Improved Upper and Lower Bounds for Büchi Disambiguation

Hrishikesh Karmarkar[1], Manas Joglekar[2], and Supratik Chakraborty[1]

[1] Department of Computer Science and Engineering, IIT Bombay
[2] Department of Computer Science, Stanford University

Abstract. We present a new ranking based construction for disambiguating non-deterministic Büchi automata and show that the state complexity tradeoff of the translation is in $O(n \cdot (0.76n)^n)$. This exponentially improves the best upper bound (i.e., $4 \cdot (3n)^n$) known earlier for Büchi disambiguation. We also show that the state complexity tradeoff of translating non-deterministic Büchi automata to strongly unambiguous Büchi automata is in $\Omega((n-1)!)$. This exponentially improves the previously known lower bound (i.e. $\Omega(2^n)$). Finally, we present a new technique to prove the already known exponential lower bound for disambiguating automata over finite or infinite words. Our technique is significantly simpler than earlier techniques based on ranks of matrices used for proving disambiguation lower bounds.

1 Introduction

Unambiguous Büchi automata over infinite words represent an interesting class of automata that are structurally situated between deterministic and non-deterministic Büchi automata, and yet are as expressive as non-deterministic Büchi automata. For notational convenience, we use UBA (respectively, NBA) to denote the class of unambiguous (respectively, non-deterministic) Büchi automata over infinite words in the rest of this paper. The expressive equivalence of UBA and NBA was first shown by Arnold [1], and later re-proven by Carton and Michel [3] and Kahler and Wilke [6]. Bousquet and Löding [2] showed that language equivalence and inclusion checking can be achieved in polynomial time for a sub-class of UBA, called strongly unambiguous Büchi automata (or SUBA), which is expressively equivalent to NBA. In later work [5], two other incomparable sub-classes of UBA were also shown to admit polynomial-time language inclusion and equivalence checking. The class of automata studied by Carton and Michel have also been called *prophetic* automata by others [4]. In a recent work, Preugschat and Wilke [11] have described a framework for characterizing fragments of linear temporal logic (LTL). Their characterization relies heavily on the use of prophetic automata and special Ehrenfeucht-Fraïssé games.

Despite the long history of studies on UBA (including several papers in recent years), important questions about disambiguation still remain open. Notable among these are the exact state complexity trade-offs in translating NBA to language-equivalent UBA or SUBA. The state complexity trade-off question asks

D. Van Hung and M. Ogawa (Eds.): ATVA 2013, LNCS 8172, pp. 40–54, 2013.

"Given an n-state NBA, how many states must a language-equivalent UBA (resp., SUBA) have as a function of n?" The present work attempts to address these questions, and makes the following contributions.

1. We show that the NBA to UBA state complexity trade-off is in $O(n \cdot (0.76n)^n)$. This is exponentially more succinct than the previously known best trade-off of $4 \cdot (3n)^n$ due to Kahler and Wilke [6]. The improved upper-bound is obtained by extending Kupferman and Vardi's ranking function based techniques [9] to the construction of unambiguous automata.
2. We show that the NBA to SUBA state complexity trade-off is in $\Omega((n-1)!)$. This is exponentially larger than the previously known best lower bound of $2^n - 1$ due to Schmidt [13]. The improved lower bound is obtained by a full-automaton technique [14].
3. We present a new technique for proving the already known fact that the NBA to UBA state complexity trade-off is at least $2^n - 1$. Our proof generalizes to all common notions of acceptance for finite and infinite words, and is conceptually simpler than the earlier proof based on ranks of matrices due to Schmidt [13].

2 Notation and Preliminaries

An NBA is a 5-tuple $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$, where Σ is a finite alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, $\delta : Q \times \Sigma \to 2^Q$ is the state transition relation and $F \subseteq Q$ is a set of accepting or final states. For notational convenience, we often use (with abuse of notation) $\delta(S, a)$ to denote $\bigcup_{q \in S} \delta(q, a)$ for $S \subseteq Q$. Given a word $\alpha \in \Sigma^\omega$ (also called an ω-word), let $\alpha(j)$ denote the j^{th} letter of α. By convention, we say that $\alpha(0)$ is the first letter of α. A run ρ of $\mathcal{A}$ on α is an infinite sequence of states $q_0 q_1 q_2 \ldots$ such that $q_{i+1} \in \delta(q_i, \alpha(i))$ and $q_i \in Q$ for all $i \geq 0$. Given a run ρ of $\mathcal{A}$ on α, let $\rho(j)$ denote the j^{th} state along ρ. The set $\mathit{inf}(\rho)$ is the set of states of $\mathcal{A}$ that appear infinitely often along ρ. A run ρ is called *final* if $\mathit{inf}(\rho) \cap F \neq \emptyset$; it is called *accepting* if it is final and $q_0 \in Q_0$. An ω-word α is said to be accepted by NBA $\mathcal{A}$ iff there is an accepting run of $\mathcal{A}$ on α. The set of ω-words accepted by $\mathcal{A}$ is called the language of $\mathcal{A}$ and is denoted $L(\mathcal{A})$. A state q_s of an NBA is called a *principal sink* if the following conditions hold: (i) q_s is non-final, (ii) every state, including q_s, has an outgoing transition on every $a \in \Sigma$ to q_s, and (iii) q_s has no outgoing transitions to any state other than q_s. It is easy to see that every NBA can be converted to a language-equivalent NBA with a principal sink by adding at most one state. Unless otherwise stated, we assume that all NBAs considered in this paper have a principal sink.

This paper concerns unambiguous and strongly unambiguous Büchi automata. An unambiguous Büchi automaton (UBA) is an NBA that has at most one accepting run for every $\alpha \in \Sigma^\omega$. An NBA is a *strongly unambiguous Büchi automaton* (SUBA) if it has at most one final run for every $\alpha \in \Sigma^\omega$. Clearly, a SUBA is a special kind of UBA.

Given an NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ and an ω-word α, the *run-DAG* of $\mathcal{A}$ on α is a directed acyclic graph, denoted $G_\alpha^\mathcal{A}$, with vertices in $Q \times \mathbb{N}$. The vertices of graph $G_\alpha^\mathcal{A}$ are defined level-wise as follows. We define *level* 0 of $G_\alpha^\mathcal{A}$ to be $L_0 = \{(q,0) \mid q \in Q_0\}$. For $i \geq 1$, *level* i of $G_\alpha^\mathcal{A}$ is defined inductively as $L_i = \{(q,i) \mid \exists (q', i-1) \in L_{i-1}$ such that $q \in \delta(q', \alpha(i-1))\}$. The run-DAG $G_\alpha^\mathcal{A}$ is given by (V, E), where $V = \bigcup_{i \geq 0} L_i$ is the set of vertices and $E = \{((q,\ i), (q',\ i+1)) \mid (q,\ i) \in V,\ (q',\ i+1) \in V,\ q' \ \in \delta(q,\ \alpha(i))\}$ is the set of edges. It is easy to see that every path in $G_\alpha^\mathcal{A}$ corresponds to a run of $\mathcal{A}$ (from an initial state) on α, and vice versa. We call vertex (q,i) an F-vertex (or *final*-vertex) if $q \in F$. Vertex (q,j) is said to be a *successor* of vertex (q,i), or *reachable* from vertex (q,i), if there is a directed path in $G_\alpha^\mathcal{A}$ from (q,i) to (q,j). If, in addition, $j = i+1$, vertex (q,j) is called an *immediate successor* of (q,i). For notational convenience, for every natural number $n \geq 1$, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$, $[n]^{odd}$ (resp., $[n]^{even}$) to denote the set of odd (resp., even) integers in $[n]$, and $\langle n \rangle$ to denote the set $[n] \cup \{\infty\}$, where $\infty > j$ for all $j \in [n]$.

2.1 Full Rankings

In [9], Kupferman and Vardi showed that given an NBA $\mathcal{A}$ with n states and a word $\alpha \in \Sigma^\omega$, there exists a family of odd ranking functions that assign ranks in $[2n]$ to the vertices of $G_\alpha^\mathcal{A}$ such that $\alpha \notin L(\mathcal{A})$ iff all infinite runs of $\mathcal{A}$ on α that start from its initial states get trapped in odd ranks. We extend the notion of odd rankings and define a *full-ranking* of $G_\alpha^\mathcal{A} = (V, E)$ as a function $r : V \to \langle 2n \rangle$ that satisfies the following conditions: (i) for every $(q,i) \in V$, if $r((q,i)) \in [2n]^{odd}$ then $q \notin F$, (ii) for every edge $((q,i),(q',i+1)) \in E$, $r((q',i+1)) \leq r((q,i))$, and (iii) every infinite path in $G_\alpha^\mathcal{A}$ eventually gets trapped in a rank in $\{\infty\} \cup [2n]^{odd}$, with at least one path trapped in ∞ iff $w \in L(\mathcal{A})$. The remainder of the discussion in this section closely parallels that in [9,7], where ranking based complementation techniques for NBA were described.

For every $\alpha \in \Sigma^\omega$, we define a unique full-ranking, $\mathsf{r}^\star_{\mathcal{A},\alpha}$, of $G_\alpha^\mathcal{A}$ along the same lines as the definition of the unique odd ranking $\mathsf{r}^{\mathsf{KV}}_{\mathcal{A},\alpha}$ in [9,7]. Specifically, we define a sequence of DAGs $G_0 \supseteq G_1 \supseteq \ldots$, where $G_0 = G_\alpha^\mathcal{A}$. A vertex v is *finite* in G_i if there are no infinite paths in G_i starting from v, while v is *F-free* in G_i if it is not finite and there is no F-vertex (q,l) that is reachable from v in G_i. The DAGs G_i are now inductively defined as follows.

- For every $i \geq 0$, $G_{2i+1} = G_{2i} \setminus \{(q,l) \mid (q,l)$ is finite in $G_{2i}\}$.
- For every $i \geq 0$, if G_{2i+1} has at least one F-free vertex, then $G_{2i+2} = G_{2i+1} \setminus \{(q,l) \mid (q,l)$ is F-free in $G_{2i+1}\}$. Otherwise, G_{2i+2} is the empty DAG.

A full-ranking function $\mathsf{r}^\star_{\mathcal{A},\alpha}$ can now be defined as follows. For every $i \geq 0$,

- If G_{2i} has at least one finite vertex, then $\mathsf{r}^\star_{\mathcal{A},\alpha}((q,l)) = 2i$ for every vertex (q,l) that is finite in G_{2i}. Otherwise, no vertex is ranked $2i$ by $r^*_{\mathcal{A},\alpha}$.
- If G_{2i+1} has at least one F-free vertex, then $\mathsf{r}^\star_{\mathcal{A},\alpha}((q,l)) = 2i+1$ for every vertex (q,l) that is F-free in G_{2i+1}. Otherwise, $\mathsf{r}^\star_{\mathcal{A},\alpha}((q,l)) = \infty$ for every vertex (q,l) in G_{2i+1}.

Using the same arguments as used in [9], it can be shown that if $\mathcal{A}$ has n states, the maximum finite (i.e., non-∞) rank in the range of $\mathsf{r}^\star_{\mathcal{A},\alpha}$ is in $[2n]$. In the subsequent discussion, we use $\mathsf{FullRankProc}$ to refer to the above "technique" for assigning full-ranks to vertices of a run-DAG.

Analogous to the concept of a *level-ranking* defined in [9], we define a *full-level ranking* as a function $f : Q \to \langle 2n \rangle \cup \{\bot\}$, such that for every $q \in Q$, if $f(q) \in [2n]^{odd}$, then $q \notin F$. Let FL represent the set of all full-level rankings and FL_∞ represent the subset of FL containing only those full-level rankings f such that $f^{-1}(\infty) \neq \emptyset$. Given two full-level rankings g_1 and g_2, and a letter $a \in \Sigma$, g_2 is said to be a *full-cover* of (g_1, a) if for all $q \in Q$ such that $g_1(q) \neq \bot$ and for all $q' \in \delta(q, a)$, $g_2(q') \leq g_1(q)$. A full-ranking r of $G^{\mathcal{A}}_\alpha$ induces a full-level ranking for every level $l \geq 0$ of $G^{\mathcal{A}}_\alpha$ such that all states not in level l of $G^{\mathcal{A}}_\alpha$ are assigned rank $\bot$. It is easy to see that if g and g' are full-level rankings for levels l and $l+1$ respectively, induced by a full-ranking r, then g' is a full-cover of $(g, \alpha(l))$. Let *max_odd*(g) (resp., *max_rank*(g)) denote the highest odd rank (resp., highest rank) in the range of full-level ranking g. A full-level ranking g is said to be *tight* if the following conditions hold: (i) *max_rank*(g) is in $[2n]^{odd} \cup \{\infty\}$, and (ii) for all $i \in [2n]^{odd}$ such that $i \leq$ *max_rank*(g), there is a state $q \in Q$ with $g(q) = i$.

The ranking $\mathsf{r}^\star_{\mathcal{A},\alpha}$ has several interesting properties that collectively characterize it. These are described in Lemma 1. Due to space restrictions, we are unable to include proofs of all lemmas and theorems in the paper. All proofs omitted from the paper can be found in [8].

Lemma 1. *Let $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ be an NBA, and $\alpha \in \Sigma^\omega$. Let (q, l) be a vertex in $G^{\mathcal{A}}_\alpha$. For every $l \in \mathbb{N}$ and $q \in Q$, we have the following.*

1. *There exists a level $l^* > 0$ such that all full-level rankings induced by $\mathsf{r}^\star_{\mathcal{A},\alpha}$ for levels $l > l^*$ are tight.*
2. *If (q, l) is not an F-vertex or $\mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) = \infty$, there exists $q' \in \delta(q, \alpha(l))$ such that $\mathsf{r}^\star_{\mathcal{A},\alpha}((q', l+1)) = \mathsf{r}^\star_{\mathcal{A},\alpha}((q, l))$.*
3. *If (q, l) is an F-vertex with rank $\mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) \in [2n]^{even}$, there exists a vertex $(q', l+1)$ such that $q' \in \delta(q, \alpha(l))$ and either $\mathsf{r}^\star_{\mathcal{A},\alpha}((q', l+1)) = \mathsf{r}^\star_{\mathcal{A},\alpha}((q, l))$ or $\mathsf{r}^\star_{\mathcal{A},\alpha}((q', l+1)) = \mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) - 1$.*
4. *If $\mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) \neq \infty$, there is no $q' \in \delta(q, \alpha(l))$ such that $\mathsf{r}^\star_{\mathcal{A},\alpha}((q', l+1)) = \infty$.*
5. *If $\mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) \in [2n]^{even}$, every path starting from (q, l) in $G^{\mathcal{A}}_\alpha$ eventually visits a vertex (q', l') such that $1 \leq \mathsf{r}^\star_{\mathcal{A},\alpha}((q', l')) < \mathsf{r}^\star_{\mathcal{A},\alpha}((q, l))$.*
6. *If $\mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) \in [2n]^{odd}$ and $\mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) > 1$, there exists a (q', l') such that (q', l') is an F-vertex reachable from (q, l) in $G^{\mathcal{A}}_\alpha$, and $\mathsf{r}^\star_{\mathcal{A},\alpha}((q', l')) = \mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) - 1$.*
7. *If $\mathsf{r}^\star_{\mathcal{A},\alpha}((q, l)) = \infty$, there exists a (q', l') such that (q', l') is an F-vertex reachable from (q, l) in $G^{\mathcal{A}}_\alpha$, and $\mathsf{r}^\star_{\mathcal{A},\alpha}((q', l')) = \infty$.*

Properties 2, 3 and 4 in the above Lemma can be checked by examining consecutive levels of the ranked run-DAG; hence these are *local* properties. In contrast, checking properties 5, 6 and 7 requires examining an unbounded fragment of the ranked run-DAG; hence these are *global* properties.

3 Improved Upper Bound by Rank Based Disambiguation

The main contribution of this section is a ranking function based algorithm, called $\mathsf{BüchiDisambiguate}$, that takes as input an NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ with $|Q| = n$, and constructs a UBA $\mathcal{U} = (\Sigma, Z, Z_0, \delta_U, F_U)$ such that (i) $L(\mathcal{U}) = L(\mathcal{A})$, and (ii) $|Z| \in O(n \cdot (0.76n)^n)$. Without loss of generality, we assume that $Q = \{q_0, q_1, \ldots q_{n-1}\}$, $Q_0 = \{q_0\}$ and $q_0 \notin F$. For notational convenience, we use "$\mathcal{A}$-states" (resp., "$\mathcal{U}$-states") to refer to states of $\mathcal{A}$ (resp., states of $\mathcal{U}$) in the following discussion.

3.1 Overview

Drawing motivation from Schewe's work [12], we define a state of $\mathcal{U}$ to be a 4-tuple (f, O, X, i), where $i \in \langle 2n \rangle$, $f : Q \to \langle 2n \rangle \cup \{\perp\}$ is a FL_∞ ranking, and O and X are subsets of Q containing $\mathcal{A}$-states that are ranked i by f and satisfy certain properties. Since every state of $\mathcal{U}$ gives a full-level ranking of $\mathcal{A}$, a run of $\mathcal{U}$ gives an infinite sequence of full-level rankings of $\mathcal{A}$, which can be "stitched" together to potentially obtain a full-ranking of $G_\alpha^{\mathcal{A}}$. The purpose of algorithm $\mathsf{BüchiDisambiguate}$ is to define the transitions and final states of $\mathcal{U}$ in such a way that a run of $\mathcal{U}$ on $\alpha \in \Sigma^\omega$ is accepting iff the "stitched" full-ranking of $G_\alpha^{\mathcal{A}}$ obtained from the run is exactly $\mathsf{r}^\star_{\mathcal{A},\alpha}$, as defined in Section 2.1.

Informally, algorithm $\mathsf{BüchiDisambiguate}$ works as follows. Suppose $\mathcal{U}$ is in state (f, O, X, i) after reading a finite prefix $\alpha(0) \ldots \alpha(k-1)$ of α. On reading the next letter, i.e. $\alpha(k)$, we want $\mathcal{U}$ to non-deterministically guess the full-level ranking, say f', induced by $\mathsf{r}^\star_{\mathcal{A},\alpha}$ at level $k+1$ of $G_\alpha^{\mathcal{A}}$. Furthermore, every such choice of f' must be a full-cover of $(f, \alpha(k))$ and must satisfy the local properties in Lemma 1. Given f, f' and $\alpha(k)$, the local properties are easy to check, and are used in algorithm $\mathsf{BüchiDisambiguate}$ to filter the full-level rankings that can serve as f'. Once a choice of f' has been made, algorithm $\mathsf{BüchiDisambiguate}$ uses the O-, X- and i-components of the current state (f, O, X, i) to *uniquely* determine the corresponding components of the next state (f', O', X', i'). In doing so, we use a technique reminiscent of that used by Miyano and Hayashi [10], and subsequently by Schewe [12], to ensure that the global properties in Lemma 1 are satisfied by the sequence of full-level rankings corresponding to an accepting run of $\mathcal{U}$. Note that the choice of f' gives rise to non-determinism in the transition relation of $\mathcal{U}$. However, in every step of the run of $\mathcal{U}$ on α, there is a unique choice of f' that can give rise to $\mathsf{r}^\star_{\mathcal{A},\alpha}$ when the full-level rankings corresponding to the run are "stitched" together.

A closer inspection of Lemma 1 shows that there are two types of global properties: those that relate to *every* path (property 5), and those that relate to *some* path (e.g., properties 6 and 7). Schewe gave a ranking based construction to enforce properties of the first type in the context of Büchi complementation [12]. We use a similar idea here for enforcing property 5. Specifically, suppose $\mathcal{U}$ is in state (f, O, X, i) after reading $\alpha(0) \ldots \alpha(k-1)$, and suppose f' has been chosen as the full-level ranking of level $k+1$ of $G_\alpha^{\mathcal{A}}$. Suppose further that we wish to enforce property 5 for all vertices $(q, k+1)$ in $G_\alpha^{\mathcal{A}}$ where q is assigned an even rank j by f'. To do so, we set i' to j, populate O' with *all* $\mathcal{A}$-states assigned rank j by f', and use the O-components of subsequent $\mathcal{U}$-states along the run to keep track of the successors (in $\mathcal{A}$) of all $\mathcal{A}$-states in O'. During this process, if we encounter an $\mathcal{A}$-state q_k with rank $< j$ in the O-component of a $\mathcal{U}$-state, we know that property 5 is satisfied for all paths ending in q_k. The state q_k is therefore removed from O, and the above process repeated until O becomes empty. The emptiness of O signifies that all paths in $G_\alpha^\mathcal{A}$ starting from $\mathcal{A}$-states with rank j at level $k+1$ eventually visit a state with rank $< j$. Once this happens, we reset O to $\emptyset$, choose the next (in cyclic order) even rank i and repeat the above process. Using the same argument as used in [12], it can be shown that a run of $\mathcal{U}$ visits a $\mathcal{U}$-state with $O = \emptyset$ and i set to the smallest even rank infinitely often iff the sequence of full-level rankings corresponding to the run satisfies property 5.

A naive way to adapt the above technique to enforce property 6 (resp., property 7) in Lemma 1 is to choose an odd rank (resp., ∞ rank) $i > 1$, populate O' with *a single non-deterministically chosen* $\mathcal{A}$-state assigned rank i by f', and track *a non-deterministically chosen single successor* of this state in the O-components of $\mathcal{U}$-states along the run until we find an $\mathcal{A}$-state that is final and assigned rank $i-1$ (resp., ∞). The problem with this naive adaptation is that the non-deterministic choice of $\mathcal{A}$-state above may lead to multiple accepting runs of $\mathcal{U}$ on α. This is undesirable, since we want $\mathcal{U}$ to be a UBA. To circumvent this problem, we choose the O-component of the next state, i.e. O' in (f', O', X', i'), deterministically, given the current $\mathcal{U}$-state (f, O, X, i) and $\alpha(k)$. Specifically, for every $\mathcal{A}$-state q_r in O, we find the $\alpha(k)$-successors of q_r in $\mathcal{A}$ that are assigned rank i by f', and choose only one of them, viz. the one with the minimum index, to stay in O'. For notational convenience, for $S \subseteq Q = \{q_0, \dots q_{n-1}\}$, let $\downarrow S$ denote the singleton set $\{q_i \mid q_i \in S \text{ and } \forall q_j \in S,\, i \leq j\}$. Then, we have $O' = \bigcup_{q_r \in O} \downarrow \{q_l \mid q_l \in \delta(q_r, \alpha(k)) \text{ and } f'(q_l) = i\}$.

Choosing O' as above has an undesired consequence: not all $\mathcal{A}$-states that are successors (in $\mathcal{A}$) of some state in O and have rank i may be tracked in the O-components of $\mathcal{U}$-states along the run. This may prevent the technique of Schewe [12] from detecting that property 6 (or property 7) is true in the sequence of full-rankings corresponding to a run of $\mathcal{U}$ on α. To rectify this situation, we use the X-component of $\mathcal{U}$-states as follows. We periodically load the X-component with a single $\mathcal{A}$-state from O, which is then removed from O. *All* successors (in $\mathcal{A}$) of the $\mathcal{A}$-state thus loaded in X are then tracked in the X-components of $\mathcal{U}$-states along the run, until we encounter a final $\mathcal{A}$-state with the desired rank ($i-1$ for property 6, and ∞ for property 7) in X. Once this happens, we empty X, load it with another $\mathcal{A}$-state (specifically, the one with the minimum index) from O, remove this chosen state from O, and repeat the process until both O and X are emptied. When both O and X become empty, we set i to the next rank i' of interest in cyclic order, load O with all $\mathcal{A}$-states assigned rank i', and repeat the entire process. Extending the reasoning used by Schewe in [12], it can be shown that a run of $\mathcal{U}$ visits a $\mathcal{U}$-state with $O = \emptyset, X = \emptyset$ and i set to the smallest rank of interest infinitely often iff the corresponding sequence of full-level rankings satisfies property 6 (or property 7, as the case may be).

3.2 Our Algorithm and Its Analysis

The pseudocode for algorithm BüchiDisambiguate is given below. Note that the checks for global properties are deferred until all full-level rankings have become tight. This is justified by property 1 in Lemma 1. The choice of initial state of $\mathcal{U}$ is motivated by the observation that q_0 is ranked ∞ in the full-level ranking induced by $\mathsf{r}^{\star}_{\mathcal{A},\alpha}$ at level 0 of $G^{\mathcal{A}}_{\alpha}$ iff $\alpha \in L(\mathcal{A})$. Finally, algorithm BüchiDisambiguate implements the following optimization when calculating the next $\mathcal{U}$-state (f', O', X', i') from a given $\mathcal{U}$-state (f, O, X, i) and $a \in \Sigma$: if i is odd or ∞ and if $\delta(X, a)$ intersects O', then X' is reset to $\emptyset$ instead of being populated with $\delta(X, a)$. This is justified because every $\mathcal{A}$-state in O' must eventually have one of its successors (in $\mathcal{A}$) with rank i moved to the X-component of a $\mathcal{U}$-state further down the run, for the run of $\mathcal{U}$ to be accepting.

Algorithm : BüchiDisambiguate
Input: NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$
Output: UBA $\mathcal{U} = (\Sigma, Z, Z_0, \delta_U, F_U)$

- States : $Z = \mathsf{FL}_\infty \times 2^Q \times 2^Q \times \langle 2n \rangle$. Furthermore, if $(f, O, X, i) \in Z$, then $O \subseteq Q$, $X \subseteq Q$, and $f \in \mathsf{FL}_\infty$ is such that $\forall q_j \in O \cup X,\ f(q_j) = i$.
- Initial State: $Z_0 = \{(f, O, X, i) \mid f(q_0) = \infty,\ \ O = X = \emptyset,\ \ i = 1 \text{ and } \forall q \in Q\,(q \neq q_0 \rightarrow f(q) = \bot)\}$.
- Transitions: For every $(f', O', X', i') \in \delta_U((f, O, X, i), a)$, where $a \in \Sigma$, the following conditions hold.
  1. Let $S = \{q_l \mid f(q_l) \neq \bot\}$. For all $q_j \notin \delta(S, a)$, $f'(q_j) = \bot$.
  2. f' is a full-cover of (f, a).
  3. For all $q_j \in Q$ such that $f(q_j) = \infty$, there is a $q_l \in \delta(q_j, a)$ such that $f'(q_l) = f(q_j)$.
  4. For all $q_j \in Q \setminus F$, there is a $q_l \in \delta(q_j, a)$ such that $f'(q_l) = f(q_j)$.
  5. For all $q_j \in F$ such that $f(q_j) \in [2n]^{even}$, there is a $q_l \in \delta(q_j, a)$ such that either $f'(q_l) = f(q_j)$ or $f'(q_l) = f(q_j) - 1$.
  6. For all $q_j \in Q$ such that $f(q_j) \neq \infty$, there is no $q_l \in \delta(q_j, a)$ such that $f'(q_l) = \infty$.
  7. In addition, O', X' and i' satisfy the following conditions.
     (a) If f is not a tight full-level ranking, then $O' = X' = \emptyset$, $i' = 1$.
     (b) If $O \cup X \neq \emptyset$, then $i' = i$. Furthermore, the following conditions hold. For notational convenience, let $O'' = \bigcup_{q_j \in O} \downarrow \{q_l \mid q_l \in \delta(q_j, a) \wedge f'(q_l) = i\}$ and let $X'' = \{q_l \mid q_l \in \delta(X, a) \wedge f'(q_l) = i\}$.
        i. If $i = 1$, then $O' = X' = \emptyset$.
        ii. If $i \in [2n]^{odd}$ and $i \neq 1$, then
           a. If $X = \emptyset$, then $X' = \downarrow O''$, $O' = O'' \setminus X'$.
           b. Else if $X'' \cap O'' \neq \emptyset$ or $(\exists q_l \in \delta(X, a) \cap F,\ f'(q_l) = (i - 1))$, then $X' = \emptyset$, $O' = O''$.
           c. Else, $X' = X''$, $O' = O''$.
        iii. If $i \in [2n]^{even}$, then $O' = \{q_l \mid q_l \in \delta(O, a) \wedge f'(q_l) = i\}$, $X' = \emptyset$.
        iv. If $i = \infty$, then
           a. If $X = \emptyset$, then $X' = \downarrow O''$, $O' = O'' \setminus X'$.
           b. Else if $X'' \cap O'' \neq \emptyset$ or $X'' \cap F \neq \emptyset$, then $X' = \emptyset$, $O' = O''$.

c. Else, $X' = X''$, $O' = O''$.

(c) If $O \cup X = \emptyset$, then $X' = \emptyset$. In addition, the following hold.

i. If $(i = 1)$ then $i' = max\_rank(f')$.
Else if $(i = \infty)$ then $i' = max(\{j \mid \exists q \in Q, f'(q) = j\} \cap [2n])$.
Else $i' = i - 1$.

ii. $O' = \{q_l \mid f'(q_l) = i'\}$.

– $F_U = \{(f, O, X, i) \mid O = X = \emptyset,\ i = 1,\ f$ is a tight full-level ranking$\}$.

End Algorithm : BüchiDisambiguate

3.3 Proof of Correctness

Let $\rho = (f_0, O_0, X_0, i_0), (f_1, O_1, X_1, i_1), \ldots$ be an accepting run of the NBA $\mathcal{U}$ constructed using algorithm BüchiDisambiguate. The run ρ *induces* a full-ranking r of $G^{\mathcal{A}}_{\alpha}$ as follows: for every $i \geq 0$, $r(q, i) = k$ iff $f_i(q) = k$ where $k \in \langle 2n \rangle$. Note that if $f_i(q) = \bot$, then q is not reachable in $\mathcal{A}$ from q_0 after reading $\alpha(0) \ldots \alpha(i-1)$.

Lemma 2. *For every vertex (q, l) in $\mathcal{G}^{\mathcal{A}}_{\alpha}$, $r((q, l)) = \mathsf{r}^{\star}_{\mathcal{A},\alpha}((q, l))$.*

Theorem 1. $L(\mathcal{U}) = L(\mathcal{A})$.

Theorem 2. *The automaton $\mathcal{U}$ is unambiguous.*

Proof. Suppose, if possible, there is a word $\alpha \in \Sigma^{\omega}$ that has two distinct accepting runs ρ_1, ρ_2 in $\mathcal{U}$. By Lemma 2, $r_1((q', l')) = \mathsf{r}^{\star}((q', l')) = r_2((q', l'))$ for every vertex (q', l') in $G^{\mathcal{A}}_{\alpha}$. Let $(f_{1,l}, O_{1,l}, X_{1,l}, i_{1,l})$ and $(f_{2,l} O_{2,l}, X_{2,l}, i_{2,l})$ be the l^{th} states reached along ρ_1 and ρ_2 respectively. We show below by induction on l that $(f_{1,l}, O_{1,l}, X_{1,l}, i_{1,l}) = (f_{2,l}, O_{2,l}, X_{2,l}, i_{2,l})$ for all $l \geq 0$.

Base Case : By our construction, $O_{1,0} = O_{2,0} = X_{1,0} = X_{2,0} = \emptyset$, $i_{1,0} = i_{2,0} = 1$. Since $r_1((q_0, 0)) = r_2((q_0, 0))$ as well, it follows that $f_{1,0} = f_{2,0}$, and hence $(f_{1,0}, O_{1,0}, X_{1,0}, i_{1,0}) = (f_{2,0}, O_{2,0}, X_{2,0}, i_{2,0})$.

Hypothesis : Assume the claim is true for $l \geq 0$. Hence, $(f_{1,l}, O_{1,l}, X_{1,l}, i_{1,l}) = (f_{2,l}, O_{2,l}, X_{2,l}, i_{2,l})$.

Induction Step : Let $q \in Q$ be such that $(q, l+1)$ is a vertex in $G^{\mathcal{A}}_{\alpha}$. Since $r_1((q', l')) = r_2((q', l')) = \mathsf{r}^{\star}_{\mathcal{A},\alpha}((q', l'))$ for every vertex (q', l') in G_{α}, it follows that $r_1((q, l+1)) = \mathsf{r}^{\star}((q, l+1)) = r_2((q, l+1))$. This implies $f_{1,l+1}(q) = \mathsf{r}^{\star}((q, l+1)) = f_{2,l+1}(q)$. For every $q' \in Q$ such that $(q', l+1)$ is not in $G^{\mathcal{A}}_{\alpha}$, by definition of level rankings, $f_{1,l+1}(q') = f_{2,l+1}(q') = \bot$. Hence, $f_{1,l+1}(s) = f_{2,l+1}(s)$ for all $s \in Q$. We thus have the following relations: (i) $f_{1,l+1} = f_{2,l+1}$, and (ii) $(f_{1,l}, O_{1,l}, X_{1,l}, i_{1,l}) = (f_{2,l}, O_{2,l}, X_{2,l}, i_{2,l})$. Since step (7) of algorithm BüchiDisambiguate uniquely determines the values of $O_{k,l+1}, X_{k,l+1}, i_{k,l+1}$ from the values of $f_{k,l}, f_{k,l+1}, O_{k,l}, X_{k,l}, i_{k,l}$, for $k \in 1, 2$, it follows that $(f_{1,l+1}, O_{1,l+1}, X_{1,l+1}, i_{1,l+1}) = (f_{2,l+1}, O_{2,l+1}, X_{2,l+1}, i_{2,l+1})$. This completes the inductive step, and we have $(f_{1,l}, O_{1,l}, X_{1,l}, i_{1,l}) = (f_{2,l}, O_{2,l}, X_{2,l}, i_{2,l})$ for all $l \geq 0$. However, this contradicts the assumption that ρ_1 and ρ_2 are distinct runs. Hence, ρ_1 and ρ_2 must be the same run of $\mathcal{U}$.

We have thus shown that for every $\alpha \in \Sigma^\omega$, there is at most one accepting run of $\mathcal{U}$. It follows that $\mathcal{U}$ is unambiguous. □

Theorem 3. *The number of states of $\mathcal{U}$ is in $O(n \cdot (0.76n)^n)$.*

Proof. Every state of $\mathcal{U}$ is a (f, O, X, i) tuple, where f is a full-level ranking. We encode (f, O, X, i) tuples as 3-tuples (p, g, i), where $p \in \{0, 1, 2\}$ and g is a modified ranking function similar to that used in [12,7].

While some states (f, O, X, i) in our construction correspond to tight full-level rankings, others do not. We first use an extension of the idea in [12] to encode (f, O, X, i) with tight full-level ranking f as a tuple (p, g, i), where $g : Q \to \{1, \ldots, r\} \cup \{-1, -2, -3, \infty\}$ and $r = max\_odd(f)$. This is done as follows. For all $q \in Q$ but $q \notin O \cup X$, we let $g(q) = f(q)$. If $q \in O \cup X$ and $f(q) \notin [r]^{odd}$, then we let $g(q) = -1$ and $i = f(q)$. This part of the encoding is similar to that used in [12]. We extend this encoding to consider cases where $q \in O \cup X$ and $f(q) = k \in [r]^{odd}$.

There are three sub-cases to consider: (i) $O \cup X \neq \{q \mid q \in Q \wedge f(q) = k\}$, (ii) $O \cup X = \{q \mid q \in Q \wedge f(q) = k\}$ and $O \neq \emptyset$, and (iii) $X = \{q \mid q \in Q \wedge f(q) = k\}$. In the first case, we let $p = 0$, $i = k$, $g(q) = -2$ for all $q \in O$ and $g(q) = -3$ for all $q \in X$. Since there exists a state $q' \in Q \setminus (O \cup X)$ with rank k, the range of g contains $k \in [r]^{odd}$ in this case. In the second case, we let $p = 1$, $i = k$, $g(q) = k$ for all $q \in O$ and $g(q) = -3$ for all $q \in X$. Finally, in the third case, we let $p = 2$, $i = k$ and $g(q) = k$ for all $q \in X$. Thus, the range of g contains $k \in [r]^{odd}$ in both the second and third cases as well. Note that the component p in (p, g, i) is used only when $i \in [r]^{odd}$. It is now easy to see that g is always onto one of the sets $A_j \cup \{1, 3, \ldots, r\}$, where A_j is a subset of $\{\infty, -1, -2, -3\}$. The total number of functions of each of the above types is $O(tight(n))$. Since $p \in \{0, 1, 2\}$ following Schewe's analysis [12], the total number of (p, g, i) tuples is upper bounded by $O(n \cdot tight(n)) = O(tight(n + 1))$.

Now, let us consider states with non-tight full-level rankings. Our construction ensures that once an odd rank i appears in a full-level ranking g along a run ρ, all subsequent full-level rankings along ρ contain every rank in $\{i, i + 2, \ldots max\_odd(g)\}$. The O, X and i components in states with non-tight full-level ranking are inconsequential; hence we ignore these. Suppose a state with non-tight full-level ranking f contains the odd ranks $\{j, \ldots, i-2, i\}$, where $1 < j \leq i$, $i = max\_odd(f)$. To encode this state, we first replace f with a level ranking g as follows. For all $k \in \{j, \ldots, i, i+1\}$ and $q \in Q$, if $f(q) = k$, then $g(q) = k - j + c$, where $c = 0$ if j is even and 1 otherwise. If $f(q) = \infty$, we let $g(q)$ be ∞. Effectively, this transforms f to a tight full-level ranking g by shifting all ranks down by $j - c$. The original state can now be represented as the tuple $(p, g, -(j - c))$. Note that the third component of a state represented as (p, g, i) is always non-negative for states with tight full-level ranking, and always negative for states with non-tight full-level ranking. Hence, there is no ambiguity in decoding the state representation. Clearly, the total no. of states with non-tight full-level rankings is $O(n \cdot tight(n)) = O(tight(n + 1))$. Thus, the total count of all (f, O, X, i) states is in $O(tight(n + 1)) = O(n \cdot tight(n))$, where $tight(n) \approx (0.76n)^n$. □

4 An Exponentially Improved Lower Bound for NBA-SUBA Translation

In this section, we prove a lower bound for the state complexity trade-off in translating an NBA to a strongly unambiguous Büchi automaton (SUBA). Our proof technique relies on the full automaton technique of Yan [14].

Definition 1 (Full automaton). *A full automaton $\mathcal{A}$ is described by the structure $\mathcal{A} = (\Sigma, Q, I, \delta, F)$ where Q is the set of states, $I \subseteq Q$ is the set of initial states, $\Sigma = 2^{(Q\times Q)}$ is the alphabet and δ is defined as follows: for all $q, q' \in Q, a \in \Sigma, \langle q, a, q'\rangle \in \Delta \Leftrightarrow \langle q, q'\rangle \in a$.*

Thus, a full automaton has a rich alphabet of size $2^{|Q|^2}$, and every automaton with $|Q|$ states has an embedding in a full automaton with the same number of states. An ω-word α over the alphabet Σ of a full automaton corresponds directly to the run-DAG $G^{\mathcal{A}}_{\alpha}$ of $\mathcal{A}$. Correspondingly, a letter $a \in \Sigma$ represents the section of the run-DAG between two successive levels, and a finite word $w \in \Sigma^*$ represents a finite section of the run-DAG.

For purposes of this section, we focus on a special family of full automata $\mathcal{F} = \{\mathcal{A}_n \mid n \geq 2\}$. Automaton $\mathcal{A}_n$ in this family is given by $\mathcal{A}_n = (\Sigma_n, Q_n, I_n, \delta_n, F_n)$, where $Q_n = \{q_0, \ldots, q_{n-1}, q_n\}$ is a set of $n+1$ states, $I_n = \{q_0, \ldots q_{n-2}\}$ is the set of initial states, and $F_n = \{q_n\}$ is the singleton set of final states. We define a *Q-ranking* for $\mathcal{A}_n$ to be a full-level ranking $r : Q_n \to \langle n-1\rangle \cup \{\bot\}$ such that (i) r is a tight full-level ranking, (ii) $r(q_n) = \bot$, (iii) $r(q_{n-1}) = \infty$, and (iv) for every $k \in [n-1]$, $|r^{-1}(k)| = 1$. The total number of Q-rankings of $\mathcal{A}_n$ is easily seen to be $(n-1)!$.

Let r_1 and r_2 be Q-rankings for $\mathcal{A}_n$. The word $w \in \Sigma_n^*$ is said to be Q-compatible with (r_1, r_2) if the following conditions are satisfied when w is viewed as a finite section of the run-DAG of $\mathcal{A}_n$, and r_1 (resp., r_2) is interpreted as the full-level ranking of states at the first (resp., last) level of w.

- There is no path from the first level to the last level of w that either starts or ends in q_n.
- There is a path from q_i in the first level of w to q_j in the last level of w iff either $r_1(q_i) > r_2(q_j)$, or $r_1(q_i) = r_2(q_j) \in [n-1]^{odd} \cup \{\infty\}$. Such a path is said to be *final* if it visits q_n; otherwise, it is *non-final.*

Lemma 3. *For every pair (r_1, r_2) of Q-rankings for $\mathcal{A}_n$, there is a word $w \in \Sigma_n^*$ that is Q-compatible with (r_1, r_2).*

Proof sketch: We show how to construct w as the concatenation of three words $w_1, w_2, w_3 \in \Sigma_n^*$. The proof that $w_1.w_2.w_3$ is Q-compatible with (r_1, r_2) follows from their construction, and uses an argument similar to that used in a related proof in [14] (specifically, proof of Lemma 2 in [14]).

The word w_1 is given by $b_1 b_2 b_3 \ldots b_{2n}$, where each $b_i \in \Sigma_n = 2^{(Q_n \times Q_n)}$ is defined as follows. For notational convenience, we use $Id_{non-final}$ below as a shorthand for $\{(q_j, q_j) \mid 0 \leq j < n\}$, i.e. identity transitions for non-final states.

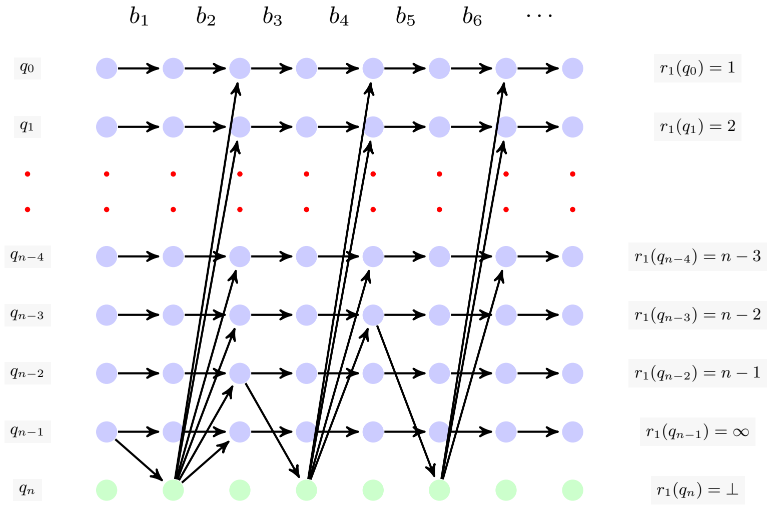

Fig. 1. Construction for w_1

- $b_1 = Id_{non-final} \cup \{(q_{n-1}, q_n)\}$
- $b_2 = Id_{non-final} \cup \{(q_n, q_j) \mid 0 \leq j < n\}$
- For $1 \leq i \leq n-1$, $b_{2i+1} = Id_{non-final} \cup \{(q_j, q_n) \mid r_1(q_j) = n-i\}$ and $b_{2i+2} = Id_{non-final} \cup \{(q_n, q_j) \mid r_1(q_j) < n-i\}$

Figure 1 shows the construction for an example word w_1. The word w_2 consists of the single letter of Σ_n given by $\{(q_i, q_j) \mid r_1(q_i) = r_2(q_j) \in [n-1]^{odd} \cup \{\infty\}\}$. The word w_3 is constructed in the same manner as w_1, but with r_2 used in place of r_1.

Lemma 4. *Let r_1, r_2 and r_3 be Q-rankings for $\mathcal{A}_n$. If $w_1 \in \Sigma_n^*$ is Q-compatible with (r_1, r_2) and $w_2 \in \Sigma_n^*$ is Q-compatible with (r_2, r_3), then $w_1 w_2$ is Q-compatible with (r_1, r_3).*

The proof follows from Lemma 3 and mimics the proof of a related result in [14] (specifically, Lemma 3 in [14]).

We now show a factorial lower bound of the NBA to SUBA state complexity trade-off. We make use of the following special class of UBA for this purpose. For notational convenience, we use $L(\mathcal{A}^{\{q\}})$ to denote the set of ω-words accepted by an NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ starting from state $q \in Q$.

Definition 2 (EUBA). *A UBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ is called a state-exclusive UBA (EUBA) if for every state $q \in Q$ either $L(\mathcal{A}^{\{q\}}) \subseteq L(\mathcal{A})$ or $L(\mathcal{A}^{\{q\}}) \subseteq \Sigma^\omega \setminus L(\mathcal{A})$. In other words, all words accepted starting from q are either in the language of $\mathcal{A}$ or in its complement.*

Theorem 4. *Every EUBA that is language equivalent to the full automaton $\mathcal{A}_n$ has at least $(n-1)!$ states.*

Proof. Let E_n be a EUBA such that $L(E_n) = L(\mathcal{A}_n)$. Let $r_1, r_2, \ldots, r_{n-1!}$ be the Q-rankings for $\mathcal{A}_n$. Repeating this sequence of full-level rankings infinitely many times, we get an infinite sequence of full-level rankings. Let w_i be the word (as constructed in the proof of Lemma 3) that is Q-compatible with (r_i, r_{i+1}), for $i \in [(n-1)! - 1]$. Let w be the infinite word $(w_1 w_2 \ldots w_{(n-1)!-1})^\omega$.

From the construction outlined in the proof of Lemma 3, there is a final path between q_{n-1} (ranked ∞) at the first level of states in w_i to the same state (ranked ∞) at the last level of states in w_i, for every $i \in [(n-1)! - 1]$. The concatenation of these paths gives a path π' that starts from q_{n-1} and visits the final state q_n infinitely often. Note that we cannot have a path in w that starts from any state other than q_{n-1} and visits q_n infinitely often. This is because a visit to q_n from any other state q_j ($\neq q_{n-1}$) with rank k must necessarily be followed by a visit to a state with rank $< k$. Hence, infinitely many visits to q_n will result in infinitely many rank reductions. This is an impossibility since ranks cannot increase along a path. Hence, paths in w starting from every state other than q_{n-1} can visit final states only finitely often.

Since q_{n-1} is not an initial state of $\mathcal{A}_n$, the path π' considered above is not an accepting run of $\mathcal{A}_n$. Hence, $w \notin L(\mathcal{A}_n)$. Let a be the letter in Σ_n that represents only the edge (q_0, q_{n-1}). Hence, the path $q_0\pi'$ is an accepting run of $\mathcal{A}_n$, and $aw \in L(\mathcal{A}_n)$.

Since $L(E_n) = L(\mathcal{A}_n)$, there must be an accepting run ρ of E_n on aw. Let k be the smallest index such that $\rho(k) \in \mathit{inf}(\rho)$ for all $i \geq k$. Let $T = (n-1)! - 1$, and suppose $|w_i| = s_i$ for all $i \in [T]$, and $s = \sum_{i=1}^{[T]} s_i$. For notational convenience, let us also assume that $s_0 = 0$. Let t be the smallest index such that $t \geq k$ and $t = p.s$ for some integer $p > 0$. Consider the sequence of indices $t + n_0.T + s_0, \ldots t + n_T.T + s_T$, where each $n_i \geq 0$ is such that ρ visits a final state of E_n between consecutive indices in the sequence. If E_n has fewer than $(n-1)!$ states, there must be a state of E_n that repeats in $\rho(t + n_0.T + s_0), \ldots \rho(t + n_T.T + s_T)$. Let $i, j \in \{0, \ldots T\}$ be such that $i < j$ and $\rho(t + n_i.T + s_i) = \rho(t + n_j.T + s_j) = z$, say. Clearly, the run ρ visits a final state of E_n between indices $t + n_i.T + s_i$ and $t + n_j.T + s_j$. Let the segment of the word aw between these two occurrences of z be v. Then, there is a path in E_n from z to itself along v that visits a final state. It follows that $v^\omega \in L(E_n^{\{z\}})$.

We now show that v^ω is also in $L(E_n)$. Consider the two Q-rankings r_i and r_j mentioned above. Clearly, the word v is Q-compatible with (r_i, r_j). Since $r_i \neq r_j$ and both are Q-rankings for $\mathcal{A}_n$, there is a state q_k for $k \in [n-2]$, such that $r_i(q_k) > r_j(q_k)$. By the construction outlined in the proof of Lemma 3, there is a final run along v from state q_k to itself. Since q_k is an initial state of $\mathcal{A}_n$, there exists an accepting run of $\mathcal{A}_n$ on v^ω. Therefore, $v^\omega \in L(\mathcal{A}_n)$; since $L(E_n) = L(\mathcal{A}_n)$, $v^\omega \in L(E_n)$ as well.

Since ρ is an accepting run of E_n, it visits at least one final state of E_n infinitely often. Therefore, there exists a strict suffix w' of w starting from the $(t + n_i.T + s_i)^{th}$ index such that the corresponding run of E_n starting at z sees at least one final state infinitely often. Hence, $w' \in L(E_n^{\{z\}})$. However, the final run of $\mathcal{A}_n$ on w' starts at q_{n-1} since this is the only state ranked ∞ in w. Since q_{n-1} is not an initial state of $\mathcal{A}_n$, $w' \notin L(\mathcal{A}_n)$ and hence $w' \notin L(E_n)$.

Thus, $v^\omega \in L(E_n^{\{z\}})$ and $v^\omega \in L(E_n)$, while $w' \in L(E_n^{\{z\}})$ and $w' \notin L(E_n)$. This contradicts the state-exclusivity property of EUBA. Hence, the number of states of E_n is at least $(n-1)!$. ☐

Theorem 5. *Every SUBA that is language equivalent to $\mathcal{A}_n$ has at least $(n-1)!$ states.*

Proof. The proof follows from the observation that every SUBA is also a EUBA by definition. ☐

5 A New Lower Bound Proof for Disambiguation

We now show an exponential lower bound for the NBA-UBA state complexity trade-off. This lower bound was already known from a result due to Schmidt [13]. However, the technique used by Schmidt involves computing ranks of specially constructed matrices. In contrast, our proof uses the full automata technique.

Definition 3 (Trim UBA). *Let $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ be a UBA. $\mathcal{A}$ is* trim *if $L(\mathcal{A}^{\{q\}}) \neq \emptyset$ for every $q \in Q$.*

Every UBA can be transformed to a language equivalent trim UBA simply by removing states from which no word can be accepted. Let $\mathcal{A}$ be a full automaton with alphabet Σ having 1 initial state, 1 final state, and n other (non-initial and non-final) states. For each non-empty subset S of non-initial and non-final states (there are $2^n - 1$ such subsets), let a_S denote the letter (in Σ) on which we have edges in $\mathcal{A}$ from the initial state to only the states in S. Similarly, let b_S denote the letter on which we have edges in $\mathcal{A}$ from only the states in S to the final state, and also from the final state to itself. It is easy to see that $a_{S_1} b_{S_2}^\omega$ is accepted by $\mathcal{A}$ if and only if $S_1 \cap S_2 \neq \emptyset$. Let D be an unambiguous and trim Büchi automaton accepting the same language as $\mathcal{A}$.

Theorem 6. *The number of states of D is at least $2^n - 1$.*

Proof. For every state q of D, let $\hat{L}(D^{\{q\}})$ denote the set of words of the form b_S^ω accepted by D, starting from q. If s_1 and s_2 are initial states of D, by definition of unambiguous automata, $\hat{L}(D^{\{s_1\}}) \cap \hat{L}(D^{\{s_2\}}) = \emptyset$. Also if any state s in D has paths to two distinct states r_1 and r_2 that are labeled by the same word $l \in \Sigma^*$, then by definition of unambiguous and trim automata, $\hat{L}(D^{\{r_1\}})$ and $\hat{L}(D^{\{r_2\}})$ must be disjoint.

For a set T of states of D, define $\hat{L}(D^T) = \bigcup_{s \in T} \hat{L}(D^{\{s\}})$. If we also have $\hat{L}(D^{\{s_1\}})$ and $\hat{L}(D^{\{s_2\}})$ disjoint for all distinct $s_1, s_2 \in T$, then $\hat{L}(D^T)$ equals the symmetric difference (or xor) of the sets $\hat{L}(D^{\{r\}})$, where r ranges over T. Now for each non-empty set K of the non-initial and non-final states of $\mathcal{A}$, consider the set K_D of states in D that have an edge from an initial state of D on the letter a_K. Then $\hat{L}(D^{K_D})$ is the set of all words of the form b_S^ω, where S is a subset of non-initial and non-final states of $\mathcal{A}$ such that $S \cap K \neq \emptyset$. Let this set of words be called $\Lambda(K)$. Since D is unambiguous, $\Lambda(K)$ can be obtained as the xor of languages $\hat{L}(D^{\{t\}})$, where t ranges over K_D. We will now show that for each non-empty subset S of non-initial and non-final states of $\mathcal{A}$, the set containing only the word b_S^ω can be obtained by xoring appropriate languages $\hat{L}(D^{\{t\}})$. Since there are $2^n - 1$ possible non-empty subsets S, this shows that by xoring appropriate languages $\hat{L}(D^{\{t\}})$, we can get up to 2^{2^n-1} different sets. This, in turn, implies that the number of distinct values taken by t, i.e. number of states in D, is at least $2^n - 1$.

We will prove the above claim by downward induction on the number of states in S. Suppose S is the set of all n non-initial and non-final states of $\mathcal{A}$. Then the set containing only b_S^ω can be obtained by xoring $\Lambda(K)$, where K ranges over all non-empty subsets of non-initial and non-final states of $\mathcal{A}$. This is because b_S^ω occurs in all $2^n - 1$ (i.e, an odd number) languages $\Lambda(K)$, where K ranges over all non-empty subsets of non-initial and non-final states of $\mathcal{A}$. However, for any other non-empty subset S' of non-initial and non-final states of $\mathcal{A}$, $b_{S'}^\omega$ occurs only in those $\Lambda(K)$s where $K \cap S' \neq \emptyset$. The latter is precisely the set of all subsets K excluding those that are disjoint from S', and the number of such subsets is even for $|S'| < n$.

Now suppose we can obtain singleton sets $\{b_S^\omega\}$ for all S with $|S| > t$. Then we can xor every $\Lambda(K)$ suitably with singleton sets containing b_S^ω for $|S| > t$, such that the resulting modified languages $\Lambda'(K)$ do not contain any b_S^ω for $|S| > t$. Now consider any set S of size t. Then take xor of the modified languages $\Lambda'(K)$ obtained above for all non-empty subsets K of S. By definition, b_S^ω occurs in all of these $\Lambda'(K)$, which are odd in count $(2^t - 1)$. For any other set S' of cardinality $\leq t$, its intersection with S is a strict subset of S. The word $b_{S'}^\omega$ doesn't occur in those $\Lambda'(K)$s where K is a non-empty subset of $S \setminus S'$; this, however, is odd in count as $S \setminus S'$ is nonempty. So the sets containing $b_{S'}^\omega$ are even in number.

This proves that all singleton sets containing only b_S^ω can be obtained for all nonempty subset S of non-initial and non-final states of $\mathcal{A}$. As argued above, this implies that D must have at least $2^n - 1$ states. □

Note that the above proof makes no use of the acceptance condition (i.e. Büchi, Müller, Streett, Rabin, parity, etc.) of the automaton, nor requires the words to be infinite. Hence it works for all acceptance conditions and even for finite words.

6 Conclusion

We now summarize our results on the state complexity trade-off in transforming an NBA to UBA and SUBA. Let $\mathsf{Size}_{NBA:\mathcal{C}}(n)$ denote the worst-case state complexity of an automaton in class $\mathcal{C}$ that accepts the same language as an NBA with n states. Table 1 shows the bounds of obtained from this paper, and compares them with previous best bounds. We propose to work towards closing the complexity gaps further in future.

Table 1. Comparison of state complexity trade-offs

Target class ($\mathcal{C}$)	$\mathsf{Size}_{NBA:\mathcal{C}}(n)$ from this paper		$\mathsf{Size}_{NBA:\mathcal{C}}(n)$ from earlier work	
	Lower bound	Upper bound	Lower bound	Upper bound
UBA	$2^n - 1$ (Thm 6)	$O(n \cdot (0.76n)^n)$ (Thm 3)	$2^n - 1$ [13]	$4 \cdot (3n)^n$ [6]
SUBA	$\Omega((n-1)!)$ (Thm 5)	-	$2^n - 1$ [13]	$O((12n)^n)$ [3]

References

1. Arnold, A.: Rational ω-languages are non-ambiguous. Theor. Comput. Sci. 26, 221–223 (1983)
2. Bousquet, N., Löding, C.: Equivalence and inclusion problem for strongly unambiguous Büchi automata. In: Dediu, A.-H., Fernau, H., Martín-Vide, C. (eds.) LATA 2010. LNCS, vol. 6031, pp. 118–129. Springer, Heidelberg (2010)
3. Carton, O., Michel, M.: Unambiguous Büchi automata. Theor. Comput. Sci. 297, 37–81 (2003)
4. Colcombet, T., Zdanowski, K.: A tight lower bound for determinization of transition labeled Büchi automata. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part II. LNCS, vol. 5556, pp. 151–162. Springer, Heidelberg (2009)
5. Isaak, D., Löding, C.: Efficient inclusion testing for simple classes of unambiguous automata. Inf. Process. Lett. 112(14-15), 578–582 (2012)
6. Kähler, D., Wilke, T.: Complementation, disambiguation, and determinization of Büchi automata unified. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 724–735. Springer, Heidelberg (2008)
7. Karmarkar, H., Chakraborty, S.: On minimal odd rankings for Büchi complementation. In: Liu, Z., Ravn, A.P. (eds.) ATVA 2009. LNCS, vol. 5799, pp. 228–243. Springer, Heidelberg (2009)
8. Karmarkar, H., Joglekar, M., Chakraborty, S.: Improved upper and lower bounds for Büchi disambiguation. Technical Report TR-13-36, Centre for Formal Design and Verification of Software, IIT Bombay (April 2013), `http://www.cfdvs.iitb.ac.in/reports/index.php`
9. Kupferman, O., Vardi, M.Y.: Weak alternating automata are not that weak. ACM Transactions on Computational Logic 2(3), 408–429 (2001)
10. Miyano, S., Hayashi, T.: Alternating finite automata on ω-words. Theor. Comput. Sci. 32, 321–330 (1984)
11. Preugschat, S., Wilke, T.: Effective characterizations of simple fragments of temporal logic using prophetic automata. In: Birkedal, L. (ed.) FOSSACS 2012. LNCS, vol. 7213, pp. 135–149. Springer, Heidelberg (2012)
12. Schewe, S.: Büchi complementation made tight. In: Proc. STACS, pp. 661–672 (2009)
13. Schmidt, E.M.: Succinctness of Description of Context-Free, Regular and Unambiguous Language. PhD thesis, Cornell University (1978)
14. Yan, Q.: Lower bounds for complementation of ω-automata via the full automata technique. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 589–600. Springer, Heidelberg (2006)