Congruence Relations for Büchi Automata

Yong Li[1], Yih-Kuen Tsay[2], Andrea Turrini[1,3], Moshe Y. Vardi[4], and Lijun Zhang[1,3,5(✉)]

[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
zhanglj@ios.ac.cn
[2] National Taiwan University, Taipei, Taiwan
[3] Institute of Intelligent Software, Guangzhou, China
[4] Rice University, Houston, USA
[5] University of Chinese Academy of Sciences, Beijing, China

Abstract. We revisit congruence relations for Büchi automata, which play a central role in automata-based formal verification. The size of the classical congruence relation is in $3^{\mathcal{O}(n^2)}$, where n is the number of states of the given Büchi automaton. We present improved congruence relations that can be exponentially coarser than the classical one. We further give asymptotically *optimal* congruence relations of size $2^{\mathcal{O}(n \log n)}$. Based on these optimal congruence relations, we obtain an *optimal* translation from a Büchi automaton to a family of deterministic finite automata (FDFA), which can be made to accept either the original language or its complement. To the best of our knowledge, our construction is the *first direct* and *optimal* translation from Büchi automata to FDFAs.

1 Introduction

Congruence relations for nondeterministic Büchi automata (NBAs) [6] are fundamental for Büchi complementation, a key operation used in the formal verification framework based on automata theory [14]. To formally verify whether the behavior of a system A satisfies a given specification B, one usually reduces this problem to a language-containment problem between the NBAs A and B; this containment problem is then reduced to the nonemptiness of the intersection of A and the complement of B. The first complementation construction for Büchi automata, proposed by Büchi [6] and widely known as Ramsey-based Büchi complementation (RBC), relies on a congruence relation with a $2^{2^{\mathcal{O}(n)}}$ blow-up, where n is the number of states of the input automaton. One can associate each equivalence class of the congruence relation with a state of the complementary automaton, similarly to the characterization provided by the Myhill-Nerode theorem for regular languages [22]. The blow-up of the congruence relation of RBC was later reduced by Sistla *et al.* [21] to $3^{\mathcal{O}(n^2)}$, without providing an explicit formal notion of congruence relation, which was later formalized by Thomas [22].

M. Huisman et al. (Eds.): FM 2021, LNCS 13047, pp. 465–482, 2021.
https://doi.org/10.1007/978-3-030-90870-6_25

Notably, current practical approaches to the containment checking for NBAs are based on the classical congruence relation given in [21,22], even though it has a larger blow-up ($3^{\mathcal{O}(n^2)}$ vs. $2^{\mathcal{O}(n \log n)}$) than other optimal complementation constructions, such as the *rank-based* complementation [15]. In fact, RABIT, the state-of-the-art tool for checking language-containment between NBAs, is also based on the classical congruence relation of [21,22] and has integrated various state-space pruning techniques for RBC, proposed in [1,2,8].

In another line of work, *families of deterministic finite automata* (FDFAs) [3] have been proposed for representing ω-regular languages, as an alternative to NBAs. By modelling a given system and specification as FDFAs, the formal verification problem can be reduced to a containment problem between two FDFAs, which can be done in polynomial time [3], in contrast to PSPACE-completeness for NBAs [14]. It has been shown that an FDFA can be induced from a congruence relation defined over a given ω-regular language, where each state of the FDFA corresponds to an equivalence class of the congruence relation [4].

In this work we show that RBC and FDFAs have an intimate connection: congruence relations for NBAs constitute the underlying concept that connects them. This connection gives us the possibility to further tighten the congruence relations for both RBC and FDFAs (see Sects. 3.2 and 4). In fact, the state-space pruning techniques developed in [1,2] for RBC [21,22] are inherently heuristics for identifying subsumption and simulation relations between congruence relations of RBC. Therefore, in order to further theoretically or empirically improve model-checking algorithms based on RBC or FDFAs, it is important to understand congruence relations for both FDFAs and RBC and, hopefully, make their congruence relations coarser. This motivates our search for better congruence relations for NBAs.

Contribution. We focus here on an in-depth study of congruence relations for NBAs and their connection to FDFAs. First, we show how to improve the classical congruence relation $\backsim$ with a blow-up of $3^{\mathcal{O}(n^2)}$, defined by the classical RBC, to congruence relations that can be exponentially tighter (Theorem 3), but can never be larger than the classical congruence relation $\backsim$ (Theorem 2). Notably, the improved congruence relations only have a blow-up of $\mathcal{O}(n^2)$ when dealing with deterministic Büchi automata (Theorem 4). Second, we further propose congruence relations for NBAs with a blow-up of only $2^{\mathcal{O}(n \log n)}$ (Lemma 11), which is then proved to be optimal (Theorem 7). Finally, we show that our congruence relations define an FDFA recognizing $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$ from an NBA $\mathcal{A}$. In particular, if $\mathcal{A}$ has n states, then our optimal congruence relations yield an FDFA $\mathcal{F}$ with an optimal complexity $2^{\mathcal{O}(n \log n)}$. Thus, to the best of our knowledge, we present the *first direct* translation from an NBA to an FDFA with *optimal* complexity. Missing proofs can be found in [18].

2 Preliminaries

Fix an *Alphabet* Σ. A *word* is a finite or infinite sequence of letters in Σ; ϵ denotes the empty word. Let Σ^* and Σ^ω denote the set of all finite and infinite

words (or ω-words), respectively. In particular, we let $\Sigma^+ = \Sigma^* \backslash \{\epsilon\}$. A *finitary language* is a subset of Σ^*; an *ω-language* is a subset of Σ^ω. Let L be a finitary language (resp., ω-language); the complementary language of L is $\Sigma^* \backslash L$ (resp., $\Sigma^\omega \backslash L$). Let ρ be a sequence; we denote by $\rho[i]$ the i-th element of ρ and by $\rho[i..k]$ the subsequence of ρ starting at the i-th element and ending at the k-th element inclusively when $i \leq k$, and the empty sequence ϵ when $i > k$. Given a finite word u and a word w, we denote by $u \cdot w$ (uw, for short) the concatenation of u and w. Given a finitary language L_1 and a finitary/ω-language L_2, the concatenation $L_1 \cdot L_2$ ($L_1 L_2$, for short) of L_1 and L_2 is the set $L_1 \cdot L_2 = \{ uw \mid u \in L_1, w \in L_2 \}$ and L_1^ω the infinite concatenation of L_1.

NBAs. A (nondeterministic) automaton is a tuple $\mathcal{A} = (Q, I, \delta, F)$, where Q is a finite set of states, $I \subseteq Q$ is a set of initial states, $\delta \colon Q \times \Sigma \to 2^Q$ 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 also extend δ to words, by letting $\delta(S, \epsilon) = S$ and $\delta(S, a_1 a_2 \cdots a_k) = \delta(\delta(S, a_1), \cdots, a_k)$, where we have $k \geq 1$ and $a_i \in \Sigma$ for $i \in \{1, \cdots, k\}$. An automaton on finite words is called a *nondeterministic finite automaton* (NFA), while an automaton on ω-words is called a *nondeterministic Büchi automaton* (NBA). An NFA $\mathcal{A}$ is said to be a *deterministic* finite automaton (DFA) if $|I| = 1$ and for each $q \in Q$ and $a \in \Sigma$, $|\delta(q, a)| \leq 1$. Deterministic Büchi automata (DBAs) are defined similarly.

A *run* of an NFA/NBA $\mathcal{A}$ on a finite word u of length $n \geq 0$ is a sequence of states $\rho = q_0 q_1 \cdots q_n \in Q^+$, such that for every $0 < i \leq n$, $q_i \in \delta(q_{i-1}, u[i])$. We write $q_0 \xrightarrow{u} q_n$ if there is a run from q_0 to q_n over u and by $q_0 \overset{u}{\Rightarrow} q_n$ if such a run also visits an accepting state. Obviously, we have that $q \xrightarrow{\epsilon} q$ for all $q \in Q$ and $q \overset{\epsilon}{\Rightarrow} q$ for all $q \in F$. A finite word $u \in \Sigma^*$ is *accepted* by an NFA $\mathcal{A}$ if there is a run $q_0 \cdots q_n$ over u such that $q_0 \in I$ and $q_n \in F$. Similarly, an *ω-run* of $\mathcal{A}$ on an ω-word w is an infinite sequence of states $\rho = q_0 q_1 \cdots$ such that $q_0 \in I$ and for every $i > 0$, $q_i \in \delta(q_{i-1}, w[i])$. Let $\mathit{Inf}(\rho)$ be the set of states that occur infinitely often in the run ρ. An ω-word $w \in \Sigma^\omega$ is *accepted* by an NBA $\mathcal{A}$ if there exists an ω-run ρ of $\mathcal{A}$ over w such that $\mathit{Inf}(\rho) \cap F \neq \emptyset$. The *finitary language* recognized by an NFA $\mathcal{A}$, denoted by $\mathcal{L}_*(\mathcal{A})$, is defined as the set of finite words accepted by it. Similarly, we denote by $\mathcal{L}(\mathcal{A})$ the *ω-language* recognized by an NBA $\mathcal{A}$, i.e., the set of ω-words accepted by $\mathcal{A}$. NFAs/DFAs accept exactly *regular* languages while NBAs recognize exactly *ω-regular* languages. The complementary automaton of an NBA $\mathcal{A}$ accepts the complementary language of $\mathcal{L}(\mathcal{A})$, i.e., $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$.

Congruence Relations. A *right congruence* (RC) relation is an equivalence relation $\backsim$ over Σ^* such that $x \backsim y$ implies $xv \backsim yv$ for all $v \in \Sigma^*$. A *congruence relation* is an equivalence relation $\backsim$ over Σ^* such that $x \backsim y$ implies $uxv \backsim uyv$ for every $x, y, u, v \in \Sigma^*$. We denote by $|\backsim|$ the index of $\backsim$, i.e., the number of equivalence classes of $\backsim$. A *finite congruence relation* is a congruence relation with a finite index. We denote by $\Sigma^*/_\backsim$ the set of equivalence classes of Σ^* under $\backsim$; we use $\Sigma^+/_\backsim$ to denote the same set of equivalence classes excluding ϵ. Given $x \in \Sigma^*$, we denote by $[x]_\backsim$ the equivalence class of $\backsim$ that x belongs to.

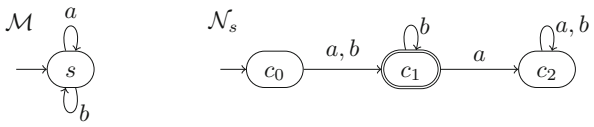

Fig. 1. An example of FDFA $\mathcal{F} = (\mathcal{M}, \{\mathcal{N}_s\})$ which is not saturated.

For a given right congruence $\backsim$ of a regular language L, it is well-known that the Myhill-Nerode theorem [19,20] defines a unique minimal DFA D of L, in which each state of D corresponds to an equivalence class defined by $\backsim$ over Σ^*. Therefore, we can construct a DFA $\mathcal{D}[\backsim]$ from $\backsim$ in a standard way.

Definition 1 ([19,20]). *Let $\backsim$ be a right congruence of finite index. The DFA $\mathcal{D}[\backsim]$ without accepting states induced by $\backsim$ is a tuple $(S, s_0, \delta_\mathcal{D}, \emptyset)$ where $S = \Sigma^*/_\backsim$, $s_0 = [\epsilon]_\backsim$, and for each $u \in \Sigma^*$ and $a \in \Sigma$, $\delta_\mathcal{D}([u]_\backsim, a) = [ua]_\backsim$.*

The DFA $\mathcal{D}[\backsim]$ is parametric on $\backsim$, indicating that it is induced by the right congruence relation $\backsim$. We may just write $\mathcal{D}$ if $\backsim$ is clear from the context.

UP-Words. The ω-regular languages accepted by NBAs can also be recognized by FDFAs by means of their *ultimately periodic words* (UP-words) [3]. A UP-word w is an ω-word of the form uv^ω, where $u \in \Sigma^*$ and $v \in \Sigma^+$. Thus $w = uv^\omega$ can be represented as a pair of finite words (u, v), called a *decomposition* of w. A UP-word can have multiple decompositions: for instance (u, v), (uv, v), and (u, vv) are all decompositions of uv^ω. For an ω-language L, let $\mathrm{UP}(L) = \{\, uv^\omega \in L \mid u \in \Sigma^* \wedge v \in \Sigma^+ \,\}$ denote the set of all UP-words in L. The set of UP-words of an ω-regular language L can be seen as the fingerprint of L, as stated below.

Theorem 1 ([7]). *(1) Every non-empty ω-regular language L contains at least one UP-word. (2) Let L and L' be two ω-regular languages. Then $L = L'$ if and only if $UP(L) = UP(L')$.*

FDFAs. Based on Theorem 1, Angluin *et al.* introduced in [3] the notion of FDFAs as another type of automata to recognize ω-regular languages.

Definition 2 (FDFAs [3]). *An FDFA is a pair $\mathcal{F} = (\mathcal{M}, \{\mathcal{N}_q\})$ consisting of a leading DFA $\mathcal{M}$ and of a progress DFA $\mathcal{N}_q$ for each state q in $\mathcal{M}$.*

Intuitively, the leading DFA $\mathcal{M}$ of $\mathcal{F} = (\mathcal{M}, \{\mathcal{N}_q\})$ for an ω-regular language L consumes the finite prefix u of a UP-word $uv^\omega \in \mathrm{UP}(L)$, reaching some state q, and for each state q of $\mathcal{M}$, the progress DFA $\mathcal{N}_q$ accepts the period v of uv^ω. An example of FDFA $\mathcal{F}$ is depicted in Fig. 1 where the leading DFA $\mathcal{M}$ has only the state s and the progress DFA associated with s is $\mathcal{N}_s$. Note that the leading DFA $\mathcal{M}$ of every FDFA does not make use of accepting states.

Let $\mathcal{D}$ be a DFA with initial state q_0 and transition function δ. Given a word $u \in \Sigma^*$, we often use $\mathcal{D}(u)$ as a shorthand for $\delta(q_0, u)$. Each FDFA $\mathcal{F}$ characterizes a set of UP-words $\mathrm{UP}(\mathcal{F})$ by following the acceptance condition.

Definition 3 (FDFA Acceptance). *Let $\mathcal{F} = (\mathcal{M}, \{\mathcal{N}_q\})$ be an FDFA and w be a UP-word. A decomposition (u, v) of w is* normalized *with respect to $\mathcal{F}$ if $\mathcal{M}(u) = \mathcal{M}(uv)$.*[1] *A decomposition (u, v) is accepted by $\mathcal{F}$ if (u, v) is normalized and we have $v \in \mathcal{L}_*(\mathcal{N}_q)$ where $q = \mathcal{M}(u)$. The UP-word w is accepted by $\mathcal{F}$ if there exists a decomposition (u, v) of w accepted by $\mathcal{F}$.*

Note that the normalized decomposition (u, v) is defined with respect to $\mathcal{F}$. We usually omit $\mathcal{F}$ and just say (u, v) is normalized when $\mathcal{F}$ is clear from the context. Consider again the FDFA $\mathcal{F}$ from Fig. 1: $(aba)^\omega$ is not accepted since no decomposition of $(aba)^\omega$ is accepted, while $(ab)^\omega$ is accepted since the decomposition (ab, ab) of $(ab)^\omega$ is such that $\mathcal{M}(ab \cdot ab) = \mathcal{M}(ab) = s$ and $ab \in \mathcal{L}_*(\mathcal{N}_s)$.

One can observe that the normalized decomposition $(ab, abab)$ of $(ab)^\omega$ is not accepted by $\mathcal{F}$, despite that (ab, ab) is accepted by $\mathcal{F}$. In the following, we define a class of FDFAs that *saturates* every accepting normalized decomposition $(ab, (ab)^k)$ of $(ab)^\omega$ (where $k \geq 1$) if (ab, ab) is accepted, which is important for FDFAs to recognize ω-regular languages [3,17].

Definition 4 (Saturation of FDFAs [3]). *Let $\mathcal{F} = (\mathcal{M}, \{\mathcal{N}_q\})$ be an FDFA and w be a UP-word in $\mathrm{UP}(\mathcal{F})$. We say $\mathcal{F}$ is* saturated *if for all normalized decompositions (u, v) and (u', v') of w, either both (u, v) and (u', v') are accepted by $\mathcal{F}$ or both are not.*

Intuitively, for a saturated FDFA $\mathcal{F}$, a UP-word w is accepted by $\mathcal{F}$ if and only if all normalized decompositions (u, v) of w are accepted by $\mathcal{F}$. From a saturated FDFA $\mathcal{F}$, one can construct an equivalent NBA $\mathcal{A}$ that recognizes $\mathrm{UP}(\mathcal{F})$ in polynomial time.

Lemma 1 (Polynomial Translation from FDFAs to NBAs [3,17]). *Let $\mathcal{F} = (\mathcal{M}, \{\mathcal{N}_q\})$ be a saturated FDFA with n states. Then, one can construct an NBA $\mathcal{A}$ with $\mathcal{O}(n^3)$ states such that $\mathrm{UP}(\mathcal{F}) = \mathrm{UP}(\mathcal{L}(\mathcal{A}))$.*

Note that an FDFA that is *not* saturated does not necessarily recognize an ω-regular language (cf. [17]), let alone permit an equivalent translation to NBAs.

In the remainder of the paper, we fix an NBA $\mathcal{A} = (Q, I, \delta, F)$, unless explicitly stated otherwise, where $\mathcal{A}$ has n states, i.e., $n = |Q|$. We call a state in an FDFA a *macrostate* to distinguish it from states of $\mathcal{A}$.

3 Improved Congruence Relations for NBAs

In this section we present congruence relations that can be used to construct NBAs accepting the language of a given NBA $\mathcal{A}$ or its complement. We first review in Sect. 3.1 the classical congruence relations defined in [21,22] and then give improved congruence relations in Sect. 3.2.

[1] We use the normalized decomposition of UP-words defined in [17], which is different from the one given in [3]. Ours is a definition for a UP-word, while their definition is applied to a decomposition. However, this difference does not affect the definition of a saturated FDFA to be given later.

3.1 Classical Congruence Relations

As mentioned in the introduction, the index of the congruence relation of RBC proposed by Büchi [6] is doubly exponential in the size of $\mathcal{A}$. Sistla, Vardi, and Wolper [21] showed how to improve RBC with a subset construction that was later presented by Thomas [22] as the following congruence relation $\backsim_{\mathcal{A}}$.

Definition 5 ([21,22]). *In the RBC construction, for all $u_1, u_2 \in \Sigma^*$, we have $u_1 \backsim_{\mathcal{A}} u_2$ if for all $q, r \in Q$, (1) $q \xrightarrow{u_1} r$ iff $q \xrightarrow{u_2} r$ and (2) $q \overset{u_1}{\Longrightarrow} r$ iff $q \overset{u_2}{\Longrightarrow} r$.*

It is easy to verify that $\backsim_{\mathcal{A}}$ is a (right-)congruence relation: given two finite words u_1 and u_2 such that $u_1 \backsim_{\mathcal{A}} u_2$, we have that $xu_1y \backsim_{\mathcal{A}} xu_2y$ holds for all $x, y \in \Sigma^*$. Moreover, we have that $\backsim_{\mathcal{A}}$ is of finite index, as stated by the next lemma. To simplify the notation, we just write $\backsim$ instead of $\backsim_{\mathcal{A}}$ as $\mathcal{A}$ is fixed.

Lemma 2 ([21,22]). *Let $\backsim$ be as given in Definition 5. Then $|\backsim| \leq 3^{n^2}$.*

Since the congruence relation $\backsim$ is defined by reachability between states, the result follows from the fact that we can map each of the n^2 pairs of states (q, r) to either both $q \overset{u}{\Rightarrow} r$ and $q \xrightarrow{u} r$, just $q \xrightarrow{u} r$, or none of them. Thus we have $|\backsim| = |\Sigma^*/_{\backsim}| \leq 3^{n^2}$. We can also establish a lower bound for $\backsim$, by means of a family of DBAs inspired by the proof of [4, Theorem 2].

Lemma 3. *There is a family of DBAs $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ such that each DBA $\mathcal{C}_n$ has $n+2$ states and the corresponding $\backsim_{\mathcal{C}_n}$ is such that $|\backsim_{\mathcal{C}_n}| \geq n!$.*

An important property we want to have is that the congruence relation $\backsim$ captures correctly the language of the NBA it corresponds to. This means that $\backsim$ must not relate words in $\mathcal{L}(\mathcal{A})$ with those in $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$, that is, for each $[u]_{\backsim} \in \Sigma^*/_{\backsim}$ and $[v]_{\backsim} \in \Sigma^+/_{\backsim}$, either $[u]_{\backsim}[v]^\omega_{\backsim} \subseteq \mathcal{L}(\mathcal{A})$ or $[u]_{\backsim}[v]^\omega_{\backsim} \subseteq \Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$. Moreover, $\backsim$ should cover the whole Σ^ω, that is, it saturates $\mathcal{L}(\mathcal{A})$, $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$, and Σ^ω. This is formalized by the following *saturation lemma* of the congruence relation $\backsim$, which is a known result from [21] that we adapt to our notation.

According to [21,22], given two classes $[u]_{\backsim} \in \Sigma^*/_{\backsim}, [v]_{\backsim} \in \Sigma^+/_{\backsim}$, the ω-language $[u]_{\backsim}[v]^\omega_{\backsim}$ is called *proper* if $[u]_{\backsim}[v]_{\backsim} \subseteq [u]_{\backsim}$ and $[v]_{\backsim}[v]_{\backsim} \subseteq [v]_{\backsim}$.

Lemma 4 (Saturation Lemma [21,22]**)**

1. *For $[u]_{\backsim} \in \Sigma^*/_{\backsim}, [v]_{\backsim} \in \Sigma^+/_{\backsim}$, if $[u]_{\backsim}[v]^\omega_{\backsim}$ is proper, then either $[u]_{\backsim}[v]^\omega_{\backsim} \cap \mathcal{L}(\mathcal{A}) = \emptyset$ or $[u]_{\backsim}[v]^\omega_{\backsim} \subseteq \mathcal{L}(\mathcal{A})$.*
2. *$\Sigma^\omega = \bigcup\{\, [u]_{\backsim}[v]^\omega_{\backsim} \mid [u]_{\backsim} \in \Sigma^*/_{\backsim} \wedge [v]_{\backsim} \in \Sigma^+/_{\backsim} \wedge [u]_{\backsim}[v]^\omega_{\backsim}$ is proper$\,\}$.*
3. *$\Sigma^\omega \backslash \mathcal{L}(\mathcal{A}) = \bigcup\{\, [u]_{\backsim}[v]^\omega_{\backsim} \mid [u]_{\backsim} \in \Sigma^*/_{\backsim} \wedge [v]_{\backsim} \in \Sigma^+/_{\backsim} \wedge [u]_{\backsim}[v]^\omega_{\backsim} \cap \mathcal{L}(\mathcal{A}) = \emptyset \wedge [u]_{\backsim}[v]^\omega_{\backsim}$ is proper$\,\}$.*

Thus, it suffices to just consider proper languages to get the languages Σ^ω (cf. Item (2) of Lemma 4) and $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$ (cf. Item (3) of Lemma 4). This means that the congruence relation $\backsim$ allows us to obtain $\mathcal{L}(\mathcal{A})$ (resp., $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$) by identifying the exact set of proper languages that are inside $\mathcal{L}(\mathcal{A})$ (resp., outside $\mathcal{L}(\mathcal{A})$). In the remainder of the paper, we show that we can obtain similar saturation lemmas (cf. Lemma 7 and Lemma 13) for the congruence relations we are going to propose to obtain $\mathcal{L}(\mathcal{A})$ or the complementary language $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$.

3.2 Improved Congruence Relations for NBAs

In this section, we introduce the relations $\backsim^i$ and $\approx_u$, for $u \in \Sigma^*$, that can never have larger index than that of the classical congruence relation $\backsim$ (cf. Lemma 5) while possibly being exponentially coarser than $\backsim$ (cf. Theorem 3). When restricted to DBAs, we reduce the worst-case blow-up from $\Omega(n!)$ (cf. Lemma 3) to $\mathcal{O}(n^2)$ (cf. Theorem 4). Still, they capture correctly $\mathcal{L}(\mathcal{A})$ and $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$ (cf. Lemma 7).

We improve the classical congruence relation $\backsim$ given in Sect. 3.1 based on the following key observations: (1) we can use different congruence relations to process the finite prefix u and the periodic word v of a UP-word uv^ω, separately, in a manner similar to FDFAs; (2) the assumption $[v]_\backsim[v]_\backsim \subseteq [v]_\backsim$ of proper languages is not necessary, according to [21]; (3) inspired by [5], we can consider only reachable states in $\mathcal{A}$, which allows us to use just *right congruences* instead of congruences such as $\backsim$. We defer the comparison of our work with [5] to Remark 2.

Instead of considering every pair of states (q, r) of $\mathcal{A}$ to define the congruence relation $\backsim$ (cf. Definition 5), we process the finite prefixes u by a simple subset construction over the states of $\mathcal{A}$, obtaining the following relation $\backsim^i$ that is obviously a right congruence.

Definition 6 (RC $\backsim^i$). *For $u_1, u_2 \in \Sigma^*$, we have $u_1 \backsim^i u_2$ if and only if $\delta(I, u_1) = \delta(I, u_2)$.*

As one can expect, by relaxing the conditions on the relation $\backsim^i$, we reduce how large its index can be, from 3^{n^2} (cf. Lemma 3) to 2^n, showing also that $\backsim^i$ is a right congruence of finite index.

Lemma 5. *Let $\backsim^i$ be the right congruence in Definition 6. Then $|\backsim^i| \leq 2^n$.*

Differently from $\backsim$ (see, e.g., Lemma 4), we will use $\backsim^i$ only to process the finite prefix u of a UP-word uv^ω; to process the period v, we now introduce the right congruence $\approx_u$, by considering only states reachable from $\delta(I, u)$.

Definition 7 (RC $\approx_u$). *For $u, v_1, v_2 \in \Sigma^*$, we have $v_1 \approx_u v_2$ if for all states $q \in \delta(I, u)$ and $r \in Q$ of $\mathcal{A}$, (1) $q \xrightarrow{v_1} r$ iff $q \xrightarrow{v_2} r$ and (2) $q \overset{v_1}{\Rightarrow} r$ iff $q \overset{v_2}{\Rightarrow} r$.*

Compared to Definition 5, we only take into account the states that can be reached from $\delta(I, u)$, as opposed to the whole set Q. In this way we obtain a right congruence relation that is coarser than $\backsim$ for the periodic finite words.

Theorem 2. *Let $\backsim$ be the congruence relation in Definition 5. For each $u, v_1, v_2 \in \Sigma^*$, we have that $v_1 \backsim v_2$ implies that $v_1 \approx_u v_2$.*

Similarly, $u_1 \backsim u_2$ implies that $u_1 \backsim^i u_2$ for all $u_1, u_2 \in \Sigma^*$.

Although the right congruence relation $\approx_u$ is coarser than its predecessor $\backsim$, it has the same upper bound for its index (cf. Lemma 2).

Lemma 6. *Given $u \in \Sigma^*$, let $\approx_u$ be as defined in Definition 7. Then $|\approx_u| \leq 3^{n^2}$.*

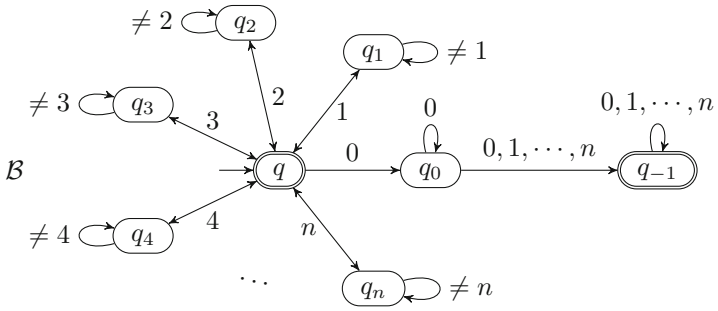

Fig. 2. The family of NBAs $\{\mathcal{B}_n\}_{n\in\mathbb{N}}$ over the alphabet $\{0, 1, \cdots, n\}$ with $n+3$ states for which $|\backsim|$ is at least $n!$ while $|\approx_u|$ is at most $(n+3)+2$ for each $u \in \Sigma^*$; the initial state is q and $F = \{q, q_{-1}\}$. We remark that this NBA is inspired by a DBA from [4]. However, our NBA is not deterministic.

Despite this common upper bound, $|\approx_u|$ can be exponentially smaller than $|\backsim|$, as witnessed by the family of NBAs $\{\mathcal{B}_n\}_{n\in\mathbb{N}}$ depicted in Fig. 2.

Theorem 3. *Given $u \in \Sigma^*$, let $\backsim$ be the congruence relation in Definition 5 and $\approx_u$ be the right congruence in Definition 7. There is a family of NBAs $\{\mathcal{B}_n\}_{n\in\mathbb{N}}$ with $n+3$ states for which $|\backsim| \geq n!$ and $|\approx_u| \leq (n+3)+2$.*

The idea underlying this result is that in $\backsim^i$, there are at most $n+4$ equivalence classes, which correspond to the singletons $\{r\}$ with $r \in Q$ and the set $\{q_{-1}, q_0\}$. For each of these classes, say $[u]_{\backsim^i} = [1]_{\backsim^i}$, the associated classes $[v]_{\approx_u}$ can correspond to at most $(n+3)+2$ configurations of (accepting) runs, like $\{q_1 \xrightarrow{v} q_1\}$ and $\{q_1 \xrightarrow{v} q_0, q_1 \overset{v}{\Rightarrow} q_0, q_1 \xrightarrow{v} q_{-1}, q_1 \overset{v}{\Rightarrow} q_{-1}\}$. On the other hand, since $\backsim$ must take care of both prefixes and periods, different permutations of $\{1, \cdots, n\}$ taken as prefixes cannot be equivalent, thus $|\backsim| \geq n!$. We refer to [18, proof of Theorem 3] for more detailed reasoning and explanations.

When working with DBAs, the overall index of the right congruence relations $\bigcup_{u\in\Sigma^*}\{\approx_u\}$ can be exponentially tighter than that of $\backsim$ (cf. Lemma 3).

Theorem 4. *Let $\mathcal{A}$ be a DBA with n states. Then $\Sigma_{[u]_{\backsim^i}\in\Sigma^*/_{\backsim^i}} |\approx_u| \in \mathcal{O}(n^2)$.*

Similarly to Lemma 4, the saturation lemma for $\backsim$, the right congruences $\backsim^i$ and $\approx_u$ with $u \in \Sigma^*$ also allow us to recognize *exactly* $\mathcal{L}(\mathcal{A})$ or its complement $\Sigma^\omega \backslash \mathcal{L}(A)$: for these relations we have again that the ω-language $[u]_{\backsim^i}[v]^\omega_{\approx_u}$ with $uv \backsim^i u$ is either completely inside $\mathcal{L}(\mathcal{A})$ or outside $\mathcal{L}(\mathcal{A})$, even if we drop the requirement $[v]_{\approx_u}[v]_{\approx_u} \subseteq [v]_{\approx_u}$.

Lemma 7 (Saturation Lemma for $(\backsim^i, \cup_{u\in\Sigma^*}\{\approx_u\})$)

1. *For $u \in \Sigma^*, v \in \Sigma^+$, if $uv \backsim^i u$, then either $[u]_{\backsim^i}[v]^\omega_{\approx_u} \cap \mathcal{L}(\mathcal{A}) = \emptyset$ or $[u]_{\backsim^i}[v]^\omega_{\approx_u} \subseteq \mathcal{L}(\mathcal{A})$.*
2. *$\Sigma^\omega = \bigcup\{\, [u]_{\backsim^i}[v]^\omega_{\approx_u} \mid [u]_{\backsim^i} \in \Sigma^*/_{\backsim^i} \wedge [v]_{\approx_u} \in \Sigma^+/_{\approx_u} \wedge uv \backsim^i u \,\}$.*

3. $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A}) = \bigcup\{ [u]_{\backsim^i}[v]^\omega_{\approx_u} \mid [u]_{\backsim^i} \in \Sigma^*/_{\backsim^i} \wedge [v]_{\approx_u} \in \Sigma^+/_{\approx_u} \wedge uv \backsim^i u \wedge [u]_{\backsim^i}[v]^\omega_{\approx_u} \cap \mathcal{L}(\mathcal{A}) = \emptyset \}$.

By definition of $\backsim^i$ and $\approx_u$, if $uv \backsim^i u$, then the set of states $\delta(I, u)$ is visited infinitely often when reading the word $w = uv^\omega$; it also implies $uv^j \backsim^i u$ for each $j \geq 0$. Moreover, if $w \in \mathcal{L}(\mathcal{A})$, then there is a run of $\mathcal{A}$ over w that is accepting, i.e., the run visits infinitely often states in F. This happens when dealing with v^ω, since u is a finite word; thus $\mathcal{A}$ visits an accepting state when reading v on the way from $\delta(I, u)$ to $\delta(\delta(I, u), v) = \delta(I, u)$.

These properties allow us to prove Item (1), i.e., that if $w \in [u]_{\backsim^i}[v]^\omega_{\approx_u} \cap \mathcal{L}(\mathcal{A})$, then for each word $w' \in [u]_{\backsim^i}[v]^\omega_{\approx_u}$ we have $w' \in \mathcal{L}(\mathcal{A})$, because w' can be written as $w' = u' \cdot v'_1 \cdot v'_2 \cdots$ with $u' \in [u]_{\backsim^i}$ and $v'_j \in [v]_{\approx_u}$ for each $j \geq 1$. Thus for w' we have that $\mathcal{A}$ visits infinitely often the set $\delta(I, u'v'_j) = \delta(I, u') = \delta(I, u)$ while visiting an accepting state on the way from $\delta(I, u')$ to $\delta(\delta(I, u'), v'_j) = \delta(I, u')$ since $v \approx_u v'_j$. Item (2) holds by considering only the UP-words (cf. Theorem 1), for which we have that for each $w = u'v'^\omega \in \mathrm{UP}(\Sigma^\omega)$, we can construct a decomposition $(u = u'v'^h, v = v'^k)$ of w with $u \backsim^i uv$ for some $h, k \geq 1$ since $\backsim^i$ is of finite index. By combining Items (1) and (2), to obtain $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$, we can just take the union of all languages $[u]_{\backsim^i}[v]^\omega_{\approx_u}$ such that $[u]_{\backsim^i}[v]^\omega_{\approx_u} \cap \mathcal{L}(\mathcal{A}) = \emptyset$.

4 Optimal Congruence Relations for NBAs

The right congruence relations we introduced in Sect. 3.2, despite improving $\backsim$, still lead to a blow-up of $3^{\mathcal{O}(n^2)}$ (cf. Lemma 6). The main cause of the exponent n^2 is that it is possible for each of the n states to be a predecessor of a state r over the word v. To avoid having to consider all such precedessors, we look for specific representatives, in order to reduce the blow-up. Inspired by [9,10], we introduce a *preorder* on the states based on the transition structure of $\mathcal{A}$; we then use the preorder to select the representatives. In particular, if the predecessors of r can be reduced to only one representative for a given v, we obtain that the blow-up reduces to $2^{\mathcal{O}(n \log n)}$. The representative we are going to use is the maximal equivalence class induced by the preorder among at most n equivalence classes. Breuers *et al.* [5] also proposed a preorder-based optimization to improve RBC; see Remark 2 for a detailed comparison.

In the remainder of this section, we present a preorder $\preceq_u$ inspired by [9,10] on the set of states $\delta(I, u)$, $u \in \Sigma^*$, yielding optimal congruence relations for NBAs. The preorder $\preceq_u$ over the set of states $\delta(I, u)$ is defined by comparing the finite runs of $\mathcal{A}$ over u from the initial states to those states.

Fix a finite word u; given a run π of $\mathcal{A}$ over u, recall that $\pi[i]$ denotes the i-th element (i.e. state) of π. For each run π, we define the function $\mathbb{1}_F(\pi) = \mathbb{1}_F(\pi[1])\mathbb{1}_F(\pi[2])\ldots$ where $\mathbb{1}_F(s) = 1$ if $s \in F$ and 0 if $s \notin F$. In other words, each run π can be encoded as a binary sequence. Given two runs π, π' of $\mathcal{A}$ over u, the two runs π and π' can be ordered by the lexicographical order with 1 being larger than 0; formally, we say that π' is *greater* than π, denoted by $\pi' > \pi$, if there is a prefix $\alpha 1$ of $\mathbb{1}_F(\pi')$ such that $\alpha 0$ is a prefix of $\mathbb{1}_F(\pi)$. That is, π' is

greater than π if there is an integer $1 \leq j \leq |u|+1$ such that $\pi'[j] \in F$, $\pi[j] \notin F$, and $\pi'[i] \in F \iff \pi[i] \in F$ for all $1 \leq i < j$ (i does not exist when $j = 1$).

Let $\Pi_{u,q}$ be the set of runs of $\mathcal{A}$ over u starting from I with the last state being q. For each state $q \in \delta(I, u)$, there may be several runs in $\Pi_{u,q}$; the set of *maximal* runs in $\Pi_{u,q}$ is defined as $\max(\Pi_{u,q}) = \{\, \pi' \in \Pi_{u,q} \mid \forall \pi \in \Pi_{u,q}, \pi \not> \pi' \,\}$. The following result is a direct consequence of the definition above.

Proposition 1. *For two runs $\pi_q, \pi'_q \in \max(\Pi_{u,q})$, for each $1 \leq i \leq |u|+1$ we have that $\pi_q[i] \in F \iff \pi'_q[i] \in F$.*

That is, all runs in $\max(\Pi_{u,q})$ have the same image under $\mathbb{1}_F$.

In the following, we define the preorder $\preceq_u$ on the set of states $P = \delta(I, u)$ by comparing the sets of maximal runs $\max(\Pi_{u,q})$ and $\max(\Pi_{u,r})$ for $q, r \in P$.

Definition 8 (Preorder $\preceq_u$). *Given $u \in \Sigma^*$,*

- *if $u = \epsilon$, then for $q, r \in I = \delta(I, u)$, we define $q \preceq_\epsilon r$ if and only if $q \in F \implies r \in F$ holds. Therefore, $q \prec_\epsilon r$ if and only if $q \notin F$ and $r \in F$;*
- *when $u \in \Sigma^+$, for $q, r \in \delta(I, u)$, $q \preceq_u r$ if the runs $\pi_q \in \max(\Pi_{u,q})$ are not greater than the runs $\pi_r \in \max(\Pi_{u,r})$. In particular, $q \prec_u r$ if the runs $\pi_r \in \max(\Pi_{u,r})$ are greater than the runs $\pi_q \in \max(\Pi_{u,q})$.*

One can verify that $\preceq_u$ is a binary relation that is reflexive (i.e., for each $q \in Q$, $q \preceq_u q$) and transitive (i.e., for each $q, r, s \in Q$, $q \preceq_u r$ and $r \preceq_u s$ implies $q \preceq_u s$), so it is a preorder; we also have $q \prec_u r$ whenever $q \preceq_u r$ and $r \not\preceq_u q$ and we write $q \simeq_u r$ whenever $q \preceq_u r$ and $r \preceq_u q$. Intuitively, we have $q \prec_u r$ if there is a run from an initial state to r on u that sees an accepting state earlier than all paths from the initial states to q on u. That is, there is a prefix $\alpha 1$ of $\mathbb{1}_F(\pi_r)$ for a run π_r to r such that $\alpha 0$ is a prefix of $\mathbb{1}_F(\pi_q)$ for all runs π_q to q.

Due to Proposition 1, if there is a run $\pi_r \in \max(\Pi_{u,r})$ greater than a run in $\max(\Pi_{u,q})$, then all runs in $\max(\Pi_{u,r})$ are greater than the runs in $\max(\Pi_{u,q})$.

Example 1. Consider the NBA $\mathcal{B}_n$ depicted in Fig. 2 and let $P = \delta(\{q\}, 00) = \{q_{-1}, q_0\}$; we have $q_0 \prec_{00} q_{-1}$ on this set since there is a run from the initial state q to q_{-1} that sees the accepting state q_{-1} after inputting the second 0, while all runs from q to q_0 on 00 do not visit an accepting state right after inputting the second 0.

Remark 1. The preorder $\preceq_u$ in Definition 8 shares the same idea of comparing the maximal runs with the *lexicographical order* of vertices at the same level of the run direct acyclic graph (DAG) over an ω-word w used in [10]; see [18, Appendix B] for a detailed comparison between their and our works. The difference between our work and the work in [10] is that the latter applies this idea to Slice-based [12] and Rank-based complementation algorithms [16] while ours is designed for RBC. A similar idea was also used in [9] for determinizing NBAs.

As an immediate consequence of Definition 8, given two states $q, r \in \delta(I, u)$ such that $q \preceq_u r$, we have that a run from an initial state to q that visits an accepting state mandates that there must be a run from an initial state to r that also visits accepting states.

Corollary 1. *Let $q, r \in \delta(I, u)$, with $q \preceq_u r$. Then $\iota_q \overset{u}{\Rightarrow} q$ for some initial state $\iota_q \in I$ implies $\iota_r \overset{u}{\Rightarrow} r$ for some initial state $\iota_r \in I$.*

Let $P = \delta(I, u)$. The preorder $\preceq_u$ defines a partition of P in which states in the same set are equivalent under $\preceq_u$. By abuse of terminology, we call the set $[r]_{\preceq_u} = \{ r' \in P \mid r' \simeq_u r \}$ the equivalence class of $r \in P$ under $\preceq_u$; we denote by $P/_{\preceq_u}$ the set of all such equivalence classes. Since every two states $q, r \in P$ are *comparable* under $\preceq_u$, we define the maximal equivalence class of P under $\preceq_u$ as $\max_{\preceq_u}(P) = \max(P/_{\preceq_u}) = \{ r \in P \mid r' \preceq_u r \text{ for all } r' \in P \}$; moreover, the equivalence classes in $P/_{\preceq_u}$ can be linearly ordered by $[r]_{\preceq_u} \trianglelefteq_u [r']_{\preceq_u} \iff r \preceq_u r'$; so we have $[r]_{\preceq_u} \triangleleft_u [r']_{\preceq_u}$ if $[r]_{\preceq_u} \trianglelefteq_u [r']_{\preceq_u}$ and $[r'] \not\trianglelefteq_u [r]_{\preceq_u}$. Here $\trianglelefteq_u$ is a partial order, not a preorder, which implies that $[r]_{\preceq_u} = [r']_{\preceq_u}$ if and only if $[r]_{\preceq_u} \trianglelefteq_u [r']_{\preceq_u}$ and $[r']_{\preceq_u} \trianglelefteq_u [r]_{\preceq_u}$. In the remainder of this section, by abuse of terminology, we just use $P/_{\preceq_u}$ to denote the preordered set of equivalence classes of P under $\preceq_u$, i.e., $P/_{\preceq_u}$ not just represents a set of equivalence classes but also linearly orders those equivalence classes with $\trianglelefteq_u$.

An interesting property of the states $\mathcal{A}$ visits on the maximal runs from the initial states I to a state $q \in \delta(I, uv)$ over the finite word uv is that they are step by step all equivalent under the preorder with respect to the prefix of uv. Let $\max(\Pi_{uv,q})|_u = \{ \pi[|u|+1] \mid \pi \in \max(\Pi_{uv,q}) \}$, i.e., the set of states reached from the initial states after inputting u on the maximal runs to q over uv.

Lemma 8. *Given $u, v \in \Sigma^*$ and $q \in \delta(I, uv)$, let $[p]_{\preceq_u} = \max_{\preceq_u}\{ [p']_{\preceq_u} \in \delta(I, u)/_{\preceq_u} \mid p' \overset{v}{\to} q \}$. Then for each $q' \in [q]_{\preceq_{uv}}$, $\max(\Pi_{uv,q'})|_u \subseteq [p]_{\preceq_u}$.*

By definition of $[q]_{\preceq_{uv}}$, all the maximal runs from the initial states to the states in $[q]_{\preceq_{uv}}$ have the same image under $\mathbb{1}_F$. As a consequence, the states on these runs reached after reading u must belong to the same equivalence class $[p]_{\preceq_u}$ under $\preceq_u$, which is also the maximal equivalence class under $\preceq_u$ that reaches $[q]_{\preceq_{uv}}$. If this would not be the case, then we would be able to find runs to $[q]_{\preceq_{uv}}$ greater than the current maximal runs by visiting $[p]_{\preceq_u}$.

A useful property of these maximal runs is that they share visits to accepting states; more precisely, if one of the maximal runs on a word uv to a state $q_1 \in [q]_{\preceq_{uv}}$ visits an accepting state while reading v, then so do all other maximal runs on the same word to some other state $q_2 \in [q]_{\preceq_{uv}}$. The motivation for this is again the maximality of the runs: if one run visits an accepting state while another does not, then the former is greater than the latter, which implies that the latter cannot be maximal. This property is formalized below.

Lemma 9. *Let $u, v \in \Sigma^*$ and $q \in \delta(I, uv)$. For each $q_1, q_2 \in [q]_{\preceq_{uv}}$, $p_1 \in \max(\Pi_{uv,q_1})|_u$, and $p_2 \in \max(\Pi_{uv,q_2})|_u$, we have $p_1 \overset{v}{\Rightarrow} q_1$ if and only if $p_2 \overset{v}{\Rightarrow} q_2$.*

Similarly to Lemma 8, a consequence of Lemma 9 is that, for a given finite word u and $q_1 \simeq_u q_2$, the maximal runs in $\max(\Pi_{u,q_1})$ and in $\max(\Pi_{u,q_2})$ visit accepting states at the same moment, i.e., they have the same image under $\mathbb{1}_F$.

The preorder $\preceq_u$ enjoys several properties about the states and maximal runs of $\mathcal{A}$ for the given finite word u. Thus, instead of tracing only the set of reachable

states, as done by the right congruence $\backsim^i$ (cf. Definition 6), we also trace the reachable states $\delta(I,u)$ with the preorder $\preceq_u$ to get the right congruence $\backsim^o$.

Definition 9 (RC $\backsim^o$). *For $u_1, u_2 \in \Sigma^*$, we have $u_1 \backsim^o u_2$ if and only if $\delta(I,u_1)/_{\preceq_{u_1}} = \delta(I,u_2)/_{\preceq_{u_2}}$.*

Example 2. Consider again the NBA $\mathcal{B}_n$ depicted in Fig. 2: we can represent $\delta(I,00)/_{\preceq_{00}}$ as the ordered sequence of sets $\langle\{q_0\},\{q_{-1}\}\rangle$ since we have $\{q_0\} \lhd_{00} \{q_{-1}\}$. Analogously, $\delta(I,000)/_{\preceq_{000}}$ can also be represented as $\langle\{q_0\},\{q_{-1}\}\rangle$ while $\delta(I,001)/_{\preceq_{001}}$ as $\langle\{q_{-1}\}\rangle$. We can see that $00 \backsim^o 000$ since $\delta(I,00)/_{\preceq_{00}} = \delta(I,000)/_{\preceq_{000}} = \langle\{q_0\},\{q_{-1}\}\rangle$ while $000 \not\backsim^o 001$ as $\delta(I,001)/_{\preceq_{001}} = \langle\{q_{-1}\}\rangle$.

Since each equivalence class $[u]_{\backsim^o}$, $u \in \Sigma^*$, can be uniquely encoded as the set $\delta(I,u)/_{\preceq_u}$, i.e., an ordered sequence of sets over Q, by [10] we have that the number of possible ordered sequences of sets over Q is $\mathcal{O}((\frac{n}{e \ln n})^n) \approx (0.53n)^n \leq n^n$. Thus we have the following upper bound for $\backsim^o$, so it is of finite index.

Lemma 10. *Let $\backsim^o$ be the right congruence in Definition 9. Then $|\backsim^o| \leq n^n$.*

Given their definitions, it is clear that $\backsim^i$ is coarser than $\backsim^o$, thus $|\backsim^i| \leq |\backsim^o|$. Nonetheless, the right congruence $\backsim^o$ allows us to define a novel right congruence relation $\approx^o_u$ of index $2^{\mathcal{O}(n \log n)}$, for a given $u \in \Sigma^*$.

Definition 10 (RC $\approx^o_u$). *Given $u, v_1, v_2 \in \Sigma^*$, we say $v_1 \approx^o_u v_2$ if and only if (1) $uv_1 \backsim^o uv_2$ and (2) for all states $q \in P'$, for $S_1 = \max_{\preceq_u}\{\,[p]_{\preceq_u} \in P/_{\preceq_u} \mid p \xrightarrow{v_1} q\,\}$ and $S_2 = \max_{\preceq_u}\{\,[p]_{\preceq_u} \in P/_{\preceq_u} \mid p \xrightarrow{v_2} q\,\}$, we have (i) $S_1 = S_2$ and (ii) $p_1 \overset{v_1}{\Longrightarrow} q$ for $p_1 \in \max(\Pi_{uv_1,q})|_u$ if and only if $p_2 \overset{v_2}{\Longrightarrow} q$ for some $p_2 \in \max(\Pi_{uv_2,q})|_u$ where $P = \delta(I,u)$ and $P' = \delta(I,uv_1) = \delta(I,uv_2)$.*

Note that the equality $\delta(I,uv_1) = \delta(I,uv_2)$ holds because, under the assumption $uv_1 \backsim^o uv_2$, we have that the sets of equivalence classes $\delta(I,uv_1)/_{\preceq_{u_1}}$ and $\delta(I,uv_2)/_{\preceq_{u_2}}$ are equal according to the definition of $\backsim^o$, which then implies that $\delta(I,uv_1)$ and $\delta(I,uv_2)$ must be equal as well.

Definition 10 formalizes the following idea for recognizing the ω-words accepted and rejected by $\mathcal{A}$. Since we want to use $(\backsim^o, \cup_{u\in\Sigma^*}\{\approx^o_u\})$ to characterize $\mathcal{L}(\mathcal{A})$ and $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$, i.e., to establish its saturation lemma in line with $\backsim$ (cf. Lemma 4) and $(\backsim^i, \cup_{u\in\Sigma^*}\{\approx_u\})$ (cf. Lemma 7), under the assumption that $uv_1 \backsim^o u$ and $u \backsim^o uv_2$, we need to guarantee that if $v_1 \approx^o_u v_2$, then $uv_1^\omega \in \mathcal{L}(\mathcal{A})$ if and only if $uv_2^\omega \in \mathcal{L}(\mathcal{A})$. To achieve this, the first condition we impose (cf. Item (1) of Definition 10) is to visit infinitely often the same states over the ω-words uv_1^ω and uv_2^ω, so we require $uv_1 \backsim^o uv_2$. The second condition is to guarantee that the images under $\mathbb{1}_F$ of the maximal runs over uv_1^k and uv_2^k, $k \geq 1$, either both contain only 0s, or both contain some 1; by this, when extending to infinite words, the images of uv_1^ω and uv_2^ω under $\mathbb{1}_F$ will both have infinitely many 1s or none of them does. This ensures that $uv_1^\omega \in \mathcal{L}(\mathcal{A})$ if and only if $uv_2^\omega \in \mathcal{L}(\mathcal{A})$. To guarantee having the property above, we first require that the maximal equivalence classes from each state $q \in \delta(I,u)$ over

both finite words v_1 and v_2 have to be the same (cf. Condition (2)-(i), together with Lemma 8); then, we demand that they share the visits to accepting states (cf. Condition (2)-(ii) and Lemma 9).

Example 3. Consider again Example 1 and let $u = \epsilon$, $v_1 = 00$, and $v_2 = 000$. We now check whether $00 \approx^o_\epsilon 000$. Clearly $\epsilon \cdot 00 \backsim^o \epsilon \cdot 000$ since $00 \backsim^o 000$. We can represent $\delta(I, \epsilon)/_{\preceq_\epsilon}$ as $\langle\{q\}\rangle$, a singleton, hence, we have $S_1 = S_2 = \{q\}$, thus we satisfy Condition (2)-(i) of Definition 10. To fulfill Condition (2)-(ii), we first have $P' = \{q_{-1}, q_0\}$. We also have $\max(\Pi_{\epsilon\cdot 00, q_{-1}}) = \{qq_0q_{-1}\}$, $\max(\Pi_{\epsilon\cdot 00, q_0}) = \{qq_0q_0\}$, $\max(\Pi_{\epsilon\cdot 000, q_{-1}}) = \{qq_0q_{-1}q_{-1}\}$, and $\max(\Pi_{\epsilon\cdot 000, q_0}) = \{qq_0q_0q_0\}$. For state $q_{-1} \in P'$, Condition (2)-(ii) is satisfied since qq_0q_{-1} and $qq_0q_{-1}q_{-1}$ both visit accepting states. For state $q_0 \in P'$, Condition (2)-(ii) is also fulfilled as qq_0q_0 and $qq_0q_0q_0$ both visit accepting state q. Therefore, we conclude that $00 \approx^o_\epsilon 000$ holds. Clearly $000 \not\approx^o_\epsilon 001$ since we already know that $\epsilon \cdot 000 \not\backsim^o \epsilon \cdot 001$.

As desired before Definition 10, the index of $\approx^o_u$ is indeed in $2^{\mathcal{O}(n \log n)}$.

Lemma 11. *Given $u \in \Sigma^*$, let $\approx^o_u$ be the right congruence from Definition 10. Then $|\approx^o_u| \leq n^n \times (n+1)^n \times 2^n \in 2^{\mathcal{O}(n \log n)}$.*

The upper bound for $|\approx^o_u|$ derives from the encoding we use for $[v]_{\approx^o_u}$. $[v]_{\approx^o_u}$ is mapped to the pair $\langle \delta(I, uv)/_{\preceq_{uv}}, f\rangle$ where the function f keeps track of the satisfaction of the states $q \in Q$ of the conditions in Definition 10, i.e., whether $q \in \delta(I, uv)$ and whether Conditions (2)-(i) and (2)-(ii) are satisfied for such states. The codomain of f has size $2n + 1 < 2(n + 1)$, so the possible different functions f are $(2(n + 1))^n = 2^n \times (n + 1)^n$, while by [10] the possible sets $\delta(I, uv)/_{\preceq_{uv}}$ are n^n, hence $|\approx^o_u| \leq n^n \times (n + 1)^n \times 2^n \in 2^{\mathcal{O}(n \log n)}$.

Similarly to Lemma 4, if we restrict ourselves to DBAs, then $|\approx^o_u|$ is exponentially better than the bound $2^{\mathcal{O}(n \log n)}$ we have for general NBAs.

Lemma 12. *Let $\mathcal{A}$ be a DBA with n states. Then $\Sigma_{[u]_{\backsim^o} \in \Sigma^*/_{\backsim^o}} |\approx^o_u| \in \mathcal{O}(n^2)$.*

This result follows from the fact that, $\mathcal{A}$ being deterministic, there are at most n classes $[u]_{\backsim^o} \in \Sigma^*/_{\backsim^o}$. By taking the same encoding as in Lemma 11, this time the index of $\approx^o_u$ is at most $2n$, and so the result follows.

Similarly to the other (right) congruence relations we considered, i.e., $\backsim$ (cf. Lemma 4) and $(\backsim^i, \cup_{u\in\Sigma^*}\{\approx_u\})$ (cf. Lemma 7), $(\backsim^o, \cup_{u\in\Sigma^*}\{\approx^o_u\})$ also enjoys its saturation lemma. As stated below, $(\backsim^o, \cup_{u\in\Sigma^*}\{\approx^o_u\})$ is able to recognize exactly $\mathcal{L}(\mathcal{A})$ and $\Sigma^\omega \setminus \mathcal{L}(\mathcal{A})$; a core property to obtain this is again that the ω-languages $[u]_{\backsim^o}[v]^\omega_{\approx^o_u}$ are included either in $\mathcal{L}(\mathcal{A})$ or in its complement $\Sigma^\omega \setminus \mathcal{L}(A)$.

Lemma 13 (Saturation Lemma for $(\backsim^o, \cup_{u\in\Sigma^*}\{\approx^o_u\})$)

1. *For $u \in \Sigma^*$ and $v \in \Sigma^+$, if $uv \backsim^o u$, then either $[u]_{\backsim^o}[v]^\omega_{\approx^o_u} \cap \mathcal{L}(\mathcal{A}) = \emptyset$ or $[u]_{\backsim^o}[v]^\omega_{\approx^o_u} \subseteq \mathcal{L}(\mathcal{A})$.*
2. *$\Sigma^\omega = \bigcup\{\, [u]_{\backsim^o}[v]^\omega_{\approx^o_u} \mid [u]_{\backsim^o} \in \Sigma^*/_{\backsim^o} \wedge [v]_{\approx^o_u} \in \Sigma^+/_{\approx^o_u} \wedge uv \backsim^o u \,\}$.*
3. *$\Sigma^\omega \setminus \mathcal{L}(\mathcal{A}) = \bigcup\{\, [u]_{\backsim^o}[v]^\omega_{\approx^o_u} \mid [u]_{\backsim^o} \in \Sigma^*/_{\backsim^o} \wedge [v]_{\approx^o_u} \in \Sigma^+/_{\approx^o_u} \wedge uv \backsim^o u \wedge [u]_{\backsim^o}[v]^\omega_{\approx^o_u} \cap \mathcal{L}(\mathcal{A}) = \emptyset \,\}$.*

The proof for this saturation lemma follows the same steps as for the other two saturation lemmas, with the appropriate adaptations that take into consideration the differences in the definitions of the right congruences.

Remark 2. In their work [21], Sistla *et al.* constructed an NBA $\mathcal{B}_{u,v}$ for each proper language $Y_{u,v} = [u]_{\backsim}[v]^{\omega}_{\backsim}$ such that $Y_{u,v} \cap \mathcal{L}(\mathcal{A}) = \emptyset$. Each $\mathcal{B}_{u,v}$ can be constructed with two copies of the DFA $\mathcal{M}[\backsim]$ induced by $\backsim$ (cf. Definition 1), where the first copy processes the finite prefix u while the second copy is modified to accept the word v^{ω}. According to [21], the resulting NBA $\mathcal{A}^c$ has $3^{\mathcal{O}(n^2)}$ states. Breuers *et al.* [5] also proposed a subset construction for improving RBC for complementing NBAs; in particular, they used the subset construction to process the finite prefix u of a UP-word uv^{ω} in $\Sigma^{\omega}\backslash\mathcal{L}(\mathcal{A})$. On the other hand, they still used the classical congruence relation $\backsim$ for recognizing the periodic word v of uv^{ω}. Differently from the algorithms proposed in [5,21], we exploit the right congruences $\approx_u$ or $\approx^o_u$ instead of the congruence $\backsim$ for accepting the period v of uv^{ω}; this can result in a considerable decrease of the index of the relation (cf. Theorem 3), which influences the number of states of the automata we build from these relations. The part for accepting v in [5] has also been optimized with a preorder and its size is also reduced to $2^{\mathcal{O}(n \log n)}$. While leading to the same upper bound, there is a difference on the automata needed to process the period v: for a given u, the construction given in [5] uses more than one automaton for recognizing v; instead, our approach needs one automaton, because the equivalence class $[u]_{\backsim^o}$ of $\backsim^o$ only relates with one right congruence relation $\approx^o_u$. This allows us to represent $(\backsim^o, \cup_{u\in\Sigma^*}\{\approx^o_u\})$ as an FDFA, as we explain in Sect. 5.

5 Connection to FDFAs

In this section, we highlight the deep connection between the congruence relations of NBAs and FDFAs. This connection allows us to use the right congruences $(\backsim^o, \bigcup_{u\in\Sigma^*}\{\approx^o_u\})$ we introduced in Sect. 4 to construct an FDFA $\mathcal{F}$ with *optimal* complexity that accepts $\Sigma^{\omega}\backslash\mathcal{L}(\mathcal{A})$. As a byproduct of this connection, we are able to prove Theorem 7; in other words, one cannot find congruence relations of index less than $2^{\mathcal{O}(n \log n)}$ that recognize $\Sigma^{\omega}\backslash\mathcal{L}(\mathcal{A})$.

We now introduce the construction of FDFAs from the right congruences. Since $\backsim^o$ (resp., $\backsim^i$) and $\approx^o_u$ (resp., $\approx_u$) with $u \in \Sigma^*$ are right congruences of finite index, by means of Definition 1 they can be used to define the transition structures of the DFAs of an FDFA $\mathcal{F}$ recognizing $\Sigma^{\omega} \setminus \mathcal{L}(\mathcal{A})$. Moreover, by Lemma 13 (resp., Lemma 7), we can identify the accepting macrostates of the progress DFAs. We now give the construction of the FDFA $\mathcal{F}$ with $\backsim^o$ and $\approx^o_u$. The construction of the FDFA with $\backsim^i$ and $\approx_u$ is similar.

Definition 11. *The FDFA $\mathcal{F}$ is a tuple $(\mathcal{M}[\backsim^o], \{\mathcal{N}_u[\approx^o_u]\})$ where*

- *$\mathcal{M}[\backsim^o]$ is the DFA induced by $\backsim^o$ according to Definition 1;*
- *for each macrostate $[u]_{\backsim^o}$ of $\mathcal{M}[\backsim^o]$, the progress DFA $\mathcal{N}_u[\approx^o_u]$ is constructed as in Definition 1 parameterized with $\approx^o_u$. The accepting macrostates of*

$\mathcal{N}_u[\approx_u^o]$ are the equivalence classes $[v]_{\approx_u^o}$ of $\approx_u^o$ such that $uv \backsim^o u$ and $[u]_{\backsim^o}[v]^\omega_{\approx_u^o} \cap \mathcal{L}(\mathcal{A}) = \emptyset$.

The FDFA constructed according to Definition 11 has the desired properties we are looking for: it accepts $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$ and has only $2^{\mathcal{O}(n \log n)}$ macrostates.

Theorem 5. *Let $\mathcal{F}$ be the FDFA constructed from $\mathcal{A}$ in Definition 11. Then (1) $UP(\mathcal{F}) = UP(\Sigma^\omega \backslash \mathcal{L}(\mathcal{A}))$; (2) $\mathcal{F}$ is saturated; and (3) $\mathcal{F}$ has $2^{\mathcal{O}(n \log n)}$ macrostates.*

The three results stated in Theorem 5 follow by the definition of $\mathcal{F}$ and the properties of $(\backsim^o, \bigcup_{u \in \Sigma^*} \{\approx_u^o\})$: Result (1) is a direct consequence of Definition 11; Result (2) is implied by the saturation lemma for $(\backsim^o, \bigcup_{u \in \Sigma^*} \{\approx_u^o\})$ (cf. Item (1) of Lemma 13); and Result (3) by the indexes of $(\backsim^o, \bigcup_{u \in \Sigma^*} \{\approx_u^o\})$ and the construction of the DFAs of $\mathcal{F}$. It is easy to see that one can also construct an FDFA accepting $\mathcal{L}(\mathcal{A})$ by setting the accepting macrostates of $\mathcal{N}_u[\approx_u^o]$ to be the equivalence classes $[v]_{\approx_u^o}$ such that $uv \backsim^o u$ and $[u]_{\backsim^o}[v]^\omega_{\approx_u^o} \cap \mathcal{L}(\mathcal{A}) \neq \emptyset$.

We are now able to formalize the optimality of our FDFA construction of $\mathcal{F}$ based on $(\backsim^o, \bigcup_{u \in \Sigma^*} \{\approx_u^o\})$. The upper bound is due to Theorem 5; the matching lower bound comes from the well-known fact [23] that there exists a family of NBAs $\{\mathcal{A}_n\}_{n \in \mathbb{N}}$ whose complementary NBAs $\{\mathcal{A}_n^c\}_{n \in \mathbb{N}}$ have $2^{\Omega(n \log n)}$ states, so the same lower bound must hold for FDFAs since there are polynomial-time translations from FDFAs to NBAs (cf. Lemma 1).

Theorem 6. *The construction of FDFAs in Definition 11 with the right congruence relations $(\backsim^o, \bigcup_{u \in \Sigma^*} \{\approx_u^o\})$ from $\mathcal{A}$ is asymptotically optimal.*

Remark 3. Here we discuss related works on FDFAs. As mentioned before, there are polynomial-time translations from FDFAs to NBAs [3,7]. The opposite translation is more challenging: the direct translations from an n-states NBA, proposed in [7] and in [13], produce an FDFA with $\mathcal{O}(4^{n^2+n})$ states and an FDFA with $\mathcal{O}(3^{n^2+n})$ states, respectively. Our construction in Definition 11 replaced with $(\backsim^i, \bigcup_{u \in \Sigma^*} \{\approx_u\})$ can even be exponentially better than these two translations; due to lack of space, the detailed reasoning can be found in [18, Appendix C]. The translation based on an intermediate determinization of NBAs to deterministic Parity automata given in [3] also yields an FDFA with the optimal complexity $2^{\mathcal{O}(n \log n)}$. Our construction (cf. Definition 11), however, is the *first direct* and *optimal* translation from an NBA to an FDFA without involving determinization of NBAs. Like in [13], our congruence relation-based translation also reveals that FDFAs are actually being applied in the language-containment checking of NBAs (cf. [1,2,11]). The specialized translation for DBAs proposed in [3] can be used to convert a DBA $\mathcal{A}$ with n states to a saturated FDFA $\mathcal{F}'$ with $n + n \times 2n \in \mathcal{O}(n^2)$ macrostates such that $\mathrm{UP}(\mathcal{F}') = \mathrm{UP}(\Sigma^\omega \backslash \mathcal{L}(\mathcal{A}))$; we remark that our construction for FDFAs in Sect. 5 degenerates to their construction when the given NBA $\mathcal{A}$ is deterministic. Given an ω-regular language L, Angluin and Fisman [4] directly operate on the language L and give congruence relations for constructing FDFAs of L. In contrast, our work takes an NBA $\mathcal{A}$

as input and defines congruence relations for recognizing $\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$ based on the transitions of $\mathcal{A}$.

We now formalize the main result of this paper as Theorem 7, which is a direct consequence of Theorem 6 since the constructed FDFA has the same number of macrostates as the index the congruence relations $(\backsim^o, \bigcup_{u\in\Sigma^*}\{\approx_u^o\})$ (cf. Definition 11).

Theorem 7. *The right congruence relations* $(\backsim^o, \bigcup_{u\in\Sigma^*}\{\approx_u^o\})$ *given in Definitions 9 and 10, respectively, are asymptotically optimal among all right congruence relations* $(\backsim, \bigcup_{u\in\Sigma^*}\{\approx_u\})$ *such that for each* $u \in \Sigma^*$ *and* $v \in \Sigma^+$*, if* $uv \backsim u$*, then either* $[u]_\backsim [v]^\omega_{\approx_u} \cap \mathcal{L}(\mathcal{A}) = \emptyset$ *or* $[u]_\backsim [v]^\omega_{\approx_u} \subseteq \mathcal{L}(\mathcal{A})$.

6 Concluding Remarks

In this work, we considered congruence relations for NBAs and we have proposed coarser relations than the classical one; we have further given asymptotically *optimal* right congruences for NBAs. To the best of our knowledge, we have given the *first direct* translation from an NBA to an FDFA with *optimal* complexity, based on our optimal right congruences.

We have shown that congruence relations relate tightly the classical RBC and FDFAs. Congruence relations are known to be able to yield the minimal DFAs for given regular languages by the Myhill-Nerode Theorem, by identifying equivalent states [19,20]. The classical congruence relation $\backsim$ has already been exploited to avoid exploration of redundant states, which explains why RBC is suitable for developing state-space pruning techniques [1,2,11]. We believe that the congruence relations we introduced in this work may further enable the reduction of the state space in practical NBA complementation construction. In future work, we plan to study whether subsumption and simulation techniques, similar to the ones developed in [1,2] for the classical congruence relation, can also be proposed for our right congruences, in the context of language-containment checking between two NBAs.

As mentioned in the introduction, formal verification based on NBAs is PSPACE-complete, which is computationally expensive. An alternative and also cheaper method is to model the system and the specification as FDFAs, so that the model-checking problem can be reduced to a containment problem between two FDFAs, which can be done in polynomial time [3]. We believe that our work benefits the community with a deep understanding of the relationship between NBAs and FDFAs, which may help with the modelling of systems as FDFAs and enhance the possibility of the use of FDFAs in formal verification.

Acknowledgements. We would like to thank the anonymous reviewers for their valuable suggestions and comments about this paper. Work supported in part by the Guangdong Science and Technology Department (Grant No. 2018B010107004), NSF grants IIS-1527668, CCF-1704883 and IIS-1830549, and an award from the Maryland Procurement Office.

References

1. Abdulla, P.A., et al.: Simulation subsumption in Ramsey-based Büchi automata universality and inclusion testing. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 132–147. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14295-6_14
2. Abdulla, P.A., et al.: Advanced Ramsey-based Büchi automata inclusion testing. In: Katoen, J.-P., König, B. (eds.) CONCUR 2011. LNCS, vol. 6901, pp. 187–202. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23217-6_13
3. Angluin, D., Boker, U., Fisman, D.: Families of DFAs as acceptors of ω-regular languages. Log. Methods Comput. Sci. **14**(1), 1–21 (2018)
4. Angluin, D., Fisman, D.: Learning regular omega languages. Theor. Comput. Sci. **650**, 57–72 (2016)
5. Breuers, S., Löding, C., Olschewski, J.: Improved Ramsey-based Büchi complementation. In: FOSSACS, pp. 150–164 (2012)
6. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Proceedings of the International Congress on Logic, Method, and Philosophy of Science 1960, pp. 1–12. Stanford University Press (1962)
7. Calbrix, H., Nivat, M., Podelski, A.: Ultimately periodic words of rational ω-languages. In: Brookes, S., Main, M., Melton, A., Mislove, M., Schmidt, D. (eds.) MFPS 1993. LNCS, vol. 802, pp. 554–566. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58027-1_27
8. Clemente, L., Mayr, R.: Efficient reduction of nondeterministic automata with application to language inclusion testing. Logic. Methods Comput. Sci. **15**(1), 12:1-12:73 (2019)
9. Fogarty, S., Kupferman, O., Vardi, M.Y., Wilke, T.: Profile trees for Büchi word automata, with application to determinization. Inf. Comput. **245**, 136–151 (2015)
10. Fogarty, S., Kupferman, O., Wilke, T., Vardi, M.Y.: Unifying Büchi complementation constructions. Logic. Methods Comput. Sci. **9**(1), 1–22 (2013)
11. Fogarty, S., Vardi, M.Y.: Efficient Büchi universality checking. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 205–220. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12002-2_17
12. Kähler, D., Wilke, T.: Complementation, disambiguation, and determinization of Büchi automata unified. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 724–735. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70575-8_59
13. Kuperberg, D., Pinault, L., Pous, D.: Coinductive algorithms for Büchi automata. In: Hofman, P., Skrzypczak, M. (eds.) DLT 2019. LNCS, vol. 11647, pp. 206–220. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24886-4_15
14. Kupferman, O., Vardi, M.Y.: Verification of fair transition systems. In: Alur, R., Henzinger, T.A. (eds.) CAV 1996. LNCS, vol. 1102, pp. 372–382. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61474-5_84
15. Kupferman, O., Vardi, M.Y.: Model checking of safety properties. Form. Methods Syst. Des. **19**(3), 291–314 (2001)
16. Kupferman, O., Vardi, M.Y.: Weak alternating automata are not that weak. ACM Trans. Comput. Logic **2**(3), 408–429 (2001)
17. Li, Y., Chen, Y.F., Zhang, L., Liu, D.: A novel learning algorithm for Büchi automata based on family of DFAs and classification trees. In: TACAS, pp. 208–226. Springer (2017). https://doi.org/10.1007/978-3-662-54577-5_12

18. Li, Y., Tsay, Y.K., Turrini, A., Vardi, M.Y., Zhang, L.: Congruence relations for Büchi automata. CoRR abs/2104.03555 (2021)
19. Myhill, J.: Finite automata and the representation of events. In: Technical Report WADD TR-57-624, pp. 112–137 (1957)
20. Nerode, A.: Linear automaton transformations. Am. Math. Soc. **9**, 541–544 (1958)
21. Sistla, A.P., Vardi, M.Y., Wolper, P.: The complementation problem for Büchi automata with applications to temporal logic. Theor. Comput. Sci. **49**(2–3), 217–237 (1987)
22. Thomas, W.: Automata on infinite objects. In: Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics, pp. 133–191. Elsevier and MIT Press (1990)
23. Yan, Q.: Lower bounds for complementation of ω-automata via the full automata technique. Logic. Methods Comput. Sci. **4**(1:5), 1–20 (2008)