ON MINIMAL ODD RANKINGS FOR BÜCHI COMPLEMENTATION

HRISHIKESH KARMARKAR AND SUPRATIK CHAKRABORTY

Department of Computer Science and Engineering, Indian Institute of Technology Bombay
e-mail address: hrishik@cse.iitb.ac.in

Department of Computer Science and Engineering, Indian Institute of Technology Bombay
e-mail address: supratik@cse.iitb.ac.in

Abstract. We study minimal odd rankings (as defined by Kupferman and Vardi[KV01]) for run-DAGs of words in the complement of a nondeterministic Büchi automaton. We present an optimized version of the ranking based complementation construction given by Friedgut, Kupferman and Vardi [FKV06] and Schewe's[Sch09] variant of it, such that every accepting run of the complement automaton assigns a minimal odd ranking to the corresponding run-DAG. This allows us to determine minimally inessential ranks and redundant slices in ranking-based complementation constructions. We exploit this to reduce the size of the complement Büchi automaton by eliminating all redundant slices. We demonstrate the practical importance of this result through a set of experiments using the NuSMV model checker.

1. Introduction

The problem of complementing nondeterministic ω-word automata is fundamental in the theory of automata over infinite words. In addition to the theoretical aspects of the study of complementation techniques, efficient complementation techniques are extremely useful in practical applications as well. Vardi's excellent survey paper on the saga of Büchi complementation spanning more than 45 years provides a brief overview of various such applications of complementation techniques [Var07]. Unfortunately, complementing nondeterministic ω-word automata is non-trivial, and existing algorithms do not scale well beyond automata with $8-10$ states on an average. This contrasts sharply with the complementation problem for nondeterministic automata over finite words, where the subset construction suffices, and automata with hundreds of states can be complemented efficiently. Thus, there is a need to optimize complementation techniques for ω-word automata, and improve their applicability in practice.

From the different types of ω-automata, nondeterministic Büchi word automata (henceforth called NBW) are particularly useful for specifying fair computations and liveness properties. The most common use of NBW complementation is in checking language inclusion for ω-regular languages. To check if the language of an automaton A is contained in that of automaton B, it suffices to check emptiness of the intersection of the languages of A and that of the complement of B. This is especially useful in formal verification where A represents the behaviour of a system and B represents the specification or property we wish

LOGICAL METHODS
IN COMPUTER SCIENCE

DOI:10.2168/LMCS-???

to verify. The availability of a number of highly efficient LTL to NBW translators, coupled with the fact that LTL is one of the most commonly used specification formalisms, is indicative of the important role played by NBW in automata-theoretic formal verification. While LTL model checking avoids NBW complementation by translating the negation of an LTL formula directly into an NBW, checking the correctness of sophisticated LTL to NBW translations requires complementing the NBW derived from the original formula [Var07]. Furthermore, formalisms like ETL include NBW within the logic. Reasoning with ETL therefore requires complementing NBW, which often limits the practical applicability of ETL. Another interesting application of NBW complementation is in refinement and optimization techniques based on language containment, as pointed out by Vardi [Var07]. Thus, improvements in complementation techniques for NBW can potentially benefit a number of applications.

Various complementation constructions for NBW have been developed over the years, starting with Büchi [Büc62] in 1962. Büchi's algorithm resulted in a complement automaton with $2^{2^{O(n)}}$ states, starting from an NBW with n states. This upper bound was improved to $2^{O(n^2)}$ by Sistla et al [SVW87]. Safra [Saf88] provided the first asymptotically optimal $n^{O(n)}$ upper bound for complementation that passes through determinization. By a theorem of Michel [Mic88], it was known that Büchi complementation has a $n!$ lower-bound. With this, Löding [Löd99] showed that Safra's construction is asymptotically optimal for Büchi determinization, and hence for complementation. The $O(n!)$ (approximately $(0.36n)^n$) lower bound for Büchi complementation was recently sharpened to $(0.76n)^n$ by Yan [Yan08] using a full-automata technique. The complementation constructions of Klarlund [Kla91], Kupferman and Vardi [KV01] and Kähler and Wilke [KW08] for nondeterministic Büchi automata are examples of determinization-free (or *Safraless* as they are popularly called) complementation constructions for Büchi automata. The best known upper bound for the problem until recently was $(0.97n)^n$, given by Kupferman and Vardi. This was recently sharpened by Schewe[Sch09] to an almost tight upper bound of $(0.76n)^n$ modulo a factor of n^2.

NBW complementation techniques based on optimized versions of Safra's determinization construction (see, for example, Piterman's recent work in [Pit07]) have been experimentally found to work well for automata of small sizes (typically $8-10$ states) [TCT$^+$08]. However, these techniques are complex and present difficulties in symbolic complementation of Büchi automata. Ranking-based complementation constructions [KV01],[FKV06],[Sch09] are simpler and more amenable to symbolic implementation. Recently, several improvements to ranking based complementation constructions (like those in [FKV06]) have been proposed Similarly, language universality and containment checking techniques that use the framework of ranking-based complementation but avoid explicit complement constructions have been successfully applied to NBW with more than 100 states [DR09, FV09]. Therefore, ranking-based complementation techniques seem to hold much promise. This motivates our study of new optimization techniques for ranking-based complementation constructions.

The main contributions of this paper can be summarized as follows.

(1) We present an improvement to the ranking based complementation constructions of Friedgut, Kupferman and Vardi [FKV06] and Schewe [Sch09] for nondeterministic Büchi automata. Our construction ensures that all accepting runs of a word in the complement language correspond to a unique minimal odd ranking.

(2) We show that minimally inessential ranks and redundant slices in the complement automaton can be efficiently identified without any language containment checks.

This allows us to efficiently identify and eliminate redundant slices, leading to a reduction in the size of the complement automaton.

(3) We present an implementation of our technique using the BDD-based symbolic model checker NuSMV to count states and identify redundant slices in the complement automaton. Our implementation permits trading off time for symbolic search in NuSMV with size of the complement automaton, without compromising the correctness of the complement construction.

The remainder of this paper is organized as follows. Section 2 presents preliminary definitions and an overview of Kupferman and Vardi's ranking-based complementation construction for NBW and related work. In Section 3, we establish a few important properties, including a minimality result, of an odd ranking function originally used by Kupferman and Vardi to prove a key result in their paper [KV01]. Section 4.1 describes a motivating example that shows how complement automata obtained using ranking-based techniques can be made exponentially more succinct by restricting accepting runs to correspond to minimal odd rankings. Section 4 presents an improved ranking-based complementation construction for NBW that exploits the above observation. We prove the correctness of our construction and show that the worst-case size of the resulting automaton is no worse than the best-known result [Sch09]. In Section 5, we define minimally inessential ranks and redundant slices of our complement automaton, and show how these can be efficiently identified. Finally, Section 7 concludes the paper and outlines some open questions for future work.

2. Preliminaries

Let $A = (Q, q_0, \Sigma, \delta, F)$ be an NBW, where Q is a set of states, $q_0 \in Q$ is an initial state, Σ is an alphabet, $\delta : Q \times \Sigma \to 2^Q$ is a transition function, and $F \in 2^Q$ is a set of accepting states. An NBW accepts a set of ω-words, where an ω-word α is an infinite sequence $\alpha_0\alpha_1\ldots$, and $\alpha_i \in \Sigma$ for all $i \geq 0$. A run ρ of A on α is an infinite sequence of states given by $\rho : \mathbb{N} \to Q$, where $\rho(0) = q_0$ and $\rho(i+1) \in \delta(\rho(i), \alpha_i)$ for all $i \geq 0$. A run ρ of A on α is called *accepting* if $\mathit{inf}(\rho) \cap F \neq \emptyset$, where $\mathit{inf}(\rho)$ is the set of states that appear infinitely often along ρ. The run ρ is called *rejecting* if $\mathit{inf}(\rho) \cap F = \emptyset$. An ω-word α is accepted by A if A has an accepting run on it, and is rejected otherwise. The set of all words accepted by A is called the *language* of A, and is denoted $L(A)$. The complementation problem for NBW is to construct an automaton A^c from a given NBW A such that $L(A^c) = \Sigma^\omega \setminus L(A)$. We will henceforth denote the complement language $\Sigma^\omega \setminus L(A)$ by $\overline{L(A)}$. An NBW is said to be *complete* if every state has at least one outgoing transition on every letter in Σ. Every NBW can be made complete without changing the accepted language by adding at most one non-accepting "sink" state. All NBW considered in the remainder of this paper are assumed to be complete.

The (possibly infinite) set of all runs of an NBW $A = (Q, q_0, \Sigma, \delta, F)$ on a word α can be represented by a directed acyclic graph $G_\alpha = (V, E)$, where V is a subset of $Q \times \mathbb{N}$ and $E \subseteq V \times V$. The root vertex of the DAG is $(q_0, 0)$. For all $i > 0$, vertex $(q, i) \in V$ iff there is a run ρ of A on α such that $\rho(i) = q$. The set of edges of G_α is $E \subseteq V \times V$, where $((q, i), (q', j)) \in E$ iff both (q, i) and (q', j) are in V, $j = i + 1$ and $q' \in \delta(q, \alpha_i)$. Graph G_α is called the *run-DAG* of α in A. A vertex $(q, l) \in V$ is called an *F-vertex* if $q \in F$, i.e, q is a final state of A. Consider a (possibly finite) DAG $G \subseteq G_\alpha$. A vertex (q, l) is said to be *F-free* in G if there is no F-vertex that is reachable from (q, l) in G. Furthermore, (q, l)

is called *finite* in G if only finitely many vertices are reachable from (q, l) in G. For every $l \geq 0$, the set of vertices $\{(q, l) \mid (q, l) \in V\}$ constitutes *level* l of G_α. An accepting path in G_α is an infinite path $(q_0, 0), (q_{i_1}, 1), (q_{i_2}, 2) \ldots$ such that $q_0, q_{i_1}, \ldots$ is an accepting run of A. The run-DAG G_α is called rejecting if there is no accepting path in G_α. Otherwise, G_α is said to be accepting.

2.1. **Ranking functions.** Kupferman and Vardi [KV01] introduced the idea of assigning ranks to vertices of run-DAGs, and described a rank-based complementation construction for alternating Büchi automata. They also showed how this technique can be used to obtain a ranking-based complementation construction for NBW, that is easier to understand and implement than the complementation construction based on Safra's determinization construction[Saf88]. In this section, we briefly overview ranking-based complementation constructions for NBW.

Let $[k]$ denote the set $\{1, 2, \ldots, k\}$, and $[k]^{odd}$ (respectively $[k]^{even}$) denote the set of all odd (respectively even) numbers in the set $\{1, 2, \ldots, k\}$. Given an NBW A with n states and an ω-word α, let $G_\alpha = (V, E)$ be the run-DAG of α in A. A *ranking* r of G_α is a function $r : V \to [2n]$ that satisfies the following conditions: (i) for all vertices $(q, l) \in V$, if $r((q, l))$ is odd then $q \notin F$, and (ii) for all edges $((q, l), (q', l+1)) \in E$, we have $r((q', l+1)) \leq r((q, l))$. A ranking associates with every vertex in G_α a rank in $[2n]$ such that the ranks along every path in G_α are non-increasing, and vertices corresponding to final states always get even ranks. A ranking r is said to be *odd* if every infinite path in G_α eventually gets trapped in an odd rank. Otherwise, r is called an *even ranking*. We use $\mathit{max\_odd}(r)$ to denote the highest odd rank in the range of r.

A *level ranking* for A is a function $g : Q \to [2n] \cup \{\perp\}$ such that for every $q \in Q$, if $g(q) \in [2n]^{odd}$, then $q \notin F$. Let $\mathcal{L}$ be the set of all level rankings for A. Given two level rankings $g, g' \in \mathcal{L}$, a set $S \subseteq Q$ and a letter σ, we say that g' covers (g, S, σ) if for all $q \in S$ and $q' \in \delta(q, \sigma)$, if $g(q) \neq \perp$, then $g'(q') \neq \perp$ and $g'(q') \leq g(q)$. For a level ranking g, we abuse notation and let $\mathit{max\_odd}(g)$ denote the highest odd rank in the range of g. A ranking r of G_α induces a level ranking for every level $l \geq 0$ of G_α. If $Q_l = \{q \mid (q, l) \in V\}$ denotes the set of states in level l of G_α, then the level ranking g induced by r for level l is as follows: $g(q) = r((q, l))$ for all $q \in Q_l$ and $g(q) = \perp$ otherwise. It is easy to see that if g and g' are level rankings induced for levels l and $l+1$ respectively, then g' covers (g, Q_l, α_l), where α_l is the l^{th} letter in the input word α. A level ranking g is said to be *tight* if the following conditions hold: (i) the highest rank in the range of g is odd, and (ii) for all $i \in [\mathit{max\_odd}(g)]^{odd}$, there is a state $q \in Q$ with $g(q) = i$.

2.2. **Odd rankings.**

Lemma 2.1 ([KV01]). *The following statements are equivalent:*

(P1) All paths of G_α see only finitely many F-vertices.

(P2) There is an odd ranking for G_α.

Kupferman and Vardi [KV01] provided a constructive proof of (P1) $\Rightarrow$ (P2) in the above Lemma. Their construction is important for some of our subsequent discussions, hence we outline it briefly here. Given an NBW A with n states, an ω-word $\alpha \in \overline{L(A)}$ and the run-DAG G_α of A on α, the proof in [KV01] inductively defines an infinite sequence of DAGs $G_0 \supseteq G_1 \supseteq \ldots$, where (i) $G_0 = G_\alpha$, (ii) $G_{2i+1} = G_{2i} \setminus \{(q, l) \mid (q, l) \text{ is finite in } G_{2i}\}$, and (iii)

$G_{2i+2} = G_{2i+1} \setminus \{(q,l) \mid (q,l)$ is F-free in $G_{2i+1}\}$, for all $i \geq 0$. We call this procedure of removing vertices from successive DAGs as $\mathsf{KVRemProc}$. An interesting consequence of this definition is that for all $i \geq 0$, G_{2i+1} is either empty or has no finite vertices. It can be shown that if all paths in G_α see only finitely many F-vertices, then G_{2n} and all subsequent G_is must be empty. A ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ of G_α can therefore be defined as follows: for every vertex (q,l) of G_α, $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}((q,l)) = 2i$ if (q,l) is finite in G_{2i}, and $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}((q,l)) = 2i+1$ if (q,l) is F-free in G_{2i+1}. Kupferman and Vardi showed that $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ is an odd ranking [KV01]. Throughout this paper, we will use $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ to denote the odd ranking computed by the above technique due to Kupferman and Vardi (hence KV in the superscript) for NBW A and $\alpha \in \overline{L(A)}$. When A and α are clear from the context, we will simply use r^{KV} for notational convenience.

The NBW complementation construction and upper size bound presented in [KV01] was subsequently tightened in [FKV06], where the following important observation was made.

Lemma 2.2 ([FKV06])**.** *Given a word $\alpha \in \overline{L(A)}$, there exists an odd ranking r of G_α and a level $l_{lim} \geq 0$, such that for all levels $l > l_{lim}$, the level ranking induced by r for l is tight.*

Lemma 2.2 led to a reduced upper bound for the size of ranking-based complementation constructions, since all non-tight level rankings could now be ignored after reading a finite prefix of the input word. Schewe[Sch09] tightened the construction and analysis further, resulting in a ranking-based complementation construction with an upper size bound that is within a factor of n^2 of the best known lower bound [Yan08]. Hence, Schewe's construction is currently the best known ranking-based construction for complementing NBW. Gurumurthy et al [GKSV03] presented a collection of practically useful optimization techniques for keeping the size of complement automata constructed using ranking techniques under control. Their experiments demonstrated the effectiveness of their optimizations for NBW with an average size of 6 states. Interestingly, their work also highlighted the difficulty of complementing NBW with tens of states in practice. Doyen and Raskin [DR09] have recently proposed powerful anti-chain optimizations in ranking-based techniques for checking universality ($L(A) =^? \Sigma^\omega$) and language containment ($L(A) \subseteq^? L(B)$) of NBW. Fogarty and Vardi [FV09] have evaluated Doyen and Raskin's technique and also Ramsey-based containment checking techniques in the context of proving size-change termination (SCT) of programs. Their results bear testimony to the effectiveness of Doyen and Raskin's anti-chain optimizations for ranking-based complementation in SCT problems, especially when the original NBW is known to have *reverse-determinism* [FV09].

Schewe's construction. Since, the ideas proposed in [KV01], [FKV06] and [Sch09] for NBW complementation are crucially important to our proposed construction we shall briefly review the construction proposed by Schewe [Sch09] first.

Let $A = (Q, q_0, \Sigma, \delta, F)$ be an NBW as before with $n = |Q|$ states. The complement NBW is $C = (Q', \Sigma, q_0', \delta', F')$ where

- $Q' = Q_1 \cup Q_2$, where $Q_1 = 2^Q$, $Q_2 = \{\ (S, O, f, i) \in 2^Q \times 2^Q \times \mathcal{T} \times \{0, 2, \ldots, 2n-2\} \mid$ f is a tight level ranking, $O \subseteq S$ and $\exists i \in \omega$ such that $O \subseteq f^{-1}(2i)\ \}$.
- $q_0' = q_0$.
- $\delta' = \delta_1 \cup \delta_2 \cup \delta_3$ where
  - $\delta_1 : Q_1 \times \Sigma \rightarrow 2^{Q_1}$ where $\delta_1(S, \sigma) = \{\delta(S, \sigma)\}$.
  - $\delta_2 : Q_2 \times \Sigma \rightarrow 2^{Q_2}$ where $(S', O, f, i) \in \delta_2(S, \sigma)$ iff $S' = \delta(S, \sigma), O = \emptyset and i = 0$.

– $\delta_3 : Q_2 \times \Sigma \to 2^{Q_2}$ where $(S', O', f', i') \in \delta_3((S, O, f, i), \sigma)$ iff $S' = \delta(S, \sigma)$, f' covers (f, S, σ), max odd rank(f) = max odd rank(f') and
  * $i' = (i + 2)$ mod (max odd rank(f')+1) and $O' = f'^{-1}(i')$ if $O = \emptyset$ or
  * $i' = i$ and $O' = \delta(O, \sigma) \cap f'^{-1}(i)$ if $O \neq \emptyset$, respectively and

– $F' = \{\emptyset\} \bigcup (2^Q \times \{\emptyset\} \times \mathcal{T} \times \omega \cap Q_2$.

The automaton C first moves around in a subset construction using states in Q_1 and transitions in δ_1. Eventually, C can use a non-deterministic jump (via a transition in δ_2) into the set of states Q_2 after all level rankings have become tight and the maximal odd rank is the same at all levels. After this the automaton stays in Q_2 and uses the transitions in δ_3. It then cycles through the set of all even ranks upto the largest odd rank of the level ranking. For every even rank k, the O-set is loaded with all states that are assigned rank k. The automaton C then computes its successor states till all states in the O-set reach a state with a rank lower than k. The critical difference between this construction and the Friedgut, Kupferman and Vardi construction is that the even ranks are monitored one at a time instead of monitoring them all together.

Given an NBW A, let $\mathsf{KVF}(A)$ be the complement NBW constructed using the Friedgut, Kupferman and Vardi construction with tight level rankings [KV01, FKV06]. For notational convenience, we will henceforth refer to this construction as KVF-construction. Similarly, let $\mathsf{KVFS}(A)$ be the complement automaton constructed using Schewe's variant[Sch09] of the KVF-construction. We will henceforth refer to this construction as KVFS-construction.

3. Minimal odd rankings

The work presented in this paper can be viewed as an optimized variant of Schewe's [Sch09] ranking-based complementation construction. The proposed method is distinct from other optimization techniques proposed in the literature (e.g. those in [GKSV03]), and adds to the repertoire of such techniques. We first show that for every NBW A and word $\alpha \in \overline{L(A)}$, the ranking r^{KV} is *minimal* in the following sense: if r is any odd ranking of G_α, then every vertex (q, l) in G_α satisfies $\mathsf{r}^{\mathsf{KV}}((q, l)) \leq r((q, l))$. We then describe how to restrict the transitions of the complement automaton obtained by the KVFS-construction, such that every accepting run of α assigns the same rank to all vertices in G_α as is assigned by r^{KV}. Thus, our construction ensures that acceptance of α happens only through minimal odd rankings. This allows us to partition the set of states of the complement automaton into *slices* such that for every word $\alpha \in \overline{L(A)}$, all its accepting runs lie in exactly one slice. Redundant slices can then be identified as those that never contribute to accepting any word in $\overline{L(A)}$. Removal of such redundant slices results in a reduced state count, while preserving the language of the complement automaton. The largest $k (> 0)$ such that there is a non-redundant slice with that assigns rank k to some vertex in the run-DAG gives the *rank of* A, as defined by [GKSV03]. Notice that our sliced view of the complement automaton is distinct from the notion of slices as used in [KW08].

Gurumurthy et al have shown [GKSV03] that for every NBW A, there exists an NBW B with $L(B) = L(A)$, such that both the KVF- and KVFS-constructions for B^c require at most 3 ranks. However, obtaining B from A is non-trivial, and requires an exponential blowup in the worst-case [FKV06]. Therefore, ranking-based complementation constructions typically focus on reducing the state count of the complement automaton starting from a given NBW A, instead of first computing B and then constructing B^c. We follow the same approach in this paper. Thus, we do not seek to obtain an NBW with the *minimum* rank for the complement of a given ω-regular language. Instead, we wish to reduce the state count of Kupferman and Vardi's rank-based complementation construction, starting from a given NBW A.

Given an NBW A and an ω-word $\alpha \in \overline{L(A)}$, an odd ranking r of G_α is said to be *minimal* if for every odd ranking r' of G_α, we have $r'((q,l)) \geq r((q,l))$ for all vertices (q,l) in G_α.

Theorem 3.1. *For every NBW A and ω-word $\alpha \in \overline{L(A)}$, the ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ is minimal.*

Proof. Let α be an ω-word in $\overline{L(A)}$. Since A and α are clear from the context, we will use the simpler notation r^{KV} to denote the ranking computed by Kupferman and Vardi's method. Let r be any (other) odd ranking of G_α, and let $V_{r,i}$ denote the set of vertices in G_α that are assigned the rank i by r. Since A is assumed to be a complete automaton, there are no finite vertices in G_α. Hence, by Kupferman and Vardi's construction, $V_{\mathsf{r}^{\mathsf{KV}},0} = \emptyset$. Note that this is consistent with our requirement that all ranking functions have range $[2n] = \{1, \ldots 2n\}$.

We will prove the theorem by showing that $V_{r,i} \subseteq \bigcup_{k=1}^{i} V_{\mathsf{r}^{\mathsf{KV}},k}$ for all $i > 0$. The proof proceeds by induction on i, and by following the construction of DAGs $G_0, G_1, \ldots$ in Kupferman and Vardi's proof of Lemma 2.1.

Base case: Consider $G_1 = G_0 \setminus \{(q',l') \mid (q',l') \text{ is finite in } G_0\} = G_\alpha \setminus \emptyset = G_\alpha$. Let (q_1, l_1) be a vertex in G_1 such that $r((q_1,l_1)) = 1$. Let (q_f, l_f) be an F-vertex reachable from (q_1, l_1) in G_α, if possible. By virtue of the requirements that F-vertices must get even ranks, and ranks cannot increase along any path, $r((q_f, l_f))$ must be < 1. However, this is impossible given that the range of r must be $[2n]$. Therefore, no F-vertex can be reachable from (q_1, l_1). In other words, (q_1, l_1) is F-free in G_1. Hence, by definition, we have $\mathsf{r}^{\mathsf{KV}}(q_1, l_1) = 1$. Thus, $V_{r,1} \subseteq V_{\mathsf{r}^{\mathsf{KV}},1}$.

Hypothesis: Assume that $V_{r,j} \subseteq \bigcup_{k=1}^{j} V_{\mathsf{r}^{\mathsf{KV}},k}$ for all $1 \leq j \leq i$.

Induction: By definition, $G_{i+1} = G_\alpha \setminus \bigcup_{s=1}^{i} V_{\mathsf{r}^{\mathsf{KV}},s}$. Let (q_{i+1}, l_{i+1}) be a vertex in G_{i+1} such that $r((q_{i+1}, l_{i+1})) = i+1$. Since r is an odd ranking, all paths starting from (q_{i+1}, l_{i+1}) must eventually get trapped in some odd rank $\leq i+1$ (assigned by r). We consider two cases.

- Suppose there are no infinite paths starting from (q_{i+1}, l_{i+1}) in G_{i+1}. This implies (q_{i+1}, l_{i+1}) is a finite vertex in G_{i+1}. We have seen earlier that for all $k \geq 0$, G_{2k+1} must be either empty or have no finite vertices. Therefore, $i+1$ must be even, and $\mathsf{r}^{\mathsf{KV}}((q_{i+1}, l_{i+1})) = i+1$ by Kupferman and Vardi's construction.
- There exists a non-empty set of infinite paths starting from (q_{i+1}, l_{i+1}) in G_{i+1}. Since $G_{i+1} = G_\alpha \setminus \bigcup_{s=1}^{i} V_{\mathsf{r}^{\mathsf{KV}},s}$, none of these paths reach any vertex in $\bigcup_{s=1}^{i} V_{\mathsf{r}^{\mathsf{KV}},s}$. Since $V_{r,j} \subseteq \bigcup_{k \in \{1,2,\ldots,j\}} V_{\mathsf{r}^{\mathsf{KV}},k}$ for all $1 \leq j \leq i$, and since ranks cannot increase along any path, it follows that r must assign $i+1$ to all vertices along each of the above paths. This, coupled with the fact that r is an odd ranking, implies that $i+1$ is odd. Since F-vertices must be assigned even ranks by r, it follows from above that (q_{i+1}, l_{i+1}) is F-free in G_α. Therefore, $\mathsf{r}^{\mathsf{KV}}((q_{i+1}, l_{i+1})) = i+1$ by Kupferman and Vardi's construction.

We have thus shown that all vertices in G_{i+1} that are assigned rank $i+1$ by r must also be assigned rank $i+1$ by r^{KV}. Therefore, $V_{r,i+1} \setminus \bigcup_{j=1}^{i} V_{\mathsf{r}^{\mathsf{KV}},j} \subseteq V_{\mathsf{r}^{\mathsf{KV}},i+1}$. In other words, $V_{r,i+1} \subseteq \bigcup_{j=1}^{i+1} V_{\mathsf{r}^{\mathsf{KV}},j}$.

By the principle of mathematical induction, it follows that $V_{r,i} \subseteq \bigcup_{k=1}^{i} V_{\mathsf{r}^{\mathsf{KV}},k}$ for all $i > 0$. Thus, if a vertex is assigned rank i by r, it must be assigned a rank $\leq i$ by r^{KV}. Hence r^{KV} is minimal.

□

Lemma 3.2. *For every ω-word α, the run-DAG G_α ranked by $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ (or simply r^{KV}) satisfies the following properties.*

1. *For every vertex (q,l) that is not an F-vertex such that $\mathsf{r}^{\mathsf{KV}}((q,l)) = k$, there must be at least one immediate successor $(q', l+1)$ such that $\mathsf{r}^{\mathsf{KV}}((q', l+1)) = k$.*
2. *For every vertex (q,l) that is an F-vertex, such that $\mathsf{r}^{\mathsf{KV}}((q,l)) = k$, there must be atleast one immediate successor $(q', l+1)$ such that $\mathsf{r}^{\mathsf{KV}}((q', l+1)) = k$ or $\mathsf{r}^{\mathsf{KV}}((q', l+1)) = k-1$.*
3. *For every vertex (q,l) such that $\mathsf{r}^{\mathsf{KV}}((q,l)) = k$, where k is odd and $k > 1$, there is a vertex (q', l') for $l' > l$ such that (q', l') is an F-vertex reachable from (q,l) and $\mathsf{r}^{\mathsf{KV}}((q', l')) = k-1$.*
4. *For every vertex (q,l) such that $\mathsf{r}^{\mathsf{KV}}((q,l)) = k$, where k is even and $k > 0$, every path starting at (q,l) eventually visits a vertex with rank less than k. Furthermore, there is a vertex (q', l') for $l' > l$ such that (q', l') is reachable from (q,l) and $\mathsf{r}^{\mathsf{KV}}((q', l')) = k-1$.*

Proof. (1) For this case, we consider two subcases.

(a) k is odd: Since (q,l) is in G_k, it is not finite in G_{k-1} and it has infinitely many vertices reachable from it in G_{k-1} and in G_k. Since, (q,l) is F-free in G_k all immediate successors of (q,l) in G_k are also F-free. Suppose, all such immediate F-free successors have an odd rank $k' < k$. The largest odd rank k' that is less than k is $k-2$, which implies the last immediate successor that became F-free did so in G_{k-2} because of KVRemProc. As a result, (q,l) is F-free in G_{k-2} and gets rank $k-2$, a contradiction. Hence, suppose there is atleast one immediate successor of (q,l) that has even rank less than k. Consider the DAG G_{k-1}. All odd ranked immediate successors of (q,l) and all even ranked successors with rank less than $k-1$ are absent in G_{k-1} since they have been removed earlier. Hence, the only immediate successors of (q,l) present in G_{k-1} are finite in G_{k-1}. If all successors are finite then (q,l) is finite and gets rank $k-1$, a contradiction. Hence, (q,l) must have atleast one immediate successor with odd rank k.

(b) k is even: Then (q,l) is finite in G_k. Suppose, all immediate successors have rank less than k. Consider the graph G_{k-1}. All immediate successors of (q,l) with rank less than $k-1$ are absent in G_{k-1} and no successor of (q,l) in G_{k-1} is finite (since any such finite successor would have been removed in G_{k-2}). Hence, the only successors of (q,l) in G_{k-1} are F-free and have rank $k-1$. But, this implies that (q,l) is F-free and gets rank $k-1$, a contradiction.

Since, both the above cases lead to contradiction, there must be a successor of (q,l) with rank k.

(2) Since, (q,l) is an F-vertex k must be even. Since $\mathsf{r}^{\mathsf{KV}}((q,l)) = k$ the vertex (q,l) is finite in G_k. Suppose all immediate successors of (q,l) have rank less than $k-1$. Consider graph G_{k-2}. All successors with rank less than $k-2$ are absent in G_{k-2} since they have been removed earlier. Hence, every successor of (q,l) in G_{k-2} must be either finite in G_{k-2} or must be infinite but not F-free in G_{k-2}. Let (q', l') be an

infinite but not F-free successor of (q,l) in G_{k-2}. The vertex (q',l') does not become F-free in G_{k-1}, else it would get rank $k-1$, which is not possible. If the vertex (q',l') is finite in G_k then it gets rank k, a contradiction. Hence, (q',l') must get a rank greater than k, which is not possible since it is a successor of (q,l), which gets rank k. Hence, the only immediate successors of (q,l) in G_{k-2} are finite in G_{k-2}. This implies that (q,l) is finite in G_{k-2} and gets rank $k-2$, a contradiction. Hence, (q,l) must have a successor with rank either k or $k-1$.

(3) Since, $\mathsf{r}^{\mathsf{KV}}((q,l)) = k$, where $1 \leq k \leq 2n-1$ is odd, the vertex (q,l) is F-free in G_k and is present in all sets $G_{k'}$ for $k' < k$. Suppose, by way of contradiction we assume that (q,l) has no F-vertices reachable from it. Then (q,l) is F-free in G_1 and gets rank 1 under r^{KV}. Similarly, if (q,l) has no F-vertices reachable from it in any $G_{k'}$ for $k' < k$ and k' odd then (q,l) gets the odd rank k'. Hence, there must be atleast one F-vertex reachable from (q,l) in every $G_{k'}$ for $k' < k$. Hence there is an F-vertex (q',l') that is reachable from (q,l) in G_{k-2}. But, (q,l) is F-free in G_k. Hence, (q',l') must have been removed from G_{k-1} while constructing G_k from G_{k-1}. We note that since (q',l') is an F-vertex it cannot be removed in G_{k-2}, as only F-free vertices are removed in G_{k-2}. Hence, the F-vertex (q',l') must have the rank $k-1$.

(4) Since $\mathsf{r}^{\mathsf{KV}}((q,l)) = k$ the vertex (q,l) is finite in G_k. We know that (q,l) was not finite in G_{k-2}, else it would have got rank $k-2$. In G_{k-1}, where $k-1$ is odd, only F-free vertices are removed. Hence, there must be atleast one F-free vertex with rank $k-1$ that is reachable from (q,l) and is removed when constructing G_k from G_{k-1}. If (q,l) is not an F-vertex then (q,l) must have an immediate successor with rank k (Condition (1)) and if (q,l) is an F-vertex then (q,l) must have an immediate successor with rank k or rank $k-1$ (Condition (2)). We construct a tree T rooted at (q,l). The vertices of the tree are vertices in the run-DAG G_α that have rank k and are reachable from (q,l) via a path on which all vertices have rank k. Let (q',l') be a vertex of T and let $(q'',l'+1)$ be another vertex of T such that $(q'',l'+1)$ is an immediate successor of (q',l') in G_α. Then $((q',l'),(q'',l'+1))$ is an edge of T. The tree T is a finitely branching tree. If T is infinite then by König's Lemma it has an infinite path starting at (q,l) such that all vertices on the path have rank k. Hence, the infinite path starting at (q,l) is present in G_k. But, (q,l) is finite in G_k, a contradiction. Hence, T is finite. Consider a leaf $(q',l+k)$, for some $k > 0$ of T. Every successor of every leaf has rank less than k. All the leaves of T together cover all paths beginning at successors of (q,l) with rank k. All other successors have rank less than k. Hence, along every path beginning at (q,l) a vertex with rank less than k is eventually visited.

□

4. Complementation with minimal ranks

4.1. **A motivating example.** We have seen above the KVFS-construction leads to almost tight worst case bounds for NBW complementation. This is a significant achievement considering the long history of Büchi complementation [Var07]. However, the KVFS-construction does not necessarily allow us to construct small complement automata for every NBW. Specifically, there exists a family of NBW $\mathcal{A} = \{ A_3, A_5, \dots \}$ such that for every $i \in \{3, 5, \dots\}$: (i) A_i has i states, (ii) each of the ranking-based complementation constructions in [KV01], [FKV06] and [Sch09] produces a complement automaton with at least $\frac{i^{(\frac{i-1}{2})}}{e^i}$ states, and (iii) a ranking-based complementation construction that assigns minimal ranks to all run-DAGs results in a complement automaton with $\Theta(i)$ states. Automata

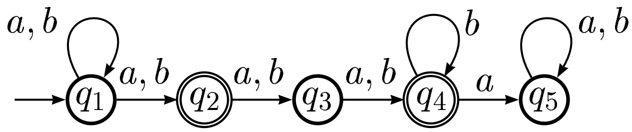

Figure 1: Automaton A_5

in the family $\mathcal{A}$ can be described as follows. Each A_i is an NBW $(Q_i, q_1, \Sigma, \delta_i, F_i)$, where $Q_i = \{q_1, q_2, \dots, q_i\}$, $\Sigma = \{a, b\}$, q_1 is the initial state and $F_i = \{q_j \mid q_j \in Q_i, j \text{ is even}\}$. The transition relation δ_i is given by: $\delta_i = \{(q_j, a, q_{j+1}), (q_j, b, q_{j+1}) \mid 1 \leq j \leq i-2\} \cup \{(q_1, a, q_1), (q_1, b, q_1),\ (q_i, a, q_i),\ (q_i, b, q_i),\ (q_{i-1}, b, q_{i-1}),\ (q_{i-1}, a, q_i)\}$. Figure 1 shows the structure of automaton A_5 defined in this manner. Note that each $A_i \in \mathcal{A}$ is a complete automaton. Furthermore, $b^\omega \in L(A_i)$ and $a^\omega \notin L(A_i)$ for each $A_i \in \mathcal{A}$. Let $\mathsf{KVF}(A_i)$ and $\mathsf{KVFS}(A_i)$ be the complement automata for A_i constructed using the KVF-construction and KVFS-construction respectively.

Lemma 4.1. *For every $A_i \in \mathcal{A}$, the number of states in $\mathsf{KVF}(A_i)$ and $\mathsf{KVFS}(A_i)$ is atleast $\frac{(i)^{(i-1)/2}}{e^i}$. Furthermore, there exists a ranking-based complementation construction for A_i that gives a complement automaton ${A_i}'$ with $\Theta(i)$ states.*

Proof. Consider the word a^ω. In the accepting runs of $\mathsf{KVF}(A^i)$ and $\mathsf{KVFS}(A^i)$ on a^ω we make the nondeterministic jump from the subset mode to the subset+ranks mode after all states $\{q_1, \dots, q_i\}$ appear in the current subset. Such a subset will appear after prefix a^i is read. After the jump is made let us assign the ranks as follows in the first state to which we make the jump : $r(q_1) = i$, $r(q_i) = 1$ and each non-accepting state other than q_1, q_i gets a unique odd rank between 1 and i. There are $(\frac{i-3}{2})!$ ways of doing this. Also, each accepting state gets a unique even rank between 1 and i. There are $(\frac{i-3}{2})!$ ways of doing this. It can be shown that starting from each of these $(\frac{i-3}{2})! \times (\frac{i-3}{2})!$ possible states one can get an infinite run on the word a^ω, where all subsequent states of the ranked run-DAG automaton are assigned consistent ranks. Hence, the number of states for $\mathsf{KVF}(A^i)$ and $\mathsf{KVFS}(A^i)$ is atleast $(\frac{i-3}{2})! \times (\frac{i-3}{2})!$, which can be approximated by $\frac{(i)^{(i-1)/2}}{e^i}$. Hence, the number of states of $\mathsf{KVF}(A^i)$ and $\mathsf{KVFS}(A^i)$ is $\Omega(i^{i-1/2})$.

The automaton ${A_i}'$ is the automaton constructed from A_i using the KVF-construction or the KVFS-construction but with the following changes. After the nondeterministic jump from the subset mode to the subset+ranks mode, we set $f(q_i) = 1$ and $f(q_1) = 3$ and $f(q_j) = 2$ for all $2 \leq j \leq i-1$. The automaton ${A_i}'$ has an accepting run that induces an odd ranking for G_α for every $\alpha \in \overline{L(A_i)}$ with just three ranks. It is not hard to see that the number of states of ${A_i}'$ is $\theta(i)$. ∎

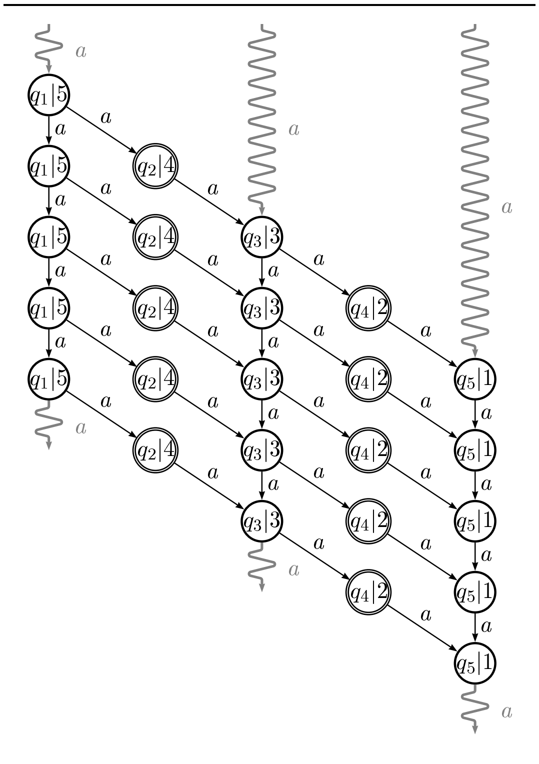

Figure 2: Run DAG for A_5 on a^ω with odd ranking r. Notation $q_i|k$ denotes that $r(q_i) = k$.

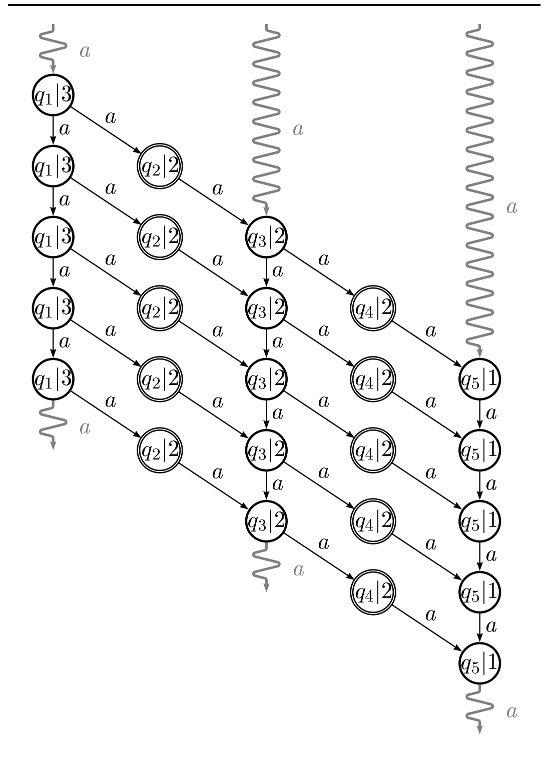

Figure 3: Run DAG for A_5 on a^ω with odd ranking f. Notation $q_i|k$ denotes that $f(q_i) = k$.

This discrepancy in the size of a sufficient rank set and the actual set of ranks used by the KVF- and KVFS-constructions motivates us to ask if we can devise a ranking-based complementation construction for NBW that uses the minimum number of ranks when accepting a word in the complement language. In this paper, we answer this question in the affirmative, by providing such a complementation construction.

4.2. **The complementation construction.** Motivated by the example described in the previous section, we now describe an optimized ranking-based complementation construction for NBW. Given an NBW A, the complement automaton A' obtained using our construction has the special property that when it accepts an ω-word α, it assigns a rank r to G_α that agrees with the ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ at every vertex in G_α. This is achieved by non-deterministically mimicking the process of rank assignment used to arrive at $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$. Our construction imposes additional constraints on the states and transitions of the complement automaton, beyond those in the KVF- and KVFS-constructions. For example, if k is the smallest rank that can be assigned to vertex (q, l), then the following conditions must hold: (i) if (q, l) is not an F-vertex then it must have a successor of the same rank at the next level, and (ii) if (q, l) is an F-vertex then it must have a successor at the next level with rank k or $k - 1$. The above observations are coded as conditions on the transitions of A', and are crucial if every accepting run of A' on $\alpha \in \overline{L(A)}$ must correspond to the unique ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ of G_α.

Algorithm 1: The complementation algorithm

Input : NBW $A = (Q, Q_0, \Sigma, \delta, F)$

Output: Complement NBW $A' = (Q', Q'_0, \Sigma, \delta', F')$

Let $A' = (Q', Q'_0, \Sigma, \delta', F')$, where

- $Q' = \{2^Q \times 2^Q \times \mathcal{R}\}$ is the state set, such that if $(S, O, f) \in Q'$, then $S \subseteq Q$ and $S \neq \emptyset$, $f \in \mathcal{R}$ is a level ranking, $O \subseteq S$, and either $O = \emptyset$ or $\exists k \in [2n-1]\ \forall q \in O,\ f(q) = k$.
- $Q'_0 = \bigcup_{i \in [2n-1]} \{(S, O, f) \mid S = \{q_0\},\ f(q_0) = i, O = \emptyset\}$ is the set of initial states.
- For every $\sigma \in \Sigma$, the transition function δ' is defined such that if $(S', O', f') \in \delta'((S, O, f), \sigma)$, the following conditions are satisfied.
  - (1) $S' = \delta(S, \sigma)$, f' covers (f, S, σ).
  - (2) For all $q \in S \setminus F$, there is a $q' \in \delta(q, \sigma)$ such that $f'(q') = f(q)$.
  - (3) For all $q \in S \cap F$ one of the following must hold
    - (a) There is a $q' \in \delta(q, \sigma)$, such that $f'(q') = f(q)$.
    - (b) There is a $q' \in \delta(q, \sigma)$, such that $f'(q') = f(q) - 1$.
  - (4) In addition, we have the following restrictions:
    - (a) If f is not a tight level ranking, then $O' = O$
    - (b) If f is a tight level ranking and $O \neq \emptyset$, then
      - (i) If for all $q \in O$, we have $f(q) = k$, where k is even, then
        - – Let $O_1 = \delta(O, \sigma) \setminus \{q \mid f'(q) < k\}$.
        - – If $(O_1 = \emptyset)$ then $O' = \{q \mid q \in Q \wedge f'(q) = k - 1\}$, else $O' = O_1$.
      - (ii) If for all $q \in O$, we have $f(q) = k$, where k is odd, then
        - – If $k = 1$ then $O' = \emptyset$
        - – If $k > 1$ then let $O_2 = O \setminus \{q \mid \exists q' \in \delta(q, \sigma),\ (f'(q') = k - 1)\}$.
          - - If $O_2 \neq \emptyset$ then $O' \subseteq \delta(O_2, \sigma)$ such that $\forall q \in O_2\, \exists q' \in O',\ q' \in \delta(q, \sigma)\ \wedge\ f'(q') = k$.
          - - If $O_2 = \emptyset$ then $O' = \{q \mid f'(q) = k - 1\}$.
    - (c) If f is a tight level ranking and $O = \emptyset$, then we set $O' = \{q \mid f'(q) = max\_odd(f')\}$.
- As in the KVF- and KVFS-constructions, the set of accepting states of A' is $F' = \{(S, O, f) \mid f$ is a tight level ranking, and $O = \emptyset\}$

Recall that in [FKV06], a state of the automaton that tracks the ranking of the run-DAG vertices at the current level is represented as a triple (S, O, f), where S is a set of Büchi states reachable after reading a finite prefix of the word, f is a tight level ranking, and the O-set checks if all even ranked vertices in a (possibly previous) level of the run-DAG have moved to lower odd ranks. In our construction, we use a similar representation of states, although the O-set is much more versatile. Specifically, the O-set is populated turn-wise with states of the same rank k, for both odd and even k. This is a generalization of Schewe's technique that uses the O-set to check if even ranked states present at a particular level have moved to states with lower odd ranks. The O-set in our construction, however, does more. It checks if every state of rank k (whether even or odd) in an O-set eventually reaches a state with rank $k-1$. If k is even. then it also checks if all runs starting at states in O eventually reach a state with rank $k-1$. When all states in an O-set tracking rank 2 reach states with rank 1, the O-set is reset and loaded with states that have the maximal odd rank in the range of the current level ranking. The process of checking ranks is then re-started. This gives rise to Algorithm (1) for constructing the complement automaton A' of A.

Note that unlike the KVF- and KVFS-constructions, the above construction does not have an initial phase of unranked subset construction, followed by a non-deterministic jump to ranked subset construction with tight level rankings. Instead, we start directly with ranked subsets of states, and the level rankings may indeed be non-tight for some finite prefix of an accepting run. The value of O is inconsequential until the level ranking becomes tight; hence it is kept as $\emptyset$ during this period. Note further that the above construction gives rise to multiple initial states in general. Since an NBW with multiple initial states can be easily converted to one with a single initial state without changing its language, this does not pose any problem, and we will not dwell on this issue any further.

Lemma 4.2. $\alpha \in L(\overline{A}) \Rightarrow \alpha \in L(A')$

Proof. Let $\alpha \in L(\mathsf{KVF}(A))$. Then α has the unique G_α-minimal odd ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$. We now construct an accepting run of A' on α i.e we show that A' accepts $(G_\alpha, \mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha})$. We consider only levels $l \geq l_{lim}$ after which all level rankings induced by $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ in G_α become tight. Let $l \geq l_{lim}$ be a level such that both the following are true -

- After reading the prefix $\alpha_0, \ldots, \alpha_{l-1}$ there exists a state (S, O, f) of A' such that $S = \{q | (q, l) \in V\}$, $O = \{q | q \in S, f(q) = k\}$, and f is a tight level ranking.
- There is atleast one vertex (q, l) in G_α such that $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}((q, l)) = k$

Such a level must exist since by the correspondence between level rankings and run-DAG rankings, the ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ must assign rank k to some vertex at the level.

- If k is odd then such a vertex (q, l) is not an F-vertex. Now, the level $l+1$ of G_α must contain one or more vertices with the same rank k, by Condition (1) on $(G_\alpha, \mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha})$ from Lemma (3.2). Thus, Condition (2) for δ' is satisfied. By Condition (4(b)ii) there exists $(S', O', f') \in \delta'((S, O, f), \sigma)$ that is a valid next state of A'. Also, O' contains all states with rank k or all states with the next even rank $k-1$, provided k was greater than 1, else we reach a final state of A'.
- If k is even then the level $l+1$ of G_α must contain one or more states with the same rank k or rank $k-1$, by Condition (2) on $(G_\alpha, \mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha})$ of Lemma (3.2). Thus, Condition (3) for δ' is satisfied and by Condition (4(b)i) there exists $(S', O', f') \in \delta'((S, O, f), \sigma)$ that is a valid next state of A' and such that O' contains all states with rank k or all states with the next odd rank $k-1$.

The above two conditions applied successively allows us to plot a run ρ through A', such that ρ induces $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$. Condition (3) on $(G_\alpha, \mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha})$ from Lemma (3.2) requires that for every vertex (q, l) ranked with odd k, has an F-vertex with even rank $k-1$ reachable from it. Condition (4) on $(G_\alpha, \mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha})$ Lemma (3.2) requires that all paths beginning at even ranked vertices eventually reach lower ranked vertices. Also, every such vertex (q, l) with even rank k has a vertex with rank $k-1$ reachable from it. These two conditions together ensure that the O-set is populated with successively lower ranked states and eventually when all states in the O-set have rank 1, it becomes empty and we restart from the maximum odd rank of the level ranking induced by $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$. It can be seen that this process must repeat infinitely often since $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ is an odd ranking and every path in G_α and hence in ρ must reach a minimum odd rank. Since, the state with an empty O-set is final in A', the run ρ sees infinitely many final states and is accepting. Hence, $\alpha \in L(A')$.

□

Lemma 4.3. $\alpha \in L(A') \Rightarrow \alpha \in L(\mathsf{KVF}(A))$

Proof. If G_α has no odd ranking then by Lemma (2.1) there is an accepting run ρ on α in the NBW A. Let ρ' be the path in G_α corresponding to ρ. For every vertex (q, l) on ρ' such that q is a final state of A ((q, l) is an F-vertex) we must have $r((q, l))$ even under any ranking r. Since, ranks do not increase and there are only finitely many ranks, eventually all F-vertices (and hence all vertices) on ρ acquire the same *even* rank say m.

Let ξ be an arbitrary run of A' on α and consider the infinite suffix of ξ after the point that ρ' acquires the final rank m. Since, the O-set of A' tracks successively lower ranks, states with rank m will eventually be added to the O-set (the run ρ will be added to the O-set in an intuitive sense). Since, all vertices on ρ' that correspond to states on ρ have the even rank m, the O-set henceforth always contains atleast one state with even rank m. As a result, the O-set is never empty and A' cannot see any more final states along ξ. Hence, ξ is non-accepting and α is rejected by A'. Hence, by the contrapositive, if α is accepted by A', then G_α must have an odd ranking. By an application of Lemma (2.1) α is also accepted by $L(\mathsf{KVF}(A))$.

□

Theorem 4.4. $L(A') = \overline{L(A)}$

The proof is established from the three sub-results: (i) every accepting run of A' on word α assigns an odd ranking to the run-DAG G_α and hence corresponds to an accepting run of $\mathsf{KVF}(A)$, (ii) the run corresponding to the ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ is an accepting run of A' on α, and (iii) $L(\mathsf{KVF}(A)) = \overline{L(A)}$[KV01]. □

Given an NBW A and the complement NBW A' constructed using the above algorithm, we now ask if A' has an accepting run on some $\alpha \in \overline{L(A)}$ that induces an odd ranking r different from $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$. We answer this question negatively in the following lemma.

Lemma 4.5. *Let $\alpha \in \overline{L(A)}$, and let r be the odd ranking corresponding to an accepting run of A' on α. Let $V_{r,i}$ (respectively, $V_{\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha},i}$) be the set of vertices in G_α that are assigned rank i by r (respectively, $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$). Then $V_{r,i} = V_{\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha},i}$ for all $i > 0$.*

Proof. We prove the claim by induction on the rank i. Since A and α are clear from the context, we will use r^{KV} in place of $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ in the remainder of the proof.
Base case: Let $(q, l) \in V_{\mathsf{r}^{\mathsf{KV}},1}$. By definition, (q, l) is F-free. Suppose $r((q, l)) = m$, where $m > 1$, if possible. If m is even, the constraints embodied in steps (2), (3) and (4(b)i) of our complementation construction, coupled with the fact that the O-set becomes $\emptyset$ infinitely often, imply that (q, l) has an F-vertex descendant (q', l') in G_α. Therefore, (q, l) is not F-free – a contradiction! Hence m cannot be even.

Suppose m is odd and > 1. The constraint embodied in step (4(b)ii) of our construction, and the fact that the O-set becomes $\emptyset$ infinitely often imply that (q, l) has a descendant (q'', l'') that is assigned an even rank by r. The constraints embodied in steps (2), (3) and (4(b)i), coupled with the fact that the O-set becomes $\emptyset$ infinitely often, further imply that (q'', l'') has an F-vertex descendant in G_α. Hence (q, l) has an F-vertex descendant, and is not F-free. This leads to a contradiction again! Therefore, our assumption must have been incorrect, i.e. $r((q,l)) \leq 1$. Since 1 is the minimum rank in the range of r, we finally have $r((q,l)) = 1$. This shows that $V_{\mathsf{r}^{\mathsf{KV}},1} \subseteq V_{r,1}$.

Now suppose $(q,l) \in V_{r,1}$. Since r corresponds to an accepting run of A', it is an odd-ranking. This, coupled with the fact that ranks cannot increase along any path in G_α, imply that all descendants of (q,l) in G_α are assigned rank 1 by r. Since F-vertices must be assigned even ranks, this implies that (q,l) is F-free in G_α. It follows that $\mathsf{r}^{\mathsf{KV}}((q,l)) = 1$. Therefore, $V_{r,1} \subseteq V_{\mathsf{r}^{\mathsf{KV}},1}$. From the above two results, we have $V_{r,1} = V_{\mathsf{r}^{\mathsf{KV}},1}$.
Hypothesis: Assume that $V_{r,j} = V_{\mathsf{r}^{\mathsf{KV}},j}$ for $1 \leq j \leq i$.
Induction: Let $(q,l) \in V_{\mathsf{r}^{\mathsf{KV}},i+1}$. Then by the induction hypothesis, (q,l) cannot be in any $V_{r,j}$ for $j \leq i$. Suppose $r((q,l)) = m$, where $m > i+1$, if possible. We have two cases.

(1) $i+1$ is odd: In this case, the constraints embodied in steps (2), (3), (4(b)i) and (4(b)ii) of our construction, coupled with the fact that the O-set becomes $\emptyset$ infinitely often, imply that vertex (q,l) has an F-vertex descendant (q',l') (possibly its own self) such that (i) $r((q',l')) = i+2$, and (ii) vertex (q',l') in turn has a descendant (q'',l'') such that $r((q'',l'')) = i+1$. The constraint embodied in (2) of our construction further implies that there must be an infinite path π starting from (q'',l'') in G_α such that every vertex on π is assigned rank $i+1$ by r, and none of these are F-vertices. Since $\mathsf{r}^{\mathsf{KV}}((q,l)) = i+1$ is odd, and since (q',l') is an F-vertex descendant of (q,l), we must have $\mathsf{r}^{\mathsf{KV}}((q',l')) \leq i$, where i is even. Furthermore, since r^{KV} is an odd ranking, every path in G_α must eventually get trapped in an odd rank assigned by r^{KV}. Hence, eventually every vertex on π is assigned an odd rank $< i$ by $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$. However, we already know that the vertices on π are eventually assigned the odd rank $i+1$ by r. Hence $V_{r,j} \neq V_{\mathsf{r}^{\mathsf{KV}},j}$ for some $j \in \{1,\ldots i\}$. This contradicts the inductive hypothesis!

(2) $i+1$ is even: In this case, the constraints embodied in steps (2), (3), (4(b)i) and (4(b)ii) of our construction, and the fact that the O-set becomes $\emptyset$ infinitely often, imply that (q,l) has a descendant (q',l') in G_α such that $r((q',l')) = i+2$ (which is odd), and there is an infinite path π starting from (q',l') such that all vertices on π are assigned rank $i+2$ by r. However, since r^{KV} is an odd ranking and since $\mathsf{r}^{\mathsf{KV}}((q,l)) = i+1$ (which is even), vertices on π must eventually get trapped in an odd rank $\leq i$ assigned by r^{KV}. This implies that $V_{r,j} \neq V_{\mathsf{r}^{\mathsf{KV}},j}$ for some $j \in \{1,\ldots i\}$. This violates the inductive hypothesis once again!

It follows from the above cases that $r((q,l)) \leq i+1$. However, $(q,l) \notin V_{\mathsf{r}^{\mathsf{KV}},j}$ for $1 \leq j \leq i$ (since $(q,l) \in V_{\mathsf{r}^{\mathsf{KV}},i+1}$), and $V_{r,j} = V_{\mathsf{r}^{\mathsf{KV}},j}$ for $1 \leq j \leq i$ (by inductive hypothesis). Therefore, $(q,l) \notin V_{r,j}$ for $1 \leq j \leq i$. This implies that $r((q,l)) = i+1$, completing the induction.

By the principle of mathematical induction, we $V_{\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha},i} = V_{r,i}$ for all $i > 0$. ☐

Theorem 4.6. *Every accepting run of A' on $\alpha \in \overline{L(A)}$ induces the unique minimal ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$.*

Proof. Follows from Lemma 4.5. ☐

Size of complement automaton. The states of A' are those in the set $\{2^Q \times 2^Q \times \mathcal{R}\}$. While some of these states correspond to tight level rankings, others do not. We first use an extension of the idea in [Sch09] to encode a state (S,O,f) with tight level ranking f as a pair (g,i), where $g : Q \to \{1,\ldots,r\} \cup \{-1,-2\}$ and $r = max\_odd(f)$. Thus, for all $q \in Q$, we have $q \notin S$ iff $g(q) = -2$. If $q \in S$ and $q \notin O$, we have $g(q) = f(q)$. If $q \in O$ and $f(q)$ is even, then we let $g(q) = -1$ and $i = f(q)$. This part of the encoding is exactly as in [Sch09]. We extend this encoding to consider cases where $q \in O$ and $f(q) = k$ is odd. There are two sub cases to consider: (i) $O \subsetneq \{q \mid q \in S \wedge f(q) = k\}$, and (ii) $O = \{q \mid q \in S \wedge f(q) = k\}$. In the first case, we let $i = k$ and $g(q) = -1$ for all $q \in O$. In the second case, we let $i = k$ and $g(q) = f(q) = k$ for all $q \in O$. Since, f is a tight level ranking, the O-set cannot be empty when we check for states with an odd rank in our construction. Therefore, there is no ambiguity in identifying the set O in both cases (i) and (ii) above. It is now easy to see that g is always onto one of the three sets $\{1, 3, \ldots, r\}$, $\{-1\} \cup \{1, 3, \ldots, r\}$ or $\{-2\} \cup \{1, 3, \ldots, r\}$. By Schewe's analysis [Sch09], the total number of such (g, i) pairs is upper bounded by $O(\mathit{tight}(n + 1))$.

Now, let us consider states with non-tight level rankings. Our construction ensures that once an odd rank i appears in a level ranking g along a run ρ, all subsequent level rankings along ρ contain every rank in $\{i, i + 2, \ldots \mathit{max\_odd}(g)\}$. The O-set in states with non-tight level ranking is inconsequential; hence we ignore this. Suppose a state with non-tight level ranking g contains the odd ranks $\{i, i - 2, \ldots, j\}$, where $1 < j \leq i = \mathit{max\_odd}(g)$. To encode this state, we first replace g with a level ranking g' as follows. For all $k \in \{j, \ldots, i\}$ and $q \in Q$, if $g(q) = k$, then $g'(q) = k - j + c$, where $c = 0$ if j is even and 1 otherwise. Effectively, this transforms g to a tight level ranking g' by shifting all ranks down by $j - c$. The original state can now be represented as the pair $(g', -(j - c))$. Note that the second component of a state represented as (g, i) is always non-negative for states with tight level ranking, and always negative for states with non-tight level ranking. Hence, there is no ambiguity in decoding the state representation. Clearly, the total no. of states with non-tight rankings is $O(n \cdot \mathit{tight}(n)) = O(\mathit{tight}(n + 1))$. Hence, the size of A' is upper bounded by $O(\mathit{tight}(n + 1))$ which differs from the lower bound of $\Omega(\mathit{tight}(n - 1))$ given by [Yan08] by only a factor of n^2.

5. Slices of complement automaton

The transitions of the complement automaton A' obtained by our construction have the property that a state (S, O, f) has a transition to another state (S', O', f') only if $\mathit{max\_odd}(f) = \mathit{max\_odd}(f')$. Consequently, the set of states Q' of A' can be partitioned into *slices* $Q_1, Q_3, \ldots, Q_{2n-1}$, such that $Q' = \bigcup_{i \in [2n-1]^{odd}} Q_i$. The set $Q_i = \{(S, O, f) | (S, O, f) \in Q' \wedge \mathit{max\_odd}(f) = i\}$ is called the i^{th} slice of A'.

The Figure4 shows the slice structure of the automaton A'. From the initial state of A' labelled Start, on reading a letter σ of the input, A' makes a nondeterministic jump to each of the states $\mathsf{Start(i)} = (S_i, O_i, f_i)$, where $S_i = \{q_0\}$, $O_i = \emptyset$ and $f_i(q_0) = i$ for $i \in [2n - 1]^{odd}$. From every such state $\mathsf{Start(i)}$ the automaton A' can move only to states such that the level ranking corresponding to that state assignes a maximum odd rank of i. Hence, once the initial nondeterministic jump is made any infinite run of A' always remains confined to the same slice, shown in Figure4 by the absence of transitions between slices.

Let ρ be an accepting run of A' on $\alpha \in \overline{L(A)}$. We say that ρ is *confined* to a slice Q_i of A' iff ρ sees only states from Q_i. If ρ is confined to slice i, and if r is the odd ranking induced by ρ, then $\mathit{max\_odd}(r) = i$.

Lemma 5.1. *All accepting runs of A' on $\alpha \in \overline{L(A)}$ are confined to the same slice.*

Proof. By Theorem 4.6, we know that every accepting run of A' on a word $\alpha \in \overline{L(A)}$ induces the unique minimal odd ranking $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$. Let k be the maximum odd rank assigned by $\mathsf{r}^{\mathsf{KV}}_{\mathsf{A},\alpha}$ to the run-DAG G_α. Then the maximum odd rank that is assigned by any level ranking along any accepting run in A' on α is equal to k. Hence, every state on every accepting run belongs to the k^{th} slice of A'.

□

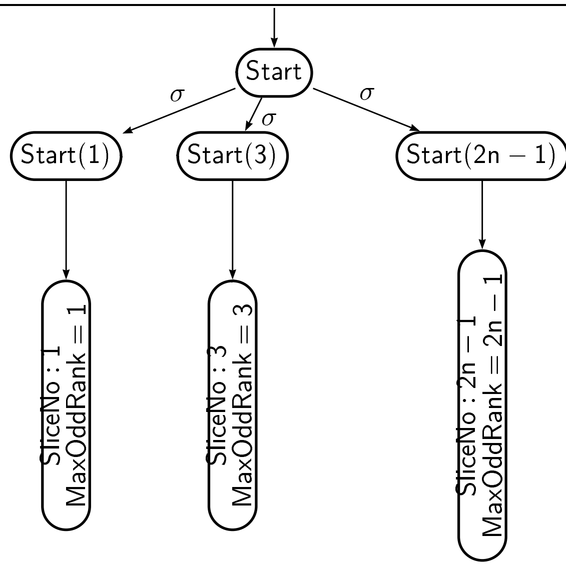

Figure 4: Slice structure of complement automaton A'

The above results indicate that if a word α is accepted by the i^{th} slice of our automaton A', then it cannot be accepted by $\mathsf{KVF}(A)$ using a tight ranking with max odd rank $< i$. It is however possible that the same word is accepted by $\mathsf{KVF}(A)$ using a tight ranking with max odd rank $> i$. Figure 1 shows an example of such an automaton, where the word a^ω is accepted by $\mathsf{KVF}(A)$ using a tight ranking with max odd rank 5, as well as with a tight ranking with max odd rank 3. But, the same word is accepted by only the 3^{rd} slice of our automaton A'. This motivates the definition of *minimally inessential ranks.*

Definition 5.2 (Minimally inessential ranks)**.** Given an NBW A with n states, odd rank i $(1 \leq i \leq 2n-1)$ is said to be minimally inessential if every word α that is accepted by $\mathsf{KVF}(A)$ using a tight ranking with max odd rank i is also accepted by $\mathsf{KVF}(A)$ using a tight ranking with max odd rank $j < i$.

An odd rank that is not minimally inessential is called minimally essential. As the example in Figure 1 shows, neither the KVF-construction nor the KVFS-construction allows us to detect minimally essential ranks in a straightforward way. Specifically, although the 5^{th} slice of $\mathsf{KVF}(A)$ for this example accepts the word a^ω, 5 is not a minimally inessential rank. In order to determine if 5 is minimally inessential, we must isolate the 5^{th} slice of $\mathsf{KVF}(A)$, and then check whether the language accepted by this slice is a subset of the language accepted by $\mathsf{KVF}(A)$ sans the 5^{th} slice. This involves complementing $\mathsf{KVF}(A)$ sans the 5^{th} slice, requiring a significant blowup. In contrast, the properties of our automaton A' allow us to detect minimally (in)essential ranks efficiently. Specifically, if we find that the i^{th} slice of A' accepts a word α, we can infer than i is minimally essential. Once all minimally essential ranks have been identified in this manner, we can prune automaton A' to retain only those slices that correspond to minimally essential ranks. This gives us a way of eliminating redundant slices (and hence states) in $\mathsf{KVF}(A)$.

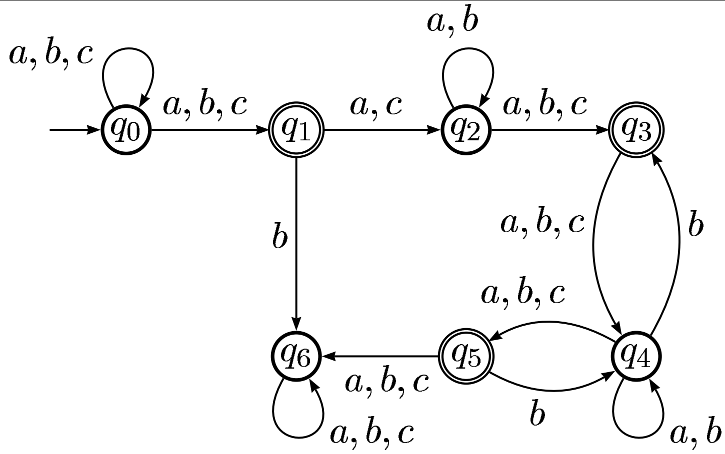

Figure 5: Example automaton with gaps in MI ranks

Another significant advantage of our construction is its ability to detect "gaps" in slices. As an example, the automaton in Figure (5) has ranks $1, 5$ as minimally inessential, while ranks 3 and 7 are minimally essential (see Table (1)). Following the definitions in [GKSV03] an integer j is a *required rank* for an NBW $\overline{A}$ if there exists a word $\beta \in L(A)$ such that some vertex in the run-DAG G_β gets rank j and the *rank* of A is the maximal rank required for A. Suppose, we compute the *rank* of the NBW A' (as suggested in [GKSV03]) and then consider slices only upto this *rank*, we will fail to detect that rank 5 is minimally inessential. Therefore, eliminating redundant slices is a stronger optimization than identifying and eliminating states with ranks greater than the rank of the NBW and can help in reducing the size of the complement NBW A' even further.

6. An implementation of our algorithm

We have implemented the complementation algorithm presented in this paper as a facility on top of the BDD-based model checker NuSMV. In our implementation, states of the complement automaton are encoded as pairs (g, i), as explained in Section 4.2. Our tool takes as input an NBW A, and generates the state transition relation of the complement automaton A' using the above encoding in NuSMV's input format. Generating the NuSMV file from a given description of NBW takes negligible time (< 0.01s). The number of boolean constraints used in expressing the transition relation in NuSMV is quadratic in n. We use NuSMV's fair CTL model checking capability to check whether there exists an infinite path in a slice of A' (corresponding to a maximum odd rank k) that visits an accepting state infinitely often. If NuSMV responds negatively to such a query, we disable all transitions

Automaton	KVFS algorithm		Our algorithm		GOAL	
(states,trans.,final)	States	MI Ranks	States	MI Ranks	WAA	SP
michel4(6,15,1)	157	{9,7}	117	{9,7}	XX	105
g15 (6,17,1)	324	{9,7,5}	155	{9,7,5}	XX	39
g47 (6,13,3)	41	{5}	29	{5,3}	XX	28
ex4 (8,9,4)	3956	$\emptyset$	39	{7,5}	XX	7
ex16 (9,13,5)	10302	$\emptyset$	909	{7}	XX	30
ex18 (9,13,5)	63605	$\emptyset$	2886	$\emptyset$	XX	71
ex20 (9,12,5)	17405	$\emptyset$	156	{7}	XX	24
ex22 (11,18,7)	258	{7,5}	23	{7,5}	XX	6
ex24 (13,16,8)	4141300	$\emptyset$	19902	{9,7}	XX	51
ex26 (15,22,11)	1042840	$\emptyset$	57540	{7,5}	XX	XX
gap1 (15,22,11)	99	{5,1}	26	{5,1}	XX	16
gap2 (15,22,11)	532	{1}	80	{5,1}	XX	48

Table 1: Experimental Results. XX:timeout (after 10 min)

to and from states in this section of the NuSMV model. This allows us to effectively detect and eliminate redundant slices of our complement automaton, resulting in a reduction of the overall size of the automaton. For purposes of comparison, we have also implemented the algorithm presented in [Sch09] in a similar manner using a translation to NuSMV. We also compared the performance of our technique with those of Safra-Piterman [Saf88, Pit07] determinization based complementation technique and Kupferman-Vardi's ranking based complementation technique [KV01], as implemented in the GOAL tool [TCT$^+$08]. Table 1 shows the results of some of our experiments. We used the CUDD BDD library with NuSMV 2.4.3, and all our experiments were run on an Intel Xeon 3GHz with 2 GB of memory and a timeout of 10 minutes. The entry for each automaton [1] lists the number of states, transitions and final states of the original automaton, the number of states of the complement automaton computed by the KVFS-construction and by our construction, and the set of minimally inessential ranks (denoted as "MI Ranks") identified by each of these techniques. In addition, each row also lists the number of states computed by the "Safra-Piterman" (denoted as "SP") technique and "Weak Alternating Automata" (denoted as "WAA") technique in GOAL.

A significant advantage of our construction is its ability to detect "gaps" in slices. As an example, the automaton in Figure (5) has ranks $1, 5$ as minimally inessential, while ranks 3 and 7 are minimally essential (see Table (1)). In this case, if we compute the *rank* of the NBW (as suggested in [GKSV03]) and then consider slices only upto this *rank*, we will fail to detect that rank 5 is minimally inessential. Therefore, eliminating redundant slices is a stronger optimization than identifying and eliminating states with ranks greater than the rank of the NBW.

Another interesting point to note about our NuSMV-based implementation is the following. Even if we comment out one or more conjunctive transition constraints that correspond to how the O set (in the (S, O, f) representation of a state) evolves when it is tracking an

[1] All example automata from this paper and the translator from automaton description to NuSMV are available at http://www.cfdvs.iitb.ac.in/reports/minrank

odd rank, the language accepted by the resulting automaton remains the same (complement of the language of the input NBW). The correctness of the construction depends on how the O set evolves when it is tracking an even rank, while the minimality of ranks depends on how the O set evolves when it is tracking an odd rank. This observation allows us to optimize the reachable state set computation of some of our complement automata. Specifically, for examples "ex20", "ex22" and "ex24", NuSMV was unable to compute the set of reachable states in our full-blown complementation construction within 10 minutes. However, on commenting out some constraints that correspond to evolution of the O-set when tracking an odd rank, we were able to get NuSMV to terminate within 10 minutes (often much earlier). Thus, the state count and eliminated ranks reported in the table for these examples can potentially be improved further by allowing NuSMV to run longer. This ability to achieve a limited tradeoff between size of the complement automaton and computational resources used without compromising the correctness of the construction, simply by commenting out appropriate constraints in the NuSMV input file, is a unique feature of our implementation. We wish to investigate the effect of combining further optimization techniques (including known ones, e.g. in [GKSV03]) with our technique in future.

7. Conclusion

In this paper, we presented a complementation algorithm for nondeterministic Büchi automata that is based on the idea of ranking functions introduced by Kupferman and Vardi[KV01]. We showed that the ranking assignment presented in [KV01] always results in a minimal odd ranking for run-DAGs of words in the complement language. We then described a complementation construction for NBW such that the complement NBW accepts only the run-DAG with the minimal odd ranking for every word in the complement. We observed that the states of the complement NBW are partioned into *slices*, and that each word in the complement is accepted by exactly one such slice. This allowed us to check for redundant slices and eliminate them, leading to a reduction the size of the complement NBW. It is noteworthy that this ability to reduce the size of the final complement NBW essentially comes for free since the worst case bounds coincide with the worst case bounds of the best known NBW

References

[Büc62] J. R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford Univ. Press, Stanford, 1962.

[DR09] Laurent Doyen and Jean-Francois Raskin. Antichains for the automata based approach to model checking. *Logical Methods in Computer Science*, 5:1–20, 2009.

[FKV06] E. Friedgut, O. Kupferman, and M. Y. Vardi. Büchi complementation made tighter. *Int. J. Found. Comput. Sci.*, 17(4):851–868, 2006.

[FV09] Seth Fogarty and Moshe Y. Vardi. Büchi complementation and size-change termination. In *Proc. TACAS*, pages 16–30, 2009.

[GKSV03] Sankar Gurumurthy, Orna Kupferman, Fabio Somenzi, and Moshe Y. Vardi. On complementing nondeterministic büchi automata. In *Proc. CHARME*, pages 96–110, 2003.

[Kla91] N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. 32nd IEEE FOCS, San Juan*, pages 358–367, 1991.

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

[KW08] Detlef Kähler and Thomas Wilke. Complementation, disambiguation, and determinization of Büchi automata unified. In *Proc. ICALP*, pages 724–735, 2008.

[Löd99] C. Löding. Optimal bounds for transformations of ω-automata. In *Proc. 19th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 97–109, 1999.

[Mic88] M. Michel. Complementation is more difficult with automata on infinite words. *CNET, Paris*, 1988.

[Pit07] N. Piterman. From nondeterministic Büchi and Streett automata to deterministic Parity automata. *Logical Methods in Computer Science*, 3(3:5):1–217, 2007.

[Saf88] S. Safra. On the complexity of ω-automata. In *Proc. 29th IEEE FOCS*, pages 319–327, 1988.

[Sch09] Sven Schewe. Büchi complementation made tight. In *Proc. STACS*, pages 661–672, 2009.

[SVW87] A. Prasad Sistla, Moshe Y. Vardi, and Pierre Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49:217–237, 1987.

[TCT+08] Yih-Kuen Tsay, Yu-Fang Chen, Ming-Hsien Tsai, Wen-Chin Chan, and Chi-Jian Luo. Goal extended: Towards a research tool for omega automata and temporal logic. In *Proc. TACAS*, pages 346–350, March 2008.

[Var07] Moshe Y. Vardi. The Büchi complementation saga. In *STACS*, pages 12–22, 2007.

[Yan08] Q. Yan. Lower bounds for complementation of omega-automata via the full automata technique. *Logical Methods in Computer Science*, 4(1):1–20, 2008.