Logical Methods in Computer Science
Vol. 4 (1:?) 2008, pp. 1–1–20
www.lmcs-online.org

Submitted Jul. 25, 2007
Published Mar. ??, 2008

LOWER BOUNDS FOR COMPLEMENTATION OF ω-AUTOMATA VIA THE FULL AUTOMATA TECHNIQUE*

QIQI YAN

Department of Computer Science and Engineering, Shanghai Jiao Tong University, 200240, Shanghai, P.R. China
e-mail address: contact@qiqiyan.com

Abstract. In this paper, we first introduce a lower bound technique for the state complexity of transformations of automata. Namely we suggest first considering the class of full automata in lower bound analysis, and later reducing the size of the large alphabet via alphabet substitutions. Then we apply such technique to the complementation of nondeterministic ω-automata, and obtain several lower bound results. Particularly, we prove an $\Omega((0.76n)^n)$ lower bound for Büchi complementation, which also holds for almost every complementation or determinization transformation of nondeterministic ω-automata, and prove an optimal $(\Omega(nk))^n$ lower bound for the complementation of generalized Büchi automata, which holds for Streett automata as well.

1. Introduction

The complementation problem of nondeterministic ω-automata, i.e. nondeterministic automata over infinite words, has various applications in formal verification. For example in automata-theoretic model checking, in order to check whether a system represented by automaton $\mathcal{A}_1$ satisfies a property represented by automaton $\mathcal{A}_2$, one checks that the intersection of $\mathcal{A}_1$ with an automaton that complements $\mathcal{A}_2$ is an automaton accepting the empty language [Kur94, VW94]. In such a process, several types of nondeterministic ω-automata are concerned, including Büchi, generalized Büchi, Rabin, Streett etc., and the complexity of complementing these automata has caught great attention.

The complementation of Büchi automata has been investigated for over forty years [Var07]. The first effective construction was given in [Büc62], and the first exponential construction was given in [SVW85] with a $2^{O(n^2)}$ state blow-up (n is the number of states of the input automaton). Even better constructions with $2^{O(n \log n)}$ state blow-ups were given in [Saf88, Kla91, KV01], which match with Michel's $n! = 2^{\Omega(n \log n)}$ lower bound

2000 ACM Subject Classification: F.4.1, F.4.3.
Key words and phrases: full automata, state complexity, automata transformation, Büchi complementation, ω-automata.

* A preliminary version of this paper appears in the proceedings of the 33rd International Colloquium on Automata, Languages and Programming, 2006.

Supported by NSFC No. 60273050.

[Mic88], and were thus considered optimal. However, a closer look reveals that the blow-up of the construction in [KV01] is $(6n)^n$, while Michel's lower bound is only roughly $(n/e)^n = (0.36n)^n$, leaving a big exponential gap hiding in the asymptotic notation[1]. Motivated by this complexity gap, the construction in [KV01] was further refined in [FKV06] to $(0.97n)^n$. On the other hand, Michel's lower bound was never improved.

For generalized Büchi, Rabin and Streett automata, the best known constructions are in [KV05b, KV05a], which are $2^{O(n \log nk)}$, $2^{O(nk \log n)}$ and $2^{O(nk \log nk)}$ respectively. Here state blow-ups are measured in terms of both n and k, where k is the index of the input automaton. Optimality problems of these constructions have been vastly open, because only $2^{\Omega(n \log n)}$ lower bounds were known by variants of Michel's proof [Löd99].

What remains missing are stronger lower bound results. Tighter lower bounds usually lead us into better understanding of the intricacy of the complementation of nondeterministic ω-automata, and are the main concern of this paper. Such understanding can suggest methods to further optimize the constructions, or to circumvent those difficult cases in practice.

To understand why we have so few strong lower bounds, we observe that at the core of almost every known lower bound is Michel's result, which was obtained in the traditional way. That is, one first constructs a particular family of automata $(\mathcal{A}_n)_{n\geq 1}$, and then proves that complementing each $\mathcal{A}_n$ requires a large state blow-up. The $\mathcal{A}_{n+1}$ of Michel's automata family is depicted in Figure 1. Although each $\mathcal{A}_{n+1}$ has a simple structure, it is not straightforward to see what language it accepts, and nor is it clear at all how we can work with this automaton for lower bound.

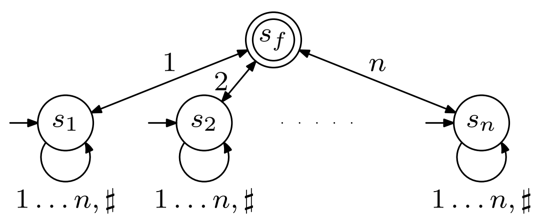

Figure 1: Michel's Automata Class

In many cases, identifying such an automata family is difficult, and is the main obstacle towards lower bounds. In this paper, we propose a new technique to circumvent this difficulty. Namely, we suggest first considering the family of full automata in lower bound analysis, and later reducing the size of the large alphabet via alphabet substitutions. A simple demonstration of such technique is presented in Section 3.

With the help of full automata, we tighten the state complexity $BC(n)$ of Büchi complementation from $(0.36n)^n \leq BC(n) \leq (0.97n)^n$ to $(0.76n)^n \leq BC(n) \leq (0.97n)^n$. Surprisingly, this $(0.76n)^n$ lower bound also holds for every complementation or determinization transformation concerning Büchi, generalized Büchi, Rabin, Streett, Muller, and parity automata. As to the complementation of generalized Büchi automata, we prove an $(\Omega(nk))^n$ lower bound, matching with the $(O(nk))^n$ upper bound in [KV05b]. This lower bound also holds for the complementation of Streett automata and the determinization of generalized

[1] In contrast, for the complementation of nondeterministic finite automata over finite words, the 2^n blow-up of the subset construction [RS59] was justified by a tight lower bound [SS78], which works even if the alphabet concerned is binary [Jir05].

Büchi automata into Rabin automata. A summary of our lower bounds is given in Section 6.

Full Automata and Sakoda and Sipser's Languages. It turns out that the notion of full automata is similar to Sakoda and Sipser's languages in [SS78]. Their language $\mathcal{B}_n$ actually corresponds to the Δ-graphs of the words accepted by some full automaton. Also as pointed to us by Christos A. Kapoutsis, the technique of alphabet substitution was somewhat implicit in Sakoda and Sipser's paper (but presented in a somewhat obscure way, refer to the paragraph before their Theorem 4.3.2). So the full automata technique is more like a new treatment of some techniques in the Sakoda and Sipser's paper, rather than a totally new invention. Compared to Sakoda and Sipser's languages, the notion of full automata enjoys a simple definition and is very handy to use. It is also more readily to be extended to other kinds of automata like alternating automata.

For unclear reasons, Sakoda and Sipser's languages were rarely applied to fields other than 2-way automata after their paper. We hope that our treatment will make a clear exposition of the techniques and demonstrate their usefulness in problems on automata over one-way inputs as well.

2. Basic Definitions

A (*nondeterministic*) *automaton* is a tuple $\mathcal{A} = (\Sigma, S, I, \Delta, *)$ with alphabet Σ, finite state set S, initial state set $I \subseteq S$, transition relation $\Delta \subseteq S \times \Sigma \times S$ and $*$ some extra components. Particularly $\mathcal{A}$ is *deterministic* if $|I| = 1$ and for all $p \in S$ and $a \in \Sigma$, $|\{q \in S \mid \langle p, a, q\rangle \in \Delta\}| \leq 1$.

For a word $w = a(0)a(1)\dots a(l-1) \in \Sigma^*$ with $length(w) = l \geq 0$, a finite run of $\mathcal{A}$ from state p to q over w is a finite state sequence $\rho = \rho(0)\rho(1)\dots\rho(l) \in S^*$ such that $\rho(0) = p$, $\rho(l) = q$ and $\langle \rho(i), a(i), \rho(i+1)\rangle \in \Delta$ for all $0 \leq i < l$. We say that ρ *visits* a state set T if $\rho(i) \in T$ for some $0 \leq i \leq l$. We write $p \xrightarrow{w} q$ if a finite run from p to q over w exists, and $p \xrightarrow[T]{w} q$ if in addition the run visits T.

A (*Nondeterministic*) *Finite Word Automaton* (NFW for short) is an automaton $\mathcal{A} = (\Sigma, S, I, \Delta, F)$ with final state set $F \subseteq S$. A finite word w is accepted by $\mathcal{A}$ if there is a finite run over w from an initial state to a final state. The language accepted by $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is the set of words accepted by $\mathcal{A}$, and its complement $\Sigma^* \backslash \mathcal{L}(\mathcal{A})$ is denoted by $\mathcal{L}^C(\mathcal{A})$.

For an ω-word $\alpha = \alpha(0)\alpha(1)\dots \in \Sigma^\omega$, i.e., an infinite sequence of letters in Σ, a (infinite) run of $\mathcal{A}$ over α is an infinite state sequence $\rho = \rho(0)\rho(1)\dots \in S^\omega$ such that $\rho(0) \in I$ and $\langle \rho(i), \alpha(i), \rho(i+1)\rangle \in \Delta$ for all $i \geq 0$. We let $Occ(\rho) = \{q \in S \mid \rho(i) = q \text{ for some } i \in \mathbb{N}\}$, $Inf(\rho) = \{q \in S \mid \rho(i) = q \text{ for infinitely many } i \in \mathbb{N}\}$, and write $\rho[l_1, l_2]$ to denote the infix $\rho(l_1)\rho(l_1+1)\dots\rho(l_2)$ of ρ.

An (nondeterministic) ω-*automaton* is an automaton $\mathcal{A} = (\Sigma, S, I, \Delta, Acc)$ with acceptance condition Acc, which is used to decide if a run ρ of $\mathcal{A}$ is successful. There are many types of ω-automata considered in the literature [Tho90]. Here we consider six of the most common types:

- *Büchi automaton*, where $Acc = F \subseteq S$ is a final state set, and ρ is successful if $Inf(\rho) \cap F \neq \emptyset$.

- *generalized Büchi automaton*, where $Acc = \{F_1, \dots, F_k\}$ is a list of final state sets, and ρ is successful if $Inf(\rho) \cap F_i \neq \emptyset$ for all $1 \leq i \leq k$.
- *Rabin automaton*, where $Acc = \{\langle G_1, B_1\rangle, \dots, \langle G_k, B_k\rangle\}$ is a list of pairs of state sets, and ρ is successful if for some $1 \leq i \leq k$, $Inf(\rho) \cap G_i \neq \emptyset$ and $Inf(\rho) \cap B_i = \emptyset$.
- *Streett automaton*, where $Acc = \{\langle G_1, B_1\rangle, \dots, \langle G_k, B_k\rangle\}$ is a list of pairs of state sets, and ρ is successful if for all $1 \leq i \leq k$, if $Inf(\rho) \cap B_i \neq \emptyset$, then $Inf(\rho) \cap G_i \neq \emptyset$.
- *Muller automaton*, where $Acc = \mathcal{F} \subseteq \mathcal{P}owerset(S)$ is a set of state sets, and ρ is successful if $Inf(\rho) \in \mathcal{F}$.
- *parity automaton*, where Acc is a mapping $c : S \to \{0 \dots l\}$, and ρ is successful if $\min\{c(q) | q \in Inf(\rho)\}$ is even.

An ω-word α is *accepted* by $\mathcal{A}$ if it has a successful run. The *ω-language* accepted by $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is the set of ω-words accepted by $\mathcal{A}$, and its complement $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$ is denoted by $\mathcal{L}^C(\mathcal{A})$. The number k, if defined, is called the *index* of $\mathcal{A}$.

We refer to the above six types of ω-automata as the *common types*. Following the convention in [KV05a], we will use acronyms like NBW, NGBW, NRW etc. to refer to Nondeterministic Büchi/generalized Büchi/Rabin/etc. Word automata. Two simple facts about these common types of ω-automata are useful for us:

fAct 2.1. [Löd99]*(1) For every* NBW $\mathcal{A}$ *and every common type $\mathcal{T}$, there exists an $\mathcal{T}$ automaton $\mathcal{A}'$ with the same number of states such that $\mathcal{A}'$ is equivalent to $\mathcal{A}$.*

(2) For every deterministic ω-automaton $\mathcal{A}$ of a common type $\mathcal{T}$ which is not Büchi nor generalized Büchi, there exists a deterministic ω-automaton $\mathcal{A}'$ of a common type (not necessarily also $\mathcal{T}$) with the same number of states (and index, if applicable) such that $\mathcal{A}'$ complements $\mathcal{A}$.

To visualize the behavior of automata over input words, we introduce the notion of Δ-graphs. If $\mathcal{A} = (\Sigma, S, I, \Delta, *)$ is an automaton, then for a finite word $w = a(0)a(1)\dots a(l-1) \in \Sigma^*$ of length l, or an ω-word $w = a(0)a(1)\dots \in \Sigma^\omega$ of length $l = \infty$, the *Δ-graph* of w under $\mathcal{A}$ is the directed graph $\mathcal{G}^{\mathcal{A}}_w = (V^{\mathcal{A}}_w, E^{\mathcal{A}}_w)$ with vertex set $V^{\mathcal{A}}_w = \{\langle p, i\rangle \mid p \in S, 0 \leq i \leq l, i \in \mathbb{N}\}$ and edge set $E^{\mathcal{A}}_w$ defined as: for all $p, q \in S$ and $0 \leq i < l$, $\langle\langle p, i\rangle, \langle q, i+1\rangle\rangle \in E^{\mathcal{A}}_w$ iff $\langle p, a(i), q\rangle \in \Delta$. For a subset T of S, we say that a vertex $\langle p, i\rangle$ is a T-vertex if $p \in T$. By definition $p \xrightarrow{w} q$ iff there is a path (in the directed sense) in $\mathcal{G}^{\mathcal{A}}_w$ from $\langle p, 0\rangle$ to $\langle q, length(w)\rangle$ and $p \xrightarrow[T]{w} q$ if furthermore the path visits some T-vertex.

Finally we define the *state complexity*[2] functions. Assume that $\mathcal{T}$ is either NFW or some common type of ω-automata. Then for a $\mathcal{T}$ automaton $\mathcal{A}$, $C_{\mathcal{T}}(\mathcal{A})$ is defined as the minimum number of states of a $\mathcal{T}$ automaton that complements $\mathcal{A}$, i.e., accepts $\mathcal{L}^C(\mathcal{A})$. For $n \geq 1$, $C_{\mathcal{T}}(n)$ is the maximum of $C_{\mathcal{T}}(\mathcal{A})$ over all $\mathcal{T}$ automata with n states. If indices are defined for $\mathcal{T}$, then $C_{\mathcal{T}}(n, k)$ is the maximum of $C_{\mathcal{T}}(\mathcal{A})$ over all $\mathcal{T}$ automata with n states and index k.

[2]In some literature, instead of merely counting the number of states, sizes of transition relations etc. are also taken into account to better measure the sizes of automata. Here we prefer state complexity because it is a measure easier to study, and its lower bound results usually imply lower bounds on "size" complexity, if the automata witnessing the lower bound are over a not too large alphabet.

3. The Full Automata Technique

In the recently emerging area of state complexity (see [Yu05] for a survey) or in the theory of ω-automata, we often concern proving theorems of such flavor:

Theorem 3.1. [Jir05] *For each $n \geq 1$, there exists an* NFW $\mathcal{A}_n$ *with n states over* $\{a, b\}$ *such that* $C_{\text{NFW}}(\mathcal{A}_n) \geq 2^n$.

In other words, we want to prove a lower bound for the state complexity of a transformation (NFW complementation in this case, can be determinization etc.), and furthermore, we hope that the automata family witnessing the lower bound ($(\mathcal{A}_n)_{n\geq 1}$ in this case) is over a fixed small alphabet. Such claims are usually difficult to prove. The apparently easy Theorem 3.1 was not proved until 2005 by a very technical proof in [Jir05][3], after the efforts in [SS78, Bir93, HK02]. To understand the difficulty involved, we first review the traditional approach people attempt at such results:

Step I: Identify an automata family $(\mathcal{A}_n)_{n\geq 1}$ with each $\mathcal{A}_n$ having n states.
Step II: Prove that to transform each $\mathcal{A}_n$ needs a large state blow-up.

Almost every known lower bound was obtained in this way, including Theorem 3.1 and the aforementioned Michel's lower bound. In such an approach, Step I is well-known to be difficult. Identifying the suitable family $(\mathcal{A}_n)_{n\geq 1}$ requires both ingenuity and luck. Even worse, most automata families that people try are natural ones with simple structures, while the ones witnessing the desired lower bound could be highly unnatural and complex. Finding the right family $(\mathcal{A}_n)_{n\geq 1}$ seems to be a major obstacle towards lower bound results.

Now we introduce the notion of full automata to circumvent this obstacle.

Definition 3.2. Given state set S, initial state set I, and extra components $*$, a *full automaton* $\mathcal{A} = (\Sigma, S, I, \Delta, *)$ is an automaton with alphabet $\Sigma = \mathcal{P}owerset(S \times S)$ and transition relation Δ defined as: for all $p, q \in S$ and $a \in \Sigma$, $\langle p, a, q\rangle \in \Delta$ iff $\langle p, q\rangle \in a$.

By definition, the alphabet contains every binary relation over S, and therefore is of a big size of $2^{|S|^2}$. Due to such rich alphabets, every automaton has some embedding in a full automaton with the same number of states. It is then not difficult to see that transforming an automaton can be reduced to transforming a full automaton, and full automata are the most difficult automata to transform.

To be specific, if we consider NFW complementation, then:

Theorem 3.3. *For all $n \geq 1$, $C_{\text{NFW}}(n) = C_{\text{NFW}}(\mathcal{A})$ for some full* NFW $\mathcal{A}$ *with n states.*

The theorem follows from the following lemma.

Lemma 3.4. *If $\mathcal{A}_1$ is an* NFW *with n states, then there is a full* NFW $\mathcal{A}_2$ *with n states such that $C_{\text{NFW}}(\mathcal{A}_2) \geq C_{\text{NFW}}(\mathcal{A}_1)$.*

Proof. By definition of C_{NFW}, it suffices to show that for some full NFW $\mathcal{A}_2$ with n states, if there is an NFW $\mathcal{CA}_2$ that complements $\mathcal{A}_2$, then there is an NFW $\mathcal{CA}_1$ complementing $\mathcal{A}_1$ with the same number of states as $\mathcal{CA}_2$.

Let $\mathcal{A}_1 = (\Sigma_1, S_1, I_1, \Delta_1, F_1)$, and consider the full NFW $\mathcal{A}_2 = (\Sigma_2, S_1, I_1, \Delta_2, F_1)$ with respect to S_1, I_1 and F_1. For each $a_1 \in \Sigma_1$, define letter $\Delta_1(a_1)$ in $\Sigma_2 = \mathcal{P}(S_1 \times S_1)$ as: $\langle p_1, q_1\rangle \in \Delta_1(a_1)$ iff $\langle p_1, a_1, q_1\rangle \in \Delta_1$, for all $p_1, q_1 \in S_1$. By definition of full automata,

[3] The result is actually slightly stronger in that his $\mathcal{A}_n$ has only one initial state. (In some literature NFWs are not allowed to have multiple initial states.)

$\langle p_1, a_2, q_1\rangle \in \Delta_2$ iff $\langle p_1, q_1\rangle \in a_2$, for all $p_1, q_1 \in S_1, a_2 \in \Sigma_2$. So we have $\langle p_1, a_1, q_1\rangle \in \Delta_1$ iff $\langle p_1, \Delta_1(a_1), q_1\rangle \in \Delta_2$, for all $a_1 \in \Sigma_1, p_1, q_1 \in S_1$. For an arbitrary word $\alpha = a(0)a(1)\dots a(l-1) \in \Sigma_1^*$, consider word $\alpha' = \Delta_1(a(0))\Delta_1(a(1))\dots\Delta_1(a(l-1)) \in \Sigma_2^*$. Then every state sequence $\rho_1 = \rho_1(0)\rho_1(1)\dots\rho_1(l) \in S_1^*$ is a run of $\mathcal{A}_1$ over α iff ρ_1 is a run of $\mathcal{A}_2$ over α'. Since $\mathcal{A}_1$ and $\mathcal{A}_2$ share the same initial and final state sets, ρ_1 is successful iff ρ_2 is successful. So $\alpha \in \mathcal{L}(\mathcal{A}_1)$ iff $\alpha' \in \mathcal{L}(\mathcal{A}_2)$.

Let $\mathcal{CA}_2 = (\Sigma_2, S_C, I_C, \Delta_C, F_C)$ be an NFW that complements $\mathcal{L}(\mathcal{A}_2)$. So $\alpha' \in \mathcal{L}(\mathcal{A}_2)$ iff $\alpha' \notin \mathcal{L}(\mathcal{CA}_2)$. Define $\mathcal{CA}_1$ to be the NFW $(\Sigma_1, S_C, I_C, \Delta'_C, F_C)$, where Δ'_C is defined as $\langle p_2, a_1, q_2\rangle \in \Delta'_C$ iff $\langle p_2, \Delta_1(a_1), q_2\rangle \in \Delta_C$, for all $p_2, q_2 \in S_C$ and $a_1 \in \Sigma_1$. Similarly every state sequence $\rho_C = \rho_C(0)\rho_C(1)\dots\rho_C(l) \in S_C^*$ is a successful run of $\mathcal{CA}_2$ over α' iff ρ_C is a successful run of $\mathcal{CA}_1$ over α. So $\alpha' \in \mathcal{L}(\mathcal{CA}_2)$ iff $\alpha \in \mathcal{L}(\mathcal{CA}_1)$.

Now for every $\alpha \in \Sigma_1^*$, $\alpha \in \mathcal{L}(\mathcal{A}_1)$ iff $\alpha \notin \mathcal{L}(\mathcal{CA}_1)$. Therefore $\mathcal{CA}_1$ with the same number of states as $\mathcal{CA}_2$ complements $\mathcal{A}_1$ as required. □

Theorem 3.3 implies that to prove a lower bound for NFW complementation (without taking the size of the alphabet into account), we can simply set $(\mathcal{A}_n)_{n\geq 1}$ to be some family of full NFWs in Step I. Similarly, the same applies to NBW complementation:

Theorem 3.5. *For all $n \geq 1$, $C_{\text{NBW}}(n) = C_{\text{NBW}}(\mathcal{A})$ for some full* NBW *$\mathcal{A}$ with n states.*

Now we apply full automata to obtain a simple proof of Theorem 3.1.

Proof. (of Theorem 3.1) We first prove a 2^n lower bound for $C_{\text{NFW}}(n)$. For each $n \geq 1$, let $\mathcal{FA}_n = (\Sigma_n, S_n, I_n, \Delta_n, F_n)$ be the full NFW with $S_n = I_n = F_n = \{s_0, \dots, s_{n-1}\}$. It suffices to prove that $C_{\text{NFW}}(\mathcal{FA}_n) \geq 2^n$.

For each subset $T \subseteq S_n$, let $Id(T)$ denote the letter $\{\langle q, q\rangle \mid q \in T\}$ and let $u_T = Id(T)$, $v_T = Id(S_n \backslash T)$. Figure 2(a) depicts one example of $u_T v_T$'s Δ-graph. Since all states in $\mathcal{FA}_n$ are both initial and final, a word w of length l is accepted by $\mathcal{FA}_n$ iff there is a path from an $\langle s_i, 0\rangle$ vertex to an $\langle s_j, l\rangle$ vertex in the Δ-graph of w under $\mathcal{FA}_n$. In particular $u_T v_T$ is not accepted by $\mathcal{FA}_n$. Suppose that some NFW $\mathcal{CA}$ complements $\mathcal{FA}_n$. So for each $T \subseteq S_n$, there is a state $\hat{q}_T$ of $\mathcal{CA}$ such that $\hat{q}_I \xrightarrow{u_T} \hat{q}_T$ and $\hat{q}_T \xrightarrow{v_T} \hat{q}_F$ for some initial state $\hat{q}_I$ and final state $\hat{q}_F$ of $\mathcal{CA}$. If we prove that $\hat{q}_{T_1} \neq \hat{q}_{T_2}$ whenever $T_1 \neq T_2$, then $\mathcal{CA}$ has at least 2^n states as required. Suppose by contradiction that $\hat{q}_{T_1} = \hat{q}_{T_2}$ for some $T_1 \neq T_2$. W.l.o.g. there is a state s of $\mathcal{FA}_n$ in $T_1 \backslash T_2$. Then $s \xrightarrow{u_{T_1}} s \xrightarrow{v_{T_2}} s$ and hence $u_{T_1} v_{T_2} \in \mathcal{L}(\mathcal{FA}_n)$. On the other hand, for some initial state $\hat{q}_I$ and final state $\hat{q}_F$ of $\mathcal{CA}$, $\hat{q}_I \xrightarrow{u_{T_1}} \hat{q}_{T_1} = \hat{q}_{T_2} \xrightarrow{v_{T_2}} \hat{q}_F$. So $u_{T_1} v_{T_2} \in \mathcal{L}(\mathcal{CA})$, contradiction.

The above proof is not fully satisfying in that the automata family witnessing the lower bound is over an exponentially growing alphabet. To fix a binary alphabet and prove Theorem 3.1, we introduce a Step III in which we do "alphabet substitution", as we now illustrate.

We first refine the above proof of $C_{\text{NFW}}(\mathcal{FA}_n) \geq 2^n$ by restricting the number of different letters involved. For two words $u, v \in \Sigma_n^*$, we say that u is *equivalent* to v with respect to $\mathcal{FA}_n$, or simply $u \sim v$, if for all $p, q \in S_n$, $p \xrightarrow{u} q$ iff $p \xrightarrow{v} q$. A little thought shows that if we substitute each $Id(T)$ letter used in the above proof by some equivalent words, the proof still works. First we consider the alphabet $\{c_i\}_{0\leq i<n}$ with $c_i = Id(S_n \backslash \{s_i\})$. Then for each $T \subseteq S_n$, $Id(T) \sim \Pi_{s \notin T} c_i$, the concatenation of all c_i's with $s_i \notin T$ in lexicographical order (any other fixed order will do). This is illustrated in Figure 2(b). Then consider the alphabet $\{a, b\}$ with $a = \{\langle s_{i+1}, s_i\rangle \mid 0 \leq i < n-1\} \cup \{\langle s_0, s_{n-1}\rangle\}$ and $b = Id(S_n \backslash \{s_0\})$,

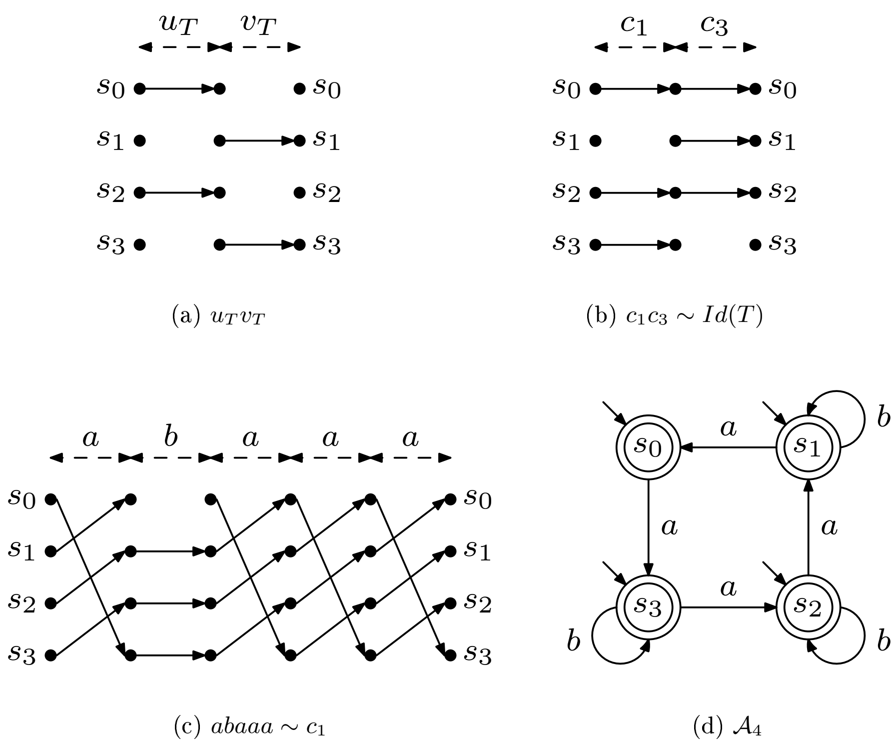

Figure 2: Examples

then for each $0 \le i < n$, $c_i \sim a^i b a^{n-i}$, as illustrated in Figure 2(c). So if we substitute each letter $Id(T)$ in the above proof by the equivalent word $\Pi_{s_i \notin T} a^i b a^{n-i}$, the proof still works.

After the above refinement of the proof, the part of $\mathcal{FA}_n$ related to letters other than $\{a, b\}$ is in fact irrelevant to the proof. So $\mathcal{A}_n = \mathcal{FA}_n \upharpoonright \{a, b\}$, the restriction of $\mathcal{FA}_n$ to $\{a, b\}$, or formally the NFW $\mathcal{A}_n = (\{a, b\}, S_n, I_n, \Delta_n \cap (S_n \times \{a, b\} \times S_n), F_n)$, also satisfies that $C_{\mathrm{NFW}}(\mathcal{A}_n) \ge 2^n$, as required ($\mathcal{A}_4$ is depicted in 2(d)). $\square$

We call the above technique of setting $(\mathcal{A}_n)_{n\ge1}$ to be a family of full automata and adding the step of alphabet substitution the "full automata technique". Setting $(\mathcal{A}_n)_{n\ge1}$ to be full automata is crucial here, which in essence delays the trouble of identifying $(\mathcal{A}_n)_{n\ge1}$ to the later analysis of transforming full automata. This makes our life easier because the latter is usually playing with words, which is clearly easier than constructing automata, especially with the rich alphabet of full automata. As to the step of alphabet substitution, our experience is that it could be technical some time, but rarely difficult.

4. Büchi Complementation

4.1. **Kupferman and Vardi's Construction.** We first briefly introduce the state-of-the-art construction for Büchi complementation by Kupferman and Vardi in [FKV06], the idea of which is important in our lower bound. Different from [FKV06], we will continue to work with our Δ-graphs rather than introducing the notion of run graphs. For $x \in \mathbb{N}$, let

$[x]$ denote the set $\{0, 1, \ldots, x\}$ and let $[x]^{odd}$ and $[x]^{even}$ denote the sets of odd and even numbers in $[x]$ respectively.

Definition 4.1. Given an NBW $\mathcal{A} = (\Sigma, S, I, \Delta, F)$ of n states, and an ω-word α, a *co-Büchi ranking* (C-Ranking for short) for $\mathcal{G}^{\mathcal{A}}_{\alpha}$ (i.e. the Δ-graph of α under $\mathcal{A}$) is a partial function f from $V^{\mathcal{A}}_{\alpha}$ to the rank set $[2n-2]$ such that:

- **(i):** For all vertices $\langle q, l\rangle \in V^{\mathcal{A}}_{\alpha}$, $f(\langle q, l\rangle)$ is undefined iff there is no path (in the directed sense) from some $\langle q_I, 0\rangle$ vertex with $q_I \in I$ to $\langle q, l\rangle$.
- **(ii):** For all vertices $\langle q, l\rangle \in V^{\mathcal{A}}_{\alpha}$, if $f(\langle q, l\rangle)$ is odd, then $q \notin F$.
- **(iii):** For all edges $\langle\langle q, l\rangle, \langle q', l+1\rangle\rangle \in E^{\mathcal{A}}_{\alpha}$, if $f(\langle q, l\rangle)$ is defined, then $f(\langle q, l\rangle) \geq f(\langle q', l+1\rangle)$.

We say that f is *odd* if for every path in $\mathcal{G}^{\mathcal{A}}_{\alpha}$, there are infinitely many vertices that are assigned odd ranks by f.

Lemma 4.2. [KV01] *The ω-word α is not accepted by $\mathcal{A}$ iff there is an odd C-ranking for $\mathcal{G}^{\mathcal{A}}_{\alpha}$.*

Proof. We prove the *if* direction here to give a sense of the idea of C-ranking. For every infinite path from a $\langle q_I, 0\rangle$ vertex for some $q_I \in I$, the ranks along the path do not increase by (iii) and so will get trapped in some fixed rank from some point on. Since f is odd, this fixed rank is odd, and thus by (ii), F-vertices are never visited since then. In other words, every run of $\mathcal{A}$ over α visits F finitely often and hence α is not accepted by $\mathcal{A}$. □

A *level ranking*[4] for $\mathcal{A}$ is a partial function $g : S \longrightarrow [2n-2]$ such that if $g(q)$ is odd, then $q \notin F$. Each C-ranking can be "sliced" into such level rankings. It was shown in [KV01] that existence of an odd C-ranking for $\mathcal{G}^{\mathcal{A}}_{\alpha}$ can be decided by an NBW $\mathcal{CA}$ which guesses an odd C-ranking level by level, and checks the validity in a local manner. By Lemma 4.2, $\mathcal{CA}$ complements $\mathcal{A}$. In the construction of $\mathcal{CA}$, distinct sets of states are used to handle different level rankings, and the number of such level rankings is the major factor of the $(6n)^n$ blow-up.

We say that a level ranking g for $\mathcal{A}$ is *tight* if (i): the maximum rank in the range of g is some odd number $2m-1$ in $[2n-2]^{odd}$, and (ii): for every $j \in [2m]^{odd}$, there is a state q with $g(q) = j$. In such a case, g is also called a TL(m)-ranking (with $1 \leq m < n$). It was further shown in [FKV06] that we can restrict attention to tight level rankings and use less states in $\mathcal{CA}$. By a careful numerical analysis [FKV06], a $(0.97)^n$ upper bound was proved for the number of states of $\mathcal{CA}$ and thus for Büchi complementation.

4.2. **Lower Bound.** We turn now to lower bound. By Theorem 3.5, it suffices to consider full NBWs. We define $\mathcal{FB}_n$ for $n > 1$ to be the full NBW $(\Sigma_n, S_n, I_n, \Delta_n, F_n)$ with $I_n = \{s_0, \ldots, s_{n-2}\}$, $F_n = \{s_f\}$ and $S_n = I_n \cup F_n$. We also use $S'_n = I_n$ to denote the "main" states.

We first try to construct an ω-word α_n not accepted by $\mathcal{FB}_n$ such that a great number of tight level rankings would have to be present in every C-ranking for $\mathcal{G}^{\mathcal{FB}_n}_{\alpha_n}$. Since the number of tight level rankings is the major factor of the state blow-up in Kupferman and Vardi's construction, this would produce a hard case for the construction. For such purpose, we

[4] Our definitions of level ranking and tight level ranking here are slightly different from [FKV06].

consider a special class of tight level rankings for $\mathcal{FB}_n$, Q-rankings. We say that a TL(m)-ranking g for $\mathcal{FB}_n$ is a $Q(m)$*-ranking* if $g(q)$ is defined for each $q \in S'_n$ and is undefined for $q = s_f$. We start defining our difficult ω-word α_n by defining its composing segments.

Lemma 4.3. *For every pair of Q-rankings (f, g), there exists a word $w_{f,g}$ such that:*

- **(i):** *For all $p, q \in S'_n$, $p \xrightarrow{w_{f,g}} q$ iff ($f_i(p) > f_{i+1}(q)$ or $f_i(p) = f_{i+1}(q) \in [2m]^{odd}$).*
- **(ii):** *For all $p, q \in S'_n$, $p \xrightarrow[F_n]{w_{f,g}} q$ iff $f_i(p) > f_{i+1}(q)$.*
- **(iii):** *For all $p, q \in S_n$, if $p \xrightarrow{w_{f,g}} q$ then $p, q \notin F_n$.*

Proof. We first illustrate the construction using a typical example depicted in Fig. 3. As in Fig. 3, the vertices of the Δ-graph of $w_{f,g}$ are separated by the wider space below $c(f, g)$ into two parts. We say that each (s_i, j) vertex in the left part is ranked $f(s_i)$ by f, and each (s_i, j) vertex in the right part is ranked $g(s_i)$ by g. So when one follows a path from a leftmost vertex v_1 to a rightmost vertex v_2, either one goes to a next vertex with the same rank, or one visits a $\langle s_f, j\rangle$ vertex and then goes to a vertex with a rank lower by one. This explains the *only if* direction of (ii). Also note that v_1 and v_2 cannot have the same even ranks because in the middle of this process, one has to go to a vertex with an odd rank to pass $c(f, g)$. So the *only if* direction in (i) holds too. For the *if* directions of (i) and (ii), suppose one wants to go from a leftmost vertex v_1 with rank r to a rightmost vertex v_2 with rank r' and that either $r > r'$ or $r = r' \in [2m]^{odd}$. Let t be an odd rank such that $r \geq t \geq r'$. Then by the construction, one can go from v_1 to some vertex with rank t in the left part, pass through $c(f, g)$ with rank t, and then continue to go to v_2 in the right part. Note that in the process, if rank ever decreases, then an $\langle s_f, j\rangle$ vertex must have been visited. So the if directions of (i) and (ii) hold as well. Condition (iii) is obviously true.

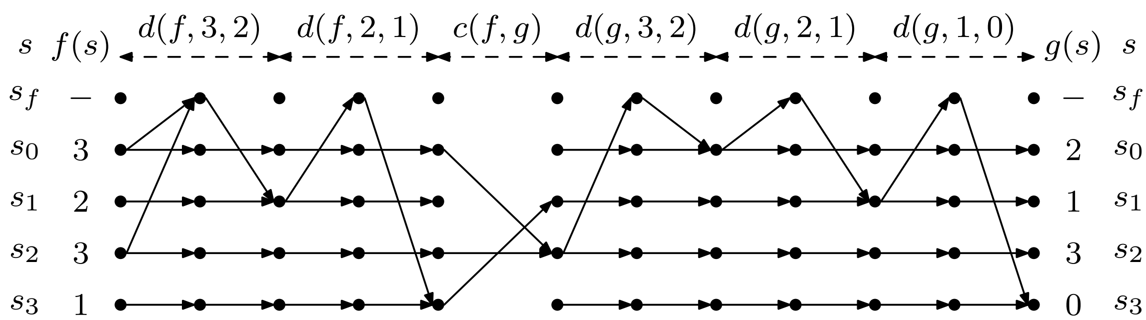

Figure 3: Δ-graph of $w_{f,g}$

For later purposes, we explicitly present our construction for $w_{f,g}$. For a $Q(m)$-ranking h, we define the state sets $Rank_h(r) = \{q \in S'_n \mid r = h(q)\}$ for $r \in [2m]$ and Odd_h to be the union of $Rank_h(r)$'s with $r \in [2m]^{odd}$. Also for each $T \subseteq S'_n$, define letters in Σ_n as $Id(T) = \{\langle q, q\rangle \mid q \in T\}$, $TtoF(T) = Id(S'_n) \cup \{\langle q, s_f\rangle \mid q \in T\}$, $FtoT(T) = Id(S'_n) \cup \{\langle s_f, q\rangle \mid q \in T\}$ and $c(f, g) = \{\langle p, q\rangle \mid f(p) = g(q) \in [2m]^{odd}, p, q \in S'_n\}$. For a $Q(m)$-ranking h and $r, r' \in [2m]$, we write $d(h, r, r')$ to denote the word $TtoF(Rank_h(r)) \cdot FtoT(Rank_h(r'))$. Then if $r_1, r_2 \ldots, r_k$ are the ranks in $[2m]$ that are images of h in descending order, we let $u_h = d(h, r_1, r_2) \cdot d(h, r_2, r_3) \cdot \cdots \cdot d(h, r_{k-1}, r_k)$. Finally, $w_{f,g}$ is defined to be $u_f \cdot c(f, g) \cdot u_g$. $\square$

Lemma 4.4. *Let $f_0, f_1, \dots, f_l$ be a list of $Q(m)$-rankings with $l > 0$, and let w be the word $w_{f_0,f_1} w_{f_1,f_2} \dots w_{f_{l-1} f_l}$. Also let $p, q \in S'_n$, then:*

(i) If $f_0(p) > f_l(q)$ or $f_0(p) = f_l(q) \in [2m]^{odd}$, then $p \xrightarrow{w} q$.

(ii) If $f_0(p) > f_l(q)$, then $p \xrightarrow[F_n]{w} q$.

Proof. If $l = 1$, then $w = w_{f_0,f_1}$, and the properties follow from Theorem 4.3 trivially. So we assume that $l > 1$. Let t be an odd rank such that $f_0(p) \geq t \geq f_l(q)$. By definition of $Q(m)$-ranking, there exists a state sequence $q_1, q_2, \dots, q_{l-1}$ such that $f_i(q_i) = t$ for all $1 \leq i \leq l-1$. So $q_i \overset{w_{f_i,f_{i+1}}}{\longrightarrow} q_{i+1}$ for all $1 \leq i < l-1$. Also because $f_0(p) \geq t \geq f_l(q)$, we have $p \overset{w_{f_0,f_1}}{\longrightarrow} q_1$ and $q_{l-1} \overset{w_{f_{l-1},f_l}}{\longrightarrow} q$. Concatenate these together, we have $p \xrightarrow{w} q$, and (i) is satisfied. If $f_0(p) > f_l(q)$, then either $f_0(p) > t$ or $t > f_l(q)$, and hence either $p \xrightarrow[F_n]{w_{f_0,f_1}} q_1$ or $q_{l-1} \xrightarrow[F_n]{w_{f_{l-1},f_l}} q$. So $p \xrightarrow[F_n]{w} q$, and (ii) is satisfied. □

Let $L(n, m)$ be the number of different $Q(m)$-rankings and let $L(n)$ be $\max_{1 \leq m < n} L(n, m)$. From now on we fix m such that $L(n) = L(n, m)$ and may simply write L for $L(n)$. Clearly there exists an infinite looping enumeration $f_0, f_1, \dots$ of $Q(m)$-rankings such that $f_i \neq f_j$ for all $i \neq j, 0 \leq i, j < L$, and $f_i = f_{jL+i}$ for all $i, j \geq 0$. Our "difficult" ω-word α_n is then the ω-word $w_0 w_1 \dots$ where $w_i = w_{f_i,f_{i+1}}$ for all $i \geq 0$.

Lemma 4.5. *The ω-word α_n is not in $\mathcal{L}(\mathcal{FB}_n)$.*

Proof. If there is a successful run ρ of $\mathcal{FB}_n$ over α_n, then there is an infinite state sequence $q_0 q_1 \dots \in S_n^\omega$ such that $q_i \xrightarrow{w_i} q_{i+1}$ for all $i \geq 0$ and $q_i \xrightarrow[F_n]{w_i} q_{i+1}$ for infinitely many $i \in \mathbb{N}$. So by the construction of $w_i = w_{f_i,f_{i+1}}$, $f_i(q_i) \geq f_{i+1}(q_{i+1})$ for all $i \geq 0$ and $f_i(q_i) > f_{i+1}(q_{i+1})$ for infinitely many $i \in \mathbb{N}$. This is impossible since $f_0(q_0)$ is finite. □

Recall that Kupferman and Vardi's construction uses distinct state sets to handle different TL(m)-rankings. It turns out that if a complement automaton of $\mathcal{FB}_n$ does not have as many states as $Q(m)$-rankings, it would be "confused" by α_n together with another complex ω-word α' derived from α_n.

Lemma 4.6. *For each $n > 1$ and each ω-automaton $\mathcal{CA}$ with less than L states, if ρ is a run of $\mathcal{CA}$ over $\alpha_n \notin \mathcal{L}(\mathcal{FB}_n)$, then there is a run ρ' of $\mathcal{CA}$ over some ω-word $\alpha' \in \mathcal{L}(\mathcal{FB}_n)$ with $Occ(\rho') = Occ(\rho)$ and $Inf(\rho') = Inf(\rho)$.*

Proof. Suppose that $\mathcal{CA} = (\Sigma_n, \hat{S}, \hat{I}, \hat{\Delta}, Acc)$ is an ω-automaton with less than L states and $\rho = \rho(0)\rho(1) \dots \in \hat{S}^\omega$ is a run of $\mathcal{CA}$ over α_n. Let $k_0, k_1, \dots$ be a number sequence such that $k_0 = 0$, $k_{i+1} - k_i = length(w_i)$ for all $i \geq 0$. So the k_i's mark the positions where the w_i's concatenate. Therefore $\rho(k_i) \xrightarrow{w_i} \rho(k_{i+1})$ for all $i \geq 0$. Define for each $0 \leq i < L$ the nonempty set:

$$\hat{Q}_i = \{\hat{q} \in \hat{S} \mid \rho(k_{jL+i}) = \hat{q} \text{ for infinitely many } j \in \mathbb{N}\}.$$

Since $\mathcal{CA}$ has less than L states, there exists some state $\hat{q}$ in $\hat{Q}_i \cap \hat{Q}_j$ for some $i \neq j, 0 \leq i, j < L$. In particular one has, by definition, $f_i \neq f_j$. W.l.o.g. there is a $q \in S'_n$ with $f_i(q) > f_j(q)$. By definitions of $\hat{Q}_i$ and $Occ(\rho)$, there is a $t_1 \in \mathbb{N}$ sufficiently large such that

$\rho(k_{t_1L+i}) = \hat{q}$, every state in $Occ(\rho)$ occurs in $\rho[0, k_{t_1L+i}]$, and that $\rho(t') \in Inf(\rho)$ for all $t' > k_{t_1L+i}$. By definitions of $Inf(\rho)$ and $\hat{Q}_j$, there is a sufficiently large $t_2 > t_1$ such that $\rho(k_{t_2L+j}) = \hat{q}$ and every state in $Inf(\rho)$ occurs in $\rho[k_{t_1L+i}, k_{t_2L+j}]$. Let $u = w_0 \dots w_{t_1L+i-1}$ and $v = w_{t_1L+i} \dots w_{t_2L+j-1}$. Finally let α' be uv^ω.

Let $q_I \in S'_n$ be such that $f_0(q_I) = 2m-1 \geq f_i(q) = f_{t_1L+i}(q)$. By Lemma 4.4, $q_I \xrightarrow{u} q$. Similarly, since $f_{t_1L+i}(q) = f_i(q) > f_j(q) = f_{t_2L+j}(q)$, by Lemma 4.4 we have $q \xrightarrow[F_n]{v} q$. Together we have $q_I \xrightarrow{u} q \xrightarrow[F_n]{v} q \xrightarrow[F_n]{v} q \dots$ and α' is accepted by $\mathcal{FB}_n$.

Finally, note that $\rho' = \rho[0, k_{t_1L+i}] \cdot (\rho[k_{t_1L+i}+1, k_{t_2L+j}])^\omega$ is a run over α', and we have guaranteed that $Occ(\rho') = Occ(\rho)$ and $Inf(\rho') = Inf(\rho)$ as required. □

Theorem 4.7. *For every $n > 1$, $L(n) \leq C_{\mathrm{NBW}}(\mathcal{FB}_n) \leq C_{\mathrm{NBW}}(n)$, where $L(n) = \Theta((0.76n)^n)$.*

Proof. By Lemma 4.6, every NBW that complements $\mathcal{FB}_n$ must have at least $L(n)$ states, otherwise both α_n and α'_n would be accepted by $\mathcal{FB}_n$, leading to contradiction. By a numerical analysis of $L(n)$ very similar to the one in [FKV06], we have that $L(n) = \Theta((0.76n)^n)$. For completeness, we present the detail of the analysis in appendix. □

4.3. **Alphabet.** Following the proof of Theorem 4.7, one constructs full NBWs witnessing the lower bound over a very large alphabet, which we rarely consider in practice. In this subsection, we show that by using alphabet substitutions like in the proof of Theorem 3.1, the NBWs witnessing the lower bound can be also over a fixed alphabet.

We say two words u and v from Σ_n^* are equivalent with respect to $\mathcal{FB}_n$, or simply $u \approx v$, if for all $p, q \in S'_n$: (i) $p \xrightarrow{u} q$ iff $p \xrightarrow{v} q$, and, (ii) $p \xrightarrow[F_n]{u} q$ iff $p \xrightarrow[F_n]{v} q$. Then if one replaces each letter involved in the lower bound proof by an equivalent word over some alphabet Γ, one shows that $\mathcal{FB}_n \restriction \Gamma$ also witnesses the same $L(n)$ lower bound.

Lemma 4.8. *There is an alphabet Γ of size 7 such that for each pair $\langle f, g\rangle$ of $Q(m)$-rankings for $\mathcal{FB}_n$, there is a word in Γ^* equivalent to $w_{f,g}$.*

Proof. Let Γ be the alphabet containing the following 7 letters:

- $rotate = \{\langle s_{i+1}, s_i\rangle \mid 0 \leq i < n-2\} \cup \{\langle s_0, s_{n-2}\rangle, \langle s_f, s_f\rangle\}$,
- $clear0 = Id(S_n \backslash \{s_0\})$,
- $swap01 = (Id(S'_n) \cup \{\langle s_0, s_1\rangle, \langle s_1, s_0\rangle\}) \backslash \{\langle s_0, s_0\rangle, \langle s_1, s_1\rangle\}$,
- $copy01 = Id(S'_n) \cup \{\langle s_1, s_0\rangle\}$,
- $0toF = Id(S_n) \cup \{\langle s_0, s_f\rangle\}$,
- $Fto0 = Id(S_n) \cup \{\langle s_f, s_0\rangle\}$,
- $clearF = Id(S'_n)$.

Only three types of letters are relevant in the proof of Theorem 4.7: $TtoF(T)$, $FtoT(T)$ and $c(f, g)$. For each $T \subseteq S'_n$, one can verify that:

- $TtoF(T) \approx clearF \cdot \prod_{s_i \in T} (rotate^i \cdot 0toF \cdot rotate^{n-1-i})$.
- $FtoT(T) \approx \prod_{s_i \in T} (rotate^i \cdot Fto0 \cdot rotate^{n-1-i}) \cdot clearF$.

As to $c(f,g)$, the task is a bit more complicated, and let us view it in a different way. For a word w, define set $r_j = \{i | s_i \xrightarrow{w} s_j, 0 \le i < n-1\}$ for every $0 \le j < n-1$. Clearly for two words u and v, the following are equivalent:

- $p \xrightarrow{u} q$ iff $p \xrightarrow{v} q$ for all $p, q \in S'_n$.
- $r_j(u) = r_j(v)$ for all $0 \le j < n-1$.

So it is sufficient to find for each $c(f,g)$ a word w over $\{rotate, clear0, swap01, copy01\}$ such that $r_j(w) = r_j(c(f,g))$ for all $0 \le j < n-1$.

Appending each letter a to the end of a word w changes the content of the $r_i(w)$'s. Consider these three types of words in Γ^*:

$$(1)\ \ swap_{i,j} = \begin{cases} rotate^i \cdot swap01 \cdot rotate^{n-1-i} & \text{if } i+1 = j \\ \begin{array}{c}(swap_{i,i+1} \cdot swap_{i+1,i+2} \cdot \cdots \cdot swap_{j-1,j}) \\ \cdot(swap_{j-2,j-1} \cdot swap_{j-3,j-2} \cdot \cdots \cdot swap_{i,i+1})\end{array} & \text{if } i+1 < j \\ swap_{j,i} & \text{if } i > j \\ \text{the empty word} & \text{if } i = j \end{cases}.$$

$$(2)\ \ copy_{i,j} = \begin{cases} swap01 \cdot copy01 \cdot swap01 & \text{if } i = 1 \text{ and } j = 0 \\ swap_{0,i} \cdot swap_{1,j} \cdot copy01 \cdot swap_{1,j} \cdot swap_{0,i} & \text{otherwise} \end{cases}.$$

$$(3)\ \ clear_i = swap_{0,i} \cdot clear0 \cdot swap_{0,i}$$

One can verify that appending a $swap_{i,j}$ to w exchanges the content of $r_i(w)$ and $r_j(w)$, appending a $copy_{i,j}$ sets $r_i(w)$ to be $r_i(w) \cup r_j(w)$, and appending a $clear_i$ empties $r_i(w)$. Obviously these three operations allow one to reach arbitrary $(r_i(w))_{0 \le i < n-1}$ configurations, including $(r_i(c(f,g)))_{0 \le i < n-1}$, as needed. □

So $\mathcal{B}_n = \mathcal{FB}_n \restriction \Gamma$, the restriction of $\mathcal{FB}_n$ to the alphabet Γ, satisfies that $C_{\mathrm{NBW}}(\mathcal{B}_n) \ge L(n)$, and we have:

Theorem 4.9. *For each $n > 1$, there exists an* NBW *$\mathcal{B}_n$ with n states over a seven letters alphabet such that $L(n) \le C_{\mathrm{NBW}}(\mathcal{B}_n)$.*

4.4. **Other Transformations.** Surprisingly, our lower bound on Büchi complementation extends to almost every complementation or determinization transformation of nondeterministic ω-automata, via a reduction making use of Lemma 4.6.

Theorem 4.10. *For each $n > 1$ and each common type $\mathcal{T}_1$ of nondeterministic ω-automata, there exists a $\mathcal{T}_1$ automaton $\mathcal{A}_n$ with n states over a fixed alphabet such that:*

- **(i):** *For each common type $\mathcal{T}_2$, every $\mathcal{T}_2$ automaton that complements $\mathcal{L}(\mathcal{A}_n)$ has at least $L(n)$ states.*
- **(ii):** *For each common type $\mathcal{T}_2$ that is not Büchi nor generalized Büchi[5], every deterministic $\mathcal{T}_2$ automaton that accepts $\mathcal{L}(\mathcal{A}_n)$ has at least $L(n)$ states.*

Proof. For each common type $\mathcal{T}_1$, by Fact 2.1, there is a $\mathcal{T}_1$ automaton $\mathcal{A}_n$ equivalent to NBW $\mathcal{FB}_n$ with also n states [Löd99]. (i) Suppose that an automaton $\mathcal{CA}$ of a common type accepts $\mathcal{L}^C(\mathcal{A}_n)$= $\mathcal{L}^C(\mathcal{FB}_n)$. Since acceptance of ω-automata of a common type only depends on the Inf set of a run, the claim can be obtained by applying Lemma 4.6. (ii) If some deterministic $\mathcal{T}_2$ automaton with less than $L(n)$ states accepts $\mathcal{L}(\mathcal{A}_n)$, and $\mathcal{T}_2$ is not Büchi or generalized Büchi, then by Fact 2.1 there is a deterministic ω-automaton of

[5] Deterministic Büchi or generalized Büchi automata are strictly weaker in expressive power than the other common types of ω-automata.

a common type (not necessarily $\mathcal{T}_2$) complementing $\mathcal{L}(\mathcal{A}_n)$ with also less than $L(n)$ states [Löd99], contrary to (i). Finally, the alphabet of $\mathcal{A}_n$ can be fixed like in the proof of Theorem 4.9. □

For the transformations involved in this theorem, less than half already had nontrivial lower bounds like $n!$ by Michel's proof or the bunch of proofs by Löding [Löd99], while the others only have trivial or weak $2^{\Omega(n)}$ lower bounds. These bounds are summarized in Section 6.

5. Complementation of Generalized Büchi Automata

We turn now to NGBW complementation. For NGBWs, state complexity is preferably measured in terms of both the number of states n and index k, where index measures the size of the acceptance condition. By applying full automata, doing a hard case analysis for the construction in [KV05b] based on GC-ranking, and using a generalization of Michel's technique, we prove an $(\Omega(nk))^n$ lower bound, matching with the $(O(nk))^n$ bound in [KV05b]. This lower bound also extends to the complementation of Streett automata and the determinization of generalized Büchi automata into Rabin automata.

5.1. Standard Full Generalized Büchi Automata $\mathcal{FB}_{n,k}$.

We first define full NGBW automata which we will show to witness our desired lower bound.

We say a generalized Büchi acceptance condition $Acc = \{F_1, F_2, \dots, F_k\}$ is *minimal*, if no F_i, F_j pair with $i \neq j$ satisfies that $F_i \subseteq F_j$. Note that if such a pair exists, F_j can be removed from Acc without altering the ω-language defined. So we will only consider minimal acceptance conditions. By the Sperner's theorem in combinatorics [Lub66], if Acc is minimal, then $k \leq \binom{n}{\lfloor n/2 \rfloor}$.

Definition 5.1. For $n > 1$ and $1 < k \leq \binom{n-1}{\lfloor (n-1)/2 \rfloor}$, the *standard full* NGBW $\mathcal{FB}_{n,k} = (\Sigma_n, S_n, I_n, \Delta_n, Acc_{n,k})$ is an NGBW with $|S_n| = n$, $I_n = S_n$ and a minimal acceptance condition $Acc_{n,k}$. Let s_{nf} be one of its state. We denote $S_n \backslash \{s_{nf}\}$ as S'_n. $Acc_{n,k}$ is defined as an arbitrary fixed set $\{F_1, F_2, \dots, F_k\} \subseteq \mathcal{P}(S'_n)$ such that: (i) $|F_i| = \lfloor (n-1)/2 \rfloor$ for each $F_i \in Acc_{n,k}$. (ii) For each $q \in S'_n$, the number of F_i's in $Acc_{n,k}$ that do not contain q is at least $\lfloor k/2 \rfloor$.

We must show that there is really such a minimal $Acc_{n,k}$ satisfying (i) and (ii). First let $Acc_{n,k}$ be a collection of arbitrary k distinct subsets of S'_n of $\lfloor (n-1)/2 \rfloor$ states and thus (i) is satisfied. Define χ_q for each $q \in S'_n$ as the number of F_i's in $Acc_{n,k}$ that contain q. By double counting, $\sum_{q \in S'_n} \chi_q = \sum_{i=1}^{k} |F_i|$. So if $|\chi_p - \chi_q| \leq 1$ for all $p, q \in S'_n$, then for all $q \in S'_n$, $\chi_q \leq \lceil \frac{k \lfloor (n-1)/2 \rfloor}{n-1} \rceil \leq \lceil k/2 \rceil$ and (ii) is also satisfied. Suppose $\chi_p - \chi_q > 1$ for some $p, q \in S'_n$. A little thought shows that there is an $F_i \in Acc_{n,k}$ such that $p \in F_i$ and $(F_i \backslash \{p\}) \cup \{q\} \notin Acc_{n,k}$. Replace F_i in $Acc_{n,k}$ by $(F_i \backslash \{p\}) \cup \{q\}$ and we make $|\chi_p - \chi_q|$ strictly smaller. Repeat this till $|\chi_p - \chi_q| \leq 1$ for all $p, q \in S'_n$. Then condition (ii) is also satisfied.

5.2. **A Generalization of Michel's Technique.** We generalize the technique used in Michel's proof for Büchi complementation [Mic88] so that a tighter analysis of NGBW complementation becomes possible.

Definition 5.2. A *generalized co-Büchi segment* (GC-segment for short) w of an NGBW $\mathcal{B}$ is a word such that $w^\omega \notin \mathcal{L}(\mathcal{B})$. Two GC-segments w_1, w_2 of $\mathcal{B}$ *conflict* if all ω-words in the form $w_1^{k_0}(w_1^{k_1}w_2^{k_2})^\omega, k_i > 0$ are in $\mathcal{L}(\mathcal{B})$. A set W of GC-segments of $\mathcal{B}$ is a *conflict set* for $\mathcal{B}$ if every two distinct GC-segments in W conflict.

Lemma 5.3. *If W is a conflict set for* NGBW *$\mathcal{B}$, then $C_{\mathrm{NGBW}}(\mathcal{B}) \geq |W|$.*

Proof. Suppose that some NGBW $\mathcal{CB} = (\Sigma, \hat{S}, \hat{I}, \hat{\Delta}, \hat{F})$ complements $\mathcal{B}$, then for each GC-segment w of $\mathcal{B}$ in W, $\mathcal{CB}$ accepts w^ω. For every two distinct GC-segments $w_1, w_2 \in W$, let $l_1 = length(w_1)$, $l_2 = length(w_2)$, and let $\rho(0)\rho(1)\ldots$ and $\rho'(0)\rho'(1)\ldots$ be $\mathcal{CB}$'s two successful runs over w_1^ω and w_2^ω respectively. Define

$$\hat{Q}_1 = \{\hat{q} \in \hat{S} \mid \rho(i \cdot l_1) = \hat{q} \text{ for infinitely many } i \in \mathbb{N}\}$$

and

$$\hat{Q}_2 = \{\hat{q} \in \hat{S} \mid \rho'(i \cdot l_2) = \hat{q} \text{ for infinitely many } i \in \mathbb{N}\}.$$

Clearly $\hat{Q}_1$ and $\hat{Q}_2$ are nonempty. It suffices to show that $\hat{Q}_1 \cap \hat{Q}_2 = \emptyset$, since it implies that the number of states of $\mathcal{CB}$ is no less than the number of GC-segments in W.

Suppose by contradiction that some $\hat{q}$ is in $\hat{Q}_1 \cap \hat{Q}_2$. By definition of $\hat{Q}_1$, there is a sufficiently large $k_0 > 0$ such that $\rho(k_0 l_1) = \hat{q}$ and for each $i \geq k_0 l_1$, $\rho(i) \in Inf(\rho)$. So $\rho[0, k_0 l_1]$ is a finite run over $w_1^{k_0}$ from some initial state $\hat{q}_I$ of $\mathcal{CB}$ to $\hat{q}$, i.e., $\hat{q}_I \xrightarrow{w_1^{k_0}} \hat{q}$. By definitions of $\hat{Q}_1$ and $Inf(\rho)$, there is a sufficiently large $k_1 > 0$ such that $\rho((k_0 + k_1)l_1) = \hat{q}$ and in addition $\rho[k \cdot l_1, (k_0 + k_1)l_1]$ is a finite run from $\hat{q}$ to $\hat{q}$ over $w_1^{k_1}$ which visits every state in $Inf(\rho)$. Similarly we have that for some k_0' and $k_2 > 0$, $\rho'[k_0' l_2, (k_0' + k_2)l_2]$ is a finite run from $\hat{q}$ to $\hat{q}$ over $w_2^{k_2}$ which visits exactly every state in $Inf(\rho')$. We construct a new run as follows:

$$\rho_{new} = \rho[0, k_0 l_1] \cdot \left(\rho[k_0 l_1 + 1, (k_0 + k_1)l_1] \cdot \rho'[k_0' l_2 + 1, (k_0' + k_2)l_2]\right)^\omega,$$

which is a run over $\alpha = w_1^{k_0}(w_1^{k_1}w_2^{k_2})^\omega$ with $Inf(\rho_{new}) = Inf(\rho) \cup Inf(\rho')$. As ρ and ρ' are both successful, ρ_{new} is also successful by definition of generalized Büchi automata. So α is accepted by $\mathcal{CB}$. However, as w_1 and w_2 conflict, α is accepted by $\mathcal{B}$ too, contradiction. □

Corollary 5.4. *If W is a conflict set for* NGBW *$\mathcal{B}$, then every* NSW *(nondeterministic Streett automaton) that complements $\mathcal{B}$ has at least $|W|$ states.*

Proof. Streett automata also satisfy that if ρ and ρ' are both successful runs, then every run ρ_{new} satisfying $Inf(\rho_{new}) = Inf(\rho) \cup Inf(\rho')$ is also successful. So the same proof as of Lemma 5.3 applies here. □

5.3. **A Conflict Set for $\mathcal{FB}_{n,k}$.** It remains to define a large conflict set for $\mathcal{FB}_{n,k}$. The following concept of pseudo generalized co-Büchi level ranking is adapted from the concept of generalized co-Büchi level ranking in the NGBW complementation construction in [KV05b].

Definition 5.5. A *pseudo generalized co-Büchi level ranking* (PGCL-ranking for short) for $\mathcal{FB}_{n,k}$ is a pair $\langle f, g\rangle$ such that f is a bijection from S_n' to $\{1, \ldots, n-1\}$ and g is a function from S_n' to $\{1, 2, \ldots, k\}$ such that each $q \in S_n'$ is not contained in $F_{g(q)}$.

By definition of $\mathcal{FB}_{n,k}$, there are at least $\lfloor k/2 \rfloor$ choices for the value of $g(q)$ for each $q \in S'_n$. So there are at least $(n-1)! \times (\lfloor k/2 \rfloor)^{n-1}$ many different PGCL-rankings, which is $(\Omega(nk))^n$ by Stirling's formula.

Let $\mathcal{G}$ be a set of state sets. In the following, we use notations in the form $p \xrightarrow[\mathcal{G},!B]{w} q$ to denote that there is a finite run over w from p to q such that the run visits every state set F in $\mathcal{G}$, but it does *not* visit B. Either $\mathcal{G}$ or B will be omitted if is empty. In the following, we set $\mathcal{F} = \{F_1, \ldots, F_k\}$.

Lemma 5.6. *For each PGCL-ranking $\langle f, g \rangle$, there exists a word $seg_{f,g}$ with the properties that for all $p, q \in S'_n$:*

- **(i):** *If $p = q$, i.e., $f(p) = f(q)$, then there is a unique finite run of $\mathcal{FB}_{n,k}$ over $seg_{f,g}$ from p to q, and it is in the form $p \xrightarrow[\mathcal{F}\backslash F_{g(p)},!F_{g(p)}]{seg_{f,g}} q$.*
- **(ii):** *If $f(p) > f(q)$, then there is a unique finite run of $\mathcal{FB}_{n,k}$ over $seg_{f,g}$ from p to q, and it is in the form $p \xrightarrow[\mathcal{F}]{seg_{f,g}} q$.*
- **(iii):** *If $f(p) < f(q)$, then there is no finite run of $\mathcal{FB}_{n,k}$ from p to q over $seg_{f,g}$.*

Proof. For notational convenience, we use notation like $\left[\begin{smallmatrix} \oplus p_1 \to p_2, \\ \ominus p_3 \to p_4, \ominus p_5 \to p_5 \end{smallmatrix}\right]$ to denote letter $\{\langle q, q \rangle \mid q \in S'_n\} \cup \{\langle p_1, p_2 \rangle\} \backslash \{\langle p_3, p_4 \rangle, \langle p_5, p_5 \rangle\}$. We also define a choice function $c(i, p)$ for each $i \in \{1, \ldots, k\}$ and state $p \in S'_n$ with $g(p) \neq i$ such that $c(i, p)$ equals to some arbitrary fixed element in $F_i \backslash F_{g(p)}$.

For each $r \in \{1, \ldots, n-1\}$, let $p \in S'_n$ be such that $f(p) = r$, and define:

$$u_r = \prod_{\substack{i \neq g(p), 1 \leq i \leq k \\ s = c(i,p)}} \begin{bmatrix} \oplus p \to s, & \ominus p \to p, \\ \oplus s \to s_{nf}, & \ominus s \to s \end{bmatrix} \begin{bmatrix} \oplus s \to p, & \ominus p \to p, \\ \oplus s_{nf} \to s, & \ominus s \to s \end{bmatrix}.$$

(Recall that ΠU means the concatenation of all words in U in lexicographical order.) Then for each $q \in S'_n$, there is a unique finite run over u_r from q to q, and it is in the form $q \xrightarrow[\mathcal{F}\backslash F_{g(p)},!F_{g(p)}]{u_r} q$ if $p = q$, or $q \xrightarrow[!F_{g(p)}]{u_r} q$ otherwise.

For each $r = \{2, 3, \ldots, n-1\}$, let $p, q, s \in S'_n$ be such that $f(p) = r$, $f(q) = r - 1$ and s be an arbitrary state in $F_{g(p)}$. Define:

$$v_r = \begin{bmatrix} \oplus p \to s\ , & \ominus s \to s\ , \\ \oplus s \to s_{nf} & \end{bmatrix} \begin{bmatrix} \oplus s \to q\ , & \ominus s \to s\ , \\ \oplus s_{nf} \to s & \end{bmatrix}.$$

Then there is a unique finite run over v_r from p to q, and it is in the form $p \xrightarrow[F_{g(p)}]{v_r} q$. Also for every $q' \in S'_n$, there is a unique finite run over v_r from q' to q', and it is in the form $q' \xrightarrow[!F_{g(p)}]{v_r} q'$.

Finally let $seg_{f,g}$ be $u_{n-1} v_{n-1} u_{n-2} v_{n-2} \ldots v_2 u_1$.

To see that $seg_{f,g}$ satisfies the required properties, first note that for all $p \in S'_n$, $p \xrightarrow[!F_{g(p)}]{u_r} p$ and $p \xrightarrow[!F_{g(p)}]{v_r} p$. For property (i), for every $p \in S'_n$ with $f(p) = r$, there exists a unique finite run over $seg_{f,g}$, and it is in the form:

$$p \xrightarrow[!F_{g(p)}]{u_{n-1} v_{n-1} \ldots u_{r+1} v_{r+1}} p \xrightarrow[\mathcal{F}\backslash F_{g(p)},!F_{g(p)}]{u_r} p \xrightarrow[!F_{g(p)}]{v_r u_{r-1} \ldots v_2 u_1} p,$$

that is, $p \xrightarrow[\mathcal{F}\backslash F_{g(p)},!F_{g(p)}]{seg_{f,g}} p$ as required. For property (ii), for every $p, q \in S'_n$ with $f(p) = r_1 > r_2 = f(q)$, let $s_r \in S'_n$ be such that $f(s_r) = r$ for each $r_1 > r > r_2$. There is a unique finite run over $seg_{f,g}$, and it is in the form:

$$p \xrightarrow{u_{n-1}v_{n-1}\dots u_{r_1+1}v_{r_1+1}} p \xrightarrow[\mathcal{F}\backslash F_{g(p)},!F_{g(p)}]{u_{r_1}} p \xrightarrow[F_{g(p)}]{v_{r_1}} s_{r_1-1}$$
$$\xrightarrow{u_{r_1-1}v_{r_1-1}} s_{r_1-2}\dots s_{r_2+1} \xrightarrow{u_{r_2+1}v_{r_2+1}} q \xrightarrow{u_{r_2}\dots v_2u_1} q,$$

that is, $p \xrightarrow[\mathcal{F}]{seg_{f,g}} q$ as required. Property (iii) is easy to verify. □

Remark 5.7. From the proof of the above lemma, it follows that an alphabet of size polynomial in n is sufficient to describe $\{seg_{f,g} | f, g \text{ are PGCL-rankings}\}$.

Lemma 5.8. *For each PGCL-ranking $\langle f, g\rangle$ for $\mathcal{FB}_{n,k}$, word $seg_{f,g}$ is a GC-segment of $\mathcal{FB}_{n,k}$.*

Proof. Let $l = length(seg_{f,g})$, and let $\rho = \rho(0)\rho(1)\dots$ be a run of $\mathcal{FB}_{n,k}$ over $seg^{\omega}_{f,g}$ in the form $\rho(0) \xrightarrow{seg_{f,g}} \rho(l) \xrightarrow{seg_{f,g}} \rho(2l)\dots$. Note that by the construction of $seg_{f,g}$, $\rho(i\cdot l) \in S'_n$ and $f(\rho(i \cdot l))$ is defined for all $i \geq 0$. Then by property (iii), $f(\rho(0)) \geq f(\rho(l)) \geq f(\rho(2l)) \geq \dots$ and then for some $t \in \mathbb{N}$, $f(\rho(t' \cdot l)) = f(\rho(t \cdot l))$ for all $t' > t$, that is $\rho(t' \cdot l) = \rho(t \cdot l)$ for all $t' > t$ since f is a bijection. Let $j = g(\rho(t \cdot l))$. By property (i), F_j is not visited in $\rho[t' \cdot l, (t' + 1) \cdot l]$ for all $t' \geq t$. So $Inf(\rho) \cap F_j = \emptyset$ and hence $seg^{\omega}_{f,g}$ is not accepted by $\mathcal{FB}_{n,k}$. □

Lemma 5.9. *The set $W = \{seg_{f,g} \mid \langle f, g\rangle$ is a PGCL-ranking for $\mathcal{FB}_{n,k}\}$ is a conflict set of size $(\Omega(nk))^n$ for $\mathcal{FB}_{n,k}$.*

Proof. Suppose $\langle f_1, g_1\rangle$ and $\langle f_2, g_2\rangle$ are two distinct PGCL-rankings. Let $w_1 = seg_{f_1,g_1}$ and $w_2 = seg_{f_2,g_2}$. There are two cases.

Case: I: f_1 and f_2 are two different bijections. So there exist $p, q \in S'_n$ such that $f_1(p) > f_1(q)$ and $f_2(p) < f_2(q)$. By property (i), $p \xrightarrow{w_1} p$, $q \xrightarrow{w_2} q$ and so $p \xrightarrow{w_1^{m-1}} p$, $q \xrightarrow{w_2^{m-1}} q$ for all $m > 0$. By property (ii), $p \xrightarrow[\mathcal{F}]{w_1} q$ and $q \xrightarrow[\mathcal{F}]{w_2} p$. So for all $m > 0$, $p \xrightarrow[\mathcal{F}]{w_1^m} q$ and $q \xrightarrow[\mathcal{F}]{w_2^m} p$. Now for every ω-word α in the form $w_1^{k_0}(w_1^{k_1}w_2^{k_2})^{\omega}$, $k_i > 0$, we construct a successful run over α as $p \xrightarrow{w_1^{k_0}} p \xrightarrow[\mathcal{F}]{w_1^{k_1}} q \xrightarrow[\mathcal{F}]{w_2^{k_2}} p \xrightarrow[\mathcal{F}]{w_1^{k_1}} q \xrightarrow[\mathcal{F}]{w_2^{k_2}} p\dots$. So α is accepted by $\mathcal{FB}_{n,k}$ and w_1 conflicts with w_2.

Case: II: $f_1 = f_2$ but $g_1 \neq g_2$. Let $p \in S'_n$ be such that $g_1(p) \neq g_2(p)$. By property (i), $p \xrightarrow[\mathcal{F}\backslash F_{g_1(p)},!F_{g_1(p)}]{w_1} p$ and $p \xrightarrow[\mathcal{F}\backslash F_{g_2(p)},!F_{g_2(p)}]{w_2} p$. As $g_1(p) \neq g_2(p)$, $p \xrightarrow[\mathcal{F}]{w_1^{k_1}w_2^{k_2}} p$ for every $k_1, k_2 > 0$. Now for every ω-word α in the form $w_1^{k_0}(w_1^{k_1}w_2^{k_2})^{\omega}$, $k_i > 0$, we construct a successful run over α as $p \xrightarrow{w_1^{k_0}} p \xrightarrow[\mathcal{F}]{w_1^{k_1}w_2^{k_2}} p \xrightarrow[\mathcal{F}]{w_1^{k_1}w_2^{k_2}} p\dots$. So α is accepted by $\mathcal{FB}_{n,k}$ and w_1 conflicts with w_2.

Finally, the size of W is just the number of different PGCL-rankings for $\mathcal{FB}_{n,k}$, which is $(\Omega(nk))^n$. □

5.4. **Results.**

Theorem 5.10. *For $n > 1$ and $1 < k \leq \binom{n-1}{\lfloor (n-1)/2 \rfloor}$, $C_{\text{NGBW}}(n,k) = (\Omega(nk))^n$.*

Proof. The theorem follows from Lemma 5.3 and Lemma 5.9 directly. □

This matches neatly[6] with the $(O(nk))^n$ construction in [KV05b], and thus settles the state complexity of NGBW complementation. Like Michel's result, this lower bound can be extended to NSW complementation and the determinization of NGBW into DRW (state complexity denoted by $D_{\text{NGBW}\to\text{DRW}}(n,k)$):

Theorem 5.11. *For all $n > 1$ and $1 < k \leq \binom{n-1}{\lfloor (n-1)/2 \rfloor}$, $C_{\text{NSW}}(n,k) = (\Omega(nk))^n$ and $D_{\text{NGBW}\to\text{DRW}}(n,k) = (\Omega(nk))^n$.*

Proof. By Fact 2.1 there is an NSW $\mathcal{S}_{n,k}$ equivalent to each $\mathcal{FB}_{n,k}$ with the same number of states and the same index. By Corollary 5.4 and Lemma 5.9, every NSW that complements $\mathcal{FB}_{n,k}$ has $(\Omega(nk))^n$ states. So $C_{\text{NSW}}(\mathcal{S}_{n,k}) = (\Omega(nk))^n$ and $C_{\text{NSW}}(n,k) = (\Omega(nk))^n$.

Suppose by contradiction that $\mathcal{R}$ is a DRW with less than $|W|$ states that accepts $\mathcal{L}(\mathcal{FB}_{n,k})$, then by Fact 2.1 there is a DSW $\mathcal{S}$ complementing $\mathcal{FB}_{n,k}$ with the same number of states as $\mathcal{R}$, contrary to Corollary 5.4. So $D_{\text{NGBW}\to\text{DRW}}(n,k) = (\Omega(nk))^n$. □

Remark 5.12. For the above lower bound, by Remark 5.7, the alphabet involved in the proof is of a size polynomial in n. It seems difficult to fix a constant alphabet, but we conjecture this to be possible if we aim at a weaker bound like $2^{\Omega(n \log nk)}$.

6. Summary

In the following table, we briefly summarize our lower bounds. Here "Any" means any common type of nondeterministic ω-automata (and the two Any's can be different). "co." means complementation and "det." means determinization. "L.B." /"U.B." stands for lower/upper bound. Weak $2^{\Omega(n)}$ lower bounds are considered trivial.

#	Transformation	Previous L.B.	Our L.B.	Known U.B.
1	NBW $\xrightarrow{\text{co.}}$ NBW	$\Omega((0.36n)^n)$ [Mic88]	$\Omega((0.76n)^n)$	$O((0.97n)^n)$ [FKV06]
2	Any $\xrightarrow{\text{co. or det.}}$ Any	trivial or $n!$ [Löd99]	$2^{\Omega(n \log n)}$	-
3	NBW $\xrightarrow{\text{det.}}$ DMW	trivial[7]	$2^{\Omega(n \log n)}$	$2^{O(n \log n)}$ [Saf89]
4	NRW $\xrightarrow{\text{co.}}$ NRW	trivial[8]	$2^{\Omega(n \log n)}$	$2^{O(nk \log n)}$ [KV05a]
5	NGBW $\xrightarrow{\text{co.}}$ NGBW	$\Omega((n/e)^n)$ [Mic88]	$(\Omega(nk))^n$	$(O(nk))^n$ [KV05b]
6	NSW $\xrightarrow{\text{co.}}$ NSW	$\Omega((n/e)^n)$ [Löd99]	$(\Omega(nk))^n$	$2^{O(nk \log(nk))}$ [KV05a]
7	NGBW $\xrightarrow{\text{det.}}$ DRW	$\Omega((n/e)^n)$ [Löd99]	$(\Omega(nk))^n$	$2^{O(nk \log(nk))}$ [Saf89]

In particular, lower bound #2 implies that the $2^{\Omega(n \log n)}$ blow-up is inherent in the complementation and determinization of nondeterministic ω-automata, corresponding to the 2^n blow-up of finite automata. The special case #3 justifies that Safra's construction is optimal in state complexity for the determinization of Büchi automata into Muller automata.

[6]The gap hidden in the notation $(\Theta(nk))^n$ can be at most c^n for some c, while the gap hidden in the more widely used notation $2^{\Theta(n \log nk)}$ can be as large as $(nk)^n$.

We single out this result because this determinization construction is touched in almost every introductory material on ω-automata, and its optimality problem was explicitly left open in [Löd99].

For many of these transformations, it is still interesting to try to narrow the complexity gap, and here we discuss three of them. First, the complexity gap of Büchi complementation, although significantly narrowed, is still exponential. By analyzing the difference between the lower and upper bounds, one can find that the gap is mainly caused by the use of the state component O in [FKV06] to maintain the states along paths that have not visited an odd vertex since the last time O has been empty. So we should investigate how many states are really necessary for such a purpose. Second, for Streett complementation, the gap is still quite large. We feel that efforts should be first taken to optimize the construction in [KV05a]. Third, it is interesting to see if an $\Omega(n^n)$ or similar lower bound exists for the determinization of NBWs into Muller or Rabin automata. Such would imply that determinization is harder than complementation for ω-automata, unlike the case of automata over finite words. Of course, one can also work on the reverse direction, trying to design ranking based constructions for determinization, which could have good complexity bound as well as better applicability to practice.

Finally, we remark that the full automata technique has been quite essential in obtaining our lower bound results. It is also possible to extend the full automata technique to other kinds of automata, like alternating automata or tree automata. We hope that the full automata technique will stimulate the discovery of new results in automata theory.

Acknowledgement. I thank Orna Kupferman and Moshe Vardi for the insightful discussion and the extremely valuable suggestions. I thank Enshao Shen for his kind support and guidance. I also thank the anonymous referees for the detailed and useful comments.

References

[Bir93] J.C. Birget. Partial orders on words, minimal elements of regular languages and state complexity (has online erratum). *Theoretical Computer Science*, 119(2):267–291, 1993.

[Büc62] J. R. Büchi. On a decision method in restricted second order arithmetic. In *Proceedings of the International Congress on Logic, Method, and Philosophy of Science*, pages 1–12. Stanford University Press, 1962.

[FKV06] E. Friedgut, O. Kupferman, and M.Y. Vardi. Büchi complementation made tighter. *International Journal of Foundations of Computer Science*, 17(4):851–868, 2006.

[HK02] M. Holzer and M. Kutrib. State complexity of basic operations on nondeterministic finite automata. In *Proceedings of 7th International Conference on Implementation and Application of Automata*, volume 2608 of *Lecture Notes in Computer Science*, pages 148–157, 2002.

[Jir05] G. Jirásková. State complexity of some operations on binary regular languages. *Theoretical Computer Science*, 330(2):287–298, 2005.

[Kla91] N. Klarlund. Progress measures for complementation of omega-automata with applications to temporal logic. In *Proceedings of 32th IEEE Symposium on Foundations of Computer Science*, pages 358–367, 1991.

[Kur94] R.P. Kurshan. *Computer-Aided Verification of Coordinating Processes: The Automata-Theoretic Approach*. Princeton Univ. Press, 1994.

[KV01] O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. *ACM Transactions on Computational Logic*, 2(3):408–429, 2001.

[KV05a] O. Kupferman and M.Y. Vardi. Complementation constructions for nondeterministic automata on infinite words. In *Proceedings of 11th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 3440 of *Lecture Notes in Computer Science*, pages 206–221, 2005.

[KV05b] O. Kupferman and M.Y. Vardi. From complementation to certification. *Theoretical Computer Science*, 345(1):83–100, 2005.
[Löd99] C. Löding. Optimal bounds for transformations of omega-automata. In *Proceedings of 19th Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 1738 of *Lecture Notes in Computer Science*, pages 97–109, 1999.
[Lub66] D. Lubell. A short proof of Sperner's lemma. *Journal of Combinatorial Theory*, 1:299, 1966.
[Mic88] M. Michel. Complementation is more difficult with automata on infinite words. CNET, Paris, 1988.
[RS59] M.O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:114–125, 1959.
[Saf88] S. Safra. On the complexity of ω-automata. In *Proceedings of 29th IEEE Symposium on Foundations of Computer Science*, pages 319–327, 1988.
[Saf89] S. Safra. *Complexity of automata on infinite objects*. PhD thesis, Weizmann Institute of Science, 1989.
[SS78] W.J. Sakoda and M. Sipser. Nondeterminism and the size of two way finite automata. In *Proceedings of 10th ACM Symposium on Theory of Computing*, pages 275–286, 1978.
[SVW85] A.P. Sistla, M.Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic (extended abstract). In *Proceedings of 12th International Colloquium on Automata, Languages and Programming*, volume 194 of *Lecture Notes in Computer Science*, pages 465–474, 1985.
[Tem93] N.M. Temme. Asymptotic estimates of stirling numbers. *Studies in Applied Mathematics*, 89:233–243, 1993.
[Tho90] W. Thomas. Automata on infinite objects. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, Formal models and semantics, pages 133–191. Elsevier, 1990.
[Var07] M.Y. Vardi. The Büchi complementation saga. In *Proceedings of 23rd International Symposium on Theoretical Aspects of Computer Science*, volume 4393 of *Lecture Notes in Computer Science*, pages 12–22, 2007.
[VW94] M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.
[Yu05] S. Yu. State complexity: Recent results and open problems. *Fundamenta Informaticae*, 64:471–480, 2005.

Appendix A. Numerical Analysis of $L(n)$

In this section, we prove that $L(n) = \Theta((0.76n)^n)$. The analysis is very similar to the one in [FKV06], but we still present it here for completeness. In the following, we write $f(n) \approx g(n)$ if two functions differ by only a polynomial factor in n. For example, by Stirling's formula, $n! \approx (n/e)^n$.

Let $T(n,m)$ denote the number of functions from $\{1 \dots n\}$ onto $\{1 \dots m\}$. The following estimate of $T(n,m)$ is implicit in Temme [Tem93]:

Lemma A.1. [Tem93]*For $0 < \beta < 1$, let x be the positive real number solving $\beta x = 1 - e^{-x}$, and let $a = -\ln x + \beta \ln(e^x - 1) - (1-\beta) + (1-\beta)\ln(1/\beta - 1)$. Then $T(n, \lfloor \beta n \rfloor) \approx (M[\beta]\, n)^n$, where $M[\beta] = e^{a-\beta}\left(\frac{\beta}{1-\beta}\right)^{1-\beta}$.*

To prove a lower bound for $L(n)$, we first express $L(n,m)$ in the following form:

Lemma A.2. $L(n,m) = \sum_{t=m}^{n-1} \binom{n-1}{t} T(t,m) m^{n-1-t}$.

Proof. To count the number of different $Q(m)$-ranking, we fix t, which denotes the number of states that have odd ranks. Then there are $\binom{n-1}{t}$ ways to choose which t states have odd ranks, and there are $T(t,m)$ ways to assign these t states the m different odd ranks.

Moreover, for each of the other $n-1-t$ states in S'_n, there are m ways to choose which even rank it is assigned. $\square$

Theorem A.3. $L(n) = \Omega((c_l n)^n)$, *where* $c_l = 0.76$.

Proof. By the previous lemma, $L(n) = \max\limits_{m=1\ldots n-1} \sum_{t=m}^{n-1} \binom{n-1}{t} T(t,m) m^{n-1-t}$. Since we do not care about polynomial factors, $\sum_{t=m}^{n-1}$ can be replaced by $\max\limits_{t=m\ldots n-1}$, and we can replace $m!$ by $(m/e)^m$ and $\binom{n-1}{t}$ by $\frac{n^n}{t^t(n-t)^{n-t}}$ as well. Also let $\gamma = m/n$ and $\beta = t/n$, then we have: $\square$

$$L(n) \approx \max_{0<\gamma\leq\beta<1} n^n (\beta n)^{-\beta n} ((1-\beta)n)^{-(1-\beta)n} \cdot (M[\gamma/\beta]\beta n)^{\beta n} \cdot (\gamma n)^{n-1-\beta n}$$
$$\approx \max_{0<\gamma\leq\beta<1} (h(\beta,\gamma)n)^n, \text{ where } h(\beta,\gamma) = (1-\beta)^{\beta-1}(M[\gamma/\beta])^\beta \gamma^{1-\beta}.$$

Computed by the Mathematica software, $h(\beta,\gamma) = 0.7645$ when $\beta = 0.7236, \gamma = 0.5744$. So $(0.76n)^n$ is an asymptotic lower bound for $L(n)$.