On the Power of Unambiguity in Büchi Complementation

Yong Li
State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences
liyong@ios.ac.cn

Moshe Y. Vardi
Rice University
vardi@cs.rice.edu

Lijun Zhang
State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences
University of Chinese Academy of Sciences
Institute of Intelligent Software, Guangzhou
zhanglj@ios.ac.cn

In this work, we exploit the power of *unambiguity* for the complementation problem of Büchi automata by utilizing reduced run directed acyclic graphs (DAGs) over infinite words, in which each vertex has at most one predecessor. We then show how to use this type of reduced run DAGs as a *unified tool* to optimize *both* rank-based and slice-based complementation constructions for Büchi automata with a finite degree of ambiguity. As a result, given a Büchi automaton with n states and a finite degree of ambiguity, the number of states in the complementary Büchi automaton constructed by the classical rank-based and slice-based complementation constructions can be improved, respectively, to $2^{\mathcal{O}(n)}$ from $2^{\mathcal{O}(n \log n)}$ and to $\mathcal{O}(4^n)$ from $\mathcal{O}((3n)^n)$.

1 Introduction

The complementation of nondeterministic Büchi automata on words (NBWs) [7] is a classic problem for NBWs and is the fundamental construction for many other important questions such as model checking [29] and program-termination analysis [15]. For instance, the complementation of NBWs is particularly valuable to model checking, when both the system A and the specification B are given as NBWs. A model-checking problem essentially asks whether the behavior of the system A satisfies the specification B. In automata-based model checking [29] framework, this model-checking problem reduces to a language-containment problem between the NBWs A and B. The standard approach to solving the language-containment problem between A and B relies on the complementation of B; one first has to construct a complementary automaton B^c such that $\mathcal{L}(B^c) = \Sigma^\omega \setminus \mathcal{L}(B)$ and then checks language emptiness of $\mathcal{L}(A) \cap \mathcal{L}(B^c)$. Various implementations of this approach with optimizations [1, 2, 11, 13] have been proposed to improve its practical performance. All the implementations above, however, directly or indirectly, resort to constructing B^c, which can be exponentially larger than B [25, 30].

The complementation of Büchi automata is also a key component in the automata-based program-termination checking framework proposed in [15]. This framework starts with a termination proof of a sample path of the given program and then generalizes that path to a Büchi automaton, whose language (by construction) represents a set of terminating paths. All these terminating paths are then removed from the program. The removal of those paths is done by automata difference operation, involved with Büchi complementation and intersection. By iteratively removing terminating paths, the framework may obtain an empty program in the end, thus also proving the termination of the program. It has been shown in [10]

J.-F. Raskin and D. Bresolin (Eds.): 11th International Symposium
on Games, Automata, Logics, and Formal Verification (GandALF'20).
EPTCS 326, 2020, pp. 182–198, doi:10.4204/EPTCS.326.12

that efficient complementation algorithms for Büchi automata can significantly improve the performance of the program-termination checking framework.

In this work, we focus on the complementation of NBWs. The complexity for complementing NBWs has been proved to be $\Omega((0.76n)^n)$ [25, 30]. A classic line of research on complementation aims at developing optimal (or close to optimal) complementation algorithms. Currently there are mainly four types of practical complementation algorithms for NBWs, namely *Ramsey-based* [26], *determinization-based* [24], *rank-based* [18] and *slice-based* [16] algorithms. These algorithms, however, all unavoidablely lead to a super-exponential growth in the size of B^c in the worst case [30].

With the growing understanding of the worst-case complexity of those algorithms, searching for specialized complementation algorithms for certain subclasses of NBWs with better complexity has become an important line of research. For instance, complementing deterministic and semi-deterministic Büchi automata can be done in $\mathcal{O}(n)$ [19] and $\mathcal{O}(4^n)$ [5], respectively. Here we follow this line of research and aim at a subclass of NBWs with restricted nondeterminism. This type of NBWs is important, as in some contexts, especially in probabilistic verification, unrestricted nondeterminism in the automata representing the properties is problematic for the verification procedure. For instance, general NBWs cannot be used directly to verify properties over Markov chains, as they will cause imprecise probabilities in the product of the system and the property [8]. In turn, it is often necessary to construct their more deterministic counterparts in terms of other types of automata for the properties, for instance semi-deterministic Büchi automata or deterministic Rabin automata, which, however, adds exponential blowups of states [12].

To avoid state-space exponential blowup, earlier work sought to use of a type of automata called *unambiguous nondeterministic Büchi automata* (UNBWs) in probabilistic verification [4,20], as UNBWs can be exponentially smaller than their equivalent deterministic automata [4]. UNBWs [9] are a subclass of NBWs that accept with at most one run for each word, while their equivalent NBWs may have more than one accepting run, or even infinitely many accepting runs. For example, by taking advantage of their unambiguity, the language-containment problem of certain proper subclasses of UNBWs has been proved to be solvable in polynomial time [6], while this problem is PSPACE-complete for NBWs [17].

The complementation problem of a more general class than UNBWs, called *finitely ambiguous nondeterministic Büchi automata* (FANBWs), which accept with finitely many runs for each word, was shown to be doable in $\mathcal{O}(5^n)$ [23], in contrast to $2^{\Omega(n\log n)}$ for general NBWs [25]. Further, checking whether an NBW is an FANBW can be done in polynomial time [21]. Therefore, once an FANBW has been identified, the specialized complementation construction for FANBWs can be applied. In this paper, we focus here on an in-depth study of the complementation problem for FANBWs.

Our main technical tool is a construction of reduced directed acyclic graphs (DAGs) of runs of FANBWs over infinite words called *co-deterministic run DAGs*, in which each vertex has at most one predecessor. This type of co-deterministic run DAGs is previously introduced in [14,23] and we defer the comparison of [14,23] and our construction to related works section. We show that such co-deterministic run DAGs can be used to simplify and improve both the classical rank-based and slice-based complementation constructions. Our contributions are the following.

- First, we apply the co-deterministic run DAGs of FANBWs over infinite words, as a *unified tool* to show how unambiguity works in Büchi complementation, to optimizing *both* rank-based complementation (RKC) and slice-based complementation (SLC).
- Second, we show that the construction of co-deterministic run DAGs in different complementation algorithms [27] helps to achieve simpler and theoretically better complementation algorithms for FANBWs. Given an FANBW with n states, we show that the number of states of the complemen-

tary NBW constructed by the classical RKC and SLC constructions can be improved, respectively, to $2^{\mathcal{O}(n)}$ from $2^{\mathcal{O}(n\log n)}$ and to $\mathcal{O}(4^n)$ from $\mathcal{O}((3n)^n)$.

- Finally, we reveal that SLC is basically an algorithm based on the construction of co-deterministic run DAGs and a specialized complementation algorithm for FANBWs. We also provide a language containment relation between states in the complementary NBWs of FANBWs, which can be used to improve the containment checking between an NBW and an (FA)NBW and also to reduce the number of redundant states in the complementary NBW.

Related work. Run DAGs were introduced in [18] and co-deterministic run DAGs were first described in [14]. In [14], Fogarty and Vardi exploit co-deterministic run DAGs to complement *reverse deterministic* Büchi automata with RKC and the Ramsey-based algorithm, while we consider RKC and SLC in this work. In a reverse deterministic Büchi automaton, each state has only one predecessor for each letter, for which all run DAGs are already co-deterministic, as explained in Section 4.2, while the run DAGs of FANBWs may not be co-deterministic without our construction described in Section 3.

Later co-deterministic run DAGs were constructed in [23] under the name of *narrow forest* for complementing FANBWs with the SLC construction only. Here we present it as co-deterministic run DAGs to serve as a unified tool for explaining concepts in both RKC and SLC constructions. A subtle difference between the construction of co-deterministic run DAGs in [23] and ours is as follows. To construct a co-deterministic run DAG over $w \in \Sigma^\omega$, Rabinovich [23] makes use of a transducer $\mathscr{T}$ that chooses one predecessor for each vertex at current level, while our construction utilizes a transition function to make the sets of successors of each pair of vertices at current level disjoint with each other, as given in Definition 2.

More significantly, for complementation, we applied co-deterministic run DAGs to *both* RKC [18] and SLC as presented in [28]. (The complementation construction proposed in [23] is a variant of SLC as introduced in [16].) The comparison of the construction in [23] and our improvement over SLC is as follows. First, the complementary NBW constructed in [23] is a UNBW with at most $\mathcal{O}(5^n)$ states; this complementary NBW is the product automaton of the transducer $\mathscr{T}$, a Büchi automaton $\mathscr{C}$ for expressing unambiguity and a Büchi automaton $\mathscr{D}$ for accepting all possible ways to construct co-deterministic DAGs over $w \notin \mathscr{L}(\mathscr{A})$. Our complementary NBW is not required to be a UNBW, since we are interested in complementation for containment checking. Thus, the bound of $\mathcal{O}(5^n)$ in [23] is exponentially higher than the bound of $\mathcal{O}(4^n)$ in this work. Indeed, the product automaton of $\mathscr{T}$ and $\mathscr{D}$ in [23] does yield a complementary NBW with $\mathcal{O}(4^n)$ states, but this construction and complexity were not explicitly given in [23].

Second, the construction in [23] and our SLC-based construction are both based on reduced DAGs in which each vertex has at most one predecessor. These two constructions, however, are technically different and have different emphases. The construction in [23] aims at building a complementary NBW $\mathscr{A}^c$ that is unambiguous, based on building product of three automata, in which each automaton fulfills part of the desired functionality for $\mathscr{A}^c$. For instance, $\mathscr{C}$ takes care of unambiguity and $\mathscr{D}$ obtains the complementary language. While our focus is on a complementation construction for containment checking. In contrast to building product automata in [23], our construction in Section 5.2 takes a tuple of sets of states of $\mathscr{A}$ as a state in the complementary automaton $\mathscr{A}^c$ of $\mathscr{A}$ and performs directly on those tuples for computing successors on-the-fly, following the idea of the NCSB complementation for semi-deterministic Büchi automata in [5]. Various subsumption relations have been proposed in [10] for this representation of states in the NCSB complementation and help to reduce the number of states in $\mathscr{A}^c$, even improving termination analysis of programs. Inspired by [10], we can also define a subsumption relation between states in $\mathscr{A}^c$ (see Corollary 2) by our construction, which can be used to improve the containment checking between an NBW and an (FA)NBW and to reduce the number of states in $\mathscr{A}^c$.

Organization of the paper. In the remainder of this paper, we first recap some definitions about Büchi automata in Section 2 and then introduce the concept of co-deterministic run DAGs in Section 3. We present our improved algorithms for the rank-based and slice-based algorithms in Section 4 and Section 5, respectively. Finally we conclude the paper with some future works in Section 6.

2 Preliminaries

We fix an *alphabet* Σ. A *word* is an infinite sequence w of letters in Σ. We denote by Σ^ω the set of all (infinite) words. A *language* is a subset of Σ^ω. Let L be a language and the complement language of L is denoted by L^c, i.e., $L^c = \Sigma^\omega \setminus L$. Let ρ be a sequence of elements: we denote by $\rho[i]$ the i-th element of ρ. Let n be a natural number; we denote by $[n]$ the set of numbers $\{0, 1, \cdots, n\}$, $[n]^{odd}$ the set of odd numbers in $[n]$ and $\langle n \rangle$ the set of numbers $[n] \setminus \{0\}$.

A *nondeterministic Büchi automaton on words* (NBW) is a tuple $\mathscr{A} = (Q, I, \delta, F)$, where Q is a finite set of states, $I \subseteq Q$ is a set of initial states, $\delta : Q \times \Sigma \to 2^Q$ is a transition function and $F \subseteq Q$ is a set of accepting states. We extend δ to sets of states, by letting $\delta(S,a) = \bigcup_{q \in S} \delta(q,a)$. We assume that each NBW $\mathscr{A}$ is *complete* in the sense that for each state $q \in Q$ and $a \in \Sigma$, $\delta(q,a) \neq \emptyset$. A *run* of $\mathscr{A}$ on a word w is an infinite sequence of states $\rho = q_0 q_1 \cdots$ such that $q_0 \in I$ and for every $i > 0$, $q_i \in \delta(q_{i-1}, a_i)$. We denote by $\mathit{inf}(\rho)$ the set of states that occur infinitely often in the run ρ. A word $w \in \Sigma^\omega$ is *accepted* by $\mathscr{A}$ if there exists a run ρ of $\mathscr{A}$ over w such that $\mathit{inf}(\rho) \cap F \neq \emptyset$. We denote by $\mathscr{L}(\mathscr{A})$ the *language* recognized by $\mathscr{A}$, i.e., the set of words accepted by $\mathscr{A}$.

Let $\mathscr{A}$ be an NBW. A complementary NBW of $\mathscr{A}$ is an NBW that accepts the complementary language $\Sigma^\omega \setminus \mathscr{L}(\mathscr{A})$ of $\mathscr{L}(\mathscr{A})$; we denote by $\mathscr{A}^S$ the automaton (Q, S, δ, F) obtained from $\mathscr{A}$ by setting its initial state set to the set $S \subseteq Q$. In particular, we use $\mathscr{A}^q$ as the shorthand for $\mathscr{A}^{\{q\}}$. We say a state q of $\mathscr{A}$ *subsumes* a state q' of $\mathscr{A}$ if $\mathscr{L}(A^{q'}) \subseteq \mathscr{L}(A^q)$. We classify $\mathscr{A}$ into following types of NBWs according to their transition structures: (1) *nondeterministic* if $|I| > 1$ or $|\delta(q,a)| > 1$ for a state $q \in Q$ and $a \in \Sigma$, (2) *deterministic* if $|I| = 1$ and for each $q \in Q$ and $a \in \Sigma$, $|\delta(q,a)| \leq 1$, and (3) *reverse deterministic* if for each state $q' \in Q$, $\mathscr{A}$ has at most one state q for each $a \in \Sigma$ such that $q' = \delta(q,a)$.

From the perspective of the number of accepting runs of $\mathscr{A}$, we have following types of NBWs.

Definition 1. *Let $\mathscr{A}$ be an NBW and k a positive integer. We say $\mathscr{A}$ is (1) finitely ambiguous (an FANBW) if for each $w \in \mathscr{L}(\mathscr{A})$, the number of accepting runs of $\mathscr{A}$ over w is finite; and (2) k-ambiguous if for each $w \in \mathscr{L}(\mathscr{A})$, the number of accepting runs of $\mathscr{A}$ over w is no greater than k, and (3) unambiguous if it is* 1*-ambiguous.*

By Definition 1, it holds that both k-ambiguous NBWs and unambiguous NBWs are special classes of FANBWs. For instance, the NBW $\mathscr{A}$ depicted in Figure 1 is a 2-ambiguous NBW, thus also an FANBW, as $(q_0)^{i+1} q_1^\omega$ and $(q_0)^{i+1} q_2 q_1^\omega$ are the only two accepting runs for accepting word $a^i b^\omega \in \mathscr{L}(\mathscr{A})$ where $i \geq 0$.

3 Co-Deterministic Run DAGs for FANBWs

In this section, we first describe the concept of run DAGs of an NBW over a word w, introduced in [18]. We then describe the co-deterministic run DAGs for FANBWs as a unified tool for both RKC and SLC

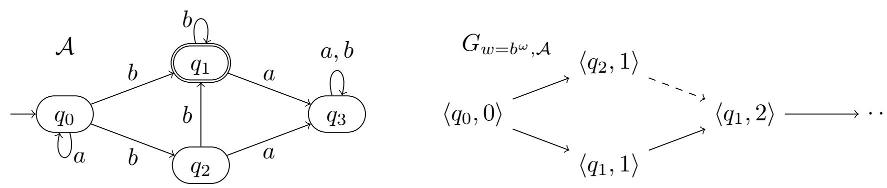

Figure 1: An FANBW $\mathscr{A}$ with $I = \{q_0\}$ and $F = \{q_1\}$ and the run DAG $G_{w,\mathscr{A}}$ over b^ω.

constructions by making use of the finite ambiguity in FANBWs. In the remainder of the paper, we use DAGs as the shorthand for run DAGs.

Let $\mathscr{A} = (Q, I, \delta, F)$ be an NBW and $w = a_0 a_1 \cdots$ be an infinite word. The DAG $G_{w,\mathscr{A}} = \langle V, E\rangle$ of $\mathscr{A}$ over w is defined as follows:

- Vertices: $V \subseteq Q \times \mathbb{N}$ is the set of vertices $\bigcup_{l\geq 0} V_l \times \{l\}$ where $V_0 = I$ and $V_{l+1} := \delta(V_l, a_l)$ for every $l \geq 0$.
- Edges: There is an edge from $\langle q, l\rangle$ to $\langle q', l'\rangle$ iff $l' = l + 1$ and $q' \in \delta(q, a_l)$.

A vertex $\langle q, l\rangle$ is said to be on level l and there are at most $|Q|$ states on each level. A vertex $\langle q, l\rangle$ is an *F-vertex* if $q \in F$. A finite/infinite sequence of vertices $\gamma = \langle q_0, 0\rangle\langle q_1, 1\rangle \cdots$ is called a *branch* of $G_{w,\mathscr{A}}$ if $q_0 \in I$ and for each $l \geq 0$, there is an edge from $\langle q_l, l\rangle$ to $\langle q_{l+1}, l+1\rangle$. An *$\omega$-branch* of $G_{w,\mathscr{A}}$ is a branch of infinite length. A *fragment* $\langle q_l, l\rangle\langle q_{l+1}, l+1\rangle \cdots$ of γ is said to be a branch from the vertex $\langle q_l, l\rangle$; a fragment $\langle q_l, l\rangle \cdots \langle q_{l+k}, l+k\rangle$ of γ is said to be a *path* from $\langle q_l, l\rangle$ to $\langle q_{l+k}, l+k\rangle$, where $k \geq 1$. A vertex $\langle q_j, j\rangle$ is *reachable* from $\langle q_l, l\rangle$ if there is a path from $\langle q_l, l\rangle$ to $\langle q_j, j\rangle$. We call a vertex $\langle q, l\rangle$ *finite* if there are no ω-branches in $G_{w,\mathscr{A}}$ starting from $\langle q, l\rangle$; and we call a vertex $\langle q, l\rangle$ *F-free* if it is not finite and no F-vertices are reachable from $\langle q, l\rangle$ in $G_{w,\mathscr{A}}$.

There is a bijection between the set of runs of $\mathscr{A}$ on w and the set of ω-branches in $G_{w,\mathscr{A}}$. To a run $\rho = q_0 q_1 \cdots$ of $\mathscr{A}$ over w corresponds an ω-branch $\hat{\rho} = \langle q_0, 0\rangle\langle q_1, 1\rangle \cdots$. Therefore, w is accepted by $\mathscr{A}$ if and only if there exists an ω-branch in $G_{w,\mathscr{A}}$ that visits F-vertices infinitely often; we say that such an ω-branch is *accepting*; $G_{w,\mathscr{A}}$ is accepting if and only if there exists an accepting ω-branch in $G_{w,\mathscr{A}}$.

Assume that $\mathscr{A}$ is an FANBW. Then an accepting ω-branch in $G_{w,\mathscr{A}}$, if exists, only merges with other (accepting) ω-branches for finitely many times. That is, there exists a level $k \geq 1$ such that all vertices after level k on an accepting ω-branch have exactly one predecessor; we call the level k a *separate level*. We formalize this property of $G_{w,\mathscr{A}}$ in Lemma 1.

Lemma 1 (Separate Levels of Accepting DAGs of FANBWs)**.** *Let $\mathscr{A}$ be an FANBW and $G_{w,\mathscr{A}}$ the accepting DAG of $\mathscr{A}$ over $w \in \mathscr{L}(\mathscr{A})$. Then there must exist a separate level $k \geq 1$ in $G_{w,\mathscr{A}}$.*

Proof. Since $\mathscr{A}$ is an FANBW, there are only finitely many accepting ω-branches in $G_{w,\mathscr{A}}$. Therefore, an accepting ω-branch in $G_{w,\mathscr{A}}$ only merges with other (accepting) ω-branches for finitely many times. It follows that given an accepting ω-branch $\hat{\rho}$ in $G_{w,\mathscr{A}}$, there must exist a separate level $h \geq 1$ such that each vertex $\hat{\rho}[i]$ with $i \geq h$ has exactly one predecessor. Otherwise, there will be infinitely many accepting branches, contradicting with the assumption that $\mathscr{A}$ is an FANBW. Assume that there are $m < \infty$ accepting ω-branches in $G_{w,\mathscr{A}}$. Then we can set the separate level k of $G_{w,\mathscr{A}}$ to $\max\{\, h_i \mid 1 \leq i \leq m \,\}$ where h_i is the separate level index of i-th accepting ω-branch. $\square$

For instance, the separate level is 2 in the accepting DAG $G_{w,\mathscr{A}}$ of $\mathscr{A}$ over b^ω in Figure 1, as each vertex $\langle q_1, i\rangle$ with $i \geq 3$ only has the predecessor $\langle q_1, i-1\rangle$.

It follows immediately from Lemma 1 that for each vertex v in $G_{w,\mathscr{A}}$ with more than one incoming edge, keeping only one of incoming edges of v will not change whether $G_{w,\mathscr{A}}$ is accepting. Assume that $Q = \{s_1, s_2, \cdots, s_n\}$. We define an *edge-reduced* DAG $G^e_{w,\mathscr{A}} = \langle V, E^e \rangle$ called co-deterministic DAG, in which each vertex only has at most one predecessor with the following policy for removing edges: if there is a vertex with multiple incoming edges in $G_{w,\mathscr{A}}$, we only keep the incoming edge from the predecessor with the minimal index. Formally, the definition of edges in $G^e_{w,\mathscr{A}}$ is given as follows.

- Edges. There is an edge from $\langle s_k, l \rangle$ to $\langle s', l' \rangle$ iff $l' = l + 1$ and $k = \min\{\, p \in \langle n \rangle \mid s' \in \delta(s_p, a_{l+1}) \,\}$.

Lemma 2 ensures that $G^e_{w,\mathscr{A}}$ is accepting if $G_{w,\mathscr{A}}$ is accepting.

Lemma 2 (Acceptance of Co-deterministic DAGs)**.** *Assume that $\mathscr{A}$ is an FANBW. Let $G^e_{w,\mathscr{A}}$ be the co-deterministic DAG of $\mathscr{A}$ over a word $w \in \Sigma^\omega$. Then w is accepted by $\mathscr{A}$ if and only if $G^e_{w,\mathscr{A}}$ is accepting.*

Proof. The proof is trivial when $G_{w,\mathscr{A}}$ is nonaccepting. Assume that $G_{w,\mathscr{A}}$ is accepting. Let $\hat{\rho}$ be an accepting ω-branch and k the separate level defined in Lemma 1. According to Lemma 1, the ω-branch from $\hat{\rho}[k+1]$ must be accepting. Moreover, $\hat{\rho}[k+1]$ is reachable from an initial vertex $\langle q, 0 \rangle$ with $q \in I$. Then there must exist an accepting ω-branch in $G^e_{w,\mathscr{A}}$ if $G_{w,\mathscr{A}}$ is accepting. Thus we conclude that w is accepted by $\mathscr{A}$ if and only if $G^e_{w,\mathscr{A}}$ is accepting. □

For instance, the co-deterministic DAG of $G_{w,\mathscr{A}}$ in Figure 1 is still accepting after deleting the edge from $\langle q_2, 1 \rangle$ to $\langle q_1, 2 \rangle$, as denoted by the dashed arrow.

By removing redundant edges, we can now define a reduced transition function $\delta^e_{w,\ell} : 2^Q \times \Sigma \to 2^Q$ over the levels in $G^e_{w,\mathscr{A}}$.

Definition 2 (Transition Function for Co-deterministic DAGs)**.** *Given the set of states $S \subseteq Q$ at level ℓ of $G^e_{w,\mathscr{A}}$ and let $S' = \delta(S, w[\ell])$ be the set of states at level $\ell + 1$. Define $S_{min} = \{\, q_m \in S \mid m \in \min\{\, k \in \langle n \rangle \mid q' \in \delta(q_k, w[\ell]) \,\}, q' \in S' \,\}$ as the minimal set of predecessors of S'. Then, for a set of states $S_1 \subseteq S$, we define $\delta^e_{w,\ell}(S_1, w[\ell]) = \delta(S_1 \cap S_{min}, w[\ell])$. We call $\delta^e_{w,\ell}$ the* reduced transition function *at level ℓ in $G^e_{w,\mathscr{A}}$.*

Example 1. *Consider again $G_{b^\omega,\mathscr{A}}$ in Figure 1 and let $S = \{q_1, q_2\}$ at level 1: we have $S' = \delta(S, b) = \{q_1\}$ and $S_{min} = \{q_1\}$. Let $\delta^e_{b^\omega,1}$ be the reduced transition function at level 1 defined from δ in Definition 2. It follows that $\delta^e_{b^\omega,1}(\{q_1\}, b) = \delta(\{q_1\} \cap S_{min}, b) = \{q_1\}$ and $\delta^e_{b^\omega,1}(\{q_2\}, b) = \delta(\{q_2\} \cap S_{min}, b) = \emptyset$.*

In general, the reduced transition function $\delta^e_{w,\ell}$ may seem to depend on the level ℓ and the word w yielding the edge connections between vertices at levels ℓ and $\ell + 1$ in $G^e_{w,\mathscr{A}}$. We claim, however, that in Definition 2, $\delta^e_{w,\ell}$ is not dependent on the level number ℓ and the word w, due to our specific choice of the set S_{min}. Thus, we can omit the level ℓ and w in our notion δ^e.

Lemma 3. *Let $S \subseteq Q$ and b be the set of states and the input letter at the level ℓ_1 in $G^e_{w_1,\mathscr{A}}$ and at the level ℓ_2 in $G^e_{w_2,\mathscr{A}}$, respectively. Then $\delta^e_{w_1,\ell_1}$ of $G^e_{w_1,\mathscr{A}}$ and $\delta^e_{w_2,\ell_2}$ of $G^e_{w_2,\mathscr{A}}$ are identical regardless of their different level numbers and infinite words.*

Proof. According to Definition 2, we can let $w_1[\ell_1] = w_2[\ell_2] = b$. Then all the subsequent computations defined for both $\delta^e_{w_1,\ell_1}$ and $\delta^e_{w_2,\ell_2}$ only depend on the set of states S and the input letter b, not their level numbers and the entire infinite words. Thus we complete the proof. □

Because of Lemma 3, we can just use the reduced transition function δ^e with respect to the set of states S and the input letter b at a level in the construction of complementary NBWs of FANBWs (see Definitions 4' and 5). We remark that one can define different co-deterministic DAGs from those constructed in this work. This is illustrated in the following example.

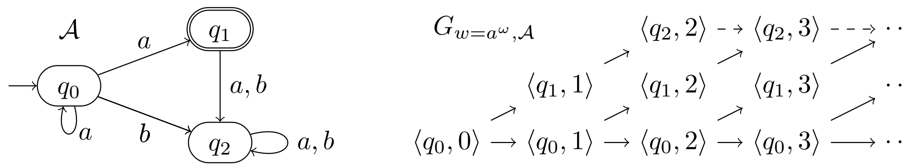

Figure 2: Another FANBW $\mathscr{A}$ with $I = \{q_0\}$ and $F = \{q_1\}$ and the run DAG $G_{w,\mathscr{A}}$ over a^ω

.

Example 2 ($\delta^e_{w,\ell}$ depending on ℓ)**.** *Consider $G_{a^\omega,\mathscr{A}}$ in Figure 2 and let $S_\ell = \{q_0, q_1, q_2\}$ at level $\ell \geq 2$: we have $S_\ell = \delta(S_\ell, a)$ as the set of states on each level $\ell \geq 2$. Rather than keeping the predecessor with the minimal index of a state in S_{min} (see Definition 2), one can define $S_{\ell,min}$ as S_{min} depending on the level ℓ as follows. We define $S_{\ell,min} = \{q_0, q_1\}$ when ℓ is an odd number and $S_{\ell,min} = \{q_0, q_2\}$ otherwise. That is, we keep the predecessor q_1 of q_2 at odd levels and q_2 at even levels. Let $\delta^e_{a^\omega,\ell}$ be the reduced transition function at level ℓ. It follows that $\delta^e_{a^\omega,\ell}(\{q_1\}, a) = \delta(\{q_1\} \cap S_{\ell,min}, a) = \{q_2\}$ when ℓ is odd and $\delta^e_{a^\omega,\ell}(\{q_1\}, a) = \emptyset$ otherwise. Clearly, the definition of $\delta^e_{a^\omega,\ell}$ is dependent on the level ℓ and the resulting co-deterministic DAG is different from the one depicted in Figure 2 where dashed arrows denote the removed edges.*

In the remainder of the paper, we may write $\delta^e(q, b)$ instead of $\delta^e(\{q\}, b)$ for an input singleton set $\{q\}$. The transition function δ^e will be used in the complementation of FANBWs since the complementation essentially constructs DAGs and then identifies accepting DAGs.

One can verify that each vertex in the co-deterministic DAG $G^e_{w,\mathscr{A}}$ of $\mathscr{A}$ over w has at most one predecessor. It follows that the number of ω-branches in a non-accepting/accepting $G^e_{w,\mathscr{A}}$ is at most $|Q|$, as stated in Lemma 4.

Lemma 4 (Finite Number of ω-Branches in Co-deterministic DAGs)**.** *Let $G^e_{w,\mathscr{A}}$ be a co-deterministic DAG of $\mathscr{A}$ over w. Then the number of ω-branches in $G^e_{w,\mathscr{A}}$ is at most $|Q|$.*

Proof. Let m_i with $i \geq 0$ be the number of vertices which are in the ω-branches (not in all branches) on level i. For instance, $m_i = 1$ for each $i \geq 1$ in Fig. 1 while the number of vertices on level 1 is 2. Since each vertex in $G^e_{w,\mathscr{A}}$ has only one predecessor, we have that $m_0 \leq m_1 \leq m_2 \leq \cdots$, i.e., the number of vertices in ω-branches on each level does not decrease over the levels. In addition, there are at most $|Q|$ states on each level. Thus there are at most $|Q|$ ω-branches since we have $m_i \leq |Q|$ for each $i \geq 0$. □

Consider the DAG $G_{w,\mathscr{A}}$ in Figure 2: one can verify that there are infinitely many ω-branches in the non-reduced DAG $G_{w,\mathscr{A}}$ over a^ω; while for the co-deterministic DAG of $G_{w,\mathscr{A}}$ where removed edges are marked with dashed arrows, there is only one ω-branch $\langle q_0, 0\rangle\langle q_0, 1\rangle \cdots \langle q_0, l\rangle \cdots$.

After redundant edges have been cut off, we obtain a DAG $G^e_{w,\mathscr{A}}$ with a finite number of ω-branches. Thus if $w \notin \mathscr{L}(\mathscr{A})$, there must exist a maximum level $l > 0$ among those ω-branches such that each F-vertex $\langle q, l'\rangle$ with $l' \geq l$ is finite, which can be used for identifying whether $G^e_{w,\mathscr{A}}$ is accepting in the complementation of FANBWs. We call a level $l > 0$ a *stable level* in $G^e_{w,\mathscr{A}}$ if each F-vertex $\langle q, l'\rangle$ with $l' \geq l$ in $G^e_{w,\mathscr{A}}$ is finite.

Lemma 5 (Stable Level in Nonaccepting Co-deterministic DAGs)**.** *Assume that $\mathscr{A}$ is an FANBW and $w \in \Sigma^\omega$. Let $G^e_{w,\mathscr{A}}$ be the co-deterministic DAG of $\mathscr{A}$ over w. Then $w \notin \mathscr{L}(\mathscr{A})$ if and only if there exists a stable level $k > 0$ in $G^e_{w,\mathscr{A}}$.*

Proof. ($\Leftarrow$) By Lemma 2, if $w \in \mathscr{L}(\mathscr{A})$ and $\mathscr{A}$ is an FANBW, there exists an accepting ω-branch in $G^e_{w,\mathscr{A}}$. It follows that if $w \in \mathscr{L}(\mathscr{A})$, there does not exist a stable level k in $G^e_{w,\mathscr{A}}$ such that each F-vertex after k is finite. Consequently, if there exists a stable level k in $G^e_{w,\mathscr{A}}$, it holds that $w \notin \mathscr{L}(\mathscr{A})$.

($\Rightarrow$) By Lemma 4, let $m \leq |Q|$ be the number of ω-branches in $G^e_{w,\mathscr{A}}$. Since $w \notin \mathscr{L}(\mathscr{A})$, all the ω-branches in $G^e_{w,\mathscr{A}}$ is nonaccepting. Therefore, for the i-th ω-branch $\hat{\rho}_i$, there is a vertex $\langle q, k_i\rangle$ such that every vertex of $\hat{\rho}_i$ reachable from $\langle q, k_i\rangle$ is not an F-vertex. It follows that we can set $k = \max\{ k_i \mid i \in \langle m\rangle \}$ and thus all the F-vertices on a level after $l \geq k$ are finite and not on ω-branches. $\square$

Consider again the DAG $G_{w,\mathscr{A}}$ in Figure 2: there does not exist a stable level in the non-reduced DAG $G_{w,\mathscr{A}}$ since each F-vertex $\langle q_1, l\rangle$ with $l \geq 1$ is not finite; while in the co-deterministic DAG of $\mathscr{A}$ over a^ω, one can verify that the stable level k is 1.

4 Rank-Based Complementation

We first introduce in Subsection 4.1 the rank-based complementation (RKC) proposed in [18], which constructs a complementary NBW $\mathscr{A}^c$ for $\mathscr{A}$ with $2^{\mathscr{O}(n \log n)}$ states. Then in Subsection 4.2, we show that if $\mathscr{A}$ is an FANBW, RKC based on the construction of co-deterministic DAGs produces a complementary NBW $\mathscr{A}^c$ with $2^{\mathscr{O}(n)}$ states.

4.1 Rank-Based Algorithm for NBWs

RKC was introduced by Kupferman and Vardi in [18] to construct a complementary NBW $\mathscr{A}^c$ of $\mathscr{A}$ by identifying the DAGs of $\mathscr{A}$ over nonaccepting words $w \notin \mathscr{L}(\mathscr{A})$. Intuitively, given a word $w \notin \mathscr{L}(\mathscr{A})$, all ω-branches of the DAG of $\mathscr{A}$ over w will eventually stop visiting F-vertices. Based on this observation, in order to identify the nonaccepting DAG of $\mathscr{A}$ over w, they introduced the notion of *level rankings* of $G_{w,\mathscr{A}}$. By assigning only even ranks to F-vertices, they showed that there exists a unique ranking function that assigns ranks in $[2n]$ to the vertices of $G_{w,\mathscr{A}}$ such that $w \notin \mathscr{L}(\mathscr{A})$ iff all ω-branches of $G_{w,\mathscr{A}}$ eventually get trapped in odd ranks.

We now define level rankings of a nonaccepting DAG. The level ranking of $G_{w,\mathscr{A}} = (V, E)$ defines a ranking function $f : V \to [2n]$ that satisfies the following conditions:

(i) for each vertex $\langle q, i\rangle \in V$ if $f(\langle q, i\rangle) \in [2n]^{odd}$, then $q \notin F$,

(ii) for each edge $(\langle q, i\rangle, \langle q', i+1\rangle) \in E$, $f(\langle q', i+1\rangle) \leq f(\langle q, i\rangle)$

The ranks along a branch decrease monotonically and F-vertices get only even ranks.

We now define a specific ranking function f of $G_{w,\mathscr{A}}$ for a given word $w \notin \mathscr{L}(\mathscr{A})$. We define a sequence of DAGs $G^0_{w,\mathscr{A}} \supseteq G^1_{w,\mathscr{A}} \supseteq \cdots$, where $G^0_{w,\mathscr{A}} = G_{w,\mathscr{A}}$, as follows. For each $i \geq 0$,

- $G^{2i+1}_{w,\mathscr{A}}$ is the DAG constructed from $G^{2i}_{w,\mathscr{A}}$ by removing all finite vertices in $G^{2i}_{w,\mathscr{A}}$ and the edges associated with them, and
- if $G^{2i+1}_{w,\mathscr{A}}$ has at least one F-free vertex, then $G^{2i+2}_{w,\mathscr{A}}$ is the DAG constructed from $G^{2i+1}_{w,\mathscr{A}}$ by removing all the F-free vertices in $G^{2i+1}_{w,\mathscr{A}}$ and the edges associated with them.

Recall that F-free vertices cannot reach F-vertices. It was shown in [18] that $G^{2n+1}_{w,\mathscr{A}}$ is empty and each vertex $\langle q, l\rangle$ is either finite in $G^{2i}_{w,\mathscr{A}}$ or F-free in $G^{2i+1}_{w,\mathscr{A}}$. Thus the sequence of DAGs generated from the definition above defines a unique ranking function f over the set of vertices in $G_{w,\mathscr{A}}$ inductively as follows. For every $i \geq 0$,

(1) $f(\langle q,l\rangle) = 2i$ for each vertex $\langle q,l\rangle$ that is finite in $G^{2i}_{w,\mathscr{A}}$, if exists.

(2) $f(\langle q,l\rangle) = 2i+1$ for each F-free vertex $\langle q,l\rangle$ in $G^{2i+1}_{w,\mathscr{A}}$, if exists.

Consequently, we have Lemma 6 for identifying nonaccepting DAGs according to [18].

Lemma 6 (Nonaccepting DAGs [18])**.** *$\mathscr{A}$ rejects a word w if and only if the unique ranking function f defined in (1) and (2) above has $2n$ as maximum rank, and all ω-branches of $G_{w,\mathscr{A}}$ eventually get trapped in odd ranks.*

We have constructed a unique ranking function above for identifying nonaccepting DAGs. To construct the complementary NBW $\mathscr{A}^c$ with such a ranking function, we have to guess the ranking level by level. Since the maximum rank is $2n$, along an input word w, we can encode a ranking function for $G_{w,\mathscr{A}}$ by utilizing a *level-ranking* function $f : Q \to [2n] \cup \{\perp\}$ for the states S at a level in the DAG $G_{w,\mathscr{A}}$ such that if $q \in S \cap F$, then $f(q)$ is even, and $f(q) = \perp$ if $q \in Q \setminus S$.

Definition 3 (Coverage Relation for Level Rankings)**.** *Let a be a letter in Σ and f, f' be two level ranking functions. We say f* covers *f'* under *letter a, denoted by $f' \leq^{\delta}_{a} f$, when for all $q, q' \in Q$, if $f(q) \geq 0$ and $q' \in \delta(q,a)$, then $0 \leq f'(q') \leq f(q)$, otherwise $f'(q') = \perp$.*

Note here that $\leq^{\delta}_{a}$ is defined based on the transition function δ. The coverage relation indicates that the level rankings f and f' of two consecutive levels of $G_{w,\mathscr{A}}$ do not increase in ranks. We denote by $\mathscr{R}$ the set of all possible level ranking functions.

In order to verify that the guess about the ranking of $G_{w,\mathscr{A}}$ is correct, RKC uses the *breakpoint construction* proposed in [22]. This construction employs a set of states $O \subseteq Q$ to check that the vertices assigned with even ranks are finite. Similarly to Lemma 5, the nonaccepting DAG $G_{w,\mathscr{A}}$ with the ranking function defined in (1) and (2) eventually reaches a stable level, after which all F-vertices are finite. Hence, a breakpoint construction suffices to verify such guesses.

The formal definition of the complementary NBW $\mathscr{A}^c$ of the input NBW $\mathscr{A}$ is given below.

Definition 4 ([18])**.** *Let $\mathscr{A} = (Q, I, \delta, F)$ be an NBW. We then define an NBW $\mathscr{A}^c = (Q^c, I^c, \delta^c, F^c)$ of $\mathscr{A}$ as follows.*

- *$Q^c \subseteq \mathscr{R} \times 2^Q$,*
- *$I^c = (f, \emptyset)$ where $f(q) = 2n$ if $q \in I$ and $f(q) = \perp$ otherwise.*
- *δ^c is defined as follows:*
  1. *if $O \neq \emptyset$, then $\delta^c((f,O),a) = \{\, (f', \delta(O,a) \setminus odd(f')) \mid f' \leq^{\delta}_{a} f \,\}$ (intuition: breakpoint O only tracks vertices assigned with even ranks),*
  2. *if $O = \emptyset$, then $\delta^c((f,O),a) = \{\, (f', even(f')) \mid f' \leq^{\delta}_{a} f \,\}$ (intuition: $O = \emptyset$ means all previous F-vertices with even ranks are finite, then verify new vertices with even ranks).*
- *$F^c = \{\, (f,O) \in Q^c \mid O = \emptyset \,\}$.*

where $odd(f) = \{\, q \in Q \mid f(q)$ is odd $\}$ and $even(f) = \{\, q \in Q \mid f(q)$ is even $\}$.

Let w be a word. Intuitively, every state (f,O) in $\mathscr{A}^c$ corresponds to a level of the DAG $G_{w,\mathscr{A}}$ over w. If w is accepted by $\mathscr{A}^c$, i.e., O becomes empty for infinitely many times, then we conclude that all the ω-branches of $G_{w,\mathscr{A}}$ eventually get trapped in odd ranks. It follows that no branches are accepting in $G_{w,\mathscr{A}}$, i.e., $w \notin \mathscr{L}(\mathscr{A})$. The other direction is also easy to prove and omitted here. Thus we conclude that $\mathscr{L}(\mathscr{A}^c) = \Sigma^\omega \setminus \mathscr{L}(\mathscr{A})$. Since $f \in \mathscr{R}$ is a function from Q to $[2n] \cup \{\perp\}$, the number of possible f functions is $(2n+2)^n \in 2^{\mathscr{O}(n \log n)}$. Therefore, the number of states in $\mathscr{A}$ is in $2^n \times 2^{\mathscr{O}(n \log n)} \in 2^{\mathscr{O}(n \log n)}$.

Lemma 7 (The Language and Size of $\mathscr{A}^c$ [18])**.** *Let $\mathscr{A}$ be an NBW with n states and $\mathscr{A}^c$ the NBW defined in Definition 4. Then $\mathscr{L}(\mathscr{A}^c) = \Sigma^\omega \setminus \mathscr{L}(\mathscr{A})$ and $\mathscr{A}^c$ has $2^{\mathscr{O}(n \log n)}$ states.*

Relation to Construction of Co-deterministic DAGs. Assume that we have two level-rankings $f' \leq_a^{\delta} f$. A state q' in the second level can have multiple a-predecessors defined in the domain of f. Then $f'(q') \leq \min\{\, f(q) \mid f(q) \neq \bot, q' \in \delta(q,a)\,\}$. Thus we can define a co-deterministic DAG out of $G_{w,\mathscr{A}}$ where each vertex only keeps one predecessor with the minimal rank in the reduced DAG, in contrast to the predecessor with minimal index in Section 3. There may, however, be multiple predecessors with the minimal rank. Consequently, the non-reduced DAG $G_{w,\mathscr{A}}$ can be mapped to multiple co-deterministic DAGs depending on which ranking function is defined on $G_{w,\mathscr{A}}$ and how predecessors are chosen. Note here that not every resulting co-deterministic DAG of $G_{w,\mathscr{A}}$ described above will be accepting if $G_{w,\mathscr{A}}$ is accepting, since each time the edges in accepting ω-branches may be deleted. Thus these co-deterministic DAGs cannot be directly applied in RKC for general NBWs.

4.2 Rank-Based Algorithm for FANBWs

In the following, we show in Lemma 8 that if $\mathscr{A}$ is an FANBW, the maximum rank of the vertices in a co-deterministic DAG of $\mathscr{A}$ is at most 2. It follows that the range of $f \in \mathscr{R}$ is $\{0,1,2\} \cup \{\bot\}$. We thus only need the maximum rank to be 2 rather than $2n$ for the co-deterministic DAG $G^e_{w,\mathscr{A}}$ of $\mathscr{A}$. Therefore, the number of states in $\mathscr{A}^c$ is in $2^n \times 4^n \in 2^{\mathscr{O}(n)}$ when the maximum rank is 2.

Lemma 8 (Maximum Rank of Co-deterministic DAGs)**.** *Assume that $\mathscr{A}$ is an FANBW and let w be a word. Let $G^e_{w,\mathscr{A}}$ be the co-deterministic DAG of $\mathscr{A}$ over w. Then $w \notin \mathscr{L}(\mathscr{A})$ iff $(G^e_{w,\mathscr{A}})^3$ is empty.*

Proof. Assume that $w \notin \mathscr{L}(\mathscr{A})$. Our goal is to prove that starting from $(G^e_{w,\mathscr{A}})^0 = G^e_{w,\mathscr{A}}$, $(G^e_{w,\mathscr{A}})^3$ is empty. By Lemma 5, there exists a stable level, say $k > 1$, such that on each level $l \geq k$, the F-vertices are finite. Therefore, $(G^e_{w,\mathscr{A}})^1$ contains only non-F-vertices after level k. It follows that $(G^e_{w,\mathscr{A}})^2$ removes all the vertices after level k. Thus if $(G^e_{w,\mathscr{A}})^2$ is not empty, $(G^e_{w,\mathscr{A}})^2$ contains only finite vertices. We then conclude that $(G^e_{w,\mathscr{A}})^3$ is empty. The other direction is trivial. □

In order to set the maximum rank to 2 in Definition 4, the underlying DAG $G_{w,\mathscr{A}}$ constructed for complementing FANBWs has to be co-deterministic. Since RKC generates rankings level by level, we have to utilize the reduced transition function δ^e for computing successors at next level. For FANBWs, the complementation construction in Definition 4 can be improved accordingly:

Definition 4'. *Let $\mathscr{A} = (Q, I, \delta, F)$ be an FANBW. We then define an NBW $\mathscr{A}^c = (Q^c, I^c, \delta^c, F^c)$, where Q^c and F^c are as in Definition 4, and I^c and δ^c are defined by:*

- *$I^c = (f, \emptyset)$ where $f(q) = 2$ if $q \in I$ and $f(q) = \bot$ otherwise.*
- *δ^c is then defined as follows:*
  1. *if $O \neq \emptyset$, then $\delta^c((f,O),a) = \{\, (f', \delta^e(O,a) \setminus odd(f')) \mid f' \leq_a^{\delta^e} f \,\}$,*
  2. *if $O = \emptyset$, then $\delta^c((f,O),a) = \{\, (f', even(f')) \mid f' \leq_a^{\delta^e} f \,\}$).*

  where δ^e is the reduced transition function at a level whose corresponding set of states and input letter are $\{\, q \in Q \mid f(q) \neq \bot \,\}$ and a, respectively.

Recall that the coverage relation between two level ranking functions f and f', parameterized with δ^e, is defined in Definition 3. Similarly to Definition 2, to compute $\delta^e(S_1, a)$, one has to first compute the minimal set S_{min} of predecessors of $S' = \delta(S,a)$ where S is the domain of f, i.e., the set of states at current level and a is the input letter at current level. Thus we have $\delta^e(S_1,a) = \delta(S_1 \cap S_{min}, a)$. Intuitively, for $w \in \Sigma^\omega$, δ^e is used to construct a co-deterministic DAG $G^e_{w,\mathscr{A}}$ over w level by level. By Lemma 8, the maximum rank of $G^e_{w,\mathscr{A}}$ is at most 2, which is sufficient in Definition 4' for constructing a ranking function to identify whether $G^e_{w,\mathscr{A}}$ is accepting. Therefore, with Definition 4', we can construct a complementary NBW $\mathscr{A}^c$ with $2^{\mathcal{O}(n)}$ states.

Theorem 1 (The Language and Size of $\mathscr{A}^c$ for FANBWs)**.** *Let $\mathscr{A}$ be an FANBW with n states and $\mathscr{A}^c$ the NBW defined in Definition 4'. Then (1) $\mathscr{L}(\mathscr{A}^c) = \Sigma^\omega \setminus \mathscr{L}(\mathscr{A})$; and (2) $\mathscr{A}^c$ has $2^{\mathcal{O}(n)}$ states.*

Proof. The proof for claim (2) is trivial and thus omitted here. By Lemma 2 and definition of ranking functions, co-deterministic DAGs of $\mathscr{A}$ over $w \in \mathscr{L}(\mathscr{A})$ will be rejected in $\mathscr{A}^c$, thus $\mathscr{L}(\mathscr{A}^c) \subseteq \Sigma^\omega \setminus \mathscr{L}(\mathscr{A})$. According to the proof of Lemma 8, there exists a unique ranking function for each rejecting co-deterministic DAG $G^e_{w,\mathscr{A}}$ of $\mathscr{A}$ over $w \notin \mathscr{L}(\mathscr{A})$. This unique ranking function can be constructed in a way similar to the one in Lemma 6. Since RKC nondeterministically guesses rankings of $G^e_{w,\mathscr{A}}$, there must be a guess of such unique ranking function. It follows that $G^e_{w,\mathscr{A}}$ must be accepting in $\mathscr{A}^c$, i.e., $\Sigma^\omega \setminus \mathscr{L}(\mathscr{A}) \subseteq \mathscr{L}(\mathscr{A}^c)$. Thus it holds that $\mathscr{L}(\mathscr{A}^c) = \Sigma^\omega \setminus \mathscr{L}(\mathscr{A})$. □

In [14], Fogarty and Vardi proved that complementing reverse deterministic NBWs with RKC is doable in $2^{\mathcal{O}(n)}$ as the non-reduced DAGs $G_{w,\mathscr{A}}$ are already co-deterministic. This is because that if $\mathscr{A}$ is reverse deterministic, then each vertex $\langle q,l \rangle$ in $G_{w,\mathscr{A}}$ has at most one predecessor, as q has only one $w[l]$-predecessor. It follows that $G_{w,\mathscr{A}}$ is co-deterministic. Similarly to Lemma 4, the number of (accepting) ω-branches in $G_{w,\mathscr{A}}$ is at most $|Q|$. According to Definition 1, reverse deterministic NBWs are a special class of FANBWs, as stated in Corollary 1.

Corollary 1. *Let $\mathscr{A}$ be a reverse deterministic NBW. Then $\mathscr{A}$ is also an FANBW.*

In contrast, an FANBW is not necessarily a reverse deterministic NBW. For instance, the FANBW $\mathscr{A}$ of Figure 1 is not reverse deterministic since q_1 has three b-predecessors, namely q_0, q_1 and q_2. We remark that the construction in [14] just sets the maximum rank to 2 in Definition 4 without modifying the transition function δ^c, which turns out to be a special case of our construction according to Corollary 1.

5 Slice-Based Algorithm

In Subsection 5.1, we first recall the *slice-based* complementation construction (SLC) described in [16, 28], adapted using our notations, which produces a complementary NBW $\mathscr{A}^c$ of $\mathscr{A}$ with $\mathcal{O}((3n)^n)$ states. Then, in Subsection 5.2, we show that for FANBWs, this construction can be simplified while yielding a complementary NBW with $\mathcal{O}(4^n)$ states.

5.1 Slice-Based Algorithm for NBWs

Let $\mathscr{A}$ be an NBW, and let w be a word. SLC uses a data structure called *slice* instead of level rankings to encode the set of vertices at the same level in $G_{w,\mathscr{A}}$. A slice in [28] is defined as an ordered sequence of disjoint sets of vertices at the same level.

We now describe SLC from the perspective of building co-deterministic DAGs. SLC does the following to construct a co-deterministic DAG $G^s_{w,\mathscr{A}}$ as it proceeds along the word w. Here the superscript s for SLC is used to distinguish the construction of co-deterministic DAGs $G^e_{w,\mathscr{A}}$ in Section 3. At level 0, we may obtain at most two vertices of $G^s_{w,\mathscr{A}}$: a vertex $\langle S_1, 0\rangle = \langle I \setminus F, 0\rangle$ and an F-vertex $\langle S_2, 0\rangle = \langle I \cap F, 0\rangle$. Recall that I and F are the set of initial states and the set of accepting states of $\mathscr{A}$, respectively. Here S_1 and S_2 are disjoint. A vertex $\langle S_j, i\rangle$ is an F-vertex if $S_j \subseteq F$, where $j \geq 1$ and $i \geq 0$. The vertices $\langle S_j, i\rangle$ on level i in $G^s_{w,\mathscr{A}}$ are ordered from left to right by their indices j where $i \geq 0$ and $1 \leq j \leq n$. During the construction, empty sets S_j are removed and the indices of remaining sets are reset according to the increasing order of their original indices.

Assume that on level i, the sequence of vertices in $G^s_{w,\mathscr{A}}$ is $\langle S_1, i\rangle, \cdots, \langle S_{k_i}, i\rangle$ where $i \geq 0$ and $1 \leq k_i \leq n$. We now describe how SLC constructs the vertices on level $i+1$. First, for a set S_j where $1 \leq j \leq k_i$, on reading the letter $w[i]$, the set of successors of S_j is partitioned into (1) a non-F set $S'_{2j-1} = \delta(S_j, w[i]) \setminus F$, and (2) an F-set $S'_{2j} = \delta(S_j, w[i]) \cap F$, as a possible new F-vertex.

This gives us a sequence of sets $S'_1, S'_2, \cdots, S'_{2k_i-1}, S'_{2k_i}$. Note that there can be some states in $\mathscr{A}$ present in multiple sets S'_j where $j \geq 1$. Here we only keep the rightmost occurrence of a state. Intuitively, different runs of $\mathscr{A}$ may merge with each other at some level and we only need to keep the right most one and cut off others, as they share the same infinite suffix. This operation does not change whether the co-deterministic DAG $G^s_{w,\mathscr{A}}$ is accepting, since at least one accepting run of $\mathscr{A}$ remains and will not be cut off. Formally, for each set S'_j where $1 \leq j \leq 2k_i$, we define a set $S''_j = S'_j \setminus \bigcup_{j<p\leq 2k_i} S'_p$. This yields a sequence of disjoint sets $S''_1, S''_2, \cdots, S''_{2k_i-1}, S''_{2k_i}$. After removing the empty sets in this sequence and reassigning the index of each set according to their positions, we finally obtain the sequence of sets of vertices on level $i+1$, denoted by $\langle S_1, l+1\rangle, \cdots, \langle S_{k_{i+1}}, l+1\rangle$. Obviously, the resulting sets at the same level are again pairwise disjoint.

Therefore, we define a co-deterministic DAG $G^s_{w,\mathscr{A}} = (V, E)$ of $\mathscr{A}$ over w for an NBW $\mathscr{A}$ as follows:

- Vertices. $V = \bigcup_{l\geq 0, 1\leq j\leq k_i} \{\langle S_j, l\rangle\}$.
- Edges. There is an edge from $\langle S_j, l\rangle$ to $\langle S_h, l+1\rangle$ iff S_h is either S''_{2j-1} or S''_{2j} as defined above where $1 \leq j \leq k_i$ and $1 \leq h \leq k_{i+1}$.

By the definition of $G^s_{w,\mathscr{A}}$, each vertex $\langle S_h, l+1\rangle$ in which S_h is either S''_{2j-1} or S''_{2j} computed from S_j has at most one predecessor $\langle S_j, l\rangle$. Thus $G^s_{w,\mathscr{A}}$ is co-deterministic. Similarly, we have the following Lemma 9.

Lemma 9 (Co-Deterministic DAGs for NBWs [28]). *Let $w \in \Sigma^\omega$ and $G^s_{w,\mathscr{A}}$ be the co-deterministic DAG as defined above. Then (1) the number of (accepting) ω-branches in $G^s_{w,\mathscr{A}}$ is at most the number of states in $\mathscr{A}$. (2) w is accepted by $\mathscr{A}$ if and only if $G^s_{w,\mathscr{A}}$ is accepting. (3) There exists a stable level $l \geq 1$ in $G^s_{w,\mathscr{A}}$ such that all F-vertices after level l are finite if and only if $w \notin \mathscr{L}(\mathscr{A})$.*

SLC for general NBWs can be viewed as consisting of two components: (1) based on the construction of co-deterministic DAGs $G^s_{w,\mathscr{A}}$ over w above, NBWs can be translated to FANBWs [21] and (2) a specialized complementation algorithm for FANBWs. In [28], SLC utilizes these two components at the same time for computing the complementary NBW $\mathscr{A}^c$.

A state of $\mathscr{A}^c$ is an ordered sequence of tuples $(S_1, l_1), \cdots, (S_h, l_h)$ where the ordered sequence $(S_1, \cdots, S_h)$ is a slice, and each vertex $\langle S_j, l\rangle$ is decorated with a label $l_j \in \{\mathsf{die}, \mathsf{inf}, \mathsf{new}\}$. The level index l is omitted during the construction of $\mathscr{A}^c$. Intuitively,

- die-labelled vertex means that those states in S_j are currently being inspected. For w to be accepted (i.e., $w \notin \mathscr{L}(\mathscr{A})$), die-labelled vertices should eventually reach empty set after a finitely many steps, thus become finite. Recall that empty sets will be removed in the construction of $G^s_{w,\mathscr{A}}$.
- inf-labelled vertex indicates all states that never reach accepting states.
- new-labelled vertex records new encountered states, that should be inspected later once the die-labelled vertex becomes empty.

Obviously, here h is at most the number n of states in $\mathscr{A}$. While for FANBWs, thanks to their finite ambiguity, the construction for co-deterministic DAGs can be simplified (see Section 3): we can even use three components (N, C, B) to compactly encode the slice and their labels. We postpone the details of the construction to the next subsection. Now we recall the complexity of the above slice based construction:

Lemma 10 (The Language and Size of $\mathscr{A}^c$ for NBWs [28]). *Let $\mathscr{A}$ be an NBW with n states and $\mathscr{A}^c$ the NBW constructed by* SLC *in Section 5. Then $\mathscr{L}(\mathscr{A}^c) = \Sigma^\omega \setminus \mathscr{L}(\mathscr{A})$ and $\mathscr{A}^c$ has $\mathscr{O}((3n)^n)$ states.*

5.2 Slice-Based Algorithm for FANBWs

We now propose the specialized complementation construction for FANBWs. Recall that, as discussed in Subsection 5.1, this construction is also the second component of SLC, used for complementing general NBWs.

We first provide some intuitions. According to Lemma 5, given a word $w \notin \mathscr{L}(\mathscr{A})$, there exists a stable level k in the co-deterministic DAG $G^e_{w,\mathscr{A}}$ such that each F-vertex on a level after k is finite. Therefore, in the construction of $\mathscr{A}^c$, we can nondeterministically guess level k and then use breakpoint construction to verify that our guess is correct, in analogy with RKC. More precisely, when constructing the complementary NBW $\mathscr{A}^c$, there are the *initial phase* and the *accepting phase*. The initial phase is purely a subset construction to trace the reachable states of each level of the co-deterministic DAG $G^e_{w,\mathscr{A}}$ over w. On reading a letter at a state of $\mathscr{A}^c$ (called *macrostate*) in the initial phase, the run of $\mathscr{A}^c$ over w (called *macrorun*) either continues to stay in the initial phase or jumps to the accepting phase. Once entering the accepting phase, we guess that the macrorun of $\mathscr{A}^c$, which consists of multiple runs of $\mathscr{A}$, has reached the stable level k. Thus in the accepting phase, we need a breakpoint construction to verify that the guess is correct, i.e., that all F-vertices after level k are finite.

In the accepting phase, we use a macrostate, represented as a triple (N,C,B), to encode the set of vertices and their labels on a level after k in the co-deterministic DAG $G^e_{w,\mathscr{A}}$ (or $G^s_{w,\mathscr{A}}$ for general NBWs accordingly), where

- the set N keeps all the reachable vertices on the level, corresponding to the set of all vertices labelled with die, inf and new;
- the set C keeps all the finite vertices on the level. That means, it contains both new-labelled vertices recording new encountered states, and die-labelled vertices being inspected now.
- the set $B \subseteq C$ as a breakpoint construction is used to verify that the guess on the set C of finite vertices is correct, corresponding to the set of vertices labelled with die.

Recall that die, inf and new are three labels of vertices used in SLC for complementing general NBWs, as described in Subsection 5.1. The specialized complementation algorithm for FANBWs is formalized below.

Definition 5. *Let $\mathscr{A} = (Q,I,\delta,F)$ be an FANBW. We then define an NBW $\mathscr{A}^c = (Q^c,I^c,\delta^c,F^c)$ as follows.*

- $Q^c \subseteq 2^Q \cup 2^Q \times 2^Q \times 2^Q$;
- $I^c = \{I\}$;
- $\delta^c = {\delta_1}^c \cup {\delta_t}^c \cup {\delta_2}^c$ *is defined as follows:*
  1. ${\delta_1}^c(S,a) = \delta^e(S,a)$ *for $S \subseteq Q$ and $a \in \Sigma$ where δ^e is the reduced transition function at current level whose corresponding set of states and input letter are S and a, respectively. (intuition: subset construction to organize the macrorun before the guess point).*
  2. ${\delta_t}^c(S,a) = {\delta_2}^c((N,C,B),a)$ *where $N = S, B = S \cap F$ and $C = B$ (intuition: make the guess point to be the macrostate (N,C,B)).*

3. $\delta_2{}^c((N,C,B),a) = (N',C',B')$ *where* δ^e *is the reduced transition function at current level whose corresponding set of states and input letter are* N *and* a*, respectively, and*
   – $N' = \delta^e(N,a)$ *(intuition: tracing the reachable states correctly),*
   – $C' = \delta^e(C,a) \cup (N' \cap F)$ *(intuition: tracing the runs which has visited accepting states after the guess point), and*
   – *if* $B \neq \emptyset$*, then* $B' = \delta^e(B,a)$ *and otherwise* $B' = C'$ *(intuition:* $B = \emptyset$ *means all runs which have visited accepting states are finite and* $B \neq \emptyset$ *indicates that previous runs are still under inspection).*

- $F^c = \{(N,C,B) \in Q^c \mid B = \emptyset\}$.

Remark 1. *As a side remark, we note that the complementary NBW constructed by Definition 5 is* limit deterministic*, as the state set* Q^c *of* $\mathscr{A}^c$ *can be partitioned into two disjoint sets* $Q^c_N \subseteq 2^Q$ *and* $Q^c_D \subseteq 2^Q \times 2^Q \times 2^Q$ *such that 1)* $F^c \subseteq Q^c_D$ *and 2) for each state* $q \in Q^c_D$ *and* $a \in \Sigma$*, we have that* $|\delta^c(q,a)| \leq 1$.

Theorem 2 (The Language and Size of $\mathscr{A}^c$ for FANBWs)**.** *Let* $\mathscr{A}$ *be an FANBW with n states and* $\mathscr{A}^c$ *be the NBW defined by Definition 5. Then (1)* $\mathscr{L}(\mathscr{A}^c) = \Sigma^\omega \setminus \mathscr{L}(\mathscr{A})$*; and (2)* $\mathscr{A}^c$ *has* $2^n + 4^n$ *states.*

Proof. We prove claim (1) as follows. Suppose $w \in \mathscr{L}(\mathscr{A})$, our goal is to prove w is not accepted by $\mathscr{A}^c$. Assume that the corresponding accepting run of $\mathscr{A}$ over w is ρ and ρ' is a macrorun of $\mathscr{A}^c$ over w. Then for the macrorun ρ': (1) if ρ' only visits states of the form $s \in 2^Q$, then ρ' is not accepted by $\mathscr{A}^c$ since no accepting $\mathscr{A}^c$-states will be visited; (2) if ρ' is a macrorun of the form $s_0, \cdots s_{k-1}, (N_k,C_k,B_k)(N_{k+1},C_{k+1},B_{k+1})\cdots$, ρ will visit some accepting state, say $q_f \in F$ infinitely often. Then at some point, say in state (N_j,C_j,B_j), we have $q_f \in B_j$ or $q_f \in C_j$. If $q_f \in B_j$, then for every $p \geq j$, we have $B_p \neq \emptyset$ according to Lemma 1; otherwise $q_f \in C_j$, then either at some point, say $p > j$, q_f will be moved to B_p when $B_{p-1} = \emptyset$, or $q_f \in C_p$ for each $p \geq j$, which indicates that $B_p \neq \emptyset$ for $p \geq j$. Therefore, w is not accepted by $\mathscr{A}^c$.

Assume that $w \notin \mathscr{L}(\mathscr{A})$, our goal is to prove that there exists an accepting macrorun ρ' of $\mathscr{A}^c$ over w. The proof idea is to analyze the co-deterministic DAG $G^e_{w,\mathscr{A}}$ of $\mathscr{A}$ over w. According to Lemma 5, there exists a stable level $k \geq 1$ such that every F-vertex on a level after k of $G^e_{w,\mathscr{A}}$ is finite. Therefore, the set B on ρ' will become empty infinitely often, i.e., w is accepted by $\mathscr{A}^c$.

We now prove claim (2). By Definition 5, the number of possible states of the form $s \in 2^Q$ is 2^n. For each state $p = (N,C,B) \in Q^c$ of $\mathscr{A}^c$, we have that $C \subseteq N$ and $B \subseteq C$. Then for a state $q \in Q$: (i) it will either be absent or present in N; (ii) for a state $q \in N$, one of the following three possibilities holds: q is only in N, q is both in C and N and q is both in B and C. Therefore $\mathscr{A}^c$ has at most $2^n + 4^n$ states. $\square$

As a consequence of Definition 5, we can define a subsumption relation between the macrostates of $\mathscr{A}^c$ below.

Corollary 2 (Subsumption Relation between Macrostates)**.** *Let* $\mathscr{A}$ *be an FANBW and* $\mathscr{A}^c$ *the complementary NBW of* $\mathscr{A}$ *defined by Definition 5, and* $m = (N,C,B)$ *and* $m' = (N',C',B')$ *are two macrostates of* $\mathscr{A}^c$ *such that* $N = N'$ *and* $C \subseteq C'$*. Then* $\mathscr{L}((\mathscr{A}^c)^{m'}) \subseteq \mathscr{L}((\mathscr{A}^c)^m)$ *or m subsumes m'.*

Proof. Let $w = a_0a_1\cdots \in \Sigma^\omega$. Let $\rho = (N_0 = N, C_0 = C, B_0 = B)(N_1,C_1,B_1)\cdots(N_k,C_k,B_k)\cdots$ be the macrorun of $(\mathscr{A}^c)^m$ over w. Similarly, the macrorun of $(\mathscr{A}^c)^{m'}$ over w is $\rho' = (N'_0 = N', C'_0 = C', B'_0 = B')(N'_1,C'_1,B'_1)\cdots(N'_k,C'_k,B'_k)\cdots$. Assume that $w \in \mathscr{L}((\mathscr{A}^c)^{m'})$, i.e., there are infinitely many empty B'-sets in ρ' according to Definition 5. It follows that the level 0 in the co-deterministic DAG $G^e_{w,\mathscr{A}^{N'}}$ of $\mathscr{A}^{N'}$ over w is a stable level, i.e., each F-vertex in $G^e_{w,\mathscr{A}^{N'}}$ is finite. (Recall that $\mathscr{A}^{N'}$ is an NBW obtained from $\mathscr{A}$ by setting the set of initial states of $\mathscr{A}$ to N'.) This is because that by Definition 5, each branch from an F-vertex in $G^e_{w,\mathscr{A}^{N'}}$ will eventually be put in the B'-set and if one such branch is not finite, the B'-set will become empty for only finitely many times, contradicting with the assumption that $w \in \mathscr{L}((\mathscr{A}^c)^{m'})$. By definition of the construction of co-deterministic DAGs in Section 3, the co-deterministic DAG $G^e_{w,\mathscr{A}^N}$ of $\mathscr{A}^N$ over w is identical to $G^e_{w,\mathscr{A}^{N'}}$ since $N = N'$. Consequently, the level 0 is also a stable level in $G^e_{w,\mathscr{A}^N}$. That is, each F-vertex in $G^e_{w,\mathscr{A}^N}$ is also finite. Since the B'-set in ρ' becomes empty and is reset to C' for infinitely many times, all branches from C' are finite. It follows that all the branches from $B \subseteq C$ are also finite since $C \subseteq C'$. Then there exists a least integer $j \geq 0$ in ρ such that $B_j = \emptyset$. Since all branches in the C-set (including new branches coming from the N-set) are finite, there are infinitely many integers $k \geq j$ such that $B_k = \emptyset$ in ρ. It follows that $w \in \mathscr{L}((\mathscr{A}^c)^m)$, which indicates that $\mathscr{L}((\mathscr{A}^c)^{m'}) \subseteq \mathscr{L}((\mathscr{A}^c)^m)$. □

Corollary 2 provides the possibility to avoid the exploration of m' when $\mathscr{L}((\mathscr{A}^c)^m)$ has already been found to be empty, when checking the language-containment between an NBW and an FANBW $\mathscr{A}$. It follows that one can also use this subsumption relation to avoid construction of redundant macrostates during the construction of $\mathscr{A}^c$, thus reducing the number of macrostates in $\mathscr{A}^c$.

6 Conclusion and Future Work

This work exploits co-deterministic DAGs over infinite words as a unified tool to optimize both RKC and SLC constructions. Consequently, we have improved the complexity of the classical RKC and SLC constructions for FANBWs, respectively, to $2^{\mathcal{O}(n)}$ from $2^{\mathcal{O}(n \log n)}$ and to $\mathcal{O}(4^n)$ from $\mathcal{O}((3n)^n)$, based on co-deterministic DAGs. As a further contribution, we view the SLC algorithm explicitly as the construction of co-deterministic DAGs and a specialized complementation algorithm for FANBWs. We then provide a subsumption relation between states in the complementary NBWs of FANBWs in hope of improving the containment checking between an NBW and an (FA)NBW.

As future work, we plan to study whether $\mathcal{O}(4^n)$ is also the lower bound for the complementation of FANBWs. An empirical evaluation on how the subsumption relation between macrostates proposed in Corollary 2 will benefit the containment checking problem is worthy of exploring. Moreover, we will also explore a Ramsey-based complementation construction based on co-deterministic DAGs. Another line of future work is studying determinization constructions for FANBWs. Finally, it is possible to use our work to improve the program-termination checking framework proposed in [15] if one generalizes a terminating path to an FANBW.

Acknowledgment We thank Rachel Faran, Yih-Kuen Tsay and anonymous reviewers for their valuable inputs at different stages to this project. This work is partially supported by Key-Area Research and Development Program of Guangdong Province (grant no. 2018B010107004), the National Natural Science Foundation of China (grant nos. 61761136011, 61532019), NSF grants IIS-1527668, CCF-1704883, IIS-1830549, and an award from the Maryland Procurement Office.

References

[1] Parosh Aziz Abdulla, Yu-Fang Chen, Lorenzo Clemente, Lukás Holík, Chih-Duo Hong, Richard Mayr & Tomás Vojnar (2010): *Simulation Subsumption in Ramsey-Based Büchi Automata Universality and Inclusion Testing*. In: *CAV*, *LNCS* 6174, Springer, pp. 132–147, doi:10.1007/978-3-642-14295-6_14.

[2] Parosh Aziz Abdulla, Yu-Fang Chen, Lorenzo Clemente, Lukás Holík, Chih-Duo Hong, Richard Mayr & Tomás Vojnar (2011): *Advanced Ramsey-Based Büchi Automata Inclusion Testing*. In: *CONCUR*, *LNCS* 6901, Springer, pp. 187–202, doi:10.1007/978-3-642-23217-6_13.

[3] Christel Baier & Joost-Pieter Katoen (2008): *Principles of model checking*. MIT press.

[4] Christel Baier, Stefan Kiefer, Joachim Klein, Sascha Klüppelholz, David Müller & James Worrell (2016): *Markov chains and unambiguous Büchi automata*. In: *CAV*, Springer, pp. 23–42, doi:10.1007/978-3-319-41528-4_2.

[5] František Blahoudek, Matthias Heizmann, Sven Schewe, Jan Strejček & Ming-Hsien Tsai (2016): *Complementing Semi-deterministic Büchi Automata*. In: *TACAS*, *LNCS* 9636, pp. 770–787, doi:10.1007/978-3-662-49674-9_49.

[6] Nicolas Bousquet & Christof Löding (2010): *Equivalence and inclusion problem for strongly unambiguous Büchi automata*. In: *LATA*, Springer, pp. 118–129, doi:10.1007/978-3-642-13089-2_10.

[7] J Richard Büchi (1990): *On a decision method in restricted second order arithmetic*. In: *The Collected Works of J. Richard Büchi*, Springer, pp. 425–435, doi:10.1007/978-1-4613-8928-6_23.

[8] D. Bustan, S. Rubin & M.Y. Vardi (2004): *Verifying omega-Regular Properties of Markov Chains*. In: *CAV*, *LNCS* 3114, Springer, pp. 189–201, doi:10.1007/978-3-540-27813-9_15.

[9] Olivier Carton & Max Michel (2003): *Unambiguous Büchi automata*. *Theoretical Computer Science* 297(1-3), pp. 37–81, doi:10.1016/S0304-3975(02)00618-7.

[10] Yu-Fang Chen, Matthias Heizmann, Ondrej Lengál, Yong Li, Ming-Hsien Tsai, Andrea Turrini & Lijun Zhang (2018): *Advanced automata-based algorithms for program termination checking*. In: *PLDI*, pp. 135–150, doi:10.1145/3192366.3192405.

[11] Lorenzo Clemente & Richard Mayr (2019): *Efficient reduction of nondeterministic automata with application to language inclusion testing*. *Logical Methods in Computer Science* 15(1), doi:10.23638/LMCS-15(1:12)2019.

[12] C. Courcoubetis & M. Yannakakis (1995): *The Complexity of Probabilistic Verification*. *J. ACM* 42(4), pp. 857–907, doi:10.1145/210332.210339.

[13] Laurent Doyen & Jean-François Raskin (2009): *Antichains for the Automata-Based Approach to Model-Checking*. *Logical Methods in Computer Science* 5(1), doi:10.2168/LMCS-5(1:5)2009.

[14] Seth Fogarty & Moshe Y. Vardi (2012): *Büchi Complementation and Size-Change Termination*. *Logical Methods in Computer Science* 8(1), doi:10.2168/LMCS-8(1:13)2012.

[15] Matthias Heizmann, Jochen Hoenicke & Andreas Podelski (2014): *Termination Analysis by Learning Terminating Programs*. In: *CAV*, pp. 797–813, doi:10.1007/978-3-319-08867-9_53.

[16] Detlef Kähler & Thomas Wilke (2008): *Complementation, disambiguation, and determinization of Büchi automata unified*. In: *ICALP*, Springer, pp. 724–735, doi:10.1007/978-3-540-70575-8_59.

[17] Orna Kupferman & Moshe Y. Vardi (1996): *Verification of Fair Transisiton Systems*. In Rajeev Alur & Thomas A. Henzinger, editors: *CAV*, *LNCS* 1102, Springer, pp. 372–382, doi:10.1007/3-540-61474-5_84.

[18] Orna Kupferman & Moshe Y Vardi (2001): *Weak alternating automata are not that weak*. *ACM Transactions on Computational Logic* 2(3), pp. 408–429, doi:10.1145/377978.377993.

[19] Robert P. Kurshan (1987): *Complementing Deterministic Büchi Automata in Polynomial Time*. *J. Comput. Syst. Sci.* 35(1), pp. 59–71, doi:10.1016/0022-0000(87)90036-5.

[20] Yong Li, Wanwei Liu, Andrea Turrini, Ernst Moritz Hahn & Lijun Zhang (2016): *An Efficient Synthesis Algorithm for Parametric Markov Chains Against Linear Time Properties*. In: *SETTA*, pp. 280–296, doi:10.1007/978-3-319-47677-3_18.

[21] Christof Löding & Anton Pirogov (2018): *On Finitely Ambiguous Büchi Automata*. In: *DLT*, pp. 503–515, doi:10.1007/978-3-319-98654-8_41.

[22] Satoru Miyano & Takeshi Hayashi (1984): *Alternating finite automata on ω-words*. *Theoretical Computer Science* 32(3), pp. 321–330, doi:10.1016/0304-3975(84)90049-5.

[23] Alexander Rabinovich (2018): *Complementation of Finitely Ambiguous Büchi Automata*. In: *DLT*, Springer, pp. 541–552, doi:10.1007/978-3-319-98654-8_44.

[24] Shmuel Safra (1988): *On the complexity of ω-automata*. In: *FOCS*, IEEE, pp. 319–327, doi:10.1109/SFCS.1988.21948.

[25] S. Schewe (2009): *Büchi Complementation Made Tight*. In: *STACS*, *LIPIcs* 3, Schloss Dagstuhl, Germany, pp. 661–672, doi:10.4230/LIPIcs.STACS.2009.1854.

[26] A Prasad Sistla, Moshe Y Vardi & Pierre Wolper (1987): *The complementation problem for Büchi automata with applications to temporal logic*. *Theoretical Computer Science* 49(2-3), pp. 217–237, doi:10.1016/0304-3975(87)90008-9.

[27] M.-H. Tsai, S. Fogarty, M.Y. Vardi & Y.-K. Tsay (2014): *State of Büchi Complementation*. *Logical Methods in Computer Science* 10(4), doi:10.2168/LMCS-10(4:13)2014.

[28] Moshe Y. Vardi & Thomas Wilke (2008): *Automata: from logics to algorithms*. In: *Logic and Automata: History and Perspectives*, pp. 629–736.

[29] Moshe Y. Vardi & Pierre Wolper (1986): *An Automata-Theoretic Approach to Automatic Program Verification (Preliminary Report)*. In: *LICS*, IEEE, pp. 332–344.

[30] Qiqi Yan (2008): *Lower Bounds for Complementation of ω-Automata Via the Full Automata Technique*. *Logical Methods in Computer Science* 4(1:5), doi:10.2168/LMCS-4(1:5)2008.