

Tight Upper Bounds for Streett and Parity Complementation*

Yang Cai¹ and Ting Zhang²

- 1 MIT Computer Science and Artificial Intelligence Laboratory
The Stata Center, 32-G696, Cambridge, MA 02139 USA
ycai@csail.mit.edu
- 2 Department of Computer Science, Iowa State University
226 Atanasoff Hall, Ames, IA 50011 USA
tingz@iastate.edu

Abstract

Complementation of finite automata on infinite words is not only a fundamental problem in automata theory, but also serves as a cornerstone for solving numerous decision problems in mathematical logic, model-checking, program analysis and verification. For Streett complementation, a significant gap exists between the current lower bound $2^{\Omega(n \lg nk)}$ and upper bound $2^{O(nk \lg nk)}$, where n is the state size, k is the number of Streett pairs, and k can be as large as 2^n . Determining the complexity of Streett complementation has been an open question since the late 80's. In this paper we show a complementation construction with upper bound $2^{O(n \lg n + nk \lg k)}$ for $k = O(n)$ and $2^{O(n^2 \lg n)}$ for $k = \omega(n)$, which matches well the lower bound obtained in [3]. We also obtain a tight upper bound $2^{O(n \lg n)}$ for parity complementation.

1998 ACM Subject Classification F.1.1 Models of Computation; F.4.1 Mathematical Logic; F.4.3 Formal Languages

Keywords and phrases Streett automata, ω -automata, parity automata, complementation, upper bounds

Digital Object Identifier 10.4230/LIPIcs.CSL.2011.112

1 Introduction

Automata on infinite words (ω -automata) have wide applications in synthesis and verification of reactive concurrent systems. Complementation plays a fundamental role in many of these applications, especially in solving the language containment problem: whether a language recognized by automaton \mathcal{A} is contained by another language represented by automaton \mathcal{B} , which is equivalent to whether the language of \mathcal{A} and the complementary language of \mathcal{B} intersect. In automata-theoretic model checking [10, 27], both system behaviors and logical specifications are represented as formal languages, and model checking by and large amounts to solving the corresponding language containment problem. As both language intersection and emptiness test are rather easy, the efficiency of complementation becomes crucial to practical deployment of model-checking tools. For this reason and many others, determining the state complexity of the complementation problem has been extensively studied in the last four decades [26].

* This research has been supported by NSF CAREER Award CCF-0954132.



Related Work

ω -automata were invented by Büchi in 1962 as a method of attack on definability and decision problems for monadic second order logic on arithmetics (S1S) [1]. That type of ω -automata are nowadays called Büchi automata. The initial Büchi complementation was not explicitly constructive and required double exponential blow-up [1]. But since then Büchi complementation has been extensively studied. The upper bound was continuously improved to $2^{O(n^2)}$ [24], $2^{O(n \lg n)}$ [21], $O((6n)^n)$ [11], $O((0.97n)^n)$ [6] and finally to $O(n^2 L(n))$ where $L(n) \approx (0.76n)^n$ [23], which matched well the lower bound $\Omega(L(n))$ [28].

Complementation for automata with rich acceptance conditions, such as Rabin automata and Streett automata, is much more sophisticated. Kupferman and Vardi showed a $2^{O(nk \lg n)}$ complementation construction for Rabin automata [14], and we showed this construction is essentially optimal [2]. This leaves Streett complementation the last classical problem where the gap between the lower and upper bounds is substantial. Besides that, Streett complementation has an importance of its own. Streett automata share identical algebraic structures with Büchi automata, except being equipped with richer acceptance conditions. A Streett acceptance condition comprises a finite list of indexed pairs of sets of states. Each pair consists of an enabling set and a fulfilling set. A run is accepting if for each pair, if the run visits states in the enabling set infinitely often, then it also visits states in the fulfilling set infinitely often. This naturally corresponds to the *strong fairness* condition that infinitely many requests are responded infinitely often, a necessary requirement for meaningful computations [5, 7]. Another advantage of Streett automata is that they are much more succinct than Büchi automata; it is unavoidable in the worst case to have 2^n state blow-up to translate a Streett automaton with $O(n)$ states and $O(n)$ index pairs to an equivalent Büchi automaton [25]. An interesting question is: to what extent does the gain from the succinctness have to be paid back at the time of complementation?

The first construction for Streett complementation was given by Safra and Vardi, and that construction required $2^{O((nk)^5)}$ state blow-up [25]. Klarlund improved this bound to $2^{O(nk \lg nk)}$ [8]. The same bound was achieved by Safra via determinization [22], by Piterman with an improved determinization construction [19], and by Kupferman and Vardi [14]. However, so far no construction has been proved to cost less than $2^{O(nk \lg nk)}$ states. The question of whether Streett complementation can be further improved from $2^{O(nk \lg nk)}$ has been constantly raised in the recent literature [14, 28, 26]. In this paper we answer this question affirmatively.

Ranking-based Complementation

A Ranking-based complementation was first proposed by Klarlund [8]. Klarlund's Büchi complementation (resp. Streett complementation) relies on *quasi co-Büchi measure* (resp. *quasi Rabin measure*), which is a ranking function on states in a run graph, measuring the progress of a run toward being accepted. By this complementation scheme, Klarlund gave a $2^{O(n \lg n)}$ Büchi complementation and a $2^{O(nk \lg nk)}$ Streett complementation [8]. Kupferman and Vardi developed a similar idea into an elegant and comprehensive framework [13, 9], obtaining complementation constructions for Büchi [11], generalized Büchi [12], Rabin and Streett [14].

Our Results

Our Streett complementation is obtained by improving Kupferman and Vardi's construction in [14]. We show that the larger the Rabin index size k , the higher the correlation between

| Type | Bound | Lower | Upper |
|-------------------|--|-------|-------|
| Büchi | $2^{\Theta(n \lg n)}$ | [17] | [21] |
| Generalized Büchi | $2^{\Theta(n \lg nk)}$ $k = O(2^n)$ | [28] | [14] |
| Streett | $2^{\Theta(n \lg n + nk \lg k)}$ $k = O(n)$ $2^{\Theta(n^2 \lg n)}$ $k = \omega(n)$ | [3] | this |
| Rabin | $2^{\Theta(nk \lg n)}$ $k = O(2^n)$ | [2] | [14] |
| Parity | $2^{\Theta(n \lg n)}$ $k = O(n)$ | [17] | this |

■ **Figure 1** The complementation complexities for ω -automata of common types.

infinite paths in a run graph satisfying a universal Rabin condition (the dual of an existential Streett condition), and characterize the correlation using two tree structures: ITS (*Increasing Tree of Sets*) and TOP (*Tree of Ordered Partitions*), both with elegant combinatorial properties. We show that our construction renders an upper bound $U(n, k)$, which is $2^{O(n \lg n + nk \lg k)}$ for $k = O(n)$ and $2^{O(n^2 \lg n)}$ for $k = \omega(n)$. $U(n, k)$ is a significant improvement from the previous best bound when $k = \omega(n)$. Speaking loosely, we gain succinctness without paying a dramatically higher price for complementation. $U(n, k)$ also matches the lower bound $L(n, k)$, which is $2^{\Omega(n \lg n + nk \lg k)}$ for $k = O(n)$ and $2^{\Omega(n^2 \lg n)}$ for $k = \omega(n)$ [3]. By a similar technique, we also obtain a $2^{O(n \lg n)}$ upper bound for parity complementation, which is essentially optimal, as parity automata generalizes Büchi automata, whose complementation lower bound is $2^{\Omega(n \lg n)}$ [17, 15]. This is surprising as the index size k (though small as $k \leq \lfloor (n+1)/2 \rfloor$) has no appearance in the asymptotical bound. We believe this is of practical interest as well, because it tells us that parity automata provide a richer acceptance condition without incurring an asymptotically higher cost on complementation. Combining the result with the one in [3] and previous findings in the literature, we now have a complete characterization of complementation complexity for ω -automata of common types. Figure 1 summarizes these results.

Paper Organization

Section 2 introduces basic notations and terminology in automata theory. Section 3 presents the framework of ranking based complementation; it introduces Büchi complementation [11], generalized Büchi complementation [12] and Streett complementation [14]. Section 4 presents our Streett complementation construction and Section 5 proves its complexity. Section 6 establishes a tight upper bound for parity complementation. Section 7 concludes with a discussion of future work. Due to space limit, all proofs are omitted, but they can be found in the full version of this paper at arXiv:1102.2960.

2 Preliminaries

Basic Notations

Let \mathbb{N} denote the set of natural numbers. We write $[i..j]$ for $\{k \in \mathbb{N} \mid i \leq k \leq j\}$, $[i..j]$ for $[i..j - 1]$, $[n]$ for $[0..n)$, and $[n]^{\text{even}}$ and $[n]^{\text{odd}}$ for even numbers and odd numbers in $[n]$, respectively. For an infinite sequence ϱ , we use $\varrho(i)$ to denote the i -th component for $i \in \mathbb{N}$. For a finite sequence α , we use $|\alpha|$ to denote the length of α , $\alpha[i]$ ($i \in [1..|\alpha|]$) to denote the object at the i -th position, and $\alpha[i..j]$ (resp. $\alpha[i..j)$) to denote the subsequence of α from position i to position j (resp. $j - 1$). When we compare finite sequences of

numbers, $>_m, \geq_m, =_m, <_m, \leq_m$ mean the corresponding standard lexicographical orderings up to position m . We reserve n and k as parameters of complementation instances (n for state size and k for index size), and define $\mu = \min(n, k)$ and $I = [1..k]$.

Automata and Runs.

A finite automaton on infinite words (ω -automaton) is a tuple $\mathcal{A} = (\Sigma, Q, Q_0, \Delta, \mathcal{F})$ where Σ is an alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is a set of initial states, $\Delta \subseteq Q \times \Sigma \times Q$ is a set of transitions, and \mathcal{F} is an acceptance condition.

An infinite word (ω -word) over Σ is an infinite sequence of letters in Σ . A *run* ϱ of \mathcal{A} over an ω -word w is an infinite sequence of states in Q such that $\varrho(0) \in Q_0$ and $\langle \varrho(i), w(i), \varrho(i+1) \rangle \in \Delta$ for $i \in \mathbb{N}$. Let $\text{Inf}(\varrho)$ be the set of states that occur infinitely many times in ϱ . An automaton accepts w if a run ϱ over w exists that satisfies \mathcal{F} , which is usually defined as a predicate on $\text{Inf}(\varrho)$. The language of \mathcal{A} , written $\mathcal{L}(\mathcal{A})$, is the set of ω -words accepted by \mathcal{A} .

Acceptance Conditions and Types

ω -automata are classified according to their acceptance conditions. Below we list automata of common types. Let G and B be functions from I to 2^Q .

- *Generalized Büchi*: $\langle B \rangle_I: \forall i \in I, \text{Inf}(\varrho) \cap B(i) \neq \emptyset$.
- *Büchi*: $\langle B \rangle_I$ with $I = \{1\}$ (i.e., $k = 1$).
- *Streett*: $\langle G, B \rangle_I: \forall i \in I, \text{Inf}(\varrho) \cap G(i) \neq \emptyset \rightarrow \text{Inf}(\varrho) \cap B(i) \neq \emptyset$.
- *Parity*: $\langle G, B \rangle_I$ with $B(1) \subset G(1) \subset \dots \subset B(k) \subset G(k)$.
- *Generalized co-Büchi*: $[B]_I: \exists i \in I, \text{Inf}(\varrho) \cap B(i) = \emptyset$.
- *Co-Büchi*: $[B]_I$ with $I = \{1\}$ (i.e., $k = 1$).
- *Rabin*: $[G, B]_I: \exists i \in I, \text{Inf}(\varrho) \cap G(i) \neq \emptyset \wedge \text{Inf}(\varrho) \cap B(i) = \emptyset$.

We use GB, B, S, P, GC, CB, and R, respectively, to denote the above acceptance conditions. By T -automata we mean the ω -automata with T -condition. Note that B and CB, GB and GC, and S and R are dual to each other, respectively. Also note that generalized Büchi and parity automata are both subclasses of Streett automata, and so are generalized co-Büchi and parity automata to Rabin automata. Let $J \subseteq I$. We use $[G, B]_J$ to denote the Rabin condition with respect to only indices in J . When J is a singleton, say $J = \{j\}$, we simply write $[G(j), B(j)]$ for $[G, B]_J$. The same convention is used for other conditions. For a Streett condition $\langle G, B \rangle_I$, we can assume that B is injective, because if $B(i) = B(i')$ for two different $i, i' \in I$, then we can replace $\langle G, B \rangle_{\{i, i'\}}$ by $\langle G(i) \cup G(i'), B(i) \rangle$. The same assumption is made for any Rabin condition $[G, B]_I$.

Δ -Graphs.

A Δ -graph of an ω -word w under \mathcal{A} is a directed graph $\mathcal{G}_w = (V, E)$ where $V = Q \times \mathbb{N}$ and $E = \{ \langle \langle q, l \rangle, \langle q', l+1 \rangle \rangle \in V \times V \mid q, q' \in Q, l \in \mathbb{N}, \langle q, w(l), q' \rangle \in \Delta \}$. By the i -th level, we mean the vertex set $Q \times \{i\}$. Let S be a subset of Q . We call a vertex $v = \langle q, l \rangle$ S -vertex if $q \in S$. When level index is of no importance in the context, we use q and v interchangeably. In particular, by an abuse of notation we write $v \in S$ to mean $v = \langle q, l \rangle$ for some $l \in \mathbb{N}$ and $q \in S$. Δ -graphs for finite words are similarly defined. The length of a finite Δ -graph is the number of levels minus 1. By unit Δ -graphs we mean Δ -graphs of length 1. A unit Δ -graph encodes all possible transitions upon reading a letter. By *width* of \mathcal{G}_w (written $\text{width}(\mathcal{G}_w)$)

we mean the maximum number of pairwise non-intersecting infinite paths in \mathcal{G}_w . Clearly, for any w , $\text{width}(\mathcal{G}_w) \leq |Q|$.

3 Ranking-based Complementation

In this section we introduce ranking-based complementation constructions developed by Klarlund, Kupferman and Vardi [8, 11, 13, 9]. Note that all complexity related notions are parameterized with n and k , but we do not list them explicitly unless required for clarity. We adopt the following naming convention: when we talk about behaviors of a source automaton, a T -condition means an *existential* one (i.e., a path in a Δ -graph that satisfies T), while in the context of complementation, a T -condition means a *universal* one (i.e., every path in a Δ -graph satisfies T).

Ranking-based Complementation Scheme

Let \mathcal{A} be a T -automaton and \mathcal{CA} a purported Büchi automaton that complements \mathcal{A} . An ω -word w is accepted by \mathcal{A} if and only if the Δ -graph \mathcal{G}_w contains an infinite path that satisfies the T -condition. Consequently, w is accepted by \mathcal{CA} if and only if *all* paths in \mathcal{G}_w satisfy the dual co- T condition (for short, \mathcal{G}_w is co- T accepting). Complementation essentially amounts to transforming a *universal* co- T condition into an *existential* Büchi condition. Rankings on Δ -graphs provide a solution; \mathcal{G}_w satisfying a universal co- T condition is precisely captured by the existence of a so-called odd co- T ranking on \mathcal{G}_w . Complementation then reduces to recognition of Δ -graphs that admit odd co- T rankings.

The general scheme goes as follows. Vertices of \mathcal{G}_w are associated with certain values. The association at a level can be viewed as a function with domain Q (with level indices dropped), called *co- T level ranking*. The values in the range of a co- T level ranking are called *co- T ranks*, and the n -tuple of co- T ranks at a level is called a *co- T level rank*. By a *co- T ranking* we mean an ω -sequence of co- T level rankings, each of which is associated with a level in \mathcal{G}_w . Co- T rankings are required to satisfy a *local* property, which holds between every two adjacent levels and is *solely* defined with respect to the unit Δ -graph of the two levels. The local property therefore can be enforced in a step-by-step check by the transitions of \mathcal{CA} . But the local property itself is not enough to ensure that a co- T condition holds universally. A special kind of co- T ranking, called *odd co- T ranking*, is singled out. A co- T ranking is *odd* if and only if every path visits certain vertices (called *odd vertices*) infinitely many times. This *global* property can be captured by a Büchi condition, using the Miyano-Hayashi breakpoint technique for universality (alternation) elimination [16].

Let $\mathcal{A} = \langle Q, Q_0, \Sigma, \Delta, \mathcal{F}^T \rangle$ be a source T -automaton. The complementation algorithm produces a target Büchi automaton $\mathcal{CA} = \langle Q', Q'_0, \Sigma, \Delta', \langle F' \rangle \rangle$. The state set Q' is $2^Q \times 2^Q \times \mathcal{R}$, where for $\langle S, O, g \rangle \in Q'$, S records the reachable states, $O \subseteq S$ records the reachable states that have an obligation to visit *odd* vertices in the future, and g is a guessed co- T level ranking, all at the current level. The transition function $\Delta' : Q' \rightarrow 2^{Q'}$ is defined such that $\Delta'(\langle S, O, g \rangle)$ is

$$\{ \langle \Delta(S, \sigma), \Delta(O, \sigma) \setminus \text{odd}(g'), g' \rangle : g' \in \text{Succ}(g, S, \sigma) \} \quad (O \neq \emptyset), \quad (1)$$

$$\{ \langle \Delta(S, \sigma), \Delta(S, \sigma) \setminus \text{odd}(g'), g' \rangle : g' \in \text{Succ}(g, S, \sigma) \} \quad (O = \emptyset), \quad (2)$$

where $\text{Succ}(g, S, \sigma)$ returns the set of legitimate level rankings provided that the current level ranking is g and the current letter is σ , and $\text{odd}(g')$ gives the set of odd vertices at the level ranked by g' . When \mathcal{CA} reads the letter $w(i)$ at level i with a level ranking f_i

and reachable state set S_i , it nondeterministically guesses a level ranking f_{i+1} such that $f_{i+1} \in \text{Succ}(f_i, S_i, w(i))$. The evolution of both S and O are done by the classic subset construction [20] with the exception that odd vertices are excluded from O (see “ $\backslash \text{odd}(g')$ ” in (1) and (2)). Once O becomes empty, it takes the value of the current S in the next stage (see the second S in (2)). The final state set F' is $2^Q \times \{\emptyset\} \times \mathcal{R}$. This Büchi condition $\langle F' \rangle$ requires that O be cleared infinitely often, which in turn enforces that every path visits odd vertices infinitely many times [16]. It is now clear that Succ represents the local property (being co- T) and \mathcal{F}' captures the global property (being odd).

► **Procedure 1 (Generic Complementation).**

Input: T -automaton $\mathcal{A} = \langle \Sigma, Q, Q_0, \Delta, \mathcal{F}^T \rangle$.

Output: Büchi automaton $\mathcal{CA} = \langle \Sigma, Q', Q'_0, \Delta', \langle F' \rangle \rangle$:

$$\begin{aligned} Q' &= 2^Q \times 2^Q \times \mathcal{R}, & Q'_0 &= Q_0 \times \{\emptyset\} \times \mathcal{R}, \\ \Delta' : Q' &\rightarrow 2^{Q'} \text{ defined as in (1) and (2)}, & F' &= 2^Q \times \{\emptyset\} \times \mathcal{R}. \end{aligned}$$

The state complexity of complementation is $|Q'|$. For every instantiation shown below, $|R|$ dominates $2^{|Q|}$ and hence the complexity is $O(|R|)$.

We now show complementation constructions for Büchi, GB and Streett, with the corresponding co- T rankings being co-Büchi, GC and Rabin, respectively. We use \mathcal{D}^T to denote the set of T -ranks, \mathcal{R}^T the set of T level rankings and \mathcal{L}^T the set of level ranks. Clearly, $|\mathcal{R}^T| = |\mathcal{L}^T|$.

Büchi Complementation

Let $\mathcal{A} = \langle Q, Q_0, \Sigma, \Delta, \langle F \rangle \rangle$ be a Büchi automaton. \mathcal{G}_w is co-Büchi accepting if every path in \mathcal{G}_w visits F -vertices finitely often. Let \mathcal{D}^{CB} (the set of co-Büchi ranks) be $[2n + 1]$.

► **Definition 1 (Co-Büchi Ranking).** A co-Büchi ranking on \mathcal{G}_w is a function $f : V \rightarrow \mathcal{D}^{\text{CB}}$ such that:

- 1.1 for all vertices $v \in V$, if $f(v) \in [2n]^{\text{odd}}$, then $v \notin F$;
- 1.2 for all edges $\langle v, v' \rangle \in E$, $f(v) \geq f(v')$.

A vertex $v \in V$ is *odd* if $f(v) \in [2n]^{\text{odd}}$. A co-Büchi ranking f is *odd* if every path in \mathcal{G}_w visits infinitely many odd vertices. A path ρ stabilizes at a rank r if $(\exists i \in \mathbb{N})(\forall j \geq i), f(\rho(j)) = f(\rho(i)) = r$ and the smallest such i is called the *stabilization point* of ρ . If \mathcal{G}_w admits an odd co-Büchi ranking f , then by (1.2) every path eventually stabilizes at an odd rank. Then by (1.1), every path eventually does not visit F -vertices; that is, \mathcal{G}_w is co-Büchi accepting.

Conversely, if \mathcal{G}_w is co-Büchi accepting, then an odd co-Büchi ranking can be constructed through a series of graph transformations. Let $\mathcal{G}_0 = \mathcal{G}_w$. Vertices with only a finite number of descendants are called *finite*. Vertices that are not F -vertices and have no F -vertices as their descendants are called *F -free*. At stage 0, we assign all finite vertices rank 0 and remove them, obtaining \mathcal{G}_1 , in which there is no finite vertices. Because \mathcal{G}_0 is co-Büchi accepting, there must exist in \mathcal{G}_1 an F -free vertex; otherwise we can select a path on which F -vertices occur infinitely often. We assign all F -free vertices rank 1 and remove them too, obtaining \mathcal{G}_2 . Now some vertices in \mathcal{G}_2 are finite due to the removal of F -free vertices in stage 0. We repeat this process in the following manner: at the first phase of stage i , we assign even rank $2i$ to finite vertices and remove them; at the second phase, we assign F -free vertices odd rank $2i + 1$ and remove them. By F -freeness, removing F -free vertices from \mathcal{G}_{2i+1} gets rid of at least one infinite path, and hence $\text{width}(\mathcal{G}_{2i+2}) < \text{width}(\mathcal{G}_{2i})$. Therefore, this process

terminates at a stage $j \leq n$. In the following summary, by $\mathcal{G} \setminus V$ we mean removing from \mathcal{G} all vertices in V and their incoming and outgoing edges.

► **Procedure 2 (Co-Büchi Ranking Assignment).**

Input: a co-Büchi accepting \mathcal{G}_0 . Output: a co-Büchi ranking f . Repeat for $i \in [0..n]$ if $\mathcal{G}_{2i} \neq \emptyset$.

- | | | | |
|-----|---|-----|---|
| 2.1 | (a) $V_{2i} = \{v \in V \mid v \text{ is finite in } \mathcal{G}_{2i}\};$ | 2.2 | (a) $V_{2i+1} = \{v \in V \mid v \text{ is } F\text{-free in } \mathcal{G}_{2i+1}\};$ |
| | (b) $f(v) = 2i$ for $v \in V_{2i};$ | | (b) $f(v) = 2i + 1$ for $v \in V_{2i+1};$ |
| | (c) $\mathcal{G}_{2i+1} = \mathcal{G}_{2i} \setminus V_{2i}.$ | | (c) $\mathcal{G}_{2i+2} = \mathcal{G}_{2i+1} \setminus V_{2i+1}.$ |

► **Lemma 2 ([11]).** \mathcal{G}_w is co-Büchi accepting if and only if \mathcal{G}_w admits an odd co-Büchi ranking.

We have $|\mathcal{D}^{\text{CB}}| = O(n)$, and hence $|\mathcal{R}^{\text{CB}}| = (O(n))^n = 2^{O(n \lg n)}$.

GB Complementation

Let $\mathcal{A} = \langle Q, Q_0, \Sigma, \Delta, \langle B \rangle_I \rangle$ be a generalized Büchi automaton. GC ranking is meant to be used for GB complementation. A \mathcal{G}_w is GC accepting if for every path ρ in \mathcal{G}_w there exists $j \in I$ such that ρ only visits $B(j)$ -vertices finitely often. Let $\mathcal{D}^{\text{GC}} = ([2n]^{\text{odd}} \times I) \cup [2n+1]^{\text{even}}$ be the set of GC ranks. We refer to values in $[2n]^{\text{odd}} \times I$ as *odd* ranks, and values in $[2n+1]^{\text{even}}$ as *even* ranks. For an odd GC rank $\langle t, u \rangle$, we call t numeric rank (r -rank) and u index rank (h -rank). Even GC ranks are just numeric ranks. The greater-than and less-than orders on GC ranks are solely defined on r -ranks. For example, $\langle t, u \rangle > \langle t', u' \rangle$ (or $\langle t, u \rangle > t'$, $t > \langle t', u' \rangle$) if and only if $t > t'$. This definition is sound with respect to its usage in this paper; as shown below, we never need to compare two odd GC ranks having the same r -rank but different h -ranks.

► **Definition 3 (GC Ranking).** A GC ranking on \mathcal{G}_w is a function $f : V \rightarrow \mathcal{D}^{\text{GC}}$ such that:

- 3.1 for every vertex $v \in V$, if $f(v) = \langle 2i+1, j \rangle$ for some $j \in I$, then $v \notin B(j)$;
 3.2 for every edge $\langle v, v' \rangle \in E$, $f(v) \geq f(v')$.

A vertex v is called *odd* (resp. *even*) if $f(v) \in [2n]^{\text{odd}} \times I$ (resp. $f(v) \in [2n+1]^{\text{even}}$). A GC ranking f is *odd* if every path in \mathcal{G}_w visits infinitely many odd vertices. Note that (3.2) implies that if two adjacent odd vertices have the same r -rank, then they have the same h -rank. As in Büchi complementation, if \mathcal{G}_w admits an odd GC ranking, then every path eventually stabilizes at an odd GC rank $\langle t, j \rangle$, and from the stabilization point on never visits $B(j)$ -vertices. Therefore, \mathcal{G}_w is GC accepting.

Conversely, if \mathcal{G}_w is GC accepting, then we can find a GC ranking by a series of graph transformations as in Büchi complementation. Each stage has two phases. We begin stage i with \mathcal{G}_{2i} ($\mathcal{G}_0 = \mathcal{G}_w$). In the first phase, finite vertices receive even rank $2i$ and are removed, resulting in \mathcal{G}_{2i+1} . Thanks to GC condition, if \mathcal{G}_{2i+1} is not empty, then for some $j \in I$ and $v \in V$, v is $B(j)$ -free. In the second phase, those $B(j)$ -free vertices receive odd rank $\langle 2i+1, j \rangle$ and are removed, producing \mathcal{G}_{2i+2} . This procedure repeats for all $i \in [0..n]$ unless \mathcal{G}_{2i} is empty. The termination condition is justified by $\text{width}(\mathcal{G}_{2i+2}) < \text{width}(\mathcal{G}_{2i})$, just as before.

► Procedure 3 (GC Ranking Assignment).

Input: a GC accepting \mathcal{G}_0 . Output: a GC ranking f . Repeat for $i \in [0..n]$ if $\mathcal{G}_{2i} \neq \emptyset$.

- 3.1 (a) $V_{2i} = \{v \in V \mid v \text{ is finite in } \mathcal{G}_{2i}\};$ 3.2 (a) $V_{2i+1} = \{v \in V \mid v \text{ is } B(j)\text{-free in } \mathcal{G}_{2i+1}\}$ for a $j \in I$ such that $B(j)$ -free vertices exist;
 (b) $f(v) = 2i$ for $v \in V_{2i};$ (b) $f(v) = \langle 2i + 1, j \rangle$ for $v \in V_{2i+1};$
 (c) $\mathcal{G}_{2i+1} = \mathcal{G}_{2i} \setminus V_{2i}.$ (c) $\mathcal{G}_{2i+2} = \mathcal{G}_{2i+1} \setminus V_{2i+1}.$

In (3.2), it does not matter which $j \in I$ is chosen. But this flexibility plays an important role in our Streett complementation construction (see Procedure 5).

► **Lemma 4** ([12]). \mathcal{G}_w is GC accepting if and only if \mathcal{G}_w admits an odd GC ranking.

We have $|\mathcal{D}^{\text{GC}}| = O(nk)$, and hence $|\mathcal{R}^{\text{GC}}| = (O(nk))^n = 2^{O(n \lg nk)}$.

Streett Complementation

Let $\mathcal{A} = \langle Q, Q_0, \Sigma, \Delta, \langle G, B \rangle_I \rangle$ be a Streett automaton. Rabin ranking is meant for Streett complementation. Let us first examine the simple case where $k = 1$, i.e., every path satisfies $[G(1), B(1)]$. Easily seen, \mathcal{G}_w admits a co-Büchi ranking, and hence we can instantiate Procedure 1 with \mathcal{R} being co-Büchi level rankings (which are also GC level rankings with index size 1). The only modification needed is to enforce that every path visits $G(1)$ -vertices, which can be easily realized by a Büchi accepting condition (see the definition of $\langle F' \rangle$ in Procedure 1). This simple procedure fails for $k > 1$, because a path visiting a finite number of $B(j)$ -vertices may not have to visit infinitely many $G(j)$ -vertices; it just satisfies $[G(j'), B(j')]$ for $j' \neq j$. Nevertheless, if we could find a way to reduce the number of Rabin pairs one by one, eventually the simple scenario has to occur. The idea in [14] is to use GC rankings to approximate Rabin accepting behaviors step by step until finally obtaining the precise characterization. As a result, Rabin ranks are tuples of GC ranks, considerably more sophisticated than GC ranks. We first put aside the formal definition of Rabin rankings and show how a Rabin ranking can be obtained provided \mathcal{G}_w is Rabin accepting. Once again, this is done through a series of graph transformations.

► Procedure 4 (Rabin Ranking Assignment).

Input: a Rabin accepting \mathcal{G}_0 . Output: a Rabin ranking f . Repeat for $i \in [0..k]$ if $\mathcal{G}_i \neq \emptyset$.

- 4.1 Assign \mathcal{G}_i a GC ranking gc_{i+1} .
 4.2 Remove all vertices v if $\text{gc}_{i+1}(v)$ is even.
 4.3 Remove all edges $\langle v, v' \rangle$ if $\text{gc}_{i+1}(v) > \text{gc}_{i+1}(v')$.
 4.4 Remove all edges $\langle v, v' \rangle$ if $\text{gc}_{i+1}(v)$ is odd with index j and v is a $G(j)$ -vertex.
 4.5 $f(v) = \langle \text{gc}_1(v), \dots, \text{gc}_{i+1}(v) \rangle$ iff v is removed from \mathcal{G}_i .

Obviously, if \mathcal{G}_w is Rabin accepting for a Rabin condition $[G, B]_I$, then it is also GC accepting for the GC condition $[B]_I$. By Lemma 4, a GC ranking gc_1 exists for \mathcal{G}_0 , which justifies Step (4.1) at stage 0. Steps (4.2)-(4.3) may break up \mathcal{G}_0 into a collection of graph components (in the undirected sense). Let \mathcal{C} be such a component. Steps (4.2)-(4.3) ensure that vertices in \mathcal{C} have the same odd rank with some index $j \in I$, and hence all are $B(j)$ -free. Step (4.4), deleting all outgoing edges from $G(j)$ -vertices, may further break up \mathcal{C} into more components. In particular, any infinite path is destroyed (i.e., broken into a collection of finite paths) if the path satisfies $[G(j), B(j)]$ (i.e., visiting infinitely many $G(j)$ -vertices but only finitely many $B(j)$ -vertices). Let $\mathcal{C}' \subseteq \mathcal{C}$ be a resulting component after Step (4.4). As a result, \mathcal{C}' should satisfy the *reduced* Rabin condition $[G, B]_{I \setminus \{j\}}$, and hence the *reduced* GC condition $[B]_{I \setminus \{j\}}$.

So after stage 0, \mathcal{G}_1 is composed of a collection of pairwise disjoint components, each of which satisfies a Rabin condition whose cardinality is at most $k - 1$. Precisely speaking, at the beginning of each stage $i \geq 1$, \mathcal{G}_i is composed of a collection of pairwise disjoint components, and at Step (4.1), gc_{i+1} is obtained by *independently* assigning each component in \mathcal{G}_i a GC ranking according to the *reduced* GC condition the component satisfies. By induction, at stage $i \geq 1$, vertices in each component in \mathcal{G}_i have been assigned the same tuple of odd GC ranks of length i and each component satisfies a Rabin condition whose cardinality is at most $k - i$. It follows that the procedure terminates and each vertex in \mathcal{G}_w eventually gets a tuple of GC ranks of length at most $k + 1$. Note that the last GC rank in a tuple is always an even GC rank (r -rank).

Let \mathcal{D}^R denote the set of Rabin ranks of the form $\langle \langle r_1, i_1 \rangle, \dots, \langle r_m, i_m \rangle, r_{m+1} \rangle$ ($m \leq k$). Ordering relations ($<_m, \leq_m, >_m, \geq_m, =_m$) on Rabin ranks are defined to be the standard lexicographical extension (up to m -th component) of orderings on GC ranks. For a Rabin rank γ of the above form, the *index projection* (or the *h-projection*) of γ , written $\text{Proj}_h \gamma$, is $\langle i_1, \dots, i_m \rangle$ and the *numeric projection* (or the *r-projection*) of γ , written $\text{Proj}_r \gamma$, is $\langle r_1, \dots, r_{m+1} \rangle$. With respect to a given function $f : V \rightarrow \mathcal{D}^R$, the *width* of $v \in V$ is the length of $f(v)$, denoted by $|v|_f$ (or $|v|$, when f is clear from the context). We say that v is *odd* (called *happy* in [14]) if $|v| > 1$ and v is a $G(|v| - 1)$ -vertex. We arrive at the formal definition of Rabin rankings.

► **Definition 5** (Rabin Ranking). A Rabin ranking is a function $f : V \rightarrow \mathcal{D}^R$ satisfying the following conditions.

- 5.1 For every vertex $v \in V$ with $|v| = m + 1 \geq 2$ and $\alpha = \text{Proj}_h f(v)$, we have
- a. for $i \in [1..m]$, $v \notin G(\alpha[i])$;
 - b. for $i \in [1..m]$, $v \notin B(\alpha[i])$.
- 5.2 For every edge $\langle v, v' \rangle \in E$ with $|v| = m + 1$, $|v'| = m' + 1$ and $m'' = \min(m, m')$, we have
- a. $f(v) \geq_{m''} f(v')$;
 - b. $f(v) \geq_{m''+1} f(v')$, or v is odd.

A Rabin ranking is *odd* if every path in \mathcal{G}_w visits infinitely many odd vertices.

► **Lemma 6** ([14]). \mathcal{G}_w is Rabin accepting if and only if \mathcal{G}_w admits an odd Rabin ranking.

We have $|\mathcal{D}^R| = (O(nk))^{k+1}$ and hence $|\mathcal{R}^R| = ((O(nk))^{k+1})^n = (nk)^{O(nk)} = 2^{O(nk \lg nk)}$.

4 Improved Streett Complementation

The above construction requires $2^{O(nk \lg nk)}$ state blow-up [14], which is substantially larger than the lower bound in [3]. In the extreme case of $k = O(2^n)$, the construction is double exponential in n . Intuitively, the larger the k , the more overlaps between $B(i)$'s and between $G(i)$'s ($i \in I$). A natural question is: can all Rabin pairs $[G(i), B(i)]$ independently impose behaviors on a Rabin accepting \mathcal{G}_w ? We showed in [2] that in Rabin complementation we can build a Streett accepting \mathcal{G}_w for which no Streett pair $\langle G(i), B(i) \rangle$ is redundant. We observed the opposite in Streett complementation; the larger the k , the higher the correlation between infinite paths that satisfy $[G, B]_I$. By exploiting this correlation, we can walk in *big* steps in approximating Rabin accepting behaviors using GC rankings. As a result, our Rabin ranks are tuples of GC ranks of length at most $\mu = \min(n, k)$. This simple but crucial observation leads to a significant improvement on the construction complexity. We elaborate on this below.

The first idea is that at Step (4.4) in stage i , in a component \mathcal{C} of \mathcal{G}_i , instead of removing all outgoing edges from $G(j)$ -vertices, we can remove all outgoing edges from $G(j')$ -vertices

for all j' such that $B(j') \subseteq B(j)$. Let $J = \{j' \in I \mid B(j') \subseteq B(j)\}$. Since vertices in \mathcal{C} are $B(j)$ -free, they are also $B(j')$ -free for any $j' \in J$. Recall that Step (4.4) is to break all infinite paths that satisfy $[G(j), B(j)]$ so that in each resulting component we have a simpler Rabin condition to satisfy. If an infinite path in \mathcal{C} satisfies $[G, B]_J$, then removing all outgoing edges from $G(j')$ -vertices (for $j' \in J$) certainly serves the same purpose, and moreover, any resulting component only needs to satisfy a Rabin condition whose cardinality is $|J|$ less than at the beginning of stage i .

The second idea is that at Step (4.1) in stage i , we can assign special GC rankings to components in \mathcal{G}_i . Recall that a GC ranking is obtained by a series of graph transformations too. In Step (3.2), we assign and remove $B(j)$ -free vertices for some $j \in I$. In fact any fixed $j \in I$ is sufficient as long as $B(j)$ -free vertices exist. Therefore, we can choose a j such that not only $B(j)$ -free vertices exist, but also for any other $j' \in I$, $B(j') \not\subseteq B(j)$, if $B(j')$ -free vertices also exist. Intuitively, we prefer a j such that $B(j)$ is minimal (with respect to set inclusion) because more vertices would be $B(j)$ -free and subject to removal.

We refine those ideas by taking into account the history of GC rankings. In stage i , right before Step (4.1), vertices in \mathcal{G}_i were assigned a tuple of GC ranks of length i . Consider a component $\mathcal{C} \subseteq \mathcal{G}_i$. No vertices in \mathcal{C} received an even GC rank in stage $i' \in [0..i)$, because otherwise they were already removed by Step (4.2) in that stage. Also all vertices in \mathcal{G}_i received the same odd GC rank in each stage $i' \in [0..i)$, for otherwise Step (4.3) in stage i' would have broken the component. Now let $\langle \langle r_1, j_1 \rangle, \dots, \langle r_i, j_i \rangle \rangle$ be the tuple that has been assigned to all vertices in \mathcal{C} . Let $B' = \cup_{t \in [1..i]} B(j_t)$ and $J' = \{j' \in I \mid B(j') \subseteq B'\}$. So all vertices in \mathcal{C} are B' -free. When we assign GC rankings for \mathcal{C} , in each stage at Step (3.2) (in Procedure 3), we choose a $j \in I \setminus J'$ such that (1) $B(j)$ -free vertices exist, (2) $B(j) \not\subseteq B'$ (we say $B(j)$ is not *covered* by B'), and (3) no $B(j')$ -free vertices exist for any other $j' \in I \setminus J'$ with $B' \cup B(j') \subseteq B' \cup B(j)$. In other words, we choose a j such that not only we can find $B(j)$ -free vertices, but also $B' \cup B(j)$ minimally extends B' . Now let \mathcal{C}' be a component right before Step (4.4) is taken and let $\langle \langle r_1, j_1 \rangle, \dots, \langle r_{i+1}, j_{i+1} \rangle \rangle$ be the tuple of ranks that has been assigned to all vertices in \mathcal{C}' (for the same reason as before, all vertices in \mathcal{C}' have received the same sequence of odd GC ranks). Let $B'' = B' \cup B(j_{i+1})$ and $J'' = \{j'' \in I \mid B(j'') \subseteq B''\}$. In Step (4.4) we remove all outgoing edges of $G(j'')$ -vertices if $B(j'') \subseteq B''$. This deletion destroys all infinite paths that satisfy $[G, B]_{J''}$ because all vertices in \mathcal{C}' are B'' -free. Let \mathcal{C}'' be a resulting component and ϱ an infinite path in \mathcal{C}'' . Then ϱ only needs to satisfy $[G, B]_{I \setminus J''}$.

Now let us assume that we have incorporated the above ideas into Procedure 4 and obtained a new Rabin rank $\gamma = \langle \langle r_1, j_1 \rangle, \dots, \langle r_m, j_m \rangle, r_{m+1} \rangle$. Let $\alpha = \text{Proj}_h \gamma$. The *non-covering* condition stated above requires α to satisfy:

$$\forall i \in [1..m] \ B(\alpha[i]) \not\subseteq \cup_{j=1}^{i-1} B(\alpha[j]), \quad (3)$$

which implies $|\alpha| \leq n$. By definition, $|\alpha| \leq k$ and so we have $|\alpha| \leq \mu$ and $|\gamma| \leq \mu + 1$. From now on we switch to terms $\mu\text{R ranks}$ (i.e., *minimal Rabin ranks*), $\mu\text{R rankings}$, $\mu\text{R level rankings}$ and $\mu\text{R level ranks}$. Their precise definitions are to be given below. We define two functions $\text{Cover} : I^* \rightarrow 2^I$ and $\text{Mini} : I^* \rightarrow 2^I$ to formalize the intuition of *minimal extension*. Cover maps tuples of indices to subsets of I such that

$$\text{Cover}(\alpha) = \{j \in I \mid B(j) \subseteq \cup_{i=1}^{|\alpha|} B(\alpha[i])\}.$$

Note that $\text{Cover}(\epsilon) = \emptyset$. Mini maps tuples of indices to subsets of I such that $j \in \text{Mini}(\alpha)$ if and only if $j \in I \setminus \text{Cover}(\alpha)$ and

$$\forall j' \in I \setminus \text{Cover}(\alpha) \ (j' \neq j \rightarrow B(j') \cup \text{Cover}(\alpha) \not\subseteq B(j) \cup \text{Cover}(\alpha)), \quad (4)$$

$$\forall j' \in I \setminus \text{Cover}(\alpha) (j' < j \rightarrow B(j') \cup \text{Cover}(\alpha) \neq B(j) \cup \text{Cover}(\alpha)). \quad (5)$$

$\text{Mini}(\alpha)$ consists of choices of indices to *minimally* enlarge $\text{Cover}(\alpha)$; ties (with respect to set inclusion) are broken by numeric minimality (Condition (5)). Before introducing μR ranking assignment, we need a new GC ranking assignment which takes a tuple of I -indices as an additional input. We call so obtained GC rankings (resp. GC ranks) μGC rankings (resp. μGC ranks).

► **Procedure 5 (μGC Ranking Assignment).**

Input: a GC accepting \mathcal{G}_0 , a tuple of I -indices α . Output: a μGC ranking f .

Repeat for $i \in [0..n]$ if $\mathcal{G}_{2i} \neq \emptyset$.

- | | | | |
|-----|---|-----|--|
| 5.1 | (a) $V_{2i} = \{v \in V \mid v \text{ is finite in } \mathcal{G}_{2i}\};$ | 5.2 | (a) $V_{2i+1} = \{v \in V \mid v \text{ is } B(j)\text{-free in } \mathcal{G}_{2i+1}\}$ for a $j \in \text{Mini}(\alpha)$ such that $B(j)$ -free vertices exist; |
| | (b) $f(v) = 2i$ for $v \in V_{2i};$ | | (b) $f(v) = \langle 2i + 1, j \rangle$ for $v \in V_{2i+1};$ |
| | (c) $\mathcal{G}_{2i+1} = \mathcal{G}_{2i} \setminus V_{2i}.$ | | (c) $\mathcal{G}_{2i+2} = \mathcal{G}_{2i+1} \setminus V_{2i+1}.$ |

As in GC ranking assignment, in (5.2) there maybe more than one j such that $B(j)$ -free vertices exist, and it does not matter which one we choose. But in case $\text{Mini}(\alpha)$ is a singleton, we have a unique j at all stages, essentially synchronizing all h -ranks in the μGC ranks obtained. This synchronization is crucial in our construction for parity complementation (see Section 6).

► **Procedure 6 (μR Ranking Assignment).**

Input: a Rabin accepting \mathcal{G}_0 . Output: a μR ranking f . Repeat for $i \in [0..\mu]$ if $\mathcal{G}_i \neq \emptyset$.

- 6.1 Assign \mathcal{G}_i a μGC ranking gc_{i+1} .
- 6.2 Remove all vertices $v \in V$ if $\text{gc}_{i+1}(v)$ is even.
- 6.3 Remove all edges $\langle v, v' \rangle \in E$ if $\text{gc}_{i+1}(v) > \text{gc}_{i+1}(v')$.
- 6.4 Remove all edges $\langle v, v' \rangle \in E$ if $v \in G(t)$ for some $t \in \text{Cover}(\text{Proj}_h(\langle \text{gc}_1, \dots, \text{gc}_{i+1} \rangle))$.
- 6.5 $f(v) = \langle \text{gc}_1(v), \dots, \text{gc}_{i+1}(v) \rangle$ iff v is removed from \mathcal{G}_i .

Note that Step (6.1) actually means that Procedure 5 is called upon for every component $\mathcal{C} \subset \mathcal{G}_i$, with the corresponding α being $\text{Proj}_h(\langle \text{gc}_1(v), \dots, \text{gc}_i(v) \rangle)$ for some $v \in \mathcal{C}$ (α is well-defined since all vertices in \mathcal{C} have received the same sequence of μGC ranks). It is time to formally define μR ranking. Let f be a function $V \rightarrow (\mathcal{D}^{\text{GC}})^{\mu+1}$. We say that v is *odd* if $|v| > 1$ and v is a $G(t)$ -vertex for some $t \in \text{Cover}(\alpha[1..|v| - 1])$ where $\alpha = \text{Proj}_h f(v)$.

► **Definition 7 (μR Ranking).** A μR ranking is a function $f : V \rightarrow (\mathcal{D}^{\text{GC}})^{\mu+1}$ satisfying the following conditions.

- 7.1 For every vertex $v \in V$ with $|v| = m + 1 \geq 2$ and $\alpha = \text{Proj}_h f(v)$, we have
 - a. for $i \in [1..m]$, $v \notin G(t)$ for $t \in \text{Cover}(\alpha[1..i]);$
 - b. for $i \in [1..m]$, $v \notin B(t)$ for $t \in \text{Cover}(\alpha[1..i]);$
 - c. for $i \in [1..m]$, $\alpha[i] \in \text{Mini}(\alpha[1..i]).$
- 7.2 For every edge $\langle v, v' \rangle \in E$ with $|v| = m + 1$, $|v'| = m' + 1$ and $m'' = \min(m, m')$, we have
 - a. $f(v) \geq_{m''} f(v');$
 - b. $f(v) \geq_{m''+1} f(v')$, or v is odd.

A μR ranking is *odd* if every infinite path in \mathcal{G}_w visits infinitely many odd vertices.

► **Lemma 8.** \mathcal{G}_w is Rabin accepting if and only if \mathcal{G}_w admits an odd μR ranking.

5 Complexity

In this section we analyze the complexity of our construction. As shown below, all complexity related notions X are also parameterized with B (besides n and k). Still, we choose to list some or all of them when clarity is needed. In particular, we use $|X|$ to abbreviate $|X(n, k)| = \max_B |X(B, n, k)|$.

Let $\mathcal{D}^{\mu R}$ be the set of μR ranks that can be produced by Procedure 6. Formally, $\mathcal{D}^{\mu R} = \cup_f \text{range}(f)$ where f ranges over all possible outputs of Procedure 6. In a similar manner, we define $\mathcal{R}^{\mu R}$ and $\mathcal{L}^{\mu R}$. We have $|\mathcal{R}^{\mu R}| = |\mathcal{L}^{\mu R}|$. We note that these notions are defined differently from their counterparts in Section 3; due to two kinds of correlations (which we refer to as *horizontal* and *vertical* correlation), $|\mathcal{L}^{\mu R}|$ is much smaller than $|\mathcal{D}^{\mu GC}|^{n(\mu+1)}$. We view $\mathcal{L}^{\mu R}$ as a set of $n \times (\mu + 1)$ matrices of μGC ranks and carry out a further simplification. Let $\mathcal{M}^{\mu R}$ be a set of $n \times \mu$ matrices obtained from $\mathcal{L}^{\mu R}$ by the following mapping: each $n \times (\mu + 1)$ matrix M is mapped to an $n \times \mu$ matrix M' by (a) deleting from M even ranks at the end of each row, (b) changing odds rank of the form $\langle 2i - 1, j \rangle$ to $\langle i, j \rangle$, and (c) aligning each row to length μ by filling $\langle 1, 0 \rangle$'s. Clearly, $|\mathcal{L}^{\mu R}| \leq n^n \cdot |\mathcal{M}^{\mu R}|$; the factor n^n suffices to compensate (a), the deletion of even ranks, and (b) and (c) have no effect to the cardinality (for (b), $i \rightarrow 2i - 1$ is one-to-one from $[1..n]$ onto $[2n]^{\text{odd}}$). Let $M \in \mathcal{M}^{\mu R}$ be called μR -matrices. We write $\text{Proj}_r M$ and $\text{Proj}_h M$ to mean, respectively, the projection of M on numeric ranks (called an r -matrix) and on index ranks (called an h -matrix). Let $\mathcal{M}^r = \text{Proj}_r \mathcal{M}^{\mu R}$, $\mathcal{M}^h = \text{Proj}_h \mathcal{M}^{\mu R}$, and \mathcal{D}^r and \mathcal{D}^h be the sets of rows occurring in matrices in \mathcal{M}^r and \mathcal{M}^h , respectively. Obviously, we have $|\mathcal{M}^{\mu R}| \leq |\mathcal{M}^r| \cdot |\mathcal{M}^h|$ and $|\mathcal{M}^h| \leq (|\mathcal{D}^h|)^n$.

► **Example 9** (μR -Matrix). Let us consider a case where $n = 3$, $k = 3$, and $Q = \{q_0, q_1, q_2\}$. Below we show that a μR level rank f corresponds to a μR -matrix M , which projects to M_r and M_h .

$$\begin{array}{c}
 \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix} \\
 Q
 \end{array}
 \begin{array}{c}
 \left| \begin{array}{ccc} \langle 1, 2 \rangle & \langle 1, 3 \rangle & 4 \\ \langle 1, 2 \rangle & \langle 3, 1 \rangle & 2 \\ \langle 1, 2 \rangle & \langle 3, 1 \rangle & \langle 3, 3 \rangle \end{array} \right| \\
 f
 \end{array}
 \begin{array}{c}
 \left| \begin{array}{ccc} \langle 1, 2 \rangle & \langle 1, 3 \rangle & \langle 1, 0 \rangle \\ \langle 1, 2 \rangle & \langle 2, 1 \rangle & \langle 1, 0 \rangle \\ \langle 1, 2 \rangle & \langle 2, 1 \rangle & \langle 2, 3 \rangle \end{array} \right| \\
 M
 \end{array}
 \begin{array}{c}
 \left| \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \end{array} \right| \\
 M_r
 \end{array}
 \begin{array}{c}
 \left| \begin{array}{ccc} 2 & 3 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 3 \end{array} \right| \\
 M_h
 \end{array}
 \end{array}$$

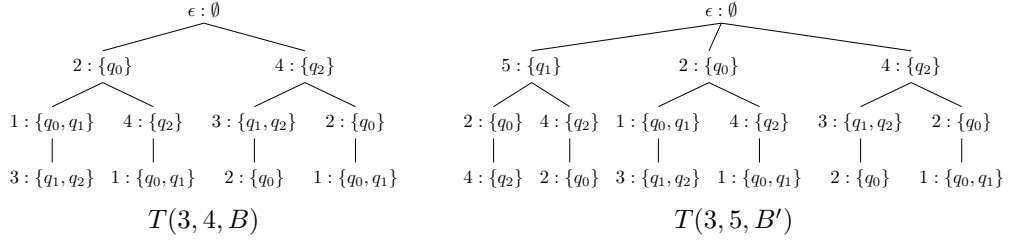
Bounding $|\mathcal{M}^h|$

It turns out that we only need to exploit a horizontal correlation to bound $|\mathcal{M}^h|$. Recall that each $\alpha \in I^*$ names a subset of Q , namely $\text{Cover}(\alpha)$. The idea is to order all α that could occur in \mathcal{D}^h into a tree structure. Consider an unordered tree where the root is labeled by ϵ and each non-root node is labeled by an index in I . With little confusion, we identify a node α with the path from the root to α and represent α by the sequence of indices on the path. So a non-root node α has $\alpha[|\alpha|]$ as its label and names $\text{Cover}(\alpha)$. We arrive at the following important notion.

► **Definition 10** (Increasing Tree of Sets (ITS)). An ITS $T(n, k, B)$ is an unordered I -labeled tree (except the root which is labeled by ϵ) such that

10.1 A non-root node α exists in $T(n, k, B)$ iff $\forall i \in [1..|\alpha|]$, $\alpha[i] \in \text{Mini}(\alpha[1..i])$.

Property (10.1) succinctly encodes three important features of ITS. First, an ITS is maximal in the sense that no node can be added. Second, if β is a direct child of α , then β must name at least one new state that has not been named by α . Third, the new contributions



■ **Figure 2** Two ITS in Example 11.

by β cannot be covered by contributions made by any another sibling β' . In particular, if more than one sibling can make the same contribution, then the one with the smallest index is selected. It follows that each tuple of n , k and B uniquely determines $T(n, k, B)$ (in the unordered sense). Note that the height of $T(n, k, B)$ (the length of the longest path in $T(n, k, B)$) is bounded by μ .

► **Example 11** (ITS). Consider $n = 3$, $k = 4$, $Q = \{q_0, q_1, q_2\}$, $B : [1..4] \rightarrow 2^Q$ and $B' : [1..5] \rightarrow 2^Q$,

$$B(1) = \{q_0, q_1\}, \quad B(2) = \{q_0\}, \quad B(3) = \{q_1, q_2\}, \quad B(4) = \{q_2\},$$

and B' extends B with $B'(5) = \{q_1\}$. $T(3, 4, B)$ and $T(3, 5, B')$ are given in Figure 2. For clarity, for each non-root node α , we also list $B(\alpha[|\alpha|])$ as the set label of α . In $T(3, 4, B)$, neither $\{q_0, q_1\}$ nor $\{q_1, q_2\}$ can appear at height 1, because $\{q_0, q_1\}$ covers $\{q_0\}$ and $\{q_1, q_2\}$ covers $\{q_2\}$. The leftmost node at the bottom level is labeled by $\{q_1, q_2\}$ instead of by $\{q_2\}$ due to the index minimality requirement. For the same reason, in $T(3, 5, B')$, we have nodes $\langle 2, 1 \rangle$, $\langle 2, 4, 1 \rangle$ and $\langle 4, 2, 1 \rangle$ all labeled with $\{q_0, q_1\}$, and nodes $\langle 2, 1, 3 \rangle$ and $\langle 4, 3 \rangle$ all labeled with $\{q_1, q_2\}$.

It is easily seen that Property (7.1c) corresponds exactly to Property (10.1). So a one-to-one correspondence exists between non-root nodes in $T(n, k, B)$ and elements in $\mathcal{D}^h(n, k, B)$. Let $|T(n, k, B)|$ denote the number of non-root nodes in $T(n, k, B)$ and $H(n, k) = \max_B |T(n, k, B)|$. Clearly, we have $|\mathcal{M}^h| \leq (H(n, k))^n$.

► **Lemma 12.** $H(n, k) = 2^{O(k \lg k)}$ for $k = O(n)$ and $H(n, k) = 2^{O(n \lg n)}$ for $k = \omega(n)$.

Bounding $|\mathcal{M}^r|$

Here we need to exploit both horizontal and vertical correlations. We show that every $n \times \mu$ r -matrix induces a 2^Q -labeled ordered tree with at most n leaves and with height at most μ . Such a tree is called an $n \times \mu$ tree. We bound $|\mathcal{M}^r|$ by counting the number of $n \times \mu$ trees.

Let M be an r -matrix. Since M comes from a μR level rank, M is associated with vertices at a level. To facilitate the discussion below, we use term *states* to specifically mean those vertices at the level where M is associated with, and use term *vertices* just as before. By rank i we simply mean a number i in M , which corresponds to the numeric μGC rank $2i - 1$.

Let us first consider ranks in column 1 of M . A state q being ranked with an odd μGC rank means that at certain stage of μGC ranking assignment, q becomes $B(j)$ -free for some $j \in I$, which implies that there exists an infinite path starting from q (recall that all finite

vertices have been removed before this odd μ GC rank is assigned). If two states q, q' are ranked with different odd μ GC ranks, say q with $\langle 2i - 1, j \rangle$ and q' with $\langle 2i' - 1, j' \rangle$ where $i > i'$, then there exist two infinite paths ϱ and ϱ' such that ϱ starts from q , ϱ' starts from q' , and ϱ and ϱ' never intersect. This is due to the nature of μ GC ranking assignment; $\langle 2i - 1, j \rangle$ is assigned to some $B(j)$ -free vertices only after those $B(j')$ -free vertices with odd μ GC rank $\langle 2i' - 1, j' \rangle$ have been removed.

Note that it is perfectly possible that an infinite path starting from q intersects another infinite path starting from q' . But a maximal subset $S^{(1)}$ of states, all with the same rank, called a *cell* at column 1, should “own” at least one *private* infinite path that does not intersect the private paths owned by any other cells at column 1. We call a path *named* if it is owned by a cell. Let $m^{(1)}$ be the maximum rank in column 1, and note that not all ranks in $[1..m^{(1)}]$ necessarily appear in column 1. But again, by the way μ GC ranking assignment is carried out, for each non-occurring rank, at least one private infinite path exists, which is called *hidden* and viewed as being owned by \emptyset . Easily seen now, each rank in $[1..m^{(1)}]$ corresponds to a non-empty set of private infinite paths.

In general, a *cell* at column l is a maximal subset of states, each of which is assigned the same tuple of ranks up to column l . Consider a cell $S^{(l)} = \{q_{i_1}, \dots, q_{i_j}\}$ at column l . Let $m^{(l+1)} = \max\{M[i_1, l+1], \dots, M[i_j, l+1]\}$. By the same reasoning as before, each rank in $[1..m^{(l+1)}]$ corresponds to a non-empty set of private infinite paths. The private paths associated with rank $M[i_{j'}, l+1]$ are owned by $S_{j'}^{(l+1)} \subseteq S^{(l)}$ which is a cell at column $l+1$ with rank $M[i_{j'}, l+1]$ ($S_{j'}^{(l+1)} = \emptyset$ if no states in $S^{(l)}$ is mapped to $M[i_{j'}, l+1]$). Moreover, none of these paths, hidden or named, should intersect private paths owned by any cell at column l that is a subset of $Q \setminus S^{(l)}$, because states in $Q \setminus S^{(l)}$ and states in $S^{(l)}$ are not in the same component at stage l (the $l+1$ -th stage) in Procedure 6. Now we are ready to show how to build an $n \times \mu$ tree from M .

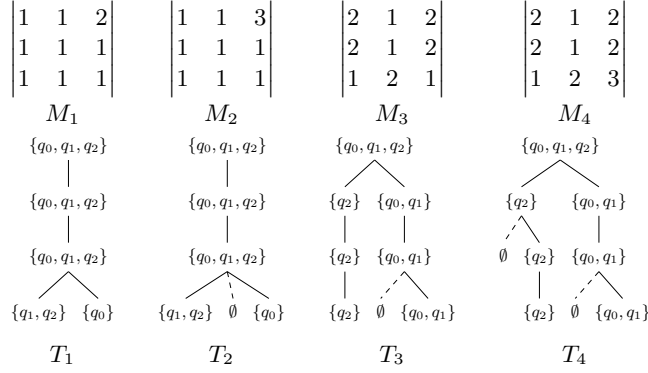
Each node in the tree is associated with a label which is a subset of Q . The root is labeled with set Q . For each rank $i \in [1..m^{(1)}]$, we add a child to the root and we order those children increasingly by the ranks associated with them. If i does not appear in column 1, the i -th child (from left to right) is labeled with \emptyset and is a terminal node (leaf). Otherwise, the child is labeled with the cell at column 1 with rank i and the child is non-terminal if its height is less than μ . We repeat the process column by column. Each maximal $S^{(l)}$ at column $l < \mu$ corresponds to a non-terminal node at height l , from which we spawn a child for each rank in $i \in [1..m^{(l)}]$, and we order and label the children using the rule stated above. After processing column μ , we obtain an $n \times \mu$ tree, because the number of leaves in the tree cannot exceed $\text{width}(\mathcal{G}_w) \leq n$, which we refer to as the *maximum width property* (MWP). Now we call an $n \times \mu$ tree a TOP (*Tree of Ordered Partitions*) and let $\mathcal{T}^r(n, k)$ denote the set of TOPs. We have $|\mathcal{M}^r| \leq |\mathcal{T}^r(n, k)|$.

► **Example 13** (TOP). Four 3×3 matrices M_1 - M_4 and their corresponding tree representations T_1 - T_4 are given in Figure 3. M_1 - M_3 obey MWP and hence T_1 - T_3 are TOPs. T_4 is not a TOP because it has more than 3 leaves.

► **Lemma 14** (Numeric Bound). $|\mathcal{T}^r(n, k)| = 2^{O(n \lg n)}$.

Since $|\mathcal{R}^{\mu R}| \leq n^n \cdot |\mathcal{M}^{\mu R}|$, $|\mathcal{M}^{\mu R}| \leq |\mathcal{M}^r| \cdot |\mathcal{M}^h|$, $|\mathcal{M}^r| \leq |\mathcal{T}^r(n, k)|$, and $|\mathcal{M}^h| \leq (H(n, k))^n$, by Lemmas 12 and 14, we have

► **Theorem 15** (Streett Upper Bound). *Streett complementation is $2^{O(n \lg n + nk \lg k)}$ for $k = O(n)$ and $2^{O(n^2 \lg n)}$ for $k = \omega(n)$.*



■ **Figure 3** Four 3×3 matrices and their corresponding tree representations in Example 13.

Note that we put bounds in the form $2^{O(\cdot)}$ just for simplicity. Even for a small k (i.e. $k = O(n)$), our upper bound is substantially smaller than the current best one $(nk)^{O(nk)}$, established by respective constructions in [8, 22, 14, 19]. Easily seen from the proofs of Lemmas 12 and 14, our upper bound is in fact $n^{O(n)} \cdot k^{O(nk)}$ when $k = O(n)$. Also note that Lemma 14 is crucial in tightening parity complementation.

6 Parity Complementation

Parity automata is a special kind of Streett automata where a Streett condition $\langle G, B \rangle_I$ is augmented with the so-called *Rabin chain condition* $B(1) \subset G(1) \subset \dots \subset B(k) \subset G(k)$. Now the short length of μR ranks is not enough to give us a better bound, because we already have $k \leq \lfloor (n+1)/2 \rfloor$. Nevertheless, the Rabin chain condition makes the GC condition $[B]_I$ degenerate to the CB condition $[B(1)]$, because being $B(1)$ -free is equivalent to being $B(i)$ -free for some $i \in I$. This coincides with the way Mini works. Intuitively, Mini synchronizes all components at a stage of μR ranking assignment. In the first stage of μR ranking assignment, in Step (4.1), Mini makes every vertex get the h -rank 1, though vertices may get different r -ranks. After disabling $G(1)$ vertices (by deleting all outgoing edges from them in Step (4.4)), we have a collection of components satisfying parity condition $\langle G, B \rangle_{[2..k]}$. Then in the second stage of μR ranking assignment, Mini gives every vertex the h -rank 2. Repeating this process, the h -projection of a final μR rank is just $\langle 1, \dots, m \rangle$ for some $m \in [1..k]$, which is completely redundant, because the only useful information (having length m) is already encoded by the corresponding r -projection. As a consequence, h -matrices contribute nothing to the complexity. A customized construction for parity complementation is given in the appendix.

► **Theorem 16** (Parity Upper Bound). *Parity complementation is in $2^{O(n \lg n)}$.*

This bound matches the lower bound of Büchi complementation, and hence it is tight as Büchi automata are a subclass of parity automata. To the best of our knowledge, the previous best upper bound is $2^{O(nk \lg n)}$, which can be easily inferred from [14] by treating parity automata as Rabin automata.

7 Concluding Remarks

In this paper we improved Kupferman and Vardi's construction and obtained tight upper bounds for Streett and parity complementation (with respect to the $2^{\Theta(X)}$ asymptotic notation). Figure 1 in the appendix rounds up the complementation complexities for ω -automata of common types.

Our inquiry also leads to some unexpected outcomes, which we believe, would help understand the strength and weakness of different types of ω -automata in modeling and specifying system behaviors.

1. Parity complementation has the same asymptotical bound as Büchi complementation while parity automata have richer and more elegant acceptance conditions than Büchi automata.
2. Streett automata are exponentially more succinct than Büchi automata while Rabin automata are not. On the other hand, Streett complementation is much easier than Rabin complementation when k is large (i.e., $k = \omega(n)$). In the extreme case where $k = \Theta(2^n)$ and $N = \Theta(nk)$ (the automata size), Streett complementation is in $O(N^{\lg^2 N}) = O(2^{\lg^3 N})$ while Rabin complementation is still in $2^{\Omega(N)}$.

Further investigation on Streett and parity complementation is desired as exponential gaps can hide in the asymptotical notations of the form $2^{\Theta(X)}$. The situation is different from that of Büchi where the best lower and upper bounds have been shown polynomially close.

We think that ITS and TOP characterize intrinsic combinatorial properties on run graphs with universal Rabin conditions. Interesting questions remain for further investigation. What would be the counterparts for run graphs with existential Streett conditions? The discovery of such combinatorial properties might help us understand the complexity of Streett determinization, for which there exists a huge gap between the current lower bound $2^{\Omega(n^2 \lg n)}$ [3] and upper bound $2^{O(nk \lg nk)}$ [19] when $k = \omega(n)$. Also of theoretical interest is whether there exists a type of ω -automata whose determinization is considerably harder than complementation. In the case of Büchi, the two operations were both proved to be in $2^{\Theta(n \lg n)}$.

Acknowledgment

We would like to thank anonymous reviewers for many useful comments, and we are grateful to Laurel Tweed and Wanwu Wang for carefully proofreading the paper.

References

- 1 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. Stanford University Press.
- 2 Y. Cai, T. Zhang, and H. Luo. An improved lower bound for the complementation of Rabin automata. In *Proc. 24th LICS*, pages 167-176, 2009.
- 3 Y. Cai and T. Zhang. A Tight lower bound for Streett complementation. Manuscript at [arXiv:1102.2963](https://arxiv.org/abs/1102.2963) [cs.LG].
- 4 Nachum Dershowitz and Shmuel Zaks. Enumerations of ordered tress. *Discrete Mathematics*, Vol. 31, No. 1 (1980) 9-28.
- 5 N. Francez and D. Kozen. Generalized fair termination. In *Proc. 11th POPL*, pages 46-53, 1984.

- 6 E. Friedgut and O. Kupferman and M.Y. Vardi. Büchi complementation made tighter. *International Journal of Foundations of Computer Science*, Vol. 17, No. 4 (2006) 851-867.
- 7 N. Francez. Fairness. *Texts and Monographs in Computer Science*. Springer-Verlag, 1986.
- 8 N. Klarlund. Progress measures for complementation of omega-automata with applications to temporal logic. In *Proc. 32th FOCS*, pages 358-367, 1991.
- 9 O. Kupferman. Avoiding Determinization. In *Proc. 21th LICS*, pages 243-254, 2006.
- 10 R.P. Kurshan. Computer aided verification of coordinating processes: an automata theoretic approach. Princeton University Press, 1994.
- 11 O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. *ACM Transactions on Computational Logic*, 2(3): 408-429, 2001.
- 12 O. Kupferman and M.Y. Vardi. From complementation to certification. In *10th TACAS*, LNCS 2988, pages 591-606, 2004.
- 13 O. Kupferman and M.Y. Vardi. Safraless decision procedures. In *Proc. 46th FOCS*, pages 531-540, 2005.
- 14 O. Kupferman and M.Y. Vardi. Complementation constructions for nondeterministic automata on infinite words. In *Proc. 11th TACAS*, pages 206-221, 2005.
- 15 C. Löding. Optimal bounds for transformations of omega-automata. In *Proc. 19th FSTTCS*, volume 1738 of *LNCS*, pages 97-109, 1999.
- 16 S. Miyano and T. Hayashi. Alternating finite automata on ω -words. *Theoretical Computer Science*, 32(3):321-330, 1984.
- 17 M. Michel. Complementation is more difficult with automata on infinite words. *CNET*, Paris, 1988.
- 18 T. V. Narayana. A Partial Order and Its Applications to Probability Theory. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, Vol. 21, (1959) 91-98.
- 19 N. Piterman. From Nondeterministic Büchi and Streett Automata to Deterministic Parity Automata. In *Proc. 21th LICS*, pages 255-264, 2006.
- 20 M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:115-125, 1959.
- 21 S. Safra. On the complexity of ω -automata. In *Proc. 29th FOCS*, pages 319-327, 1988.
- 22 S. Safra. Exponential Determinization for ω -Automata with Strong-Fairness Acceptance Condition. In *Proc. 24th STOC*, pages 275-327, 1992.
- 23 S. Schewe. Büchi complementation made tight. In *Proc. 26th STACS*, pages 661-672, 2009.
- 24 A. P. Sistla, M.Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49:217-327, 1987.
- 25 S. Safra and M.Y. Vardi. On ω -Automata and Temporal Logics. In *Proc. 29th STOC*, pages 127-137, 1989.
- 26 M.Y. Vardi. The Büchi complementation saga. In *Proc. 24th STACS*, pages 12-22, 2007.
- 27 M.Y. Vardi. and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proc. 1st LICS*, pages 332-334, 1986.
- 28 Q. Yan. Lower bound for complementation of ω -automata via the full automata technique. In *Proc. 33th ICALP*, volume 4052 of *LNCS*, pages 589-600, 2006.