Complementation Constructions for Nondeterministic Automata on Infinite Words

Orna Kupferman[1,⋆] and Moshe Y. Vardi[2,⋆⋆]

[1] Hebrew University, School of Engineering and Computer Science,
Jerusalem 91904, Israel
orna@cs.huji.ac.il,
http://www.cs.huji.ac.il/~orna

[2] Rice University, Department of Computer Science, Houston,
TX 77251-1892, U.S.A
vardi@cs.rice.edu,
http://www.cs.rice.edu/~vardi

Abstract. The complementation problem for nondeterministic automata on infinite words has numerous applications in formal verification. In particular, the language-containment problem, to which many verification problems are reduced, involves complementation. Traditional optimal complementation constructions are quite complicated and have not been implemented. Recently, we have developed an analysis techniques for runs of co-Büchi and generalized co-Büchi automata and used the analysis to describe simpler optimal complementation constructions for Büchi and generalized Büchi automata. In this work, we extend the analysis technique to Rabin and Streett automata, and use the analysis to describe novel and simple complementation constructions for them.

1 Introduction

The complementation problem for nondeterministic automata on infinite words 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 specification [Kur94, VW94]. While it is easy to complement specifications given in terms of formulas in temporal logic, complementation of specifications given in terms of automata is so problematic, that in practice the user is required to describe the specification in terms of a deterministic automaton (it is easy to complement a deterministic automaton) [Kur87, HHK96], or to supply the automaton for the negation of the specification [Hol97]. Language containment is also useful in the context of abstraction,

[⋆] Supported in part by BSF grant 9800096, and by a grant from Minerva.

[⋆⋆] 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.

N. Halbwachs and L. Zuck (Eds.): TACAS 2005, LNCS 3440, pp. 206–221, 2005.

where a large system is replaced by an abstraction whose language is richer, yet its state space is smaller. Such abstractions are particularly useful in the context of parametric verification, where a parallel composition of an unbounded number of processes is abstracted by a composition of a finite number of them [KP00, KPP03], and in the context of inheritance and behavioral conformity in object-oriented analysis and design [HK02]. Other applications have to do with the fact that language equivalence is checked by two language-containment tests. For example, the translators from LTL into automata have reached a remarkable level of sophistication (cf. [GBS02]), and it is useful to check their correctness, which involves a language-equivalence test.

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 [Büc62]. Thus, complementing an automaton with n states resulted in an automaton with $2^{2^{O(n)}}$ states. In 1988, Safra introduced an optimal determinization construction, which also enabled a $2^{O(n \log n)}$ complementation construction [Saf88], matching the known lower bound [Mic88]. Another $2^{O(n \log n)}$ construction was suggested by Klarlund in [Kla91], which circumvented the need for determinization. The optimal constructions in [Saf88, Kla91] found theoretical applications in the establishment of decision procedures (cf. [EJ91]), but the intricacy of the constructions makes their implementation difficult. We know of no implementation of Klarlund's algorithm, and the implementation of Safra's algorithm [THB95] has to cope with the rather involved structure of the states in the complementary automaton. In [KV01] we described a simple, optimal complementation of nondeterministic Büchi automata, based on the analysis of runs of universal co-Büchi automata. A report on an implementation of this construction can be found in [GKSV03]. The construction was extended to nondeterministic generalized Büchi automata in [KV04]. Beyond its simplicity, the construction has other attractive properties: it can be implemented symbolically [Mer00, KV01], it is amenable to optimizations [GKSV03] and improvements [FKV04], and it naturally generates certificates to the verification task [KV04].

Many of the applications described above for the language-containment problem involve Rabin and Streett automata; cf. [LPS81, KPSZ02]. In particular, applications that involve the composition of processes and objects are typically applied to systems augmented with a strong-fairness condition, which corresponds to the Streett acceptance condition. Since nondeterministic Büchi automata recognize all the ω-regular languages, the complementation procedure in [KV01] can be used in order to complement richer classes of automata on infinite words, like nondeterministic Rabin and Streett automata: given a Rabin or Streett automaton $\mathcal{A}$, one can first translate $\mathcal{A}$ to a nondeterministic Büchi automaton $\mathcal{A}'$, and then complement $\mathcal{A}'$. While such an approach is reasonable for Rabin automata, it is not reasonable for Streett automata. Indeed, given a Rabin automaton $\mathcal{A}$ with n states and index k, the automaton $\mathcal{A}'$ has $O(nk)$ states, resulting in a complementary Büchi automaton with $2^{O(nk \log nk)}$ states. When $\mathcal{A}$ is a Streett automaton, however, $\mathcal{A}'$ has $O(n2^k)$ states [SV89], resulting in a complementary Büchi automaton with $2^{O(nk2^k \log n)}$ states. The fact that going through Büchi automata leads to a doubly-exponential construction makes the complementation problem for nondeterministic Streett automata much harder than the corresponding problem for Rabin automata. The first exponential complementation construction for Streett automata was given in [SV89]. their bound for the size of complementary automaton is 2^{m^5}, where m is the size of the input automaton. Only in [Kla91, Saf92], Klarlund and Safra came up with an improved construction, where the complementary automaton has $2^{O(nk \log nk)}$ states (optimality of this bound is still open). As has been the case with the early optimal constructions for Büchi automata, the constructions in [Kla91, Saf92] are quite complicated, and quite difficult to understand and teach.

In this work, we generalize the approach of [KV01, KV04] to nondeterministic Rabin and Streett automata, and describe novel and simple complementation construction for them. Given a nondeterministic Rabin automaton $\mathcal{A}$ with n states and index k, the complementary automaton $\tilde{\mathcal{A}}$ we construct is a nondeterministic Büchi automaton with $2^{O(nk \log n)}$ states. When $\mathcal{A}$ is a Streett automaton, $\tilde{\mathcal{A}}$ has $2^{O(nk \log nk)}$ states. Our construction is based on an analysis of the runs of the universal dual of $\mathcal{A}$, by means of ranks associated with states. In this sense, it is closely related to the *progress-measures* introduced in [Kla90]. Note that while the constructions (Theorems 2 and 4) are simple, the analysis (Lemmas 3 and 4) is quite nontrivial. As in the case of Büchi, the state space of the complementary automaton consists of subsets of the state space of original automaton and ranking functions for them, thus our constructions can be implemented symbolically[1], and we expect them to be optimizable. Note that in the case of Streett automata, our blow-up matches the one of Klarlund and Safra, whereas in the case of Rabin automata, we improve the known $2^{O(nk \log nk)}$ bound exponentially. At any rate, the main advantage of our approach is in the simplicity of the construction; the complexity analysis shows that, furthermore, there is no "penalty" for this simplicity.

2 Preliminaries

Automata on Infinite Words. Given an alphabet Σ, an *infinite word over Σ* is an infinite sequence $w = \sigma_0 \cdot \sigma_1 \cdot \sigma_2 \cdots$ of letters in Σ. We denote by w^l the suffix $\sigma_l \cdot \sigma_{l+1} \cdot \sigma_{l+2} \cdots$ of w. 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 not *deterministic*. If 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 $r(0) \in Q_{in}$ and for every $l \geq 0$, we have $r(l+1) \in \rho(r(l), \sigma_l)$. 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 such thing "last state", and acceptance is defined according to the set $\mathit{Inf}(r)$ of states that r visits *infinitely often*, i.e., $\mathit{Inf}(r) = \{q \in Q \ : \text{for infinitely many } l \in \mathbb{N}, \text{we have } r(l) = q\}$. As Q is finite, it is guaranteed that $\mathit{Inf}(r) \neq \emptyset$. The way we refer to $\mathit{Inf}(r)$ depends on

[1] In contrast, the state space of the complementary automata in [Kla91, Saf92] consist of labeled ordered trees, making a symbolic implementation difficult.

the acceptance condition of $\mathcal{A}$. Several acceptance conditions are studied in the literature. We consider here five:

- *Büchi automata*, where $\alpha \subseteq Q$, and r is accepting iff $\mathit{Inf}(r) \cap \alpha \neq \emptyset$.
- *co-Büchi automata*, where $\alpha \subseteq Q$, and r is accepting iff $\mathit{Inf}(r) \cap \alpha = \emptyset$.
- *Generalized Büchi automata*, where $\alpha = \{G_1, G_2, \ldots, G_k\}$ and r is accepting iff $\mathit{Inf}(r) \cap G_i \neq \emptyset$ for all $1 \leq i \leq k$.
- *Generalized co-Büchi automata*, where $\alpha = \{B_1, B_2, \ldots, B_k\}$ and r is accepting iff $\mathit{Inf}(r) \cap B_i = \emptyset$ for some $1 \leq i \leq k$.
- *Rabin automata*, where $\alpha = \{\langle G_1, B_1\rangle, \langle G_2, B_2\rangle, \ldots, \langle G_k, B_k\rangle\}$, and r is accepting iff for some $1 \leq i \leq k$, we have that $\mathit{Inf}(r) \cap G_i \neq \emptyset$ and $\mathit{Inf}(r) \cap B_i = \emptyset$.
- *Streett automata*, where $\alpha = \{\langle B_1, G_1\rangle, \langle B_2, G_2\rangle, \ldots, \langle B_k, G_k\rangle\}$, and r is accepting iff for all $1 \leq i \leq k$, if $\mathit{Inf}(r) \cap B_i \neq \emptyset$, then $\mathit{Inf}(r) \cap G_i \neq \emptyset$.

The number k of sets in the generalized Büchi and co-Büchi acceptance conditions or pairs in the Rabin and Streett acceptance conditions is called the *index* of α (or $\mathcal{A}$). Note that the Büchi and the co-Büchi conditions are dual, in the sense that a run r satisfies a Büchi condition α iff r does not satisfy α when regarded as a co-Büchi condition. Similarly, generalized Büchi and generalized co-Büchi are dual, and so are Rabin and Streett.

Since $\mathcal{A}$ is not 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 a *nondeterministic* automaton, 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 and is one of "B" (Büchi), "C" (co-Büchi), "GB" (generalized Büchi), "GC" (generalized co-Büchi), "S" (Streett), and "R" (Rabin). The third letter designates the objects accepted by the automata; 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. For the case of Streett and Rabin automata we sometimes also indicate the index of the automaton. For example, USW[1] is a universal Streett word automaton with one pair in its acceptance condition.

In [KV01], we suggested the following approach for complementing nondeterministic automata: in order to complement a nondeterministic automaton, first dualize the transition function and the acceptance condition, and then translate the resulting universal automaton back to a nondeterministic one. By [MS87], the dual automaton accepts the complementary language, and so does the nondeterministic automaton we end up with. In the special case of Büchi automata, one starts with an NBW, dualize it to a UCW, which accepts the complementary language, and then translates the UCW to an equivalent NBW. Thus, rather than determinization, complementation is based on a translation of universal automata to nondeterministic ones, which turned out to be much simpler. In this paper, we extend this approach to Rabin and Streett automata.

Run DAG*s* Consider a universal word automaton $\mathcal{A} = \langle \Sigma, Q, Q_{in}, \delta, \alpha \rangle$. Let $|Q| = n$. 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) $\mathcal{G}_r = \langle V, E \rangle$, where

- $V \subseteq Q \times \mathbb{N}$ is such that $\langle q, l \rangle \in V$ iff some run of $\mathcal{A}$ on w has $r(l) = q$. For example, the first level of $\mathcal{G}_r$ 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, $\mathcal{G}_r$ embodies exactly all the runs of $\mathcal{A}$ on w. We call $\mathcal{G}_r$ the *run* DAG of $\mathcal{A}$ on w, and we say that $\mathcal{G}_r$ is *accepting* if all its paths satisfy the acceptance condition α. Note that $\mathcal{A}$ accepts w iff $\mathcal{G}_r$ 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 $\mathcal{G}_r$ is an *S-vertex* if $q \in S$.

Consider a (possibly finite) DAG $\mathcal{G} \subseteq \mathcal{G}_r$. 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. Finally, we say that the *width* of $\mathcal{G}$ is d if d is the maximal number for which there are infinitely many levels l such that there are d vertices of the form $\langle q, l \rangle$ in $\mathcal{G}$. Note that the width of $\mathcal{G}_r$ is at most n.

Runs of UCW and UGCW were studied in [KV01, KV04]. 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. Consider a generalized co-Büchi condition $\alpha = \{B_1, \ldots, B_k\}$. Let $I = \{1, \ldots, k\}$, and let $\Omega_I = [2n]^{even} \cup ([2n]^{odd} \times I)$. We refer to the members of Ω_I in $[2n]^{even}$ as *even ranks* and refer to the members of Ω_I in $[2n]^{odd} \times \{j\}$ as *odd ranks with index* j. The members of Ω_I are ordered according to their element in $[2n]$. Thus, $r \leq r'$, $\langle r, i \rangle \leq r'$, and $r \leq \langle r', i' \rangle$ iff $r \leq r'$. In addition, $\langle r, i \rangle \leq \langle r', i' \rangle$ iff $r < r'$ or $\langle r, i \rangle = \langle r', i' \rangle$.

Generalized Co-Büchi Ranking. Recall that a run r satisfies α if there is some $j \in I$ such that $inf(r) \cap B_j = \emptyset$. A *generalized co-Büchi ranking* (*GC-ranking*, for short) for $\mathcal{G}_r$ is a function $f : V \to \Omega_I$ that satisfies the following conditions:

1. For all vertices $\langle q, l \rangle \in V$, if $f(\langle q, l \rangle) = \langle r, j \rangle$, then $q \notin B_j$.
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 ranking associates with each vertex in $\mathcal{G}_r$ a rank in Ω_I so that ranks along paths are not increased, and B_j-vertices cannot get an odd rank with index j. Note that each path in $\mathcal{G}_r$ eventually gets trapped in some rank. We say that the ranking f is an *odd GC-ranking* if all the paths of $\mathcal{G}_r$ 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 $\mathcal{G}_r$, 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 $\mathcal{G}_r$ has infinitely many vertices with odd ranks.

Lemma 1. [KV04] *The following are equivalent.*

1. *All the paths of $\mathcal{G}_r$ satisfy the generalized co-Büchi condition $\{B_1, \ldots, B_k\}$.*
2. *There is an odd GC-ranking for $\mathcal{G}_r$.*

Proof: Assume first that there is an odd GC-ranking for $\mathcal{G}_r$. Then, every path in $\mathcal{G}_r$ eventually gets trapped in an odd rank with index j, for some $j \in I$. Hence, as B_j-vertices cannot get an odd rank with index j, all the paths of $\mathcal{G}_r$ has some $j \in I$ for which they visit B_j only finitely often, and we are done.

For the other direction, given an accepting run DAG $\mathcal{G}_r$, we define an infinite sequence $\mathcal{G}_0 \supseteq \mathcal{G}_1 \supseteq \mathcal{G}_2 \supseteq \ldots$ of DAGs inductively as follows. For $\mathcal{G} \subseteq \mathcal{G}_r$ and $j \in I$, we say that j is *helpful* for $\mathcal{G}$ if $\mathcal{G}$ contains a B_j-free vertex.

- $\mathcal{G}_0 = \mathcal{G}_r$.
- $\mathcal{G}_{2i+1} = \mathcal{G}_{2i} \setminus \{\langle q, l \rangle \mid \langle q, l \rangle$ is finite in $\mathcal{G}_{2i}\}$.
- Let $j \in I$ be the minimal[2] index helpful for $\mathcal{G}_{2i+1}$, if exists.
  Then, $\mathcal{G}_{2i+2} = \mathcal{G}_{2i+1} \setminus \{\langle q, l \rangle \mid \langle q, l \rangle$ is B_j-free in $\mathcal{G}_{2i+1}\}$.

It can be shown that for every $i \geq 0$, unless the DAG $\mathcal{G}_{2i+1}$ is empty, then there is some $j \in I$ that is helpful for $\mathcal{G}_{2i+1}$. Since the successors of a B_j-free vertex are also B_j-free, and since all the vertices in $\mathcal{G}_{2i+1}$ have at least one successor, the transition from $\mathcal{G}_{2i+1}$ to $\mathcal{G}_{2i+2}$ involves the removal of an infinite path from $\mathcal{G}_{2i+1}$. 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.

Each vertex $\langle q, l \rangle$ in $\mathcal{G}_r$ has a unique index $i \geq 1$ such that $\langle q, l \rangle$ is either finite in $\mathcal{G}_{2i}$ or B_j-free in $\mathcal{G}_{2i+1}$, for some $j \in I$. Thus, the sequence of DAGs induces a function $f : V \to \Omega_I$, where $f(\langle q, l \rangle)$ is $2i$, if $\langle q, l \rangle$ is finite in $\mathcal{G}_{2i}$, and is $\langle 2i+1, j \rangle$, if j is the minimal index helpful for $\mathcal{G}_{2i+1}$ and $\langle q, l \rangle$ is B_j-free in $\mathcal{G}_{2i+1}$. It can be shown that the function f is an odd GC-ranking[3]. □

A *co-Büchi-ranking* for $\mathcal{G}_r$ (*C-ranking*, for short) can be defined as a special case of GC-ranking. Since $I = \{1\}$, we omit the indices from the odd ranks, thus a C-ranking is a function $f : V \to [2n]$. It can be shown (a special case of Lemma 1, see [KV01] for details) that all the paths of $\mathcal{G}_r$ have only finitely many α-vertices iff there is an odd C-ranking for $\mathcal{G}_r$.

3 NRW Complementation

In this section we analyze runs of USW and use the analysis in order to translate USW to NBW. The translation is then used for NRW complementation. We start with USW[1], and then generalize to USW with an arbitrary index.

[2] The fact that j is minimal is not important, any choice will do.

[3] The proof in [KV04] refers to a slightly different definition of GC-ranking, but it is easy to modify it to the definition we use here.

Streett[1]-ranking. We first consider USW[1], where $\alpha = \{\langle B, G\rangle\}$ contains a single pair, and $\mathcal{G}_r$ is accepting iff all paths in $\mathcal{G}_r$ have finitely many B-vertices or infinitely many G-vertices.

A *Streett[1]-ranking* for $\mathcal{G}_r$ (*S[1]-ranking*, for short) is a function $f : V \rightarrow [2n]$ 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 $q \notin B$.
2. For all edges $\langle\langle q, l\rangle, \langle q', l+1\rangle\rangle \in E$, either $f(\langle q', l+1\rangle) \leq f(\langle q, l\rangle)$ or $q \in G$.

Thus, an S[1]-ranking associates with each vertex in $\mathcal{G}_r$ a rank in $[2n]$ so that the ranks along paths may increase only when a G-vertex is visited, and no B-vertex is odd. Note that each path in $\mathcal{G}_r$ either visit G-vertices infinitely often or eventually gets trapped in some rank. We say that the S[1]-ranking f is an *odd S[1]-ranking* if all the paths of $\mathcal{G}_r$ either visit G-vertices infinitely often or eventually gets 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 $\mathcal{G}_r$, either $q_l \in G$ for infinitely many $l \geq 0$ or 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 $\mathcal{G}_r$ has infinitely many G-vertices or infinitely many odd vertices.

Lemma 2. *The following are equivalent.*

1. *All the paths of $\mathcal{G}_r$ satisfy the Streett[1] condition $\{\langle B, G\rangle\}$.*
2. *There is an odd S[1]-ranking for $\mathcal{G}_r$.*

Lemma 2 implies that $\mathcal{A}$ accepts a word w iff there is a ranking for the run DAG $\mathcal{G}_r$ of $\mathcal{A}$ on w such that every infinite path in $\mathcal{G}_r$ has infinitely many G-vertices or infinitely many odd vertices. Intuitively, the lemma suggests that the two requirements that the Streett[1] condition involves (finitely many B or infinitely many G) can be reduced to a new condition of only one type (infinitely often, for odd or G-vertices). This intuition is formalized in the translation of USW[1] to NBW, which is described (as a special case of a translation of USW to NBW) in Theorem 2.

Theorem 1. *Let $\mathcal{A}$ be a* USW[1] *with n states. There is an* NBW *$\mathcal{A}'$ with $2^{O(n \log n)}$ states such that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Streett-ranking. We now turn to consider a general Streett condition $\alpha = \{\langle B_1, G_1\rangle, \ldots, \langle B_k, G_k\rangle\}$, where $\mathcal{G}_r$ is accepting iff all paths in $\mathcal{G}_r$ have, for all $1 \leq i \leq k$, finitely many B_i-vertices or infinitely many G_i-vertices.

Consider a function $f : V \rightarrow [2n]^k$. For an index $1 \leq i \leq k$, we use $f(v)[i]$ to denote the i-th element in $f(v)$. We call $f(v)[i]$ the *i-rank of v* (according to f). A *Streett-ranking* (*S-ranking*, for short) for $\mathcal{G}_r$ is a function $f : V \rightarrow [2n]^k$ that satisfies the following two conditions:

1. For all vertices $\langle q, l\rangle \in V$ and $1 \leq i \leq k$, if $f(\langle q, l\rangle)[i]$ is odd, then $q \notin B_i$.
2. For all edges $\langle\langle q, l\rangle, \langle q', l+1\rangle\rangle \in E$ and $1 \leq i \leq k$, either $f(\langle q', l+1\rangle)[i] \leq f(\langle q, l\rangle)[i]$ or $q \in G_i$.

Thus, an S-ranking f associates with each vertex in $\mathcal{G}_r$ a vector of k ranks in $[2n]$ so that for all $1 \leq i \leq k$, the projection $f[i]$ of f is an S[1]-ranking with respect to $\langle B_i, G_i\rangle$.

We say that the ranking f is an *odd S-ranking* if, for all $1 \leq i \leq k$, the S[1]-ranking $f[i]$ is odd. Thus, for all $1 \leq i \leq k$, all the paths of $\mathcal{G}_r$ either visit G_i-vertices infinitely often or eventually get trapped in an odd i-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 $\mathcal{G}_r$ and for all $1 \leq i \leq k$, either $q_l \in G_i$ for infinitely many $l \geq 0$ or there is $l \geq 0$ such that $f(\langle q_l, l\rangle)[i]$ is odd, and for all $l' \geq l$, we have $f(\langle q_{l'}, l'\rangle)[i] = f(\langle q_l, l\rangle)[i]$. Note that, equivalently, f is odd if every path of $\mathcal{G}_r$ has, for all $1 \leq i \leq k$ infinitely many G_i-vertices or infinitely many vertices with an odd i-rank.

Lemma 3. *The following are equivalent.*

1. *All the paths of $\mathcal{G}_r$ satisfy the Streett condition $\{\langle B_1, G_1\rangle, \ldots, \langle B_k, G_k\rangle\}$.*
2. *There is an odd S-ranking for $\mathcal{G}_r$.*

Proof: Immediate from Lemma 2 and the definition of an odd S-ranking as the composition of k odd S[1]-rankings for the pairs in the Streett condition. □

From USW to NBW. A USW $\mathcal{A}$ with $\alpha = \{\langle B_1, G_1\rangle, \langle B_2, G_2\rangle, \ldots, \langle B_k, G_k\rangle\}$ is equivalent to the intersection of the k USW[1] $\mathcal{A}_i$ obtained from $\mathcal{A}$ by taking the acceptance condition to be $\langle B_i, G_i\rangle$. It is not surprising, then, that the definition of an odd S-ranking f requires f to be an odd S[1]-ranking with respect to all pairs in α. Following this approach, translating $\mathcal{A}$ to an NBW $\mathcal{A}'$ can proceed by first translating each USW[1] $\mathcal{A}_i$ into an equivalent NBW $\mathcal{A}'_i$ as described in Theorem 1, and then defining $\mathcal{A}'$ as the product of the $\mathcal{A}'_i$'s (see [Cho74] for the product construction for NBW). Such a product would have at most $k \cdot 3^{nk} \cdot (2n+1)^{nk}$ states. We now describe a direct construction, which follows from the analysis of S-ranking, and which is exponentially better.

Theorem 2. *Let $\mathcal{A}$ be a* USW *with n states and index k. There is an* NBW *$\mathcal{A}'$ with $2^{O(nk \log n)}$ states such that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof: Let $\mathcal{A} = \langle \Sigma, Q, Q_{in}, \delta, \{\langle B_1, G_1\rangle, \ldots, \langle B_k, G_k\rangle\}\rangle$ When $\mathcal{A}'$ reads a word w, it guesses an odd S-ranking for the run DAG $\mathcal{G}_r$ of $\mathcal{A}$ on w. At a given point of a run of $\mathcal{A}'$, it keeps in its memory a whole level of $\mathcal{G}_r$ and a guess for the rank of the vertices at this level. In order to make sure that for all $1 \leq i \leq k$, all the paths of $\mathcal{G}_r$ either visit i-odd or G_i-vertices infinitely often, $\mathcal{A}'$ has a flag $1 \leq i \leq k$ and it remembers the set of states that owe a visit to i-odd or G_i-vertices. Once the set becomes empty, i is changed to $(i \bmod k) + 1$.

Before we define $\mathcal{A}'$, we need some notations. A *level ranking* for $\mathcal{A}$ is a function $g : Q \to [2n]^k$, such that for all $1 \leq i \leq k$, if $g(q)[i]$ is odd, then $q \notin B_i$. 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 $\mathcal{G}_r$, for $l \geq 0$, contains the states in S, and the $(l+1)$-th letter in w is σ, then level $l+1$ of $\mathcal{G}_r$ contains the states in $\delta(S, \sigma)$.

For two level rankings g and g' in $\mathcal{R}$ and a letter σ, we say that g' *covers* $\langle g, \sigma\rangle$ if for all q and q' in Q, if $q' \in \delta(q, \sigma)$, then for all $1 \leq i \leq k$, either $q \in G_i$ or $g'(q')[i] \leq g(q)[i]$. Thus, if g describes the ranks of the vertices of level l, and the $(l+1)$-th letter in w is σ, then g' is a possible level ranking for level $l+1$. Finally, for $g \in \mathcal{R}$ and $1 \leq i \leq k$, let $good(g, i) = G_i \cup \{q : g(q)[i] \in [2n]^{odd}\}$. Thus, a state of Q is in $good(g, i)$ if it belongs to G_i or has an i-odd rank.

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

- $Q' = 2^Q \times 2^Q \times \mathcal{R} \times \{1, \ldots, k\}$, where a state $\langle S, O, g, i \rangle \in Q'$ indicates that the current level of the DAG contains the states in S, the pair that is now examined is i, the set $O \subseteq S$ contains states along paths that have not visited a G_i-vertex or an i-odd vertex since the last time O has been empty, and g is the guessed level ranking for the current level.[4]
- $Q'_{in} = Q_{in} \times \{\emptyset\} \times \mathcal{R} \times \{1\}$.
- δ' is defined, for all $\langle S, O, g, i \rangle \in Q'$ and $\sigma \in \Sigma$, as follows.
    - If $O \neq \emptyset$, then $\delta'(\langle S, O, g, i \rangle, \sigma) = \{\langle \delta(S, \sigma), \delta(O, \sigma) \setminus \mathit{good}(g', i), g', i \rangle : g' \text{ covers } \langle g, \sigma \rangle\}$.
    - If $O = \emptyset$, then $\delta'(\langle S, O, g, i \rangle, \sigma) = \{\langle \delta(S, \sigma), \delta(S, \sigma) \setminus \mathit{good}(g', (i \bmod k) + 1), g', (i \bmod k) + 1 \rangle : g' \text{ covers } \langle g, \sigma \rangle\}$.
- $\alpha' = 2^Q \times \{\emptyset\} \times \mathcal{R} \times \{1, \ldots, k\}$.

Since there are at most $(2n+1)^{nk}$ level rankings, the number of states in $\mathcal{A}'$ is at most $k \cdot 3^n \cdot (2n+1)^{nk} = 2^{O(nk \log n)}$. □

For the proof of Theorem 1, note that when $\mathcal{A}$ is a USW[1], there is no need for the index component in the state space, and $\mathcal{A}'$ has $2^{O(n \log n)}$ states.

Theorem 3. *Let $\mathcal{A}$ be an NRW with n states and index k. There is an NBW $\tilde{\mathcal{A}}$ with $2^{O(nk \log n)}$ states such that $\mathcal{L}(\tilde{\mathcal{A}}) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{A})$.*

Proof: The automaton $\tilde{\mathcal{A}}$ is obtained by translating the USW that dualizes $\mathcal{A}$ to an NBW. □

Note that the previous complementation constructions for NRW involve a $2^{O(nk \log nk)}$ blow up, as they first translate the NRW into an NBW with $O(nk)$ states, and complementing an NBW with m states results in an NBW with $2^{O(m \log m)}$ states [Saf88]. Thus, our construction eliminates the term k from the exponent. In addition, the constants hiding in the $O()$ notation are exponentially better in our approach. Indeed, the number of states of an NBW equivalent to an NRW[k] with n states may be $2nk$. On the other hand, our ranks refer to the original state space of the automaton, and there is no need to double it for each pair. For example, when $k = 1$, going through NBW results in a complementary NBW with at most $3^{2n} \cdot (4n+1)^{2n}$ states, whereas our direct construction results in an NBW with at most $3^n \cdot (2n+1)^n$ states.

4 NSW Complementation

In this section we analyze runs of URW and use the analysis in order to translate URW to NBW. The translation is then used for NSW complementation.

[4] Note that a naive direct construction results in an NBW whose state space contains k subsets of Q, each acting as the "O component" of a pair in α. Since, however, the O component of all pairs should become empty infinitely often, it is possible to optimize the naive construction and keep track of a single pair (and its corresponding O component) at a time.

Rabin-ranking. Consider a Rabin condition $\alpha = \{\langle G_1, B_1\rangle, \langle G_2, B_2\rangle, \ldots, \langle G_k, B_k\rangle\}$. Let $I = \{1, \ldots, k\}$, and let $\Omega_I = [2n]^{even} \cup ([2n]^{odd} \times I)$. Recall that a run r satisfies α iff there is $1 \leq i \leq k$ such that $Inf(r) \cap G_i \neq \emptyset$ and $Inf(r) \cap B_i = \emptyset$. A *Rabin rank* is a tuple $\langle\langle r_1, i_1\rangle, \ldots, \langle r_{m-1}, i_{m-1}\rangle, r_m\rangle$ of m ranks in Ω_I, for $1 \leq m \leq k+1$. The i_j's are distinct, and except for the last rank, which is even, all the ranks are odd. We refer to m as the *width* of the rank, and to the j-th element of a Rabin rank γ as $\gamma[j]$. Let $\mathcal{D}_I$ denote the set of Rabin ranks (with respect to α).

A *Rabin ranking* (*R-ranking*, for short) for $\mathcal{G}_r$ is a function $f : V \to \mathcal{D}_I$ that satisfies the following conditions:

1. For all $\langle q, l\rangle \in V$, let m be the width of $f(\langle q, l\rangle)$. Then,
   (a) For all $1 \leq j < m-1$, if $f(\langle q, l\rangle)[j] = \langle r_j, i_j\rangle$, then $q \notin G_{i_j}$.
   (b) For all $1 \leq j < m$, if $f(\langle q, l\rangle)[j] = \langle r_j, i_j\rangle$, then $q \notin B_{i_j}$.
2. For all edges $\langle\langle q, l\rangle, \langle q', l+1\rangle\rangle \in E$, let m and m' be the widths of $f(\langle q, l\rangle)$ and $f(\langle q', l+1\rangle)$, respectively, and let $m'' = \min\{m, m'\}$. Then,
   (a) For all $1 \leq j \leq m''-1$, if $f(\langle q', l+1\rangle)[h] = f(\langle q, l\rangle)[h]$ for all $1 \leq h < j$, then $f(\langle q', l+1\rangle)[j] \leq f(\langle q, l\rangle)[j]$.
   (b) If $f(\langle q', l+1\rangle)[h] = f(\langle q, l\rangle)[h]$ for all $1 \leq h < m''$, then either $f(\langle q', l+1\rangle)[m''] \leq f(\langle q, l\rangle)[m'']$, or $m'' > 1$, $f(\langle q, l\rangle)[m''-1] = \langle r_{m''-1}, i_{m''-1}\rangle$, and $q \in G_{i_{m''-1}}$.

Thus, if $f(\langle q, l\rangle) = \gamma$ and $f(\langle q', l+1\rangle) = \gamma'$, then Condition 2 guarantees that for all $1 \leq j \leq m''-1$, if $\gamma'[j] > \gamma[j]$, then there is $1 \leq h < j$ such that $\gamma'[h] \neq \gamma[h]$. In addition, if $\gamma'[m''] > \gamma[m'']$, then either there is $1 \leq h < j$ such that $\gamma'[h] \neq \gamma[h]$, or $m'' > 1$, $f(\langle q, l\rangle)[m''-1] = \langle r_{m''-1}, i_{m''-1}\rangle$, and $q \in G_{i_{m''-1}}$. We refer to the latter conjunction as the *bridge disjunct* of Condition 2b.

For a vertex $v \in V$, the width of v, denoted $width(v)$, is the width of $f(v)$. A vertex with width 1 is *even*, and a vertex with width at least 2 is *odd*. We say that a vertex $\langle q, l\rangle$ is *happy* (with respect to f) if $f(\langle q, l\rangle) = \langle\langle r_1, i_1\rangle, \ldots, \langle r_{m-1}, i_{m-1}\rangle, r_m\rangle$ for some $m > 1$ and $q \in G_{i_{m-1}}$. Note that all happy vertices are odd. An R-ranking is an *odd R-ranking* if all infinite paths have infinitely many happy vertices.

Lemma 4. *The following are equivalent.*

1. *All the paths of $\mathcal{G}_r$ satisfy the Rabin condition $\{\langle G_1, B_1\rangle, \ldots, \langle G_k, B_k\rangle\}$.*
2. *There is an odd R-ranking for $\mathcal{G}_r$.*

Intuitively, Lemma 4 suggests that the requirements that the Rabin condition involves, which are of different types (infinitely often, for the G_i elements, and finitely often, for the B_i elements), can be reduced to a new condition of only one type (infinitely often, for happy vertices). This intuition is formalized in the construction below. Note that while the proof of Lemma 4 is complicated, the construction that follows is simple.

Theorem 4. *Let $\mathcal{A}$ be a* URW *with n states and index k. There is an* NBW *$\mathcal{A}'$ with $2^{O(nk \log nk)}$ states such that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof: Let $\mathcal{A} = \langle \Sigma, Q, Q_{in}, \delta, \{\langle G_1, B_1\rangle, \ldots, \langle G_k, B_k\rangle\}\rangle$ When $\mathcal{A}'$ reads a word w, it guesses an odd R-ranking for the run DAG $\mathcal{G}_r$ of $\mathcal{A}$ on w. At a given point of a run of

$\mathcal{A}'$, it keeps in its memory a whole level of $\mathcal{G}_r$ and a guess for the ranks of the vertices at this level. In order to make sure that all the infinite paths of $\mathcal{G}_r$ visit happy vertices infinitely often, $\mathcal{A}'$ remembers the set of states that owe a visit to happy vertices.

Before we define $\mathcal{A}'$, we need to adjust our notations to ranks in $\mathcal{D}_I$. A *level ranking* for $\mathcal{A}$ is a function $g : Q \to \mathcal{D}_I$, such that for all $q \in Q$ with $width(g(q)) = m$, and for all $1 \leq j < m - 1$, if $g(q)[j] = \langle r_j, i_j \rangle$, then $q \notin G_{i_j}$. Also, for all $1 \leq j < m$, if $g(q)[j] = \langle r_j, i_j \rangle$, then $q \notin B_{i_j}$. The correspondence between the above conditions and Condition 1 in the definition of R-ranking guarantees that g describes possible ranks for vertices in some level of $\mathcal{G}_r$. Let $\mathcal{R}$ be the set of all level rankings. Note that since a Rabin rank in $\mathcal{D}_I$ can be characterized by at most k elements in $[2n]^{odd}$, one element in $[2n]^{even}$, and a permutation of I, the size of $\mathcal{D}_I$ is at most $n^k \cdot (n + 1) \cdot k!$. Accordingly, there are at most $2^{O(nk \log nk)}$ level rankings. For two level rankings g and g' in $\mathcal{R}$, a subset $S \subseteq Q$, and a letter σ, we say that g' *covers* $\langle g, S, \sigma \rangle$ if for all $q \in S$ and $q' \in \delta(q, \sigma)$, the following holds. Let m and m' be the widths of $g(q)$ and $g(q')$, respectively, and let $m'' = \min\{m, m'\}$. Then,

1. For all $1 \leq j \leq m'' - 1$, if $g'(q')[h] = g(q)[h]$ for all $1 \leq h < j$, then $g'(q')[j] \leq g(q)[j]$.
2. If $g(q')[h] = g(q)[h]$ for all $1 \leq h < m''$, then either $g'(q')[m''] \leq g(q)[m'']$, or $m'' > 1$, $g(q)[m'' - 1] = \langle r_{m''-1}, i_{m''-1} \rangle$, and $q \in G_{i_{m''-1}}$.

The correspondence between the above conditions and Condition 2 in the definition of R-ranking guarantees that if S is the set of states in level l, the $(l + 1)$-th letter in the word is σ, g describes the ranks of vertices of level l, and g' covers $\langle g, S, \sigma \rangle$, then g' is a possible level ranking for level $l + 1$. Finally, for $g \in \mathcal{R}$, let $good(g) \subseteq Q$ be the set of states q such that the width of $g(q)$ is $m > 1$, $g(q)[m - 1] = \langle r_{m-1}, i_{m-1} \rangle$ for some $r_{m-1} \in [2n]^{odd}$, and $q \in G_{i_{m-1}}$.

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

- $Q' = 2^Q \times 2^Q \times \mathcal{R}$, 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 a happy 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 $\delta'(\langle S, O, g \rangle, \sigma) = \{\langle \delta(S, \sigma), \delta(O, \sigma) \setminus good(g'), g' \rangle : g'$ covers $\langle g, S, \sigma \rangle\}$.
  - If $O = \emptyset$, then $\delta'(\langle S, O, g \rangle, \sigma) = \{\langle \delta(S, \sigma), \delta(S, \sigma) \setminus good(g'), g' \rangle : g'$ covers $\langle g, S, \sigma \rangle\}$.
- $\alpha' = 2^Q \times \{\emptyset\} \times \mathcal{R}$.

Since there are at most $2^{O(nk \log nk)}$ level rankings, the number of states in $\mathcal{A}'$ is at most $3^n \cdot 2^{O(nk \log nk)} = 2^{O(nk \log nk)}$. □

Remark 1. Below we discuss some variants of R-ranking, which still satisfy Lemma 4, and therefore, with a corresponding adjustment of the definition of "covers", can be used in order to translate URW to NBW. First, it can be shown that Condition 1a is not essential. In other words, the proof of Lemma 4 stays valid when we allow a vertex $\langle q, l\rangle$ with $q \in G_{i_j}$ to have $f(\langle q, l\rangle)[j] = \langle r_j, i_j\rangle$, for $j < \mathit{width}(\langle q, l\rangle)$. Condition 1a, however, has the advantage that it restricts the state space of the NBW. Second, the indices i_j of a Rabin rank $\langle\langle r_1, i_1\rangle, \ldots, \langle r_{m-1}, i_{m-1}\rangle, r_m\rangle$ need not be distinct. Again, the proof stays valid if we allow an index to repeat. As with Condition 1a, the fact the indices are distinct restricts the state space. On the other hand, in a symbolic implementation, such a restriction may cause complications.

Theorem 5. *Let $\mathcal{A}$ be an NSW with n states and index k. There is an NBW $\tilde{\mathcal{A}}$ with $2^{O(nk \log nk)}$ states such that $\mathcal{L}(\tilde{\mathcal{A}}) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{A})$.*

Proof: The automaton $\tilde{\mathcal{A}}$ is obtained by translating the URW that dualizes $\mathcal{A}$ to an NBW. □

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. We first show how an automaton that complements $\mathcal{A}_2$, when constructed using our construction, can be optimized in the process of its intersection with $\mathcal{A}_1$. We then describe the product $\mathcal{P}$ of $\mathcal{A}_1$ with the complementing automaton, namely the automaton whose emptiness should be tested in order to check whether $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$. Our goal in describing $\mathcal{P}$ is to highlight the simplicity of the language-containment algorithm. To the best of our knowledge, this is the first time that such a product $\mathcal{P}$ is described in a few lines.

5.1 Optimizations That Depend on $\mathcal{A}_1$

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 $\tilde{\mathcal{A}}_2$. Let n be the number of states on $\mathcal{A}_2$. The range of the ranks is $0, \ldots, 2n$, and, depending on the type of $\tilde{\mathcal{A}}_2$, they may be associated with indices, and/or arranged in tuples. The bound $2n$ on the maximal rank follows from the fact that the width of the run DAG is bounded by n. To see the latter, consider a run DAG $\mathcal{G}_r$ that embodies all the runs of $\tilde{\mathcal{A}}_2$ on a word $w = \sigma_0 \cdot \sigma_1 \cdots$. A level $l \geq 0$ of $\mathcal{G}_r$ contains exactly all vertices $\langle q, l\rangle$ such that a run of $\mathcal{A}_2$ on w visits q after reading the prefix $\sigma_0 \cdot \sigma_1 \ldots \sigma_{l-1}$. Thus, since there are n different states, there may be at most n different such vertices in each level.

In fact, we can tighten the width of $\mathcal{G}_r$ 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 [RS59]: 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 $2r^{max}$. 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 to appear 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 designated for checking the containment of $\mathcal{L}(\mathcal{A}_1)$ in $\mathcal{L}(\mathcal{A}_2)$, to $2r^{max}_{\mathcal{A}_1}$.

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 $\mathcal{L}(\mathcal{A}^s_2) = \emptyset$, we remove s from the range of δ_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 $\mathcal{A}^d_2$ 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 $\mathcal{L}(\mathcal{A}_1)$.

5.2 The Product Automaton

We describe the construction for the most complicated case, where $\mathcal{A}_1$ and $\mathcal{A}_2$ are Streett automata. Other cases are similar, with modified definitions for $\mathcal{R}$, *covers*, and *good*, as in the proofs of Theorems 3 and 5.

Let $\mathcal{A}_1 = \langle \Sigma, Q_1, Q^1_{in}, \delta_1, \alpha_1 \rangle$ and $\mathcal{A}_2 = \langle \Sigma, Q_2, Q^2_{in}, \delta_2, \alpha_2 \rangle$. Also, let $\mathcal{R}$, *covers*, and *good*, be as in the proof Theorem 5, with respect to the components of $\mathcal{A}_2$. As explained in Section 5.1, the ranks in the range $\mathcal{D}_I$ of the level rankings in $\mathcal{R}$ can be restricted to Rabin ranks in which $\Omega_I = [2r^{max}_{\mathcal{A}_1}]^{even} \cup ([2r^{max}_{\mathcal{A}_1}]^{odd} \times I)$. We define the product of $\mathcal{A}_1$ and $\tilde{\mathcal{N}_2}$ as an NSW $\mathcal{P} = \langle \Sigma, Q', Q'_{in}, \delta', \alpha' \rangle$, where

- $Q' = Q_1 \times 2^{Q_2} \times 2^{Q_2} \times \mathcal{R}$.
- $Q'_{in} = Q^1_{in} \times \{Q^2_{in}\} \times \{\emptyset\} \times \mathcal{R}$.
- δ' is defined, for all $\langle q, S, O, g \rangle \in Q'$ and $\sigma \in \Sigma$, as follows.
  - If $O \neq \emptyset$, then $\delta'(\langle q, S, O, g \rangle, \sigma) =$ $\{\langle q', \delta(S, \sigma), \delta(O, \sigma) \setminus good(g'), g' \rangle : q' \in \delta_1(q, \sigma)$ and g' covers $\langle g, S, \sigma \rangle\}$.
  - If $O = \emptyset$, then $\delta'(\langle q', S, O, g \rangle, \sigma) =$ $\{\langle q', \delta(S, \sigma), \delta(S, \sigma) \setminus good(g'), g' \rangle : q' \in \delta_1(q, \sigma)$ and g' covers $\langle g, S, \sigma \rangle\}$.
- $\alpha' = (\bigcup_{\langle G, B \rangle \in \alpha_1} \{\langle G \times 2^{Q_2} \times 2^{Q_2} \times \mathcal{R}, B \times 2^{Q_2} \times 2^{Q_2} \times \mathcal{R} \rangle\}) \times \{\langle Q', Q_1 \times 2^{Q_2} \times \{\emptyset\} \times \mathcal{R} \rangle\}$.

6 Discussion

Complementation is a key construction in formal verification. At the same time, complementation of automata on infinite words is widely perceived to be rather difficult, unlike the straightforward subset construction for automata on finite words [RS59]. Checking the syllabi of several formal-verification courses, one finds that while most mention the closure under complementation for automata on infinite words, only a few actually teach a complementation construction. Indeed, not too many researchers are sufficiently familiar with the details of known constructions, and many believe that most of the students would not be able to follow the intricate technical details.

This situation has led to a perception that complementation constructions for automata on infinite words are rather impractical. Indeed, an attempt to implement Safra's construction led support to this perception [THB95]. Consequently, there is extensive work on simulation-based abstraction and refinement, cf. [LT87, AL91, DHW91], and research has focused on ways in which fair simulation can approximate language containment [HKR02], and ways in which the complementation construction can be circumvented by manually bridging the gap between fair simulation and language containment [Att99, KPP03].

We believe that this perception ought to be challenged. It is true that language containment is PSPACE-complete [MS73], whereas simulation can be solved in polynomial time [HHK95]. Nevertheless, the exponential blow-up of complementation, which is the reason underlying the PSPACE-hardness of language containment, is a worst-case analysis. As we have learned recently in the context of reasoning about automata on finite words, worst-case blow-ups rarely occur in typical practice [EKM98]. This is confirmed by our recent experience with the complementation construction for Büchi automata [GKSV03]. It is worth remembering also that the translation from LTL to Büchi automata [VW94] was for several years considered impractical because of its worst-case exponential blow-up. We also found the construction of [KV01] quite easy to teach, covering it in a two-hour lecture[5]. We believe that the complementation problem for automata on infinite words ought to be investigated further by the research community, in order to make complementation constructions routinely applicable in formal verification. We hope that our results here for Rabin and Streett automata would constitute a significant contribution in that direction.

References

[AL91] M. Abadi and L. Lamport. The existence of refinement mappings. *TCS*, 82(2):253–284, 1991.

[Att99] P. Attie. Liveness-preserving simulation relations. In *Proc. 18th PODC*, pages 63–72, 1999.

[Büc62] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. Internat. Congr. Logic, Method. and Philos. Sci. 1960*, pages 1–12, Stanford, 1962.

[Cho74] Y. Choueka. Theories of automata on ω-tapes: A simplified approach. *Journal of CSS*, 8:117–141, 1974.

[DHW91] D.L. Dill, A.J. Hu, and H. Wong-Toi. Checking for language inclusion using simulation relations. In *Proc. 3rd CAV*, LNCS 575, pages 255–265, 1991

[EJ91] E.A. Emerson and C. Jutla. Tree automata, μ-calculus and determinacy. In *Proc. 32nd FOCS* pages 368–377, 1991.

[EKM98] J. Elgaard, N. Klarlund, and A. Möller, Mona 1.x: new techniques for WS1S and WS2S. In *Proc. 10th CAV*, LNCS 1427, pages 516–520, 1998.

[FKV04] E. Friedgut, O. Kupferman, and M.Y. Vardi. Büchi complementation made tighter. In *Proc. 2nd ATVA*, LNCS 3299, pages 64–78, 2004.

[GBS02] S. Gurumurthy, R. Bloem, and F. Somenzi. Fair simulation minimization. In *Proc. 14th CAV*, LNCS 2404, pages 610–623, 2002.

[GKSV03] S. Gurumurthy, O. Kupferman, F. Somenzi, and M.Y. Vardi. On complementing nondeterministic Büchi automata. In *Proc. 12th CHARME*, LNCS 2860, pages 96–110, 2003.

[HHK95] M.R. Henzinger, T.A. Henzinger, and P.W. Kopke. Computing simulations on finite and infinite graphs. In *Proc. 36th FOCS*, pages 453–462, 1995.

[HHK96] R.H. Hardin, Z. Har'el, and R.P. Kurshan. COSPAN. In *Proc. 8th CAV*, LNCS 1102, pages 423–427, 1996.

[HK02] D. Harel and O. Kupferman. On the behavioral inheritance of state-based objects. *IEEE TSE*, 28(9):889–903, 2002.

[HKR02] T.A. Henzinger, O. Kupferman, and S. Rajamani. Fair simulation. *I&C*, 173(1):64–81, 2002.

[Hol97] G.J. Holzmann. The model checker SPIN. *IEEE TSE*, 23(5):279–295, May 1997.

[Kla90] N. Klarlund. *Progress Measures and finite arguments for infinite computations*. PhD thesis, Cornell University, 1990.

[5] Lecture notes can be found in www.wisdom.weizmann.ac.il/∼vardi/av (Moshe Vardi) and. www7.in.tum.de/lehre/automaten2/SS99/ (Javier Esparza).

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

[KP00] Y. Kesten and A. Pnueli. Verification by augmented finitary abstraction. *I&C*, 163(1):203–243, 2000.

[KPP03] Y. Kesten, N. Piterman, and A. Pnueli. Bridging the gap between fair simulation and trace containment. In *Proc. 15th CAV*, LNCS 2725, pages 381–393, 2003.

[KPSZ02] Y. Kesten, A. Pnueli, E. Shahar, and L. Zuck. Network invariant in action. In *Proc. 13th CONCUR*, LNCS 2421, pages 101–115, 2002.

[Kur87] R.P. Kurshan. Complementing deterministic Büchi automata in polynomial time. *Journal of CSS*, 35:59–71, 1987.

[Kur94] R.P. Kurshan. *Computer Aided Verification of Coordinating Processes*. Princeton Univ. Press, 1994.

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

[KV04] O. Kupferman and M.Y. Vardi. From complementation to certification. In *10th TACAS*, LNCS 2988, pages 591-606, 2004.

[LPS81] D. Lehman, A. Pnueli, and J. Stavi. Impartiality, justice, and fairness – the ethics of concurrent termination. In *Proc. 8th ICALP*, LNCS 115, pages 264–277, 1981.

[LT87] N. A. Lynch and M.R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proc. 6th PODC*, pages 137–151, 1987.

[Mer00] S. Merz. Weak alternating automata in Isabelle/HOL. In *Proc. 13th TPiHOL*, LNCS 1869, pages 423–440, 2000.

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

[MS73] A.R. Meyer and L.J. Stockmeyer. Word problems requiring exponential time: Preliminary report. In *Proc. 5th STOC*, pages 1–9, 1973.

[MS87] D.E. Muller and P.E. Schupp. Alternating automata on infinite trees. *TCS*, 54:267–276, 1987.

[RS59] M.O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:115–125, 1959.

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

[Saf92] S. Safra. Exponential determinization for ω-automata with strong-fairness acceptance condition. In *Proc. 24th STOC*, 1992.

[SV89] S. Safra and M.Y. Vardi. On ω-automata and temporal logic. In *Proc. 21st STOC*, pages 127–137, 1989.

[THB95] S. Tasiran, R. Hojati, and R.K. Brayton. Language containment using nondeterministic ω-automata. In *Proc. 8th CHARME*, LNCS 987, pages 261–277, 1995.

[VW94] M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *I&C*, 115(1):1–37, November 1994.