belongs to  $W^J \cap E$ . Lemma 50 implies that for  $n = 0$  the invariant holds. Assume that  $n$  steps of the construction have been done and a state  $p_n$  is defined. Since  $p_n \rightarrow q$  does not hold,  $\forall$  has a winning strategy  $\sigma_{\forall}^S$  in  $\mathcal{G}^{\text{safe}}$  from  $(p_n, q)$ . Consider the play of  $\mathcal{G}^{\text{safe}}$  resulting from  $\forall$  playing  $\sigma_{\forall}^S$  and  $\exists$  using her strategy  $\sigma_{\exists}^J$  from  $\mathcal{G}^{\text{Joker}}$ .

Since  $\sigma_{\forall}^S$  is winning, after a finite word  $w_n$  has been played the two constructed runs are  $p_n \xrightarrow{w_n} p'_n$  and  $q \xrightarrow{w_n} q'_n$  with the first path containing a rejecting transition of  $\mathcal{A}^J$  and the second one not containing any rejecting transition of  $\mathcal{A}^J$ . Since the automaton  $\mathcal{A}^J$  is normalised, we know that there exists a path  $q'_n \xrightarrow{w'_n} q$  without any rejecting transition. Let  $\forall$  proceed along this path and  $\exists$  play using  $\sigma_{\exists}^J: p'_n \xrightarrow{w'_n} p_{n+1}$ . After this finite play we reached the position

$(p_{n+1}, q)$  and the invariant that  $(p_{n+1}, q) \in W^J \cap E$  holds because we simulated the winning strategy  $\sigma_{\exists}^J$  of  $\exists$ , and only took transitions of  $\mathcal{A}^J$ .

After infinitely many steps of the above construction we obtain a play of  $\mathcal{G}^{\text{Joker}}$  that is consistent with  $\sigma_{\exists}^J$ . In this play  $\forall$  never plays JOKER, the constructed  $\omega$ -word  $\alpha$  is  $w_0 w'_0 w_1 w'_1 \dots$ , the pseudo-run  $\tau$  over  $\alpha$  does not contain any rejecting transition of  $\mathcal{A}^J$  while the run  $\rho$  over  $\alpha$  contains infinitely many rejecting transitions of  $\mathcal{A}^J$ . Therefore,  $\tau$  is an accepting run over  $\alpha$  of  $\mathcal{A}$  as well and  $\rho$  is not accepting. Therefore, we obtain a contradiction with the fact that  $\sigma_{\exists}^J$  is winning.  $\square$

**Lemma 54.** *If  $p \rightarrow q$  and  $q \rightarrow r$  then  $p \rightarrow r$ .*

*Proof.* We need to provide a winning strategy of  $\exists$  in  $\mathcal{G}^{\text{safe}}$  from the position  $(p, r)$ . Since the winning condition of  $\mathcal{G}^{\text{safe}}$  is safety, it is enough to show how  $\exists$  can survive one round. The invariant is that  $(p, q) \in W^S$ ,  $(q, r) \in W^S$ , and the current position is  $(p, r)$ . Assume that  $\forall$  plays a letter  $a \in A$ . We will simulate two plays of  $\mathcal{G}^{\text{safe}}$  from  $(p, q)$  and  $(q, r)$ . Let the strategy  $\sigma_{\exists}^S$  move from  $(p, q)$  over  $a$  to  $p'$  and from  $(q, r)$  over  $a$  to  $q'$ . Let  $\exists$  play from  $(p, r)$  the state  $p'$  and assume that  $\forall$  replied by  $r'$ .

Since  $q \xrightarrow{a} q'$  so  $q'$  is a valid reply of  $\forall$  in the play from  $(p, q)$ . Similarly,  $r \xrightarrow{a} r'$  so  $r'$  is a valid reply in the play from  $(q, r)$ . The only possibility for  $\exists$  to lose in this round would be if the transition  $r \xrightarrow{a} r'$  was non-rejecting in  $\mathcal{A}^J$  but the transition  $p \xrightarrow{a} p'$  was rejecting in  $\mathcal{A}^J$ . But in that case, the transition  $q \xrightarrow{a} q'$  cannot be rejecting in  $\mathcal{A}^J$  (otherwise  $(q, r) \notin W^S$ ) and therefore neither  $p \xrightarrow{a} p'$  can be rejecting in  $\mathcal{A}^J$  (otherwise  $(p, q) \notin W^S$ ). Therefore,  $\exists$  did not lose in this round,  $(p', q') \in W^S$  and  $(q', r') \in W^S$  so the invariant holds.  $\square$

The following corollary follows directly from the two above lemmas and the fact that  $Q^J$  is finite.

**Corollary 55.** *For every  $q \in Q^J$  there exists  $p \in Q^J$  such that  $p \rightarrow p$  and  $p \rightarrow q$ .*

**Lemma 56.** *If  $p \rightarrow q$  then  $L^{\text{safe}}(\mathcal{A}^J, p) \supseteq L^{\text{safe}}(\mathcal{A}^J, q)$ .*

*Proof.* Consider  $\alpha \in L^{\text{safe}}(\mathcal{A}^J, q)$  and a play of  $\mathcal{G}^{\text{safe}}$  from  $(p, q)$  where  $\exists$  plays according to her winning strategy and  $\forall$  plays  $\alpha$  together with a run  $\tau$  of  $\mathcal{A}^J$  from  $q$  over  $\alpha$  that does not contain any rejecting transition. Since  $\exists$  plays according to her winning strategy, we know that the run  $\rho$  over  $\alpha$  proposed by  $\exists$  neither contains a rejecting transition. Therefore,  $\alpha \in L^{\text{safe}}(\mathcal{A}^J, p)$ .  $\square$

## E.6 Deterministic part

Now we will focus on states  $p \in Q^J$  such that  $p \rightarrow p$ . Similarly as in Section B.5 we will define a structure of a deterministic automaton  $\mathcal{D}$  on this set of states. The automaton will be a *safety* automaton — this time we allow its transition function to be partial, and we consider the run non-accepting if it is finite and

accepting if it is infinite. Notice that we could equivalently define  $\mathcal{D}$  as a co-Büchi automaton by adding a sink state with rejecting self-loop, and all other transitions non-rejecting.

We will use  $D \stackrel{\text{def}}{=} \{p \in Q^J \mid p \rightarrow p\}$  as the set of states of  $\mathcal{D}$ .

**Lemma 57.** *If  $p \in D$  and  $\sigma_{\exists}^S(p, p, a) = p'$  with the transition  $p \xrightarrow{a} p'$  not rejecting in  $\mathcal{A}^J$  then  $p' \in D$  (i.e.  $p' \rightarrow p'$ ).*

*Proof.* If  $\forall$  plays  $a$  in the position  $(p, p)$  of  $\mathcal{G}^{\text{safe}}$  then the reply of  $\sigma_{\exists}^S$  is  $p'$  as above. This round can be finished by  $\forall$  playing  $p'$  as well and reaching the position  $(p', p')$ . If the transition  $p \xrightarrow{a} p'$  is non-rejecting in  $\mathcal{A}^J$  then this round does not finish the game and therefore  $\exists$  is able to win from  $(p', p')$  (i.e.  $(p', p') \in W^S$  and therefore  $p \rightarrow p'$ ).  $\square$

Now we define the transitions of the automaton  $\mathcal{D}$ : let  $(p, a) \in D \times A$ , we define  $\delta^{\mathcal{D}}(p, a) \stackrel{\text{def}}{=} \sigma_{\exists}^S(p, p, a)$  if  $p \xrightarrow{a} \sigma_{\exists}^S(p, p, a)$  is non-rejecting, and  $\delta^{\mathcal{D}}(p, a)$  undefined (noted  $\perp$ ) otherwise.

This deterministic automaton  $\mathcal{D}$  has safety condition, meaning a run is accepting if and only it is infinite. The initial state  $q_{\mathcal{I}}^{\mathcal{D}}$  of  $\mathcal{D}$  is any state  $p$  such that  $p \rightarrow q_{\mathcal{I}}^{\mathcal{A}}$  (such a state exists by Lemma 53).

**Lemma 58.** *For all  $p \in D$  we have  $L^{\text{safe}}(\mathcal{A}^J, p) = L(\mathcal{D}, p)$ .*

*Proof.* Clearly the  $\supseteq$  containment is trivial — an accepting run of  $\mathcal{D}$  from  $p$  over  $\alpha$  is a run of  $\mathcal{A}^J$  from  $p$  over  $\alpha$  that does not contain any rejecting transition. It remains to prove that if  $\alpha \in L^{\text{safe}}(\mathcal{A}^J, p)$  then  $\alpha \in L(\mathcal{D}, p)$ . The proof is inductive proving that  $\mathcal{D}$  does not get blocked when reading  $\alpha$  from  $p$ . The invariant is that  $\mathcal{D}$  is in a state  $p \in D$  and  $\alpha \in L^{\text{safe}}(\mathcal{A}^J, p)$ .

Let  $\alpha = a\alpha'$  and  $\rho$  be a run of  $\mathcal{A}^J$  from  $p$  over  $\alpha$  witnessing that  $\alpha \in L^{\text{safe}}(\mathcal{A}^J, p)$ . Now let  $p' = \delta^{\mathcal{D}}(p, a)$  and  $q' = \rho(1)$ . Since  $\rho'$  does not contain any rejecting transition over  $a\alpha$ , we know that the transition  $p \xrightarrow{a} q'$  is a non-rejecting transition of  $\mathcal{A}^J$ . Since  $p \rightarrow p$ , the transition  $p \xrightarrow{a} p'$  cannot be rejecting, otherwise  $\exists$  would lose in the position  $(p, p)$  of  $\mathcal{G}^{\text{safe}}$ . What remains is to prove that  $\alpha' \in L^{\text{safe}}(\mathcal{A}^J, p')$ . But  $\alpha' \in L^{\text{safe}}(\mathcal{A}^J, q')$  and since  $(p, p) \in W^S$ , also  $(p', q') \in W^S$  (i.e.  $p' \rightarrow q'$ ) as  $\forall$  can lead the strategy of  $\exists$  from  $(p, p)$  to  $(p', q')$ . Therefore, by Lemma 56 we know that  $\alpha' \in L(\mathcal{A}^J, p')$ . Thus, the invariant holds.  $\square$

## E.7 Building a GFG automaton

We will now build a co-Büchi GFG automaton  $\mathcal{B}$  on top of  $\mathcal{D}$ , recognising  $L(\mathcal{A})$ . The set of states of  $\mathcal{B}$  is  $D$  and the initial state is  $q_{\mathcal{I}}^{\mathcal{D}}$ . Consider a state  $p \in D$  and a letter  $a \in A$ . For each transition  $p \xrightarrow{a} q'$  of  $\mathcal{A}^J$  and each  $(p', q') \in D \times Q^J \cap E$  we have a transition  $p \xrightarrow{a} p'$  in  $\Delta^{\mathcal{B}}$ . Moreover, if  $\delta^{\mathcal{D}}(p, a) = p'$  (in particular it is defined) then the transition is non-rejecting, otherwise it is rejecting. Note that if  $\delta^{\mathcal{D}}(p, a) = p'$  is defined then by the definition of  $E$  there is a (non-rejecting) transition  $p \xrightarrow{a} p'$  in  $\mathcal{B}$ .

Intuitively, the automaton  $\mathcal{B}$  follows  $\mathcal{D}$  as long as possible, and at any time it can jump to any  $E$ -equivalent state via a rejecting transition.

By the definition of  $\mathcal{B}$  we obtain the following fact.

**Fact 59.** *For every  $p \in D$  we have  $L^{\text{safe}}(\mathcal{B}, p) = L(\mathcal{D}, p)$ .*

**Lemma 60.** *The following relations between paths in  $\mathcal{A}^J$  and  $\mathcal{B}$  hold.*

1. *Assume that  $(p, q) \in D \times Q^J \cap E$  and there is a path  $q \xrightarrow{u} q'$  in  $\mathcal{A}^J$  with  $u \neq \epsilon$  then*

$$\{p' \in D \mid p \xrightarrow{u} p' \text{ is a path of } \mathcal{B}\} = \{p' \in D \mid (p', q') \in E\} \neq \emptyset.$$

2. *Conversely, if  $(p, q) \in D \times Q^J \cap E$  and there is a path  $p \xrightarrow{u} p'$  in  $\mathcal{B}$  then there exists  $q' \in Q^J$  and a path  $q \xrightarrow{u} q'$  in  $\mathcal{A}^J$  such that  $(p', q') \in E$ .*

*In particular, the above properties hold for  $p = q_{\mathbf{I}}^{\mathcal{B}}$  and  $q = q_{\mathbf{I}}^{\mathcal{A}}$  because  $(q_{\mathbf{I}}^{\mathcal{B}}, q_{\mathbf{I}}^{\mathcal{A}}) \in E$ .*

*Proof.* Both items are proved by induction on  $u$ . In Item 2 it is important that  $\mathcal{A}^J$  can always perform at least one transition from a given state  $q \in Q^J$  over a letter  $a \in A$ .  $\square$

**Lemma 61.**  $L(\mathcal{B}) = L(\mathcal{A})$ .

*Proof.* Let  $\alpha \in L(\mathcal{A}) = L(\mathcal{A}^J)$ . We know that there exists a decomposition  $\alpha = ua'$  and a path  $q_{\mathbf{I}}^{\mathcal{A}} \xrightarrow{u} q'$  in  $\mathcal{A}^J$  such that  $\alpha' \in L^{\text{safe}}(\mathcal{A}^J, q')$ . Let  $p' \in D$  such that  $p' \rightarrow q'$  (in particular  $(p', q') \in E$ ). By Lemma 56 we have  $\alpha' \in L^{\text{safe}}(\mathcal{A}^J, p')$ . Therefore, by Lemma 58 we know that  $\alpha' \in L(\mathcal{D}, p')$  and by Fact 59 also  $\alpha' \in L^{\text{safe}}(\mathcal{B}, p')$ . Item 1 of Lemma 60 implies that there exists a path  $q_{\mathbf{I}}^{\mathcal{B}} \xrightarrow{u} p'$  in  $\mathcal{B}$ . This path together with the fact that  $\alpha' \in L^{\text{safe}}(\mathcal{B}, p')$  provides an accepting run of  $\mathcal{B}$  over  $ua' = \alpha$ .

Conversely, let  $\alpha \in L(\mathcal{B})$ . Since the acceptance condition of  $\mathcal{B}$  is co-Büchi, we know that there exists a state  $p' \in D$  and a decomposition  $\alpha = ua'$  such that  $q_{\mathbf{I}}^{\mathcal{B}} \xrightarrow{u} p'$  and  $\alpha' \in L^{\text{safe}}(\mathcal{B}, p')$ . In particular  $\alpha' \in L^{\text{safe}}(\mathcal{A}^J, p') \subseteq L(\mathcal{A}^J, p')$ . Item 2 of Lemma 60 implies that there exists a path  $q_{\mathbf{I}}^{\mathcal{A}} \xrightarrow{u} q'$  in  $\mathcal{A}^J$  with  $q' \in Q^J$  and  $(p', q') \in E$ . Since  $(p', q') \in E$ , we know that  $\alpha' \in L(\mathcal{A}^J, q')$ . Therefore, we can find an accepting run of  $\mathcal{A}^J$  over  $ua' = \alpha$  that starts with the path  $q_{\mathbf{I}}^{\mathcal{A}} \xrightarrow{u} q'$ . This run is an accepting run of  $\mathcal{A}^J$  over  $\alpha$  and therefore  $\alpha \in L(\mathcal{A})$ .  $\square$

## E.8 $\mathcal{B}$ is GFG

We now define a strategy  $\sigma^{\text{GFG}}$  in the game  $\mathcal{G}(\mathcal{B})$  proving that  $\mathcal{B}$  is in fact GFG. Intuitively, the strategy will try the non-rejecting paths in  $\mathcal{B}$  (we can deterministically follow these paths as in  $\mathcal{D}$ ) one after another. Consider the memory structure  $M = D^{\leq |D|} \setminus \{\epsilon\}$  — finite non-empty sequences of states from  $D$ . The invariant of our strategy is that if we are in a position  $p$  of  $\mathcal{G}(\mathcal{B})$  and our memory state is  $m = d_0 \dots d_\ell$  then:

- $p = d_0$ ,
- each state appears at most once in  $m$ ,
- for all  $i, j \leq \ell$  we have  $(d_i, d_j) \in E$ .

The initial memory value is  $m_0 = \{q_I^{\mathcal{B}}\}$ . The memory states  $m$  can be seen as a simplification of the structure of Last-Appearance-Record from [Bü83].

Consider a position  $p$  in  $\mathcal{G}(\mathcal{B})$  when the memory state is  $m = d_0 d_1 \dots d_\ell \in M$  and  $\forall$  plays a letter  $a \in A$ . For each  $i = 0, \dots, \ell$  let us define  $d'_i$  as:

- $\epsilon$  if  $\delta^{\mathcal{D}}(d_i, a) = \perp$ ,
- $\epsilon$  if  $\delta^{\mathcal{D}}(d_i, a)$  is among  $d'_j$  for  $j < i$ ,
- $\delta^{\mathcal{D}}(d_i, a)$  otherwise.

Let  $p' = \sigma_{\exists}^S(p, p, a)$  and let  $r \in D^*$  be a list of all states  $q \in D$  such that  $(p', q) \in E$  and  $q$  is not any of  $d'_i$ . Now  $m' = d'_0 \dots d'_\ell \cdot r$  (clearly  $p'$  appears in this list, either as  $d'_0$  or later if  $d'_0 = \epsilon$ ). By the construction, each state in  $m'$  appears at most once and all of them are  $E$ -equivalent. Now let  $\exists$  move in such a case to the position  $m'(0)$  and set the memory state to  $m'$ . By Lemma 60 and the fact that all  $d'_i$  are  $E$ -equivalent we know that there exists a transition  $p \xrightarrow{a} m'(0)$  of  $\mathcal{B}$ .

Note that the transition taken by  $\exists$  according to the above strategy may be rejecting in  $\mathcal{B}$  if  $\delta^{\mathcal{D}}(d_0, a) = \perp$ . Clearly if  $d$  appears in  $m$  at a position  $i$  and  $\delta^{\mathcal{D}}(d, a) = d' \neq \perp$  then  $d'$  appears in  $m'$  at a position  $i'$  such that  $i \geq i'$ . If the transition played by  $\exists$  in the given round is rejecting in  $\mathcal{B}$  then  $i > i'$ .

**Lemma 62.** *The strategy  $\sigma^{\text{GFG}}$  is a winning strategy in  $\mathcal{G}(\mathcal{B})$ .*

*Proof.* Consider a play in which  $\forall$  proposed an  $\omega$ -word  $\alpha$  and  $\exists$  produced an infinite run  $\rho$  of  $\mathcal{B}$  over  $\alpha$ . Assume that  $\alpha \in L(\mathcal{B})$ , in particular  $\alpha = u\alpha'$  with  $q_I^{\mathcal{B}} \xrightarrow{u} p$  a path in  $\mathcal{B}$  and  $\alpha' \in L^{\text{safe}}(\mathcal{B}, p)$ . Let  $\rho$  be a run of  $\mathcal{B}$  from  $p$  over  $\alpha$  witnessing that  $\alpha' \in L^{\text{safe}}(\mathcal{B}, p)$  —  $\rho$  does not contain any rejecting transition of  $\mathcal{B}$ . Therefore,  $\rho$  is an accepting run of  $\mathcal{D}$  from  $p$  over  $\alpha$ .

For  $n = 0, \dots$  let  $m_n$  be the memory state of the above strategy of  $\exists$  after  $\forall$  has played  $u\alpha(0) \cdot \alpha(n-1)$ . By Lemma 60, we know that the state  $p = \rho(0)$  appears in  $m_0$  at some position  $i_0$ . By the definition of  $\sigma^{\text{GFG}}$  it will always be there — for every  $n$  the state  $\rho(n)$  appears in  $m_n$  at a position  $i_n$ . As we observed above, we know that  $i_n \geq i_{n+1}$  and whenever  $\exists$  plays a rejecting transition of  $\mathcal{B}$  then  $i_n > i_{n+1}$ . Therefore, there can be at most  $i_0$  such transitions played after  $\forall$  has played the word  $u$ .  $\square$

## E.9 Deciding GFGness of $\mathcal{A}$

Putting together the results from this section, we get the following theorem:

**Theorem 63.** *Given a co-Büchi automaton  $\mathcal{A}$ , there is a polynomial time algorithm that either*

- answers “ $\mathcal{A}$  is not GFG” (if  $\forall$  wins the Joker game  $\mathcal{G}^{\text{Joker}}$ )

- builds a co-Büchi GFG automaton  $\mathcal{B}$  in polynomial time, of the same size and language as  $\mathcal{A}$ .

Now Theorem 11 follows by applying Theorem 14 to the automata  $\mathcal{A}$  and  $\mathcal{B}$  in the second case above.

## References

- BKKS13. Udi Boker, Denis Kuperberg, Orna Kupferman, and Michał Skrzypczak. Nondeterminism in the presence of a diverse or unknown future. In *ICALP (2)*, pages 89–100, 2013.
- BL69. Julius Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969.
- Büc83. Julius Richard Büchi. State-strategies for games in F-sigma-delta intersected G-delta-sigma. *J. Symb. Log.*, 48(4):1171–1198, 1983.
- Col09. Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *ICALP (2)*, volume 5556 of *LNCS*, pages 139–150, 2009.
- Col12. Thomas Colcombet. Forms of determinism for automata (invited talk). In *STACS*, pages 1–23, 2012.
- EJ91. Allen Emerson and Charanjit Jutla. Tree automata, mu-calculus and determinacy. In *FOCS’91*, pages 368–377, 1991.
- HP06. Thomas A. Henzinger and Nir Piterman. Solving games without determinization. In *CSL*, LNCS, pages 395–410. Springer, 2006.
- KMBK14. Joachim Klein, David Müller, Christel Baier, and Sascha Klüppelholz. Are good-for-games automata good for probabilistic model checking? In *LATA, Madrid, Spain, March 10-14, 2014. Proceedings*, pages 453–465, 2014.
- McN66. Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966.
- Mos91. Andrzej W. Mostowski. Games with forbidden positions. Technical report, University of Gdańsk, 1991.
- PP06. Nir Piterman and Amir Pnueli. Faster solutions of rabin and streett games. In *Logic in Computer Science (LICS)*, pages 275–284, 2006.
- Rab72. Michael Oser Rabin. *Automata on Infinite Objects and Church’s Problem*. American Mathematical Society, Boston, MA, USA, 1972.
- SE89. Robert S. Streett and E. Allen Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81(3):249–264, 1989.
- Wal02. Igor Walukiewicz. Monadic second-order logic on tree-like structures. *Theoretical Computer Science*, 275(1–2):311–346, 2002.