Rank-based complementation construction (0.97n)^n
Based on Kupferman, Vardi: Weak alternating automata are not that weak, 2001

International Journal of Foundations of Computer Science
Vol. 17, No. 4 (2006) 851–867

World Scientific
www.worldscientific.com

BÜCHI COMPLEMENTATION MADE TIGHTER*

EHUD FRIEDGUT
Hebrew University,
Institute of Mathematics, Jerusalem 91904, Israel
Email: ehud@math.huji.ac.il

ORNA KUPFERMAN
Hebrew University,
School of Engineering and Computer Science, Jerusalem 91904, Israel
Email: orna@cs.huji.ac.il

MOSHE Y. VARDI
Rice University,
Department of Computer Science, Houston, TX 77251-1892, U.S.A.
Email:vardi@cs.rice.edu

Received 3 May 2005
Accepted 16 March 2006
Communicated by Farn Wang

ABSTRACT

The complementation problem for nondeterministic word automata has numerous applications in formal verification. In particular, the language-containment problem, to which many verification problems is reduced, involves complementation. For automata on finite words, which correspond to safety properties, complementation involves determinization. The 2^n blow-up that is caused by the subset construction is justified by a tight lower bound. For Büchi automata on infinite words, which are required for the modeling of liveness properties, optimal complementation constructions are quite complicated, as the subset construction is not sufficient. From a theoretical point of view, the problem is considered solved since 1988, when Safra came up with a determinization construction for Büchi automata, leading to a $2^{O(n \log n)}$ complementation construction, and Michel came up with a matching lower bound. A careful analysis, however, of the exact blow-up in Safra's and Michel's bounds reveals an exponential gap in the constants hiding in the $O()$ notations: while the upper bound on the number of states in Safra's complementary automaton is n^{2n}, Michel's lower bound involves only an $n!$ blow up, which is roughly $(n/e)^n$. The exponential gap exists also in more recent complementation constructions. In particular, the upper bound on the number of states in the complementation construction of Kupferman and Vardi, which avoids determinization, is $(6n)^n$. This is in contrast with the case of automata on finite words, where the upper and lower bounds coincides. In this work we describe an improved complementation construction for nondeterministic Büchi automata and analyze its complexity. We show that the new construction results in an automaton with at most $(0.96n)^n$ states. While this leaves the problem about the exact blow up open, the gap is now exponentially smaller.

*A preliminary version of this paper appears in the proceedings of the 2nd International Symposium on Automated Technology for Verification and Analysis, LNCS 3299, 2004.

From a practical point of view, our solution enjoys the simplicity of the construction of Kupferman and Vardi, and results in much smaller automata.

Keywords: Nondeterministic Büchi Automata, Complementation.

1. Introduction

The complementation problem for nondeterministic word automata has numerous applications in formal verification. In order to check that the language of an automaton $\mathcal{A}_1$ is contained in the language of a second automaton $\mathcal{A}_2$, one checks that the intersection of $\mathcal{A}_1$ with an automaton that complements $\mathcal{A}_2$ is empty. Many problems in verification and design are reduced to language containment. In model checking, the automaton $\mathcal{A}_1$ corresponds to the system, and the automaton $\mathcal{A}_2$ corresponds to the property we wish to verify [13, 23]. While it is easy to complement properties given in terms of formulas in temporal logic, complementation of properties given in terms of automata is not simple. Indeed, a word w is rejected by a nondeterministic automaton $\mathcal{A}$ if all the runs of $\mathcal{A}$ on w rejects the word. Thus, the complementary automaton has to consider all possible runs, and complementation has the flavor of determinization. For automata on finite words, determinization, and hence also complementation, is done via the subset construction [17]. Accordingly, if we start with a nondeterministic automaton with n states, the complementary automaton may have 2^n states. The exponential blow-up that is caused by the subset construction is justified by a tight lower bound: it is proved in [19] that for every $n > 1$, there exists a language L_n that is recognized by a nondeterministic automaton with n states, yet a nondeterministic automaton for the complement of L_n has at least 2^n states (see also [2], for a similar result, in which the languages L_n are over an alphabet of size 4).

For Büchi automata on infinite words, which are required for the modeling of liveness properties, optimal complementation constructions are quite complicated, as the subset construction is not sufficient. Due to the lack of a simple complementation construction, the user is typically required to specify the property by a deterministic Büchi automaton [13] (it is easy to complement a deterministic Büchi automaton), or to supply the automaton for the negation of the property [9]. Similarly, specification formalisms like ETL [24], which have automata within the logic, involve complementation of automata, and the difficulty of complementing Büchi automata is an obstacle to practical use [1]. In fact, even when the properties are specified in LTL, complementation is useful: the translators from LTL into automata have reached a remarkable level of sophistication (c.f., [4, 21, 5, 6]). Even though complementation of the automata is not explicitly required, the translations are so involved that it is useful to checks their correctness, which involves complementation[a]. Complementation is interesting in practice also because it enables refinement and optimization techniques that are based on language

[a]For an LTL formula ψ, one typically checks that both the intersection of $\mathcal{A}_\psi$ with $\mathcal{A}_{\neg\psi}$ and the intersection of their complementary automata are empty. As shown in [7], the complementation construction in [12] is feasible for automata obtained from LTL formulas.

containment rather than simulation[b]. Thus, an effective algorithm for the complementation of Büchi automata would be of significant practical value.

Efforts to develop simple complementation constructions for nondeterministic automata started early in the 60s, motivated by decision problems of second-order logics. Büchi suggested a complementation construction for nondeterministic Büchi automata that involved a complicated combinatorial argument and a doubly-exponential blow-up in the state space [3]. Thus, complementing an automaton with n states resulted in an automaton with $2^{2^{O(n)}}$ states. In [20], Sistla et al. suggested an improved construction, with only $2^{O(n^2)}$ states, which is still, however, not optimal. Only in [18], Safra introduced a determinization construction, which also enabled a $2^{O(n \log n)}$ complementation construction, matching a lower bound described by Michel [15]. Thus, from a theoretical point of view, the problem is considered solved since 1988. A careful analysis, however, of the exact blow-up in Safra's and Michel's bounds reveals an exponential gap in the constants hiding in the $O()$ notations: while the upper bound on the number of states in the complementary automaton constructed by Safra is n^{2n}, Michel's lower bound involves only an $n!$ blow up, which is roughly $(n/e)^n$. The exponential gap exists also in more recent complementation constructions. In particular, the upper bound on the number of states in the complementation construction in [12], which avoids determinization, is $(6n)^n$. A similar bound was given in [14], which uses the analysis in [12]. This is in contrast with the case of automata on finite words, where, as mentioned above, the upper and lower bounds coincide.

In this work we describe an improved complementation construction for nondeterministic Büchi automata and analyze its complexity. The construction is based on new observations on runs of nondeterministic Büchi automata: a run of a nondeterministic Büchi automaton $\mathcal{A}$ is accepting if it visits the set α of accepting states infinitely often. Accordingly, $\mathcal{A}$ rejects a word w if every run of $\mathcal{A}$ visits α only finitely often. The runs of $\mathcal{A}$ can be arranged in a DAG (directed acyclic graph). It is shown in [12] that $\mathcal{A}$ rejects w iff it is possible to label the vertices of the DAG by ranks in $0, \ldots, 2n$ so that some local conditions on the ranks of vertices and their successors are met. Intuitively, as in the progress measure of [11], the ranks measure the distance to a position from which no states in α are visited. We show here that the ranks that label vertices of the same level in the DAG have an additional property: starting from some limit level $l_{lim} \geq 0$, if a vertex in level $l \geq l_{lim}$ is labeled by an odd rank j, then all the odd ranks in $1, \ldots, j$ label vertices in level l. It follows that the complementary automaton, which considers all the possible *level rankings* (i.e., ranks that vertices of some level in the DAG are labeled with), may restrict attention to a special class of level rankings. Using some estimates on the asymptotics of Stirling numbers of the second kind we are able to bound the size of this class and describe a complementation construction with only $(3cn)^n$ states, for $c < 0.76$. We then tighten the analysis further and show that our complementary

[b]Since complementation of Büchi automata is complicated, current research is focused on ways in which fair simulation can approximate language containment [8], and ways in which the complementation construction can be circumvented by manually bridging the gap between fair simulation and language containment [10].

automaton has at most $(0.96n)^n$ states. While this leaves the problem about the exact blow up that complementation involves open, the gap is now exponentially smaller: instead of an upper bound of $(6n)^n$ states, we now have at most $(0.96n)^n$ states. From a practical point of view, our solution enjoys the simplicity of [12], and results in much smaller automata. Moreover, the optimization constructions described in [7] for the construction of [12] can be applied also in our new construction, leading in practice to further reduction of the state space. Finally, for the application of the complementation construction to language containment, we show how the complementing automaton can be optimized further – an optimization that depends on both arguments of the complementation problem.

2. Preliminaries

Given an alphabet Σ, an *infinite word over* Σ is an infinite sequence $w = \sigma_0 \cdot \sigma_1 \cdots$ of letters in Σ. An *automaton on infinite words* is $\mathcal{A} = \langle \Sigma, Q, Q_{in}, \rho, \alpha \rangle$, where Σ is the input alphabet, Q is a finite set of states, $\rho : Q \times \Sigma \to 2^Q$ is a transition function, $Q_{in} \subseteq Q$ is a set of initial states, and α is an acceptance condition (a condition that defines a subset of Q^ω). Intuitively, $\rho(q, \sigma)$ is the set of states that $\mathcal{A}$ can move into when it is in state q and it reads the letter σ. Since the transition function of $\mathcal{A}$ may specify many possible transitions for each state and letter, $\mathcal{A}$ is in general not *deterministic.* If $|Q_{in}| = 1$ and ρ is such that for every $q \in Q$ and $\sigma \in \Sigma$, we have that $|\rho(q, \sigma)| = 1$, then $\mathcal{A}$ is a deterministic automaton.

A *run* of $\mathcal{A}$ on w is a function $r : \mathbb{N} \to Q$ where $r(0) \in Q_{in}$ (i.e., the run starts in an initial state) and for every $l \geq 0$, we have $r(l+1) \in \rho(r(l), \sigma_l)$ (i.e., the run obeys the transition function). In automata over finite words, acceptance is defined according to the last state visited by the run. When the words are infinite, there is no "last state", and acceptance is defined according to the set $Inf(r)$ of states that r visits *infinitely often*, i.e.,

$$Inf(r) = \{q \in Q \ : \text{for infinitely many } l \in \mathbb{N}, \text{we have } r(l) = q\}.$$

As Q is finite, it is guaranteed that $Inf(r) \neq \emptyset$. The way we refer to $Inf(r)$ depends on the acceptance condition of $\mathcal{A}$. In *Büchi automata*, $\alpha \subseteq Q$, and r is accepting iff $Inf(r) \cap \alpha \neq \emptyset$. Dually, in *co-Büchi automata*, $\alpha \subseteq Q$, and r is accepting iff $Inf(r) \cap \alpha = \emptyset$.

Since $\mathcal{A}$ is not necessarily deterministic, it may have many runs on w. In contrast, a deterministic automaton has a single run on w. There are two dual ways in which we can refer to the many runs. When $\mathcal{A}$ is an *existential* automaton (or simply a *nondeterministic* automaton, as we shall call it in the sequel), it accepts an input word w iff there exists an accepting run of $\mathcal{A}$ on w. When $\mathcal{A}$ is a *universal* automaton, it accepts an input word w iff all the runs of $\mathcal{A}$ on w are accepting.

We use three-letter acronyms to describe types of automata. The first letter describes the transition structure and is one of "D" (deterministic), "N" (nondeterministic), and "U" (universal). The second letter describes the acceptance condition; in this paper we only consider "B" (Büchi) and "C" (co-Büchi). The third letter describes the objects on which the automata run; in this paper we are only concerned with "W" (infinite words). Thus, for example, NBW designates a nondeterministic Büchi word automaton and UCW designates a universal co-Büchi word automaton.

In [12], we suggested the following approach for NBW complementation: in order to complement an NBW, first dualize the transition function and the acceptance condition, and then translate the resulting UCW automaton back to a nondeterministic one. By [16], the dual automaton accepts the complementary language, and so does the nondeterministic automaton we end up with. Thus, rather than determinization, complementation is based on a translation of universal automata to nondeterministic ones, which turned out to be much simpler.

Consider a UCW $\mathcal{A} = \langle \Sigma, Q, Q_{in}, \delta, \alpha \rangle$. The runs of $\mathcal{A}$ on a word $w = \sigma_0 \cdot \sigma_1 \cdots$ can be arranged in an infinite DAG (directed acyclic graph) $\mathcal{G} = \langle V, E \rangle$, where

- $V \subseteq Q \times \mathbb{N}$ is such that $\langle q, l \rangle \in V$ iff some run r of $\mathcal{A}$ on w has $r(l) = q$. For example, the first level of $\mathcal{G}$ contains the vertices $Q_{in} \times \{0\}$.
- $E \subseteq \bigcup_{l \geq 0} (Q \times \{l\}) \times (Q \times \{l+1\})$ is such that $E(\langle q, l \rangle, \langle q', l+1 \rangle)$ iff $\langle q, l \rangle \in V$ and $q' \in \delta(q, \sigma_l)$.

Thus, $\mathcal{G}$ embodies exactly all the runs of $\mathcal{A}$ on w. We call $\mathcal{G}$ the *run* DAG of $\mathcal{A}$ on w, and we say that $\mathcal{G}$ is *accepting* if all its paths satisfy the acceptance condition α. Note that $\mathcal{A}$ accepts w iff $\mathcal{G}$ is accepting. We say that a vertex $\langle q', l' \rangle$ is a *successor* of a vertex $\langle q, l \rangle$ iff $E(\langle q, l \rangle, \langle q', l' \rangle)$. We say that $\langle q', l' \rangle$ is *reachable* from $\langle q, l \rangle$ iff there exists a sequence $\langle q_0, l_0 \rangle, \langle q_1, l_1 \rangle, \langle q_2, l_2 \rangle, \ldots$ of successive vertices such that $\langle q, l \rangle = \langle q_0, l_0 \rangle$, and there exists $i \geq 0$ such that $\langle q', l' \rangle = \langle q_i, l_i \rangle$. For a set $S \subseteq Q$, we say that a vertex $\langle q, l \rangle$ of $\mathcal{G}$ is an *S-vertex* if $q \in S$.

Consider a (possibly finite) DAG $\mathcal{G} \subseteq \mathcal{G}$. We say that a vertex $\langle q, l \rangle$ is *finite* in $\mathcal{G}$ if only finitely many vertices in $\mathcal{G}$ are reachable from $\langle q, l \rangle$. For a set $S \subseteq Q$, we say that a vertex $\langle q, l \rangle$ is *S-free* in $\mathcal{G}$ if all the vertices in $\mathcal{G}$ that are reachable from $\langle q, l \rangle$ are not S-vertices. Note that, in particular, an S-free vertex is not an S-vertex. We say that a level l of $\mathcal{G}$ is of *width* $d \geq 0$ if there are d vertices of the form $\langle q, l \rangle$ in $\mathcal{G}$. Finally, the *width* of $\mathcal{G}$ is the maximal $d \geq 0$ such that there are infinitely many levels l of width d. The *α-less* width of a level of $\mathcal{G}$ is defined similarly, restricted to vertices $\langle q, l \rangle$ for which $q \notin \alpha$. Note that the width of $\mathcal{G}$ is at most n and the α-less width of $\mathcal{G}$ is at most $n - |\alpha|$.

Runs of UCW were studied in [12]. For $x \in \mathbb{N}$, let $[x]$ denote the set $\{0, 1, \ldots, x\}$, and let $[x]^{odd}$ and $[x]^{even}$ denote the set of odd and even members of $[x]$, respectively. A *co-Büchi-ranking* for $\mathcal{G}$ (*C-ranking*, for short) is a function $f : V \to [2n]$ that satisfies the following two conditions:

1. For all vertices $\langle q, l \rangle \in V$, if $f(\langle q, l \rangle)$ is odd, then $q \notin \alpha$.
2. For all edges $\langle \langle q, l \rangle, \langle q', l+1 \rangle \rangle \in E$, we have $f(\langle q', l+1 \rangle) \leq f(\langle q, l \rangle)$.

Thus, a C-ranking associates with each vertex in $\mathcal{G}$ a rank in $[2n]$ so that the ranks along paths do not increase, and α-vertices get only even ranks. We say that a vertex $\langle q, l \rangle$ is an *odd vertex* if $f(\langle q, l \rangle)$ is odd. Note that each path in $\mathcal{G}$ eventually gets trapped in some rank. We say that the C-ranking f is an *odd C-ranking* if all the paths of $\mathcal{G}$ eventually get trapped in an odd rank. Formally, f is odd iff for all paths $\langle q_0, 0\rangle, \langle q_1, 1\rangle, \langle q_2, 2\rangle, \ldots$ in $\mathcal{G}$, there is $l \geq 0$ such that $f(\langle q_l, l\rangle)$ is odd, and for all $l' \geq l$, we have $f(\langle q_{l'}, l'\rangle) = f(\langle q_l, l\rangle)$. Note that, equivalently, f is odd if every path of $\mathcal{G}$ has infinitely many odd vertices.

Lemma 1 [12] *The following are equivalent.*

1. *All the paths of $\mathcal{G}$ have only finitely many α-vertices.*
2. *There is an odd C-ranking for $\mathcal{G}$.*

Proof. Assume first that there is an odd C-ranking for $\mathcal{G}$. Then, every path in $\mathcal{G}$ eventually gets trapped in an odd rank. Hence, as α-vertices get only even ranks, all the paths of $\mathcal{G}$ visit α only finitely often, and we are done.

For the other direction, given an accepting run DAG $\mathcal{G}$, we define an infinite sequence $\mathcal{G}_0 \supseteq \mathcal{G}_1 \supseteq \mathcal{G}_2 \supseteq \ldots$ of DAGs inductively as follows.

- $\mathcal{G}_0 = \mathcal{G}$.
- $\mathcal{G}_{2i+1} = \mathcal{G}_{2i} \setminus \{\langle q, l\rangle \mid \langle q, l\rangle \text{ is finite in } \mathcal{G}_{2i}\}$.
- $\mathcal{G}_{2i+2} = \mathcal{G}_{2i+1} \setminus \{\langle q, l\rangle \mid \langle q, l\rangle \text{ is } \alpha\text{-free in } \mathcal{G}_{2i+1}\}$.

It is shown in [12] that for every $i \geq 0$, the transition from $\mathcal{G}_{2i+1}$ to $\mathcal{G}_{2i+2}$ involves the removal of an infinite path from $\mathcal{G}_{2i+1}$. Intuitively, it follows from the fact that as long as $\mathcal{G}_{2i+1}$ is not empty, it contains a at least one α-free vertex, from which an infinite path of α-free vertices start. Since the width of $\mathcal{G}_0$ is bounded by n, it follows that the width of $\mathcal{G}_{2i}$ is at most $n - i$. Hence, $\mathcal{G}_{2n}$ is finite, and $\mathcal{G}_{2n+1}$ is empty. In fact, as argued in [7], the α-less width of $\mathcal{G}_{2i}$ is at most $n - (|\alpha| + i)$, implying that $\mathcal{G}_{2(n-|\alpha|)+1}$ is already empty. Since $|\alpha| \geq 1$, we can therefore assume that $\mathcal{G}_{2n-1}$ is empty.

Each vertex $\langle q, l\rangle$ in $\mathcal{G}$ has a unique index $i \geq 1$ such that $\langle q, l\rangle$ is either finite in $\mathcal{G}_{2i}$ or α-free in $\mathcal{G}_{2i+1}$. Thus, the sequence of DAGs induces a function $rank : V \rightarrow [2n-2]$, defined as follows.

$$rank(q, l) = \left[\begin{array}{ll} 2i & \text{If } \langle q, l\rangle \text{ is finite in } \mathcal{G}_{2i}. \\ 2i+1 & \text{If } \langle q, l\rangle \text{ is } \alpha\text{-free in } \mathcal{G}_{2i+1}. \end{array}\right.$$

It is shown in [12] that the function *rank* is an odd C-ranking. □

We now use C-ranking in order to translate UCW to NBW:

Theorem 1 [12, 7] *Let $\mathcal{A}$ be a* UCW *with n states. There is an* NBW *$\mathcal{A}'$ with at most $3^n \cdot (2n-1)^n$ states such that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof. Let $\mathcal{A} = \langle \Sigma, Q, Q_{in}, \delta, \alpha\rangle$. When $\mathcal{A}'$ reads a word w, it guesses an odd C-ranking for the run DAG $\mathcal{G}$ of $\mathcal{A}$ on w. At a given point of a run of $\mathcal{A}'$, it keeps in its memory a whole level of $\mathcal{G}$ and a guess for the rank of the vertices at this level. In order to make sure that all the paths of $\mathcal{G}$ visit infinitely many odd vertices, $\mathcal{A}'$ remembers the set of states that owe a visit to an odd vertex.

Before we define $\mathcal{A}'$, we need some notations. A *level ranking* for $\mathcal{A}$ is a function $g : Q \to [2n-2]$, such that if $g(q)$ is odd, then $q \notin \alpha$. Let $\mathcal{R}$ be the set of all level rankings. For a subset S of Q and a letter σ, let $\delta(S, \sigma) = \bigcup_{s \in S} \delta(s, \sigma)$. Note that if level l in $\mathcal{G}$, for $l \geq 0$, contains the states in S, and the $(l+1)$-th letter in w is σ, then level $l+1$ of $\mathcal{G}$ contains the states in $\delta(S, \sigma)$.

For two level rankings g and g' in $\mathcal{R}$, a set $S \subseteq Q$, and a letter σ, we say that g' *covers* $\langle g, S, \sigma \rangle$ if for all $q \in S$ and $q' \in \delta(q, \sigma)$, we have $g'(q') \leq g(q)$. Thus, if the vertices of level l contain exactly all the states in S, g describes the ranks of these vertices, and the $(l+1)$-th letter in w is σ, then g' is a possible level ranking for level $l+1$. Finally, for $g \in \mathcal{R}$, let $odd(g) = \{q : g(q) \in [2n-2]^{odd}\}$. Thus, a state of Q is in $odd(g)$ if it has an odd rank.

Now, $\mathcal{A}' = \langle \Sigma, Q', Q'_{in}, \delta', \alpha' \rangle$, where

- $Q' \subseteq 2^Q \times 2^Q \times \mathcal{R}$, where a state $\langle S, O, g \rangle \in Q'$ indicates that the current level of the DAG contains the states in S, the set $O \subseteq S$ contains states along paths that have not visited an odd vertex since the last time O has been empty, and g is the guessed level ranking for the current level.
- $Q'_{in} = \{Q_{in}\} \times \{\emptyset\} \times \mathcal{R}$.
- δ' is defined, for all $\langle S, O, g \rangle \in Q'$ and $\sigma \in \Sigma$, as follows.
  - If $O \neq \emptyset$, then
  $$\delta'(\langle S, O, g \rangle, \sigma) = \{\langle \delta(S, \sigma), \delta(O, \sigma) \setminus odd(g'), g' \rangle : g' \text{ covers } \langle g, S, \sigma \rangle\}.$$
  - If $O = \emptyset$, then
  $$\delta'(\langle S, O, g \rangle, \sigma) = \{\langle \delta(S, \sigma), \delta(S, \sigma) \setminus odd(g'), g' \rangle : g' \text{ covers } \langle g, S, \sigma \rangle\}.$$
- $\alpha' = 2^Q \times \{\emptyset\} \times \mathcal{R}$. Thus, a run has to visit a state with an empty O-component infinitely many times. Note that this implies that the guessed C-ranking is such that each path of the run †visits infinitely many vertices with an odd rank. Thus, the guessed C-ranking is odd.

Consider a state $\langle S, O, g \rangle \in Q'$. Since $O \subseteq S$, there are at most 3^n pairs S and O that can be members of the same state. In addition, since there are at most $(2n-1)^n$ level rankings, the number of states in $\mathcal{A}'$ is at most $3^n \cdot (2n-1)^n$. □

Corollary 1 *Let $\mathcal{A}$ be an NBW with n states. There is an NBW $\tilde{\mathcal{A}}$ with at most $3^n \cdot (2n-1)^n$ states such that $\mathcal{L}(\tilde{\mathcal{A}}) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{A})$.*

3. An Improved Upper Bound

In this section we show how the $3^n \cdot (2n-1)^n$ bound described in Section 2 can be improved.

Consider a UCW $\mathcal{A}$ and a word $w \in \Sigma^\omega$ accepted by $\mathcal{A}$. For the run r of $\mathcal{A}$ on w, let $max\_rank(r)$ be the maximal rank that a vertex in $\mathcal{G}$ gets.

We now prove that there is a level l in $\mathcal{G}$ such that all the odd ranks below $max\_rank(r)$ appear in all the levels after l. Intuitively, it follows from the fact that odd ranks correspond to vertices that are α-free, and there is a level l that has an α-free vertex in all the intermediate DAGs G_i, for an add i below $max\_rank(r)$.

Lemma 2 *There is a level $l \geq 0$ such that for each level $l' > l$, and for all ranks $j \in [max\_rank(r)]^{odd}$, there is a vertex $\langle q, l' \rangle$ such that $rank(q, l') = j$.*

Proof. Let k be the minimal index for which $\mathcal{G}_{2k}$ is finite. For every $0 \leq i \leq k-1$, the DAG $\mathcal{G}_{2i+1}$ contains an α-free vertex. Let l_i be the minimal level such that $\mathcal{G}_{2i+1}$ has an α-free vertex $\langle q, l_i \rangle$. Since $\langle q, l_i \rangle$ is in $\mathcal{G}_{2i+1}$, it is not finite in $\mathcal{G}_{2i}$. Thus, there are infinitely many vertices in $\mathcal{G}_{2i}$ that are reachable from $\langle q, l_i \rangle$. Hence, by König's Lemma, $\mathcal{G}_{2i}$ contains an infinite path $\langle q, l_i \rangle, \langle q_1, l_i + 1 \rangle, \langle q_2, l_i + 2 \rangle, \ldots$. For all $j \geq 1$, the vertex $\langle q_j, l_i + j \rangle$ has infinitely many vertices reachable from it in $\mathcal{G}_{2i}$ and thus, it is not finite in $\mathcal{G}_{2i}$. Therefore, the path $\langle q, l_i \rangle, \langle q_1, l_i + 1 \rangle, \langle q_2, l_i + 2 \rangle, \ldots$ exists also in $\mathcal{G}_{2i+1}$. Recall that $\langle q, l_i \rangle$ is α-free. Hence, being reachable from $\langle q, l_i \rangle$, all the vertices $\langle q_j, l_i + j \rangle$ in the path are α-free as well. It follows that for every $0 \leq i \leq k-1$ there exists a level l_i such that for all $l' \geq l_i$, there is a vertex $\langle q, l' \rangle$ that is α-free in $\mathcal{G}_{2i+1}$, and for which $rank(q, l')$ would therefore be $2i+1$. Since the maximal odd member in $[max\_rank(r)]^{odd}$ is $2k-1$, taking $l = \max_{0 \leq i \leq k-1}\{l_i\}$ satisfies the lemma's requirements. $\square$

Recall that a level ranking for $\mathcal{A}$ is a function $g : Q \to [2n-2]$, such that if $g(q)$ is odd, then $q \notin \alpha$. Let $max\_odd(g)$ be the maximal odd number in the range of g.

Definition 1 *We say that a level ranking g is* tight *if*

1. *the maximal rank in the range of g is odd, and*
2. *for all $j \in [max\_odd(g)]^{odd}$, there is a state $q \in Q$ with $g(q) = j$.*

Lemma 3 *There is a level $l \geq 0$ such that for each level $l' > l$, the level ranking that corresponds to l' is tight.*

Proof. Lemma 2 implies that for all the levels l' beyond some level l_1, the level ranking that corresponds to l' satisfies the second condition in Definition 1. Let g be the level ranking in level l_1. Since even ranks label finite vertices, only a finite number of levels $l' \geq l_1$ have even ranks greater than $max\_odd(g)$ in their range. The level l required in the lemma is then the level beyond l_1 in which these even ranks "evaporate". $\square$

We refer to the minimal level l that satisfies the conditions in Lemma 3 as the *limit level* of r, denoted $limit(r)$.

It follows that we can improve the construction described in the proof of Theorem 1 and restrict the set $\mathcal{R}$ of possible level rankings to the set of tight level rankings. Since, however, the tightness of the level ranking is guaranteed only beyond the limit level of r, we also need to guess this level, and proceed with the usual subset construction until we reach it. Formally, we suggest the following modified construction.

Theorem 2 *Let $\mathcal{A}$ be a* UCW *with n states. Let $tight(n)$ be the number of tight level rankings. There is an* NBW *$\mathcal{A}'$ with at most $3^n \cdot tight(n)$ states such that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof. Let $\mathcal{A} = \langle \Sigma, Q, Q_{in}, \delta, \alpha \rangle$, and let $\mathcal{R}_{tight}$ be the set of tight level rankings for $\mathcal{A}$. Then, $\mathcal{A}' = \langle \Sigma, Q', Q'_{in}, \delta', \alpha' \rangle$, where

- $Q' \subseteq 2^Q \cup (2^Q \times 2^Q \times \mathcal{R}_{tight})$, where a state $S \in Q'$ indicates that the current level of the DAG contains the states in S (and is relevant for levels before the limit of r), and a state $\langle S, O, g \rangle \in Q'$ is similar to the states in the construction in the proof of Theorem 1 (and is relevant for levels beyond the limit of r). In particular, $O \subseteq S$.

- $Q'_{in} = \{Q_{in}\}$. Thus, the run starts in a "subset mode", corresponding to a guess that the limit level has not been reached yet.

- For all states in Q' of the form $S \in 2^Q$ and $\sigma \in \Sigma$, we have that

  $$\delta'(S, \sigma) = \{\delta(S, \sigma)\} \cup \{\langle \delta(S, \sigma), O, g \rangle : O \subseteq \delta(S, \sigma) \text{ and } g \in \mathcal{R}_{tight}\}.$$

  Thus, at each point in the subset mode, $\mathcal{A}'$ may guess that the current level is the limit level, and move to a "subset+ranks" mode, where it proceeds as the NBW constructed in the proof of Theorem 1. Thus, for states of the form $\langle S, O, g \rangle$, the transition function is as described there, only that rank levels are restricted to tight ones.

- $\alpha' = 2^Q \times \{\emptyset\} \times \mathcal{R}_{tight}$. Thus, as in the proof of Theorem 1, $\mathcal{A}'$ is required to visit infinitely many states in which the O component is empty. In particular, this force $\mathcal{A}'$ to eventually switch from the subset mode to the subset+ranks mode.

We prove that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$. In order to prove that $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}')$, we prove that $\mathcal{L}(\mathcal{A}'') \subseteq \mathcal{L}(\mathcal{A}')$, for the NBW $\mathcal{A}''$ constructed in Theorem 1, and for which we know that $\mathcal{L}(\mathcal{A}'') = \mathcal{L}(\mathcal{A})$. Consider a word $w \in \Sigma^\omega$ accepted by $\mathcal{A}''$. Let r'' be the accepting run of $\mathcal{A}''$ on w. By Lemma 3, the point *limit*(r) exists, and all the level rankings beyond *limit*(r) are tight. Therefore, the run r' obtained from r'' by projecting the states corresponding to levels up to *limit*(r) on their S component is a legal and accepting run of $\mathcal{A}'$ on w.

It is left to prove that $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$. Consider a word $w \in \Sigma^\omega$ accepted by $\mathcal{A}'$. Let $\mathcal{G}$ be the run DAG of $\mathcal{A}$ on w. We prove that there is an odd C-ranking $f : V \to [2n]$ for $\mathcal{G}$. Then, by Lemma 1, $\mathcal{A}$ accepts w. Let r' be the accepting run of $\mathcal{A}'$ on w. By the definition of δ, the projection of the states of r' on the first S component corresponds to the structure of $\mathcal{G}$. Since the initial state of $\mathcal{A}'$ is $\{Q_{in}\}$ whereas α' contains only states in $2^Q \times 2^Q \times \mathcal{R}_{tight}$, there must be a level l in which r' switches from a state mode to a state+ranks mode, and from which, according to the definition of δ', it stayed forever in that mode. The tight level rankings that r' describes for levels beyond l induce the C-ranking for vertices in these levels. For levels $l' < l$, we can define $f(\langle q, l' \rangle) = 2n$ for all $q \in Q$. Note that the ranks of all vertices in levels up to l is even, and f does not increase up to this point. In addition, since the maximal element in the range of a level ranking $g \in \mathcal{R}_{tight}$ is at most $2n - 1$, the ranking f does not increase also in level l. Finally, since each path eventually reaches the point l, from which the level rankings induce an odd C-ranking, f is odd. $\square$

Corollary 2 *Let $\mathcal{A}$ be an NBW with n states. There is an NBW $\tilde{\mathcal{A}}$ with at most $3^n \cdot tight(n)$ states such that $\mathcal{L}(\tilde{\mathcal{A}}) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{A})$.*

While the improved construction involves an extra copy of 2^Q in the state space, the restriction to tight rank assignments is significant. Indeed, as we show now, $tight(n)$ is bounded by $(cn)^n$, for $c \leq 0.76$, which gives a $(3cn)^n$ bound, for $c \leq 0.76$, on the number of states of the complementary automaton.

The general idea is roughly as follows. Recall that we wish to bound from above the number of tight level rankings – functions $f : Q \to [2n-2]$ that are onto $[2l-1]^{odd}$, with $2l-1$ being the maximal number in the range of f. As a first step we need a bound on the number of functions from the set $\{1, \ldots, n\}$ onto the set $\{1, \ldots, m\}$. This is nothing else but $m!$ times the *Stirling number of the second kind*, $S(n,m)$. The asymptotics of these numbers is known, e.g. in [22], where the following is implicit[c].

Lemma 4 (Temme) *For $0 \leq \beta \leq 1$, let x be the unique positive real number satisfying $\beta x = 1 - e^{-x}$, and let $a = -\ln x + \beta \ln(e^x - 1) - (1-\beta) + (1-\beta)\ln(\frac{1-\beta}{\beta})$. The number of functions from $\{1, \ldots, n\}$ onto $\{1, \ldots, \beta n\}$ is at most*

$$[(1 + o(1))(M[\beta]n)]^n,$$

where

$$M[\beta] = (\frac{\beta}{1-\beta})^{1-\beta} e^{a-\beta}.$$

Now, let

$$p(\ell, n) = \max_r \left\{ M\left[\frac{r+\ell}{n}\right] \left(\frac{\ell}{r}\right)^{\frac{r}{n}} \left(\frac{\ell}{\ell - r}\right)^{\frac{\ell-r}{n}} \right\}, \qquad (*)$$

where the maximum is over all $r \leq \ell, n - \ell$. The value of $p(\ell, n)$ depends only on the ratio $\frac{\ell}{n}$. To see this, note that if we allow r to assume a real value rather than only integer values, we still get an upper bound. Thus, we can assume that $r = \alpha n$ and $\ell = \gamma n$ for some α and γ. Then, all the terms in the bound are functions of α and γ, where we are maximizing on α. Therefore, the bound we get is only a function of $\gamma = \frac{\ell}{n}$. Let $h(\frac{\ell}{n}) = p(\ell, n)$. Then:

Theorem 3 *The number of functions from $\{1, \ldots, n\}$ to $\{0, \ldots, 2\ell - 1\}$ that are onto the ℓ odds is no more than $n\left[(1 + o(1))h(\frac{\ell}{n})n\right]^n$.*

Proof. Fixing r, one chooses which r evens are going to be hit ($\binom{\ell}{r}$ possibilities) and then chooses a function from $\{1, \ldots, n\}$ onto the set of the ℓ odds union with the set of the r chosen evens. Clearly, the number of such functions is equal to the number of functions from $\{1, \ldots, n\}$ onto $\{1, \ldots, \ell + r\}$. By Lemma 4 and Stirling's approximation we get the expression that appears in (*). Choosing the "worst" r gives us the upper bound. $\square$

[c]The version of this lemma that appears in the preliminary version of this paper ignores the e^α factor in the definition of $M[\beta]$, and hence results in a weaker bound. This was pointed out to us by Qiqi Yan.

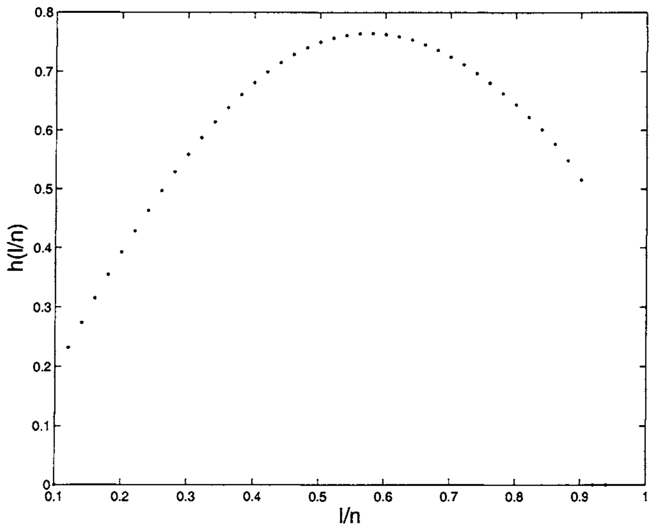

Fig. 1. The function $h(\frac{\ell}{n})$, for $0 < \frac{\ell}{n} < 1$.

Recall that a tight level ranking is a function $g : Q \to \{0, \ldots, 2n-2\}$ such that the maximal rank in the range of g is some odd $2\ell - 1$, and g is onto the ℓ odds $\{1, 3, \ldots, 2\ell - 1\}$. Thus, the expression in Theorem 3 counts the number of tight level rankings as a function of $\frac{\ell}{n}$.

In Figure 1 we describe the behavior of $h(\frac{\ell}{n})$ for $0 < \frac{\ell}{n} < 1$, as plotted by Matlab. A simple analysis shows that the maximal value that $h(\frac{\ell}{n})$ can get is at most 0.76, for $\frac{\ell}{n} = 0.574$, implying an upper bound of $(0.76n)^n$ to the number of tight level rankings[d]

4. A Tighter Analysis

In Section 3, we described an improved complementation construction and showed that the state space of the complementary automaton is bounded by $(3cn)^n$, for $c \leq 0.76$. Our analysis was based on the observation that the state space of the complementary automaton consists of triples $\langle S, O, g \rangle$ in which S and O are subsets of the state space of the original automaton, with $O \subseteq S$, and g is a tight level ranking. Accordingly, we multiplied 3^n – the number of pairs $\langle S, O \rangle$ as above with *tight*(n) – the number of tight level rankings. This analysis, while significantly improving the known $3^n(2n-1)^n$ upper bound, ignores possible relations between the pair $\langle S, O \rangle$ and the tight level ranking g associated with it. In this section we point to such relations and show that, indeed, they lead to a tighter analysis.

Consider a state $\langle S, O, g \rangle$ of the NBW $\mathcal{A}'$ constructed in Theorem 2. Note that while $g : Q \to [2n-2]$ has Q as its domain, we can, given S, restrict attention to

[d]In general, sampling may not be sufficient for bounding the maximal value of a function. When, however, it is possible to bound the derivative of the function, a bound on its maximal value does follow from sampling at close enough intervals, which is the case with the function h and the analysis in Figure 1.

level rankings in which all states not in S are mapped to 0. To see this, note that the requirement about g being a tight level ranking stays valid for g with $g(s) = 0$ for all $g \notin S$. Also, the definition of when a level ranking covers another level ranking is parametrized with S. In addition, as O maintains the set of states that have not been visited an odd vertex, g maps all the states in O to an even rank.

Let *tighter*(n) be the number of triples $\langle S, O, f\rangle$, with $O \subseteq S \subseteq \{1, \ldots, n\}$, such that there exists ℓ so that $f : S \to \{1, \ldots, 2\ell + 1\}$ is onto the odds and $f(x)$ is even for $x \in O$. By the above discussion, we have the following.

Corollary 3 *Let $\mathcal{A}$ be an NBW with n states. There is an NBW $\tilde{\mathcal{A}}$ with at most tighter(n) states such that $\mathcal{L}(\tilde{\mathcal{A}}) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{A})$.*

We now calculate *tighter*(n) and conclude that the more careful analysis is significant – the bound on the state space that follows from Corollary 3 is better than the one that follows from Corollary 2.

For a triple $\langle S, O, f\rangle$ as above, let $T \subseteq S$ be the inverse image of the odd integers under f, and let $O' = S \setminus T$. Let α, β, and γ be such that $|S| = \alpha n$, $|T| = \beta n$, and $\ell = \gamma n$. Also, let $\Psi_n(\alpha, \beta, \gamma)$ denote the number of triples $\langle S, O, f\rangle$ for a fixed triple $\langle \alpha, \beta, \gamma\rangle$. We are interested in $\sum_{\alpha,\beta,\gamma} \Psi_n(\alpha, \beta, \gamma)$, where $0 \leq \gamma \leq \beta \leq \alpha \leq 1$, and all three numbers are integer multiples of $1/n$. Clearly, this is at most $n^3 \max_{\alpha,\beta,\gamma} \Psi_n(\alpha, \beta, \gamma)$. Let us therefore compute Ψ_n for a fixed $\langle \alpha, \beta, \gamma\rangle$.

In order to count, we start by choosing S, then choose T, next we choose the value of ℓ, then define f, and finally choose $O \subseteq O'$.

The number of ways to choose S is $\binom{n}{\alpha n}$ which, using Stirling's factorial approximation formula, is

$$[(1 + o(1))\alpha^{-\alpha}(1 - \alpha)^{\alpha-1}]^n.$$

Note that in the above calculation we should use the convention $0^0 = 1$. The number of ways to choose T inside S is $\binom{\alpha n}{\beta n}$, which is approximately

$$[(1 + o(1))(\beta/\alpha)^{-\beta/\alpha}(1 - \frac{\beta}{\alpha})^{\frac{\beta}{\alpha}-1}]^{\alpha n}.$$

The number of ways to choose the values of f on T, according to Lemma 4, is

$$(M[\gamma/\beta]\beta n)^{\beta n}.$$

The number of ways to choose the values of f for the elements of O' is $\ell^{|O'|}$, which is

$$(\gamma n)^{(\alpha-\beta)n}.$$

The number of choices for O is

$$2^{|O'|} = 2^{(\alpha-\beta)n}.$$

Using the notation $\psi_n(\alpha, \beta, \gamma) = \sqrt[n]{\Psi_n(\alpha, \beta, \gamma)}$ and multiplying all of the above, we get $\psi_n(\alpha, \beta, \gamma) =$

$$((1 + o(1))\alpha^{-\alpha}(1 - \alpha)^{\alpha-1})((\beta/\alpha)^{-\beta/\alpha}(1 - \frac{\beta}{\alpha})^{\frac{\beta}{\alpha}-1})^{\alpha}(M[\gamma/\beta]\beta)^{\beta}(2\gamma)^{\alpha-\beta}n^{\alpha}.$$

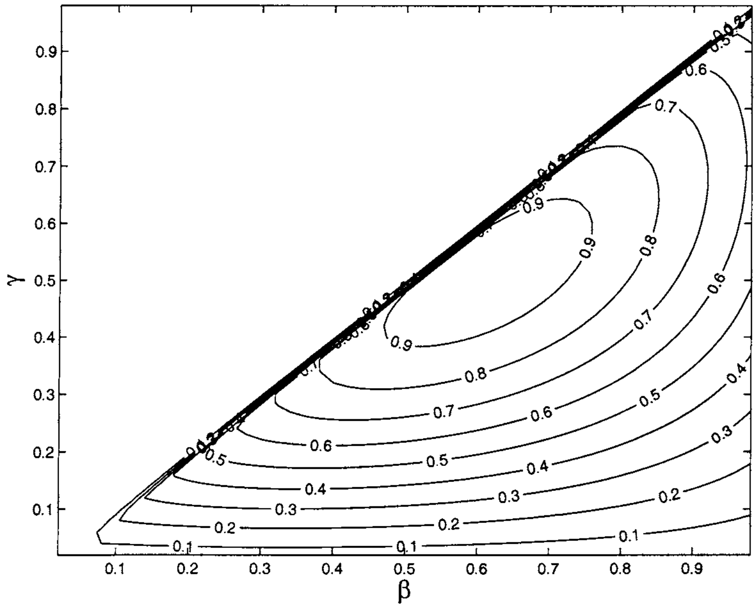

Fig. 2. Contour lines of $h'(\beta, \gamma)$, for $0 \le \gamma \le \beta \le 1$.

For fixed values of β and γ, the asymptotic maximum of the above is achieved for $\alpha = 1$. It see this, recall that $\alpha \le 1$ and note that all the terms in $\psi_n(\alpha, \beta, \gamma)$, except for n^α are bounded (as n goes to infinity). Therefore, for any $\alpha < 1$, we have that $\psi_n(\alpha, \beta, \gamma)$ behaves like $O(n^\alpha)$, which is smaller than n^1, which is the order of magnitude when $\alpha = 1$. Setting $\alpha = 1$, we get

$$\max_\alpha \frac{\psi_n(\alpha, \beta, \gamma)}{n} = h'(\beta, \gamma) = (1 + o(1))\beta^{-\beta}(1-\beta)^{\beta-1}(M[\gamma/\beta]\beta)^\beta(2\gamma)^{1-\beta}.$$

Since $\sqrt[n]{n^3} \to 1$ this is also the asymptotic value of $\frac{1}{n}\sqrt[n]{\sum_{\alpha,\beta,\gamma} \Psi_n(\alpha, \beta, \gamma)}$.

In Figure 2 we describe the behavior of $h'(\beta, \gamma)$ for $0 \le \gamma \le \beta \le 1$, as plotted by Matlab. A simple analysis shows that the maximal value that $h'(\beta, \gamma)$ can get is at most 0.9624, for $\beta \approx 0.6115$ and $\gamma \approx 0.5082$, implying an upper bound of $(0.9624n)^n$ for *tighter*(n).

5. Language Containment

Recall that a primary application of complementation constructions is language containment: in order to check that the language of an automaton $\mathcal{A}_1$ is contained in the language of a second automaton $\mathcal{A}_2$, one checks that the intersection of $\mathcal{A}_1$ with an automaton that complements $\mathcal{A}_2$ is empty. In this section we demonstrate the simplicity and advantage of our construction with respect to this application and show how an automaton that complements $\mathcal{A}_2$, when constructed using our construction, can be optimized in the process of its intersection with $\mathcal{A}_1$.

Consider a language-containment problem $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$. The solution that follows from our approach is to start by dualizing $\mathcal{A}_2$, translate the result (a universal automaton $\tilde{\mathcal{A}}_2$) to a nondeterministic automaton $\tilde{\mathcal{N}_2}$, which complements $\mathcal{A}_2$, and check the emptiness of the product $\mathcal{A}_1 \times \tilde{\mathcal{N}_2}$. Consider the universal automaton $\tilde{\mathcal{A}}_2$. Our translation of $\tilde{\mathcal{A}}_2$ to $\tilde{\mathcal{N}_2}$ is based on ranks we associate with vertices that appear in run DAGs of $\tilde{\mathcal{A}}_2$. For $\mathcal{A}_2$ with n states, the range of the ranks is $0, \ldots, 2n-2$, where the bound $2n-2$ on the maximal rank follows from the fact that the width of the run DAG $\mathcal{G}$ is bounded by n.

In fact, we can tighten the width of $\mathcal{G}$ further. Indeed, the structure of $\mathcal{A}_2$ may guarantee that some states may not appear together in the same level. For example, if q_0 and q_1 are reachable only after reading even-length and odd-length prefixes of w, respectively, then q_0 and q_1 cannot appear together in the same level in the run DAG of $\mathcal{A}_2$ on w, which enables us to bound its width by $n-1$. In general, since the construction of $\tilde{\mathcal{N}_2}$ takes into an account all words $w \in \Sigma^\omega$, we need to check the "mutual exclusiveness" of q_0 and q_1 with respect to all words. This can be done using the subset construction [17]: let $\mathcal{A}_2 = \langle \Sigma, Q_2, Q^2_{in}, \delta_2, \alpha_2 \rangle$, and let $\mathcal{A}^d_2 = \langle \Sigma, 2^{Q_2}, \{Q^2_{in}\}, \delta^d_2 \rangle$ be the automaton without acceptance condition obtained by applying the subset construction to $\mathcal{A}_2$. Thus, for all $S \in 2^{Q_2}$, we have that $\delta^d_2(S, \sigma) = \bigcup_{s \in S} \delta_2(s, \sigma)$. Now, let $reach(\mathcal{A}_2) \subseteq 2^{Q_2}$ be the set of states reachable in $\mathcal{A}^d_2$ from $\{Q^2_{in}\}$. Thus, $S \subseteq Q_2$ is in $reach(\mathcal{A}_2)$ iff there is a finite word $w \in \Sigma^*$ such that $\delta^d_2(\{Q^2_{in}\}, w) = S$. Then, $reach(\mathcal{A}_2)$ contains exactly all sets S of states such that all the states in S may appear in the same level of some run DAG of $\mathcal{A}_2$. Accordingly, we can tighten our bound on the maximal width a run DAG may have to $r^{max} = \max_{S \in reach(\mathcal{A}_2)} |S|$, and tighten our bound on the maximal rank to $2r^{max} - 2$. If $Q_2 \in reach(\mathcal{A}_2)$, then $r^{max} = n$, and we do not optimize. Often, however, the structure of $\mathcal{A}_2$ does prevent some states from appearing together on the same level. As we shall explain now, the presence of $\mathcal{A}_1$ can make the above optimization even more effective.

It is easy to see that some states may be mutual exclusive (i.e., cannot appear in the same level in the run DAG) with respect to some words and not be mutual exclusive with respect to other words. The definition of r^{max} requires mutual exclusiveness with respect to all words. On the other hand, checking $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$, we only have to consider mutual exclusiveness with respect to words in $\mathcal{L}(\mathcal{A}_1)$. Note that the fewer words we have to consider, the more likely we are to get mutual exclusiveness, and then tighten the bound further. Checking mutual exclusiveness with respect to $\mathcal{L}(\mathcal{A}_1)$ can be done by taking the product of $\mathcal{A}_1$ with $\mathcal{A}^d_2$. Formally, let $\mathcal{A}_1 = \langle \Sigma, Q_1, Q^1_{in}, \delta_1, \alpha_1 \rangle$, and let $reach(\mathcal{A}_{2|\mathcal{A}_1}) \subseteq 2^{Q_2}$ be the set of states that are reachable in the product of $\mathcal{A}_1$ with $\mathcal{A}^d_2$, projected on the state space of $\mathcal{A}^d_2$. Thus, $S \subseteq Q_2$ is in $reach(\mathcal{A}_{2|\mathcal{A}_1})$ iff there is a finite word $w \in \Sigma^*$ and a state $s' \in Q_1$ such that $s' \in \delta_1(Q^1_{in}, w)$ and $\delta^d_2(\{Q^2_{in}\}, w) = S$. Note that $reach(\mathcal{A}_{2|\mathcal{A}_1})$ excludes from $reach(\mathcal{A}_2)$ sets that are reachable in $\mathcal{A}_2$ only via words that are not reachable in $\mathcal{A}_1$. Accordingly, we can tighten our bound on the maximal width a run DAG of $\mathcal{A}_2$ on a word in $\mathcal{L}(\mathcal{A}_1)$ may have to $r^{max}_{\mathcal{A}_1} = \max_{S \in reach(\mathcal{A}_{2|\mathcal{A}_1})} |S|$, and tighten our bound on the maximal rank in the construction of $\tilde{\mathcal{N}_2}$, which is needed for checking the containment of $\mathcal{L}(\mathcal{A}_1)$ in $\mathcal{L}(\mathcal{A}_2)$, to $2r^{max}_{\mathcal{A}_1} - 2$.

Note that since we actually need to consider only accepting run DAGs, we can optimize further by removal of empty states from the participating automata. For example, if a state $s \in Q_2$ is such that $\mathcal{L}(\mathcal{A}_2^s) = \emptyset$, we can remove s from the range of δ_2 without changing the language accepted by $\mathcal{A}_2$. In particular, it follows that $\mathcal{A}_2$ has no rejecting sinks, and the range of δ_2 may contain the empty set. This removes from $reach(\mathcal{A}_2)$ sets S that may appear in the same level in a rejecting run DAG of $\mathcal{A}_2$ but cannot appear in the same level in an accepting run DAG. Consequently, r^{max} may become smaller. Similarly, by removing (in addition) empty states from $\mathcal{A}_1$, we restrict $reach(\mathcal{A}_{2|\mathcal{A}_1})$ to sets S of states such that all the states in S may appear in the same level of some (accepting) run DAG of $\mathcal{A}_2$ on a word in $\mathcal{L}(\mathcal{A}_1)$. Finally, we can also remove from $reach(\mathcal{A}_{2|\mathcal{A}_1})$ sets S induced only by pairs $\langle s, S\rangle \in Q_1 \times 2^{Q_2}$ for which the product of $\mathcal{A}_1$ and $\mathcal{A}_2^d$ with initial state $\langle s, S\rangle$ is empty. Indeed, such sets cannot appear in the same level of an accepting run DAG of $\mathcal{A}_2$ on a word in $\mathcal{L}(\mathcal{A}_1)$.

Finally, recall that the bound on the maximal rank that a vertex can get actually depends on the α-less width of $\mathcal{G}$, which we have approximated from above thoughout the paper by $n-1$. The considerations that enables us to take the α-less width (see [7] for details) are orthogonal to the considerations that enable us to ignore mutually exclusive states, thus we can tighten our bound on the maximal rank in the construction of $\tilde{\mathcal{N}_2}$ to $2r^{max}_{\mathcal{A}_1} - |\alpha_2| - 1$.

Acknowledgment

We thank Raz Kupferman for Matlab-analysis services. We thank Qiqi Yan for pointing out an error in Lemma 4 in the preliminary version.

Orna Kupferman is supported in part by BSF grant 9800096, and by a grant from Minerva. Moshe Y. Vardi is supported in part by NSF grants CCR-9988322, CCR-0124077, CCR-0311326, IIS-9908435, IIS-9978135, EIA-0086264, and ANI-0216467, by BSF grant 9800096, by Texas ATP grant 003604-0058-2003, and by a grant from the Intel Corporation.

References

1. R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singerman, A. Tiemeyer, M.Y. Vardi, and Y. Zbar. The For-Spec temporal logic: A new temporal property-specification logic. In *Proc. 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 2280 of *Lecture Notes in Computer Science*, pages 296–211, Grenoble, France, April 2002. Springer-Verlag.
2. J.C. Birget. Partial orders on words, minimal elements of regular languages, and state complexity. *Theoretical Computer Science*, 119:267–291, 1993.
3. J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. International Congress on Logic, Method, and Philosophy of Science. 1960*, pages 1–12, Stanford, 1962. Stanford University Press.
4. N. Daniele, F. Guinchiglia, and M.Y. Vardi. Improved automata generation for linear temporal logic. In *Computer Aided Verification, Proc. 11th International Conference*, volume 1633 of *Lecture Notes in Computer Science*, pages 249–260. Springer-Verlag, 1999.

5. P. Gastin and D. Oddoux. Fast LTL to büchi automata translation. In *Computer Aided Verification, Proc. 13th International Conference*, volume 2102 of *Lecture Notes in Computer Science*, pages 53–65. Springer-Verlag, 2001.
6. S. Gurumurthy, R. Bloem, and F. Somenzi. Fair simulation minimization. In *Computer Aided Verification, Proc. 14th International Conference*, volume 2404 of *Lecture Notes in Computer Science*, pages 610–623. Springer-Verlag, 2002.
7. S. Gurumurthy, O. Kupferman, F. Somenzi, and M.Y. Vardi. On complementing nondeterministic Büchi automata. In *12th Advanced Research Working Conference on Correct Hardware Design and Verification Methods*, volume 2860 of *Lecture Notes in Computer Science*, pages 96–110. Springer-Verlag, 2003.
8. T.A. Henzinger, O. Kupferman, and S. Rajamani. Fair simulation. *Information and Computation*, 173(1):64–81, 2002.
9. G.J. Holzmann. The model checker SPIN. *IEEE Trans. on Software Engineering*, 23(5):279–295, May 1997. Special issue on Formal Methods in Software Practice.
10. Y. Kesten, N. Piterman, and A. Pnueli. Bridging the gap between fair simulation and trace containment. In *Computer Aided Verification, Proc. 15th International Conference*, volume 2725 of *Lecture Notes in Computer Science*, pages 381–393. Springer-Verlag, 2003.
11. N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, pages 358–367, San Juan, October 1991.
12. O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. on Computational Logic*, 2(2):408–429, July 2001.
13. R.P. Kurshan. *Computer Aided Verification of Coordinating Processes.* Princeton Univ. Press, 1994.
14. C. Löding and W. Thomas. Alternating automata and logics over infinite words. In *Theoretical Compter Science - Exploring New Frontiers of Theoretical Informatics*, volume 1872 of *Lecture Notes in Computer Science*, pages 521–535. Springer-Verlag, 2000.
15. M. Michel. Complementation is more difficult with automata on infinite words. CNET, Paris, 1988.
16. D.E. Muller and P.E. Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987.
17. M.O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:115–125, 1959.
18. S. Safra. On the complexity of ω-automata. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 319–327, White Plains, October 1988.
19. W. Sakoda and M. Sipser. Non-determinism and the size of two-way automata. In *Proc. 10th ACM Symp. on Theory of Computing*, pages 275–286, 1978.
20. A.P. Sistla, M.Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49:217–237, 1987.
21. F. Somenzi and R. Bloem. Efficient Büchi automata from LTL formulae. In *Computer Aided Verification, Proc. 12th International Conference*, volume 1855 of *Lecture Notes in Computer Science*, pages 248–263. Springer-Verlag, 2000.
22. N.M. Temme. Asimptotic estimates of Stirling numbers. *Stud. Appl. Math.*, 89:233–243, 1993.

23. M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, November 1994.
24. P. Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1–2):72–99, 1983.