Singly Exponential Translation of Alternating Weak Büchi Automata to Unambiguous Büchi Automata

Yong Li
University of Liverpool, UK
SKLCS, Institute of Software, Chinese Academy of Sciences, Beijing, China

Sven Schewe
University of Liverpool, UK

Moshe Y. Vardi
Rice University, Houston, TX, USA

Abstract

We introduce a method for translating an alternating weak Büchi automaton (AWA), which corresponds to a Linear Dynamic Logic (LDL) formula, to an unambiguous Büchi automaton (UBA). Our translations generalise constructions for Linear Temporal Logic (LTL), a less expressive specification language than LDL. In classical constructions, LTL formulas are first translated to alternating *very weak* automata (AVAs) – automata that have only singleton strongly connected components (SCCs); the AVAs are then handled by efficient disambiguation procedures. However, general AWAs can have larger SCCs, which complicates disambiguation. Currently, the only available disambiguation procedure has to go through an intermediate construction of nondeterministic Büchi automata (NBAs), which would incur an exponential blow-up of its own. We introduce a translation from *general* AWAs to UBAs with a *singly* exponential blow-up, which also immediately provides a singly exponential translation from LDL to UBAs. Interestingly, the complexity of our translation is *smaller* than the best known disambiguation algorithm for NBAs (broadly $(0.53n)^n$ vs. $(0.76n)^n$), while the input of our construction can be exponentially more succinct.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Theory of computation → Verification by model checking

Keywords and phrases Büchi automata, unambiguous automata, alternation, weak, disambiguation

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2023.37

Related Version *Full Version*: https://arxiv.org/abs/2305.09966

Funding This work has been supported in part by the EPSRC through grants EP/X021513/1 and EP/X017796/1, NSFC grant 62102407, NSF grants IIS-1527668, CCF-1704883, IIS-1830549, CNS-2016656, DoD MURI grant N00014-20-1-2787, and an award from the Maryland Procurement Office.

Acknowledgements We thank the anonymous reviewers for their valuable feedback.

1 Introduction

Automata over infinite words were first introduced by Büchi [8]. The automata used by Büchi (thus called *Büchi automata*) accept an infinite word if they have a run over the word that visits accepting states infinitely often. Nondeterministic Büchi automata (NBAs) are nowadays recognized as a standard tool for model checking transition systems against temporal specification languages like Linear Temporal Logic (LTL) [1, 11, 13, 26].

NBAs belong to a larger class of automata over infinite words, also known as ω-automata. Translations between different types of ω-automata play a central role in automata theory, and many of them have gained practical importance, too. For example, researchers have started to pay attention to a kind of automata called *alternating automata* [20, 22] in the 80s.

Alternating automata not only have existential, but also *universal* branching. In alternating automata, the transition function no longer maps a state and a letter to a set of states, but to a positive Boolean formula over states. An alternating Büchi automaton accepts an infinite word if there is a run graph over the word, in which all traces visit accepting states infinitely often. Every NBA can be seen as a special type of alternating Büchi automaton (ABA), while the translation from ABAs to NBAs may incur an exponential blow-up in the number of states [20]. Indeed, ABAs can be exponentially more succinct than their counterpart NBAs [6]. Apart from their succinctness, another reason why alternating automata have become popular in our community is their tight connection to specification logics. There is a straight forward translation from Linear Dynamic Logic (LDL) [12, 25] to *alternating weak Büchi automata* (AWAs), both recognizing exactly the ω-regular languages. AWAs are a special type of ABAs in which every strongly connected component (SCC) contains either only accepting states or only rejecting states. (AWAs have also been applied to the complementation of Büchi automata [17].) Further, there is a one-to-one mapping [5, 7, 11] between LTL and *very weak* alternating Büchi automata (AVAs) [23] – special AWAs where every SCC has only one state.

Automata over infinite words with different branching mechanisms all have their place in building the foundation of automata-theoretic model checking. This paper adds another chapter to the success story of efficient automata transformations: we show how to efficiently translate AWAs to unambiguous Büchi automata (UBAs) [10], and thus also the logics that tractably reduce to AWAs, e.g., LDL. UBAs are a type of NBAs that have at most one accepting run for each word and have found applications in probabilistic verification [2][1].

Our approach can be viewed as a generalization of earlier work on the disambiguation of AVAs [4, 14]. The property of the very weakness has proven useful for disambiguation: to obtain an unambiguous generalized Büchi automaton (UGBA) from an AVA, it essentially suffices to use the nondeterministic power of the automaton to guess, in every step, the precise set of states from which the automaton accepts. There is only one correct guess (which provides unambiguity), and discharging the correctness of these guesses is straight forward. AVAs with n states can therefore be disambiguated to UGBAs with 2^n states and n accepting sets, and thus to UBAs with $n2^n$ states.

Unfortunately, this approach does not extend easily to the disambiguation of AWAs: while there would still be exactly one correct guess, the straight-forward way to discharging its correctness would involve a breakpoint construction [20], which is *not* unambiguous.

The technical contribution of this paper is to replace these breakpoint constructions by *total preorders*, and showing that there is a *unique* correct way to choose these orders. We provide two different reductions, one closer to the underpinning principles – and thus better for a classroom (cf. Section 3.4) – and a more efficient approach (cf. Section 4).

Given that we track total preorders, the worst-case complexity arises when all, or almost all, states are in the same component. To be more precise, if $\mathsf{tpo}(n)$ denotes the number of total preorders on sets with n states, then our construction provides UBAs of size $\mathcal{O}\big(\mathsf{tpo}(n)\big)$. As $\mathsf{tpo}(n) \approx \frac{n!}{2(\ln 2)^{n+1}}$ [3], we have that $\lim_{n\to\infty} \frac{\sqrt[n]{\mathsf{tpo}(n)}}{n} = \frac{1}{e \ln 2} \approx 0.53$, which is a better bound than the best known bound for Büchi disambiguation [16] (and complementation [24]), where the latter number is ≈ 0.76.

[1] We note that specialized model checking algorithm for Markov chains against AWAs/LDL, without constructing UBAs, has been proposed in [9] without implementations. Nonetheless, our translation can potentially be used as a third party tool that constructs UBA from an AWA/LDL formula for PRISM model checker [18] without changing the underlying model checking algorithm [2].

While it is not surprising that a direct construction of UBAs for AWAs is superior to a construction that goes through nondeterminization (and thus incurs two exponential blow-ups on the way), we did not initially expect a construction that leads to a smaller increase in the size when starting from an AWA compared to starting from an NBA, as AWAs can be exponentially more succinct than NBAs, but not vice versa (See [17] for a quadratic transformation).

As a final test for the quality of our construction, we briefly discuss how it behaves on AVAs, for which efficient disambiguation is available. We show that the complexity of our construction can be improved to $n2^n$ when the input is an AVA, leading to the same construction as the classic disambiguation construction for LTL/AVAs [4, 14] (cf. Section 5). We also discuss how to adjust it so that it can produce the same transition based UGBA in this case, too. The greater generality we obtain comes therefore at no additional cost.

Related work. Disambiguation of AVAs from LTL specifications have been studied in [4, 14]. Our disambiguation algorithm can be seen as a more general form of them. The disambiguation of NBAs was considered in [15], which has a blow-up of $\mathcal{O}((3n)^n)$; the complexity has been later improved to $\mathcal{O}(n \cdot (0.76n)^n)$ in [16]. Our construction can also be used for disambiguating NBAs, by going through an intermediate construction of AWAs from NBAs; however, the intermediate procedure itself can incur a quadratic blow-up of states [14]. Nonetheless, if the input is an AWA, our construction improves the current best known approach exponentially by avoiding the alternation removal operation for AWAs [6, 20].

2 Preliminaries

For a given set X, we denote by $\mathcal{B}^+(X)$ the set of *positive Boolean* formulas over X. These are the formulas obtained from elements of X by only using $\wedge$ and $\vee$, where we also allow $\mathtt{tt}$ and $\mathtt{ff}$. We use $\mathtt{tt}$ and $\mathtt{ff}$ to represent tautology and contradiction, respectively. For a set $Y \subseteq X$, we say Y satisfies a formula $\theta \in \mathcal{B}^+(X)$, denoted as $Y \models \theta$, if the Boolean formula θ is evaluated to $\mathtt{tt}$ when we assign $\mathtt{tt}$ to members of Y and $\mathtt{ff}$ to members of $X \setminus Y$. For an infinite sequence ρ, we denote by $\rho[i]$ the i-th element in ρ for some $i \geq 0$; for $i \in \mathbb{N}$, we denote by $\rho[i \cdots] = \rho[i]\rho[i+1] \cdots$ the suffix of ρ from its i-th letter.

An *alternating* Büchi automaton (ABA) $\mathcal{A}$ is a tuple $(\Sigma, Q, \iota, \delta, F)$ where Σ is a finite alphabet, Q is a finite set of states, $\iota \in Q$ is the initial state, $\delta : Q \times \Sigma \to \mathcal{B}^+(Q)$ is the transition function, and $F \subseteq Q$ is the set of accepting states. ABAs allow both non-deterministic and universal transitions. The disjunctions in transition formulas model the non-deterministic choices, while conjunctions model the universal choices. The existence of both nondeterministic and universal choices can make ABAs exponentially more succinct than NBAs [6]. We assume w.l.o.g. that every ABA is *complete*, in the sense that there is a next state for each $s \in Q$ and $\sigma \in \Sigma$. Every ABA can be made complete as follows. Fix a state $s \in Q$ and a letter $\sigma' \in \Sigma$. If $\delta(s, \sigma') = \mathtt{ff}$, we can add a sink rejecting state $\perp$, and set $\delta(s, \sigma') = \perp$ and $\delta(\perp, \sigma) = \perp$ for every $\sigma \in \Sigma$; If $\delta(s, \sigma') = \mathtt{tt}$, we can similarly add a sink accepting state $\top$, and set $\delta(s, \sigma') = \top$ and $\delta(\top, \sigma) = \top$ for every $\sigma \in \Sigma$. For a state $s \in Q$, we denote by $\mathcal{A}^s$ the ABA obtained from $\mathcal{A}$ by setting the initial state to s.

The *underlying graph* $\mathcal{G}_{\mathcal{A}}$ of an ABA $\mathcal{A}$ is a graph $\langle Q, E \rangle$, where the set of vertices is the set Q of states in $\mathcal{A}$ and $(q, q') \in E$ if q' appears in the formula $\delta(q, \sigma)$ for some $\sigma \in \Sigma$. We call a set $C \subseteq Q$ a *strongly connected component* (SCC) of $\mathcal{A}$ if, for every pair of states $q, q' \in C$, q and q' can reach each other in $\mathcal{G}_{\mathcal{A}}$.

A *nondeterministic Büchi automaton* (NBA) $\mathcal{A}$ is an ABA where $\mathcal{B}^+(Q)$ only contains the $\vee$ operator; we also allow *multiple* initial states for NBAs. We usually denote the transition function δ of an NBA $\mathcal{A}$ as a function $\delta : Q \times \Sigma \to 2^Q$ and the set of initial states as I. Let $w = w[0]w[1] \cdots \in \Sigma^\omega$ be an (infinite) *word* over Σ.

A *run* of the NBA $\mathcal{A}$ over w is a state sequence $\rho = q_0 q_1 \cdots \in Q^\omega$ such that $q_0 \in I$ and, for all $i \in \mathbb{N}$, we have that $q_{i+1} \in \delta(q_i, w[i])$. We denote by $\mathsf{inf}(\rho)$ the set of states that occur in ρ infinitely often. A run ρ of the NBA $\mathcal{A}$ is *accepting* if $\mathsf{inf}(\rho) \cap F \neq \emptyset$. An NBA $\mathcal{A}$ accepts a word w if there is an accepting run ρ of $\mathcal{A}$ over w. An NBA $\mathcal{A}$ is said to be *unambiguous* (abbreviated as UBA) [10] if $\mathcal{A}$ has at most *one* accepting run for every word.

Since ABA have universal branching (or conjunctions in δ), a run of an ABA is no longer an infinite sequence of states; instead, a run of an ABA $\mathcal{A}$ over w is a run directed acyclic graph (run DAG) $\mathcal{G}_w = (V, E)$ formally defined below:

- $V \subseteq Q \times \mathbb{N}$ where $\langle \iota, 0 \rangle \in V$.
- $E \subseteq \bigcup_{\ell > 0} (Q \times \{\ell\}) \times (Q \times \{\ell + 1\})$ where, for every vertex $\langle q, \ell \rangle \in V, \ell \geq 0$, we have that $\{\, q' \in Q \mid (\langle q, \ell \rangle, \langle q', \ell + 1 \rangle) \in E \,\} \models \delta(q, w[\ell])$.

A vertex $\langle q, \ell \rangle$ is said to be *accepting* if $q \in F$. An infinite sequence $\rho = \langle q_0, 0 \rangle \langle q_1, 1 \rangle \cdots$ of vertices is called an *ω-branch* of $\mathcal{G}_w$ if $q_0 = \iota$ and for all $\ell \in \mathbb{N}$, we have $(\langle q_\ell, \ell \rangle, \langle q_{\ell+1}, \ell + 1 \rangle) \in E$. We also say the fragment $\langle q_i, i \rangle \langle q_{i+1}, i + 1 \rangle \cdots$ of ρ is an *ω-branch* from $\langle q_i, i \rangle$. We say a run DAG $\mathcal{G}_w$ is *accepting* if *all* its ω-branches visit accepting vertices infinitely often. An ω-word w is *accepting* if there is an accepting run DAG of $\mathcal{A}$ over w.

Let $\mathcal{A}$ be an ABA. We denote by $\mathcal{L}(\mathcal{A})$ the set of words accepted by $\mathcal{A}$.

It is known that both NBAs and ABAs recognise exactly the ω-regular languages. ABAs can be transformed into language-equivalent NBAs in exponential time [20]. In this work, we consider a special type of ABAs, called *alternating weak Büchi automata* (AWAs) where, for every SCC C of an AWA $\mathcal{A} = (\Sigma, Q, \iota, \delta, F)$, we have either $C \subseteq F$ or $C \cap F = \emptyset$. We note that different choices of equivalent transition formulas, e.g., $\delta(p, \sigma) = q_1$ and $\delta(p, \sigma) = q_1 \wedge (q_1 \vee q_2)$, will result in different SCCs. However, as long as the input ABA is weak[2], our proposed translation still applies.

One can transform an ABA to its equivalent AWA with only quadratic blow-up of the number of states [17]. A nice property of an AWA $\mathcal{A}$ is that we can easily define its dual AWA $\widehat{\mathcal{A}} = (\Sigma, Q, \iota, \widehat{\delta}, \widehat{F})$, which has the same statespace and the same underlying graph as $\mathcal{A}$, as follows: for a state $q \in Q$ and $a \in \Sigma$, $\widehat{\delta}(q, a)$ is defined from $\delta(q, a)$ by exchanging the occurrences of ff and $\mathtt{tt}$ and the occurrences of $\wedge$ and $\vee$, and $\widehat{F} = Q \setminus F$. It follows that:

▶ **Lemma 1** ([21]). *Let $\mathcal{A}$ be an AWA and $\widehat{\mathcal{A}}$ its dual AWA. For every state $q \in Q$, we have $\mathcal{L}(\mathcal{A}^q) = \Sigma^\omega \setminus \mathcal{L}(\widehat{\mathcal{A}}^q)$.*

In the remainder of the paper, we call a state of an NBA a *macrostate* and a run of an NBA a *macrorun* in order to distinguish them from those of ABA.

3 From AWAs to UBAs

In this section, we will present a construction of UBA $\mathcal{B}_u$ from an AWA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{B}_u) = \mathcal{L}(\mathcal{A})$. We will first introduce the construction of an NBA from an AWA given in [9] and show that this construction does *not* necessarily yield a UBA (Section 3.1). Nonetheless, we extract the essence of the construction and show that we can associate a *unique* sequence to each word (Section 3.2).

[2] To make ABAs as weak as possible, one solution would be computing minimal satisfying assignments to the transition formulas, which is well defined and results in minimal possible SCCs.

We then enrich this unique sequence with additional, similarly unique, information, which we subsequently abstract into the essence of a unique accepting macrorun of $\mathcal{B}_u$. Developing this into a UBA whose macrorun can be uniquely mapped to the sequence (Section 3.4) is then just a simple technical exercise.

3.1 From AWAs to NBAs

As shown in [20], we can obtain an equivalent NBA $\mathcal{N}(\mathcal{A})$ from an ABA $\mathcal{A}$ with an exponential blow-up of states, which is widely known as the *breakpoint construction*. In [9], the authors define a different construction of NBAs $\mathcal{B}$ from AWAs $\mathcal{A}$, which can be seen as a combination of the NBAs $\mathcal{N}(\mathcal{A})$ and $\mathcal{N}(\widehat{\mathcal{A}})$. Below we will first introduce the construction in [9] and extract its essence as a unique sequence of sets of states for each word.

The macrostate of $\mathcal{B}$ is encoded as a *consistent* tuple (Q_1, Q_2, Q_3, Q_4) such that $Q_2 = Q \setminus Q_1, Q_3 \subseteq Q_1 \setminus F$, and $Q_4 \subseteq Q_2 \setminus \widehat{F}$.

The formal translation is defined as follows.

▶ **Definition 2** ([9]). *Let $\mathcal{A} = (\Sigma, Q, \iota, \delta, F)$ be an AWA. We define an NBA $\mathcal{B} = (\Sigma, Q_{\mathcal{B}}, I_{\mathcal{B}}, \delta_{\mathcal{B}}, F_{\mathcal{B}})$ where*

- *$Q_{\mathcal{B}}$ is the set of consistent tuples over $2^Q \times 2^Q \times 2^Q \times 2^Q$.*
- *$I_{\mathcal{B}} = \{ (Q_1, Q_2, Q_3, Q_4) \in Q_{\mathcal{B}} \mid \iota \in Q_1 \}$*[3]*,*
- *Let (Q_1, Q_2, Q_3, Q_4) be a macrostate in $Q_{\mathcal{B}}$ and $\sigma \in \Sigma$.*
  Then $(Q'_1, Q'_2, Q'_3, Q'_4) \in \delta_{\mathcal{B}}((Q_1, Q_2, Q_3, Q_4), \sigma)$ if $Q'_1 \models \wedge_{s\in Q_1} \delta(s, \sigma)$ and $Q'_2 \models \wedge_{s\in Q_2} \widehat{\delta}(s, \sigma)$ and either
  - *$Q_3 = Q_4 = \emptyset, Q'_3 = Q'_1 \setminus F$ and $Q'_4 = Q'_2 \setminus \widehat{F}$,*
  - *$Q_3 \neq \emptyset$ or $Q_4 \neq \emptyset$, there exists $Y_3 \subseteq Q'_1$ such that $Y_3 \models \wedge_{s\in Q_3} \delta(s, \sigma)$ and $Q'_3 = Y_3 \setminus F$, and there exists $Y_4 \subseteq Q'_2$ such that $Y_4 \models \wedge_{s\in Q_4} \widehat{\delta}(s, \sigma)$ and $Q'_4 = Y_4 \setminus \widehat{F}$.*
- *$F_{\mathcal{B}} = \{ (Q_1, Q_2, Q_3, Q_4) \in Q_{\mathcal{B}} \mid Q_3 = Q_4 = \emptyset \}$.*

Intuitively, the resulting NBA performs two breakpoint constructions: one breakpoint construction macrostate (Q_1, Q_3) for $\mathcal{A}$ and the other breakpoint construction macrostate (Q_2, Q_4) for $\widehat{\mathcal{A}}$. Let $w \in \Sigma^\omega$. The tuple (Q_1, Q_3) in the construction uses Q_1 to keep track of the reachable states of $\mathcal{A}$ in a run DAG $\mathcal{G}_w$ over w and exploits the set Q_3 to check whether all ω-branches end in accepting SCCs. If all ω-branches in Q_3 have visited accepting vertices, Q_3 will fall empty, as Q_3 only contains non-accepting states. Once Q_3 becomes empty, we reset the set with $Q'_3 = Q'_1 \setminus F$ since we need to also check the ω-branches that newly appear in Q_1. If Q_3 becomes empty for infinitely many times, we know that every ω-branch in $\mathcal{G}_w$ is accepting, i.e., all ω-branches visit accepting vertices infinitely often. Hence w is accepted by $\mathcal{A}$ since there is an accepting run DAG from $\mathcal{A}^\iota$. We can similarly reason about the breakpoint construction for $\widehat{\mathcal{A}}$.

Besides that $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A})$, Bustan, Rubin, and Vardi [9] have also shown the following:

▶ **Lemma 3** ([9]). *Let $\mathcal{B}$ be the NBA constructed as in Definition 2. Then*

- *Let $S \subseteq Q$, we have that*

$$\mathcal{L}(\mathcal{B}^{(S, Q\setminus S, Q_3, Q_4)}) = \bigcap_{s\in S} \mathcal{L}(\mathcal{A}^s) \cap \bigcap_{s\in Q\setminus S} \mathcal{L}(\widehat{\mathcal{A}}^s)$$

  where $Q_3 \subseteq S$ and $Q_4 \subseteq Q \setminus S$;

[3] $I_{\mathcal{B}}$ is not present in [9] and we added it for the completeness of the definition.

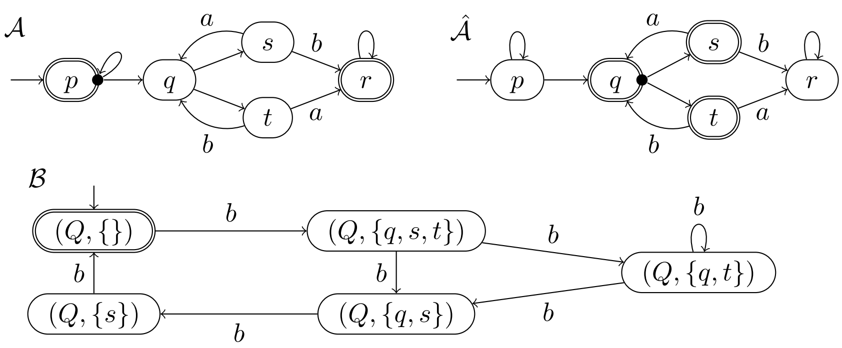

Figure 1 An example of an AWA $\mathcal{A}$, its dual $\widehat{\mathcal{A}}$ and *incomplete* part of the constructed $\mathcal{B}$ over b^ω, where for instance the transition $((Q, \{q, s\}), b, (Q, \{t\}))$ is missing.

- *Let* (Q_1, Q_2, Q_3, Q_4) *and* (Q'_1, Q'_2, Q'_3, Q'_4) *be two macrostates of* $\mathcal{B}$*, we have that*
  - $\mathcal{L}(\mathcal{B}^{(Q_1,Q_2,Q_3,Q_4)}) \cap \mathcal{L}(\mathcal{B}^{(Q'_1,Q'_2,Q'_3,Q'_4)}) = \emptyset$ *if* $Q_1 \neq Q'_1$*, and*
  - $\mathcal{L}(\mathcal{B}^{(Q_1,Q_2,Q_3,Q_4)}) = \mathcal{L}(\mathcal{B}^{(Q'_1,Q'_2,Q'_3,Q'_4)})$ *if* $Q_1 = Q'_1$.

Let $w \in \mathcal{L}(\mathcal{B})$ and $\rho = (Q_1^0, Q_2^0, Q_3^0, Q_4^0)(Q_1^1, Q_2^1, Q_3^1, Q_4^1)\cdots$ be an accepting macrorun of $\mathcal{B}$ over w. According to Lemma 3, it is easy to see that the Q_1-set sequence $Q_1^0 Q_1^1 \cdots$ is in fact *unique* for every accepting macrorun over w. If there are two accepting macroruns, say ρ_1 and ρ_2, of $\mathcal{B}$ over w that have two different Q_1-set sequences, there must be a position $j \geq 0$ such that their Q_1-sets differ. By Lemma 3, the suffix $w[j \cdots]$ cannot be accepted from both macrostates $\rho_1[j]$ and $\rho_2[j]$, leading to contradiction. Therefore, every accepting macrorun of $\mathcal{B}$ over w corresponds to a unique sequence of Q_1-sets. However, $\mathcal{B}$ does not necessarily have only one accepting macrorun over w, because there is *nondeterminism* in developing the breakpoints.

▶ **Lemma 4.** *The NBA* $\mathcal{B}$ *defined as in Definition 2 is not necessarily unambiguous.*

Proof. We prove Lemma 4 by giving an example AWA $\mathcal{A}$ for which the constructed $\mathcal{B}$ is *not* unambiguous. The example AWA $\mathcal{A}$ and its dual $\widehat{\mathcal{A}}$ are given in Figure 1 where accepting states are depicted with double circles, initial states are marked with an incoming arrow and universal transitions are originated from a black filled circle. The transitions are by default labelled with $\Sigma = \{a, b\}$ unless explicitly labelled otherwise. We let $Q = \{p, q, s, t, r\}$. First, we can see that $b^\omega \in \mathcal{L}(\mathcal{A}^p) \cap \mathcal{L}(\mathcal{A}^q) \cap \mathcal{L}(\mathcal{A}^s) \cap \mathcal{L}(\mathcal{A}^t) \cap \mathcal{L}(\mathcal{A}^r)$. So the unique Q_1-sequence of all accepting macroruns in $\mathcal{B}$ over b^ω should be Q^ω, according to Lemma 3. We only depict an *incomplete* part of $\mathcal{B}$ over b^ω where we ignore the Q_2 and Q_4 sets because we have constantly $Q_2 = \{\}$ and $Q_4 = \{\}$ by definition. One of the initial macrostates is $m_0 = (Q, \{\})$, which is also accepting. When reading the letter b, we always have $\{p, q, s, t, r\} \models \wedge_{c \in Q} \delta(c, b)$. Thus, the successor of m_0 over b is $m_1 = (Q, Q \setminus \{p, r\}) = (Q, \{q, s, t\})$ since the breakpoint set Q'_3 needs to be reset to $Q'_1 \setminus F$ when $Q_3 = \{\}$. When choosing the successor set Q'_3 for $Q_3 = \{q, s, t\}$ at m_1, we have two options, namely $\{q, s\}$ and $\{q, t\}$, since q has nondeterministic choices upon reading letter b. Consequently, $\mathcal{B}$ can transition to either $m_2 = (Q, \{q, s\})$ or $m_3 = (Q, \{q, t\})$, upon reading b in m_1. In fact, all the nondeterminism of $\mathcal{B}$ in Figure 1 is due to nondeterministic choices at q. We can continue to explore the state space of $\mathcal{B}$ according to Definition 2 and obtain the incomplete part of $\mathcal{B}$ depicted in Figure 1. Note that, we have ignored some outgoing transitions from $(Q, \{q, s\})$ since the present part already suffices to prove Lemma 4. It is easy to see that $\mathcal{B}$ has at least two accepting macroruns over b^ω. Thus we have proved Lemma 4. ◀

In fact, based on Definition 2, it is easy to compute a unique sequence of sets of states for each given word, which builds the foundation of our proposed construction.

3.2 Unique sequence of sets of states for each word

In the remainder of the paper, we fix an AWA $\mathcal{A} = (\Sigma, Q, \iota, \delta, F)$. For every word $w \in \Sigma^\omega$, we define a *unique* sequence of sets of states associated with it as the sequence $Q_1^0 Q_1^1 Q_1^2 \cdots$ such that, for every $i \geq 0$, we have that:

P1 $Q_1^i \subseteq Q$,
P2 for every state $q \in Q_1^i$, $w[i \cdots] \in \mathcal{L}(\mathcal{A}^q)$ and
P3 for every state $q \in Q \setminus Q_1^i$, $w[i \cdots] \notin \mathcal{L}(\mathcal{A}^q)$ (or, similarly, $w[i \cdots] \in \mathcal{L}(\widehat{\mathcal{A}}^q)$).

These properties immediately entail the weaker *local* consistency requirements:

L2 for every state $q \in Q_1^i$, $Q_1^{i+1} \models \delta(q, w[i])$ (entailed by P2) and
L3 for every state $q \in Q \setminus Q_1^i$, $Q \setminus Q_1^{i+1} \models \widehat{\delta}(q, w[i])$ (entailed by P3).

It is obvious that, for every state $s \in Q$, $\Sigma^\omega = \mathcal{L}(\mathcal{A}^s) \uplus \overline{\mathcal{L}(\mathcal{A}^s)} = \mathcal{L}(\mathcal{A}^s) \uplus \mathcal{L}(\widehat{\mathcal{A}}^s)$ holds. We define $Q_w = \{ s \in Q \mid w \in \mathcal{L}(\mathcal{A}^s) \}$. This clearly provides $Q \setminus Q_w = \{ s \in Q \mid w \in \mathcal{L}(\widehat{\mathcal{A}}^s) \}$. For every $w \in \Sigma^\omega$, we therefore have

$$w \in \bigcap_{s \in Q_w} \mathcal{L}(\mathcal{A}^s) \cap \bigcap_{s \in Q \setminus Q_w} \overline{\mathcal{L}(\mathcal{A}^s)} \text{ or, equivalently, } w \in \bigcap_{s \in Q_w} \mathcal{L}(\mathcal{A}^s) \cap \bigcap_{s \in Q \setminus Q_w} \mathcal{L}(\widehat{\mathcal{A}}^s).$$

For every $i \geq 0$, P2 and P3 are then equivalent to the requirement $Q_1^i = Q_{w[i \cdots]}$.

To see how the local constraints L2 and L3 can be obtained from P2 and P3, respectively, we fix an integer $i \geq 0$. Let $s \in Q_1^i$, so we know that $\mathcal{A}^s$ accepts $w[i \cdots]$. Let S^{i+1} be the set of successors of s in an accepting run DAG of $\mathcal{A}^s$ over $w[i \cdots]$, i.e., $S^{i+1} \models \delta(s, w[i])$. As the run DAG is accepting, we in particular have, for every $t \in S^{i+1}$, that $\mathcal{A}^t$ accepts $w[i+1 \cdots]$, which implies $S^{i+1} \subseteq Q_1^{i+1}$. With $S^{i+1} \models \delta(s, w[i])$, this provides $Q_1^{i+1} \models \delta(s, w[i])$, and thus L2.

Similarly, we can also show that, for every state $q \in Q \setminus Q_1^i$, we have $Q \setminus Q_1^{i+1} \models \widehat{\delta}(q, w[i])$. As before, $\widehat{\mathcal{A}}^q$ accepts $w[i \cdots]$ for every $q \in Q \setminus Q_1^i$ by definition. We let S^{i+1} be the set of successors of q in an accepting run DAG of $\widehat{\mathcal{A}}^q$. This implies at the same time $S^{i+1} \models \widehat{\delta}(q, w[i])$ (local constraints for the run DAG) and $S^{i+1} \subseteq Q \setminus Q_1^{i+1}$ (as the subgraphs starting there must be accepting). Together, this provides $Q \setminus Q_1^{i+1} \models \widehat{\delta}(q, w[i])$, and thus L3 also holds.

Moreover, every set Q_1^i is uniquely defined based on the word $w[i \cdots]$. Therefore, the sequence $\mathtt{R}_w = Q_1^0 Q_1^1 \cdots$ we have defined above indeed is the unique sequence satisfying P1, P2, and P3. Let us consider again the NBA construction of Definition 2: obviously, it enforces the local consistency requirements L2 and L3 on the definition of the transition relation $\delta_{\mathcal{B}}$, which is the necessary condition for the Q_1-sequence being unique; the sufficient condition that $Q_1^i = Q_{w[i \cdots]}$ for all $i \in \mathbb{N}$ is guaranteed with the two breakpoint constructions.

In the remainder of the paper, we denote this unique sequence for a given word w by $\mathtt{R}_w$. The UBA we will construct has to guess (not only) this unique sequence correctly on the fly, but also when it leaves each SCC, as shown later.

3.3 Unique distance functions

As discussed before, we have a unique sequence $\mathtt{R}_w = Q_1^0 Q_1^1 \cdots$ for w. However, as we have seen in Section 3.1, $\mathtt{R}_w$ alone does not suffice to yield an UBA. The construction from Section 3.1, for example, validates that all rejecting SCCs can be left using breakpoints, and we have shown how that leaves leeway w.r.t. how these breakpoints are met. In this section, we discuss a different, an unambiguous (but not finite) way to check the correctness of $\mathtt{R}_w$ by making the minimal time it takes from a state, for the given input word, to leave the rejecting SCC of $\mathcal{A}$ or $\widehat{\mathcal{A}}$ on every branch of this run DAG. For instance, in Figure 1, it is possible to select different successors for state q when reading a b, going to either s or t. One of them will lead to leaving this SCC immediately, either s (when reading a b) or t (when reading an a). For acceptance, the choice does not matter – so long as the correct choice is eventually made. On the word b^ω, for example in $\mathcal{A}$, we could go to t the first 20 times, and to s only in the 21^{st} attempt; the answer to the question 'how long does it take to leave the SCC starting in q on this run DAG?' would be 42.

The *shortest* time, however, is well defined. In the example automaton $\mathcal{A}$, it depends on the next letter: if it is a, then the distance is 1 from t, 2 from q, and 3 from s, and when it is b, then the distance is 1 from s, 2 from q, and 3 from t.

To reason about the minimal number of steps it takes from a state within a rejecting SCC that needs to leave it, we will define a *distance function*.

Formally, we denote by R the set of states in all rejecting SCCs of $\mathcal{A}$ and A the set of states in all accepting SCCs of $\mathcal{A}$. For a given word w and its unique sequence $\mathtt{R}_w$, we identify the unique distance[4] to leave a rejecting SCCs at each level i in $\mathcal{G}_w$ by defining a distance function $d_i : (Q_1^i \cap R) \uplus (A \setminus Q_1^i) \to \mathbb{N}^{>0}$ for each $i \in \mathbb{N}$.

▶ **Definition 5.** *Let w be a word and $\mathtt{R}_w = Q_1^0 Q_1^1 \cdots$ be its unique sequence of sets of states. We say $\Phi_w = (Q_1^0, d_0)(Q_1^1, d_1)\cdots$ is* consistent *if, for every $i \in \mathbb{N}$, we have (Q_1^i, d_i) and (Q_1^{i+1}, d_{i+1}) satisfy the following rules:*

R1. *For every state $p \in R \cap Q_1^i$ that belongs to a rejecting SCC C in $\mathcal{A}$,*

$$a: \ (Q_1^{i+1} \setminus C) \cup \{q \in C \cap Q_1^{i+1} \mid d_{i+1}(q) \leq d_i(p) - 1\} \models \delta(p, w[i]) \ \textit{and}$$

$$b: \ \textit{if } d_i(p) > 1, (Q_1^{i+1} \setminus C) \cup \{q \in C \cap Q_1^{i+1} \mid d_{i+1}(q) \leq d_i(p) - 2\} \not\models \delta(p, w[i]) \ \textit{hold.}$$

R2. *For every state $p \in A \setminus Q_1^i$ that belongs to an accepting SCC C in $\mathcal{A}$,*

$$a: \ \big(Q \setminus (Q_1^{i+1} \cup C)\big) \cup \{q \in C \setminus Q_1^{i+1} \mid d_{i+1}(q) \leq d_i(p) - 1\} \models \widehat{\delta}(p, w[i]) \ \textit{and}$$

$$b: \ \textit{if } d_i(q) > 1, \big(Q \setminus (Q_1^{i+1} \cup C)\big) \cup \{q \in C \setminus Q_1^{i+1} \mid d_{i+1}(q) \leq d_i(p) - 2\} \not\models \widehat{\delta}(p, w[i]) \ \textit{hold.}$$

Intuitively, the distance function defines a *minimal* number of steps to escape from rejecting SCCs over different accepting run DAGs and *maximal* over different branches of one such run DAG.

For instance, when $d_i(p) = 1$, we have that $Q_1^{i+1} \setminus C \models \delta(p, w[i])$ if $p \in Q_1^i \cap R$, otherwise $Q \setminus (Q_1^{i+1} \cup C) \models \widehat{\delta}(p, w[i])$ if $p \in A \setminus Q_1^i$. It means that p can escape from C within one step from an accepting run DAG $\mathcal{G}_{w[i\cdots]}$ starting from $\langle p, 0 \rangle$.

▶ **Lemma 6.** *For each $w \in \Sigma^\omega$, there is a unique consistent sequence $\Phi_w = (Q_1^0, d_0)(Q_1^1, d_2)\cdots$ where $Q_1^0 Q_1^1 Q_1^2 \cdots$ is $\mathtt{R}_w$ and $d_0 d_1 \cdots$ is the sequence of distance functions.*

One can easily construct a consistent sequence of distance functions as follows. Let C be a rejecting SCC of $\mathcal{A}$; the case for a rejecting SCC of $\widehat{\mathcal{A}}$ is entirely similar. Below, we describe how to obtain a sequence of distance values for each state $q \in C \cap Q_1^i$ with $i \geq 0$ in order to

---

[4] Note that, while the distance is unique, the way does not have to be. To see this, we could just expand the alphabet of $\mathcal{A}$ by adding a letter c, and by adding c to the transitions from both s and t to r. Then there are two equally short (length 2) ways from q to r whenever the next letter is c.

form a consistent sequence Φ_w. For $q \in C \cap Q_1^i$ at the level i, we first obtain an accepting run DAG $\mathcal{G}_{w[i\cdots]}$ over $w[i\cdots]$ starting from $\langle q, 0\rangle$. One can define the maximal distance, say K, over *all* branches from $\langle q, 0\rangle$ to escape the rejecting SCC C. Such a maximal distance value must exist and be a finite value, since all branches will eventually get trapped in accepting SCCs. For all accepting run DAGs $\mathcal{G}'_{w[i\cdots]}$ over $w[i\cdots]$ starting from the vertex $\langle q, 0\rangle$, there are only finitely many run DAGs of depth K from $\langle q, 0\rangle$; we denote the finite set of such run DAGs of depth K by $P_{q,i}$. We then denote the maximal distance over one *finite* run DAG $G_{q,i,K} \in P_{q,i}$ by $K_{G_{q,i,K}}$. (Note that we set the distance to ∞ for a finite branch in $G_{q,i,K}$ if it does not visit a state outside C.) We then set $d_i(q) = \min\{K_{G_{q,i,K}} : G_{q,i,K} \in P_{q,i}\} \leq K$. One of $G_{q,i,K}$ must provide the *minimal* value, so that $d_i(q)$ is well defined. This way, we can define the sequence of distance functions $\mathbf{d} = d_0 d_1 \cdots$ for the sequence $\mathtt{R}_w$. We can also prove that the sequence $\mathtt{R}_w \times \mathbf{d}$ is consistent by an induction on all the distance values $k > 0$; We refer to [19] for the details.

The proof for the uniqueness of $\mathbf{d}$ to $\mathtt{R}_w$ can also be obtained by an induction on the distance value $k > 0$; See [19] for details. The intuition is that every consistent sequence of distance functions $\mathbf{c}$ does not have smaller distance values than $\mathbf{d}$ for every state $q \in C \cap Q_1^i$ (see the construction of $\mathbf{d}$ above), and if $\mathbf{c}$ does have greater distance values for some state, a violation of the consistency constraints in Definition 5 will be found, leading to contradiction.

3.4 Unique total preorders

The range of the sequence $\mathbf{d} = d_0 d_1 d_2 \ldots$ of distance functions for $\mathtt{R}_w$ is not a priori bounded by any given *finite* number when ranging over all infinite words. Therefore, we may need *infinite* amount of memory to store d. To allow for an abstraction of d that preserves uniqueness and needs only finite memory, we will abstract the values of each function d_i as families of total *preorders*, $\{\preceq_C^i\}_{C \in \mathcal{S}}$, where $\mathcal{S}$ denotes the set of SCCs in the graph of $\mathcal{A}$. For a given SCC $C \in \mathcal{S}$, a total preorder $\preceq_C^i$ is a relation defined over $H^i \times H^i$, where $H^i = C \cap Q_1^i$ if $C \subseteq R$ or $H^i = C \setminus Q_1^i$ if $C \subseteq A$; As usual, $\preceq_C^i$ is *reflexive* (i.e., for each $q \in H^i$, $q \preceq_C^i q$) and *transitive* (i.e., for each $q, r, s \in H^i$, $q \preceq_C^i r$ and $r \preceq_C^i s$ implies $q \preceq_C^i s$). We also have $q \prec_C^i r$ whenever $q \preceq_C^i r$ but $r \not\preceq_C^i q$. We write $q \simeq_C^i r$ if we have $q \preceq_C^i r$ and $r \preceq_C^i q$. Since $\preceq_C^i$ is total, for every two states $p, q \in H^i$, we have $p \preceq_C^i q$ or $q \preceq_C^i p$. Note that $\prec_C^i$ is acyclic: it is impossible for two states $q, p \in H^i$ satisfying $p \prec_C^i q$ and $q \prec_C^i p$.

Formally, we define a consistent sequence of total preorders as below.

▶ **Definition 7.** *Let $w \in \Sigma^\omega$ and $\mathtt{R}_w = Q_1^0 Q_1^1 \cdots$ be its unique sequence of sets of states. We say $\mathcal{P}_w = (Q_1^0, \{\preceq_C^0\}_{C \in \mathcal{S}})(Q_1^1, \{\preceq_C^1\}_{C \in \mathcal{S}}) \cdots$ is* consistent *if, for every $i \in \mathbb{N}$, we have that $(Q_1^i, \{\preceq_C^i\}_{C \in \mathcal{S}})$ and $(Q_1^{i+1}, \{\preceq_C^{i+1}\}_{C \in \mathcal{S}})$ satisfy the following rules:*

R1'. *$\forall q, q' \in C \cap Q_1^i \subseteq R$, we have that $q \prec_C^i q'$ iff there exists $r \in C \cap Q_1^{i+1}$ such that*

$$a: \ \{r' \in C \cap Q_1^{i+1} \mid r' \prec_C^{i+1} r\} \cup (Q_1^{i+1} \setminus C) \models \delta(q, w[i]) \text{ and}$$

$$b: \ \{r' \in C \cap Q_1^{i+1} \mid r' \prec_C^{i+1} r\} \cup (Q_1^{i+1} \setminus C) \not\models \delta(q', w[i]) \text{ hold,}$$

where $C \subseteq R$ is a rejecting SCC of $\mathcal{A}$.

R2'. *$\forall q, q' \in C \setminus Q_1^i \subseteq A$, we have $q \prec_C^i q'$ iff there exists $r \in C \setminus Q_1^{i+1}$ such that*

$$a: \ \{r' \in C \setminus Q_1^{i+1} \mid r' \prec_C^{i+1} r\} \cup \big(Q \setminus (Q_1^{i+1} \cup C)\big) \models \widehat{\delta}(q, w[i]) \text{ and}$$

$$b: \ \{r' \in C \setminus Q_1^{i+1} \mid r' \prec_C^{i+1} r\} \cup \big(Q \setminus (Q_1^{i+1} \cup C)\big) \not\models \widehat{\delta}(q', w[i]) \text{ hold,}$$

where $C \subseteq A$ is an accepting SCC of $\mathcal{A}$.

As the names indicate, the Rules R1' and R2' correspond to Rules R1 and R2, respectively, from Definition 5. We will first show that there is a consistent sequence of total preorders for each word.

▶ **Lemma 8.** *For each word $w \in \Sigma^\omega$, there exists a consistent sequence $\mathcal{P}_w = (Q_1^0, \{\preceq_C^0\}_{C\in\mathcal{S}})(Q_1^1, \{\preceq_C^1\}_{C\in\mathcal{S}})\cdots$, where $Q_1^0 Q_1^1 \cdots$ is the unique sequence $\mathtt{R}_w$.*

Proof. It is simple to derive a consistent sequence $\mathcal{P}_w = (Q_1^0, \{\preceq_C^0\}_{C\in\mathcal{S}})(Q_1^1, \{\preceq_C^1\}_{C\in\mathcal{S}})\cdots$ from $\Phi_w = (Q_1^0, d_0)(Q_1^1, d_1)\cdots$ as given in Lemma 6: We can simply select, for all $i \in \mathbb{N}$ and $C \in \mathcal{S}$, $\preceq_C^i$ is the total preorder over $C \cap Q_1^i$ (if $C \subseteq R$) or $C \setminus Q_1^i$ (if $C \subseteq A$) with $p \preceq_C^i q$ iff $d_i(p) \leq d_i(q)$. In particular, $p \prec_C^i q$ iff $d_i(p) < d_i(q)$.

It is easy to verify that the sequence $\mathcal{P}_w$ as defined above is indeed consistent. For instance, for all $q, q' \in C \cap Q_1^i \subseteq R$, if $q \prec_C^i q'$, then $d_i(q) < d_i(q')$ by definition. Then we can choose the r-state in Definition 7 (Rule R1') such that $d_{i+1}(r) = d_i(q') - 1$. (Note that some such a state r must exist since $d_i(q') > d_i(q) \geq 1$.)

Combining Definition 5 (R1) and Definition 7 (R1'), we have that Rule R1b now entails R1'b, and Rule R1a entails R1'a, because $\{r' \in C \cap Q_1^{i+1} \mid r' \prec_C^{i+1} r\} \supseteq \{r' \in C \cap Q_1^{i+1} \mid d_{i+1}(r') \leq d_i(q) - 1\}$, because $d_i(q) - 1 \leq d_i(q') - 2 < d_i(q') - 1 = d_{i+1}(r)$.

The argument for accepting SCCs is using rules R2 and R2' in the same way. ◀

After discussing how such a sequence can be obtained, we now establish that it is unique. Note, however, that it is unique for the correct sequence $\mathtt{R}_w$, while there may be sequences of total preorders that work with incorrect sequences of sets of states. (For example, a total preorder can accommodate an infinite distance for all states, where the obligation to leave a rejecting SCC cannot be discharged, while the local consistency constraints can be met.) Nonetheless, a breakpoint construction ensures to obtain the unique sequence $\mathtt{R}_w$.

▶ **Lemma 9.** *Let w be a word in Σ^ω and $\Phi_w = (Q_1^0, d_0)(Q_1^1, d_1)\cdots$ be its unique consistent sequence of distance functions. Let $\mathcal{P}_w = (Q_1^0, \{\preceq_C^0\}_{C\in\mathcal{S}})(Q_1^1, \{\preceq_C^1\}_{C\in\mathcal{S}})\cdots$ be a sequence satisfying Definition 7. Then*

- *For every two states $q, q' \in C \cap Q_1^i \subseteq R$, if $q \preceq_C^i q'$, then $d_i(q) \leq d_i(q')$, and in particular if $q \prec_C^i q'$, then $d_i(q) < d_i(q')$. (C is a rejecting SCC)*
- *For every two states $q, q' \in C \setminus Q_1^i \subseteq A$, if $q \preceq_C^i q'$, then $d_i(q) \leq d_i(q')$, and in particular if $q \prec_C^i q'$, then $d_i(q) < d_i(q')$. (C is an accpting SCC)*

Proof. We only prove the first claim; the proof of the second claim is entirely similar.

Let C be a rejecting SCC and i be a natural number. We let q and q' be two states in $C \cap Q_1^i$. In order to prove that $q \preceq_C^i q'$ implies $d_i(q) \leq d_i(q')$, we can just prove its contraposition that $d_i(q') < d_i(q)$ implies $q' \prec_C^i q$ for all distance values $k > 0$ with $d_i(q') \leq k$. We can similarly prove that $q \prec_C^i q'$ implies $d_i(q) < d_i(q')$.

Our goal is then to prove that, for all $k > 0$, $d_i(q') < d_i(q) \implies q' \prec_C^i q$ and $d_i(q') \leq d_i(q) \implies q' \preceq_C^i q$ when $d_i(q') \leq k$. In the remainder of the proof, we will prove it by induction over the distance value $k > 0$. Note that our claim is quantified over all natural numbers i.

For the **induction basis** ($k = 1$), we have $d_i(q') \leq k$ by assumption. So, $d_i(q') = 1$. But then $Q_1^{i+1} \setminus C \models \delta(q', w[i])$. Consequently, by Rule R1'b, q' must be a minimal element of $\preceq_C^i$. Hence, we have $q' \preceq_C^i q$. Since by assumption that $d_i(q) > d_i(q') = 1$, Rule R1 supplies $Q_1^{i+1} \setminus C \not\models \delta(q, w[i])$. We can therefore choose r from Rule R1' as a minimal element of $\preceq_C^{i+1}$ to get $S^{i+1} = \{\, r' \in C \cap Q_1^{i+1} \mid r' \prec_C^{i+1} r \,\} = \emptyset$. It follows that $S^{i+1} \cup (Q_1^{i+1} \setminus C) \models \delta(q', w[i])$ (R1'a) but $S^{i+1} \cup (Q_1^{i+1} \setminus C) \not\models \delta(q, w[i])$ (R1'b). By Definition 7, we have $q' \prec_C^i q$. Hence, for $k = 1$ and $d_i(q') \leq k = 1$, it holds that $d_i(q') < d_i(q)$ implies $q' \prec_C^i q$.

When $d_i(q) = d_i(q') = k = 1$, it directly follows that $q \not\prec^i_C q'$ and $q' \not\prec^i_C q$ by Definition 7, thus also $q' \simeq^i_C q$ since $\preceq^i_C$ is a total preorder. Therefore, if $d_i(q') \leq d_i(q)$, then $q' \preceq^i_C q$, thus also $q \prec^i_C q'$ implies $d_i(q) < d_i(q')$.

For the **induction step** $k \mapsto k+1$, we have $d_i(q') = k+1$ and we want to prove $q' \prec^i_C q$ when $k+1 = d_i(q') < d_i(q)$, and prove $q' \simeq^i_C q$ when $d_i(q') = d_i(q)$ (hence $d_i(q') \leq d_i(q) \implies q' \preceq^i_C q$). We only give the high level proof idea here and refer to [19] for details.

Recall that in the induction basis, we proved that q' is a minimal element with respect to $\preceq^i_C$ when $d_i(q') \leq k$. Our key observation is that, for all $k > 0$, all elements in $\{\, p \in C \cap Q^i_1 \mid d_i(p) = k+1 \,\}$ are minimal with respect to $\preceq^i_C$ in the set $\{\, p \in C \cap Q^i_1 \mid d_i(p) > k \,\}$ (See [19] for proof details). The intuition is that our claim is equivalent to that for every two states $q, q' \in C \cap Q^i_1 \subseteq R$, $q \preceq^i_C q'$ if and only if $d_i(q) \leq d_i(q')$ (Since $\preceq^i_C$ is a preorder, we also have $q \prec^i_C q'$ iff $d_i(q) < d_i(q')$). Hence, the minimal elements in $\{\, p \in C \cap Q^i_1 \mid d_i(p) > k \,\}$ (i.e., $\{\, p \in C \cap Q^i_1 \mid d_i(p) = k+1 \,\}$) must also be the minimal elements with respect to $\preceq^i_C$, based on our induction hypothesis.

Let $S = \{p \in C \cap Q^i_1 \mid d_i(p) > k\}$. First, we know that q' is a minimal element with respect to $\preceq^i_C$ in the set S, as $d_i(q') = k+1$ by assumption. Since by assumption that $k < d_i(q') = k+1 < d_i(q)$, we know that q is also in S. Hence, $q' \preceq^i_C q$ holds.

We still need to prove that $q' \prec^i_C q$ under the assumption that $d_i(q') < d_i(q)$. By assumption that $d_i(q) > d_i(q') = k+1$, we pick a state r' that is minimal w.r.t. $\preceq^{i+1}_C$ in the set $\{p \in C \cap Q^{i+1}_1 \mid d_{i+1}(p) > k\}$ (and hence $d_{i+1}(r') = k+1$). We then prove that the selected state r' is the r-state that witnesses $q' \prec^i_C q$ for R1' of Definition 7. The observation is that, by Definition 5, we have $Q^{i+1}_1 \setminus C \cup \{\, p \in C \cap Q^{i+1}_1 \mid d_{i+1}(p) \leq d_i(q') - 1 = d_{i+1}(r') - 1 \,\} \models \delta(q', w[i])$ but $Q^{i+1}_1 \setminus C \cup \{\, p \in C \cap Q^{i+1}_1 \mid d_{i+1}(p) \leq d_{i+1}(r') - 1 \,\} \not\models \delta(q, w[i])$. By induction hypothesis, for all states $p \in C \cap Q^{i+1}_1$ such that $d_{i+1}(p) \leq d_{i+1}(r') - 1 = k$ (i.e., $d_{i+1}(p) < d_{i+1}(r')$), we also have $p \prec^i_C r'$. It then follows that by Definition 7 that $q' \prec^i_C q$ holds. Hence, $d_i(q') < d_i(q) \implies q' \prec^i_C q$.

To prove that $q \prec^i_C q'$ implies $d_i(q) < d_i(q')$, we also prove its contraposition, i.e., $d_i(q') \leq d_i(q)$ implies $q' \preceq^i_C q$ for all $i \in \mathbb{N}$. We have already shown that $d_i(q') < d_i(q)$ implies $q' \prec^i_C q$. Moreover, if $d_i(q') = d_i(q) = k+1$, then $q' \simeq^i_C q$, since both q' and q are minimal element w.r.t. $\preceq^i_C$ in the set $\{p \in C \cap Q^i_1 \mid d_i(p) > k\}$. It then follows that $q \prec^i_C q'$ implies $d_i(q) < d_i(q')$. Hence, we have completed the proof. ◀

By Lemma 9, for states $p, q \in H^i$, we have both $p \simeq^i_C q \iff d_i(p) = d_i(q)$ and $p \prec^i_C q \iff d_i(p) < d_i(q)$ hold for all $i \in \mathbb{N}$, where $H^i = C \cap Q^i_1$ if $C \subseteq R$ and $H^i = C \setminus Q^i_1$ if $C \subseteq A$. Then Corollary 10 follows immediately from Lemma 6.

▶ **Corollary 10.** *For each $w \in \Sigma^\omega$, there is a unique consistent sequence of sets of states and total preorders $\mathcal{P}_w = (Q^0_1, \{\preceq^0_C\}_{C \in \mathcal{S}})(Q^1_1, \{\preceq^1_C\}_{C \in \mathcal{S}}) \cdots$ where $Q^0_1 Q^1_1 Q^2_1 \cdots$ is the unique sequence $\mathtt{R}_w$.*

In order to lift this unique set to an UBA, we need to discharge the correctness of the sequence $Q^0_1 Q^1_1 Q^2_1 \cdots$. This is, however, a relatively simple task: for the correct sequence, the total preorders provide the same rational way of creating the same accepting runs on the tails $w[i \cdots]$ of w from the states marked as accepting in $\mathcal{A}$ by inclusion in Q^i_1, or as accepting from $\widehat{\mathcal{A}}$ by non-inclusion in Q^i_1.

To prepare such a construction, we first define an arbitrary (but fixed) order on the SCCs of $\mathcal{A}$, as well as a $\mathtt{next}$ operator for cycling through SCCs, and fix an initial SCC $C_0 \in \mathcal{S}$. Recall that $\mathcal{S}$ is the set of all SCCs in $\mathcal{A}$. Note that we assume that the graph of $\mathcal{A}$ has at least one SCC. If this is not the case, we can simply build an unambiguous safety automaton that guesses $\mathtt{R}_w$. Then, our construction of UBA is formalized below.

▶ **Definition 11.** *Let $\mathcal{A} = (\Sigma, Q, \iota, \delta, F)$ be an AWA. We define an NBA $\mathcal{B}_u = (\Sigma, Q_u, I_u, \delta_u, F_u)$ as follows.*

- *The macrostates of Q_u are tuples $(Q_1, Q_2, \{\preceq_C\}_{C\in\mathcal{S}}, S, D)$ such that*
  - *Q_1 and Q_2 partition Q, i.e., $Q_2 = Q \setminus Q_1$*
  - *for all $C \in \mathcal{S}$, if $C \subseteq R$ then $\preceq_C$ is a total preorder over $Q_1 \cap C$*
  - *for all $C \in \mathcal{S}$, if $C \subseteq A$ then $\preceq_C$ is a total preorder over $Q_2 \cap C$*
  - *$S \in \mathcal{S}$ is an SCC in the graph of $\mathcal{A}$*
  - *D is a downwards closed set w.r.t. the total preorder $\preceq_S$: if $q \in D$ then (1) $q \in Q_1 \cap S$ if $S \subseteq R$ resp. $q \in Q_2 \cap S$ if $S \subseteq A$, and (2) $q' \preceq_S q$ implies $q' \in D$,*
- *$I_u = \{\, (Q_1, Q_2, \{\preceq_C\}_{C\in\mathcal{S}}, S, D) \in Q_u \mid \iota \in Q_1, S = C_0, D = \emptyset \,\}$,*
- *Let $(Q_1, Q_2, \{\preceq_C\}_{C\in\mathcal{S}}, S, D)$ be a macrostate in Q_u and $\sigma \in \Sigma$. Then we have that $(Q'_1, Q'_2, \{\preceq'_C\}_{C\in\mathcal{S}}, S', D') \in \delta_u\big((Q_1, Q_2, \{\preceq_C\}_{C\in\mathcal{S}}, S, D), \sigma\big)$ if*
  - *$Q'_1 \models \wedge_{s\in Q_1}\delta(s, \sigma)$ and $Q'_2 \models \wedge_{s\in Q_2}\widehat{\delta}(s, \sigma)$ (local consistency)*
  - *for all $C \in \mathcal{S}$, $(Q_1, \preceq_C)$ and $(Q'_1, \preceq'_C)$ satisfy the requirements of Rule R1' (if $C \subseteq R$) resp. Rule R2' (if $C \subseteq A$)*
  - *if $D = \emptyset$, then $S' = \mathsf{next}(S)$ and $D' = Q'_1 \cap S'$ if $S' \subseteq R$ resp. $D' = Q'_2 \cap S'$ if $S' \subseteq A$,*
  - *if $D \neq \emptyset$, then $S' = S$ and D' is the smallest downwards closed set (see above) such that $D' \cup (Q'_1 \setminus S) \models \wedge_{s\in D}\delta(s, \sigma)$ if $S \subseteq R$ resp. $D' \cup (Q'_2 \setminus S) \models \wedge_{s\in D}\widehat{\delta}(s, \sigma)$ if $S \subseteq A$,*
- *$F_u = \{\, (Q_1, Q_2, \{\preceq_C\}_{C\in\mathcal{S}}, S, D) \in Q_u \mid D = \emptyset \,\}$.*

The new construction uses D as the breakpoint to ensure that the correct unique sequence $\mathtt{R}_w$ for each word w is obtained. The nondeterminism of the construction lies only in choosing Q'_1 (which entails Q'_2) and in updating the total preorders. From an accepting macrorun of $\mathcal{B}_u$ over a word w, one can actually construct an accepting run DAG $\mathcal{G}_w$ of $\mathcal{A}$ by selecting successors in the next level for each state q only the ones in the smallest downwards closed set D satisfying $\delta(q, \sigma)$. This way, all branches of $\mathcal{G}_w$ by construction will eventually get trapped in an accepting SCC, since D will become empty infinitely often. Hence, $\mathcal{L}(\mathcal{B}_u) \subseteq \mathcal{L}(\mathcal{A})$. Moreover, one can construct from the unique sequence of preorders Φ_w of a word $w \in \mathcal{L}(\mathcal{A})$ as given in Corollary 10 a unique infinite macrorun ρ of $\mathcal{B}_u$. Wrong guesses of the preorders for $\mathtt{R}_w$ will result in discontinued macroruns once a violation to R1' (or R2') has been detected. That is, there are no consistent ways to update the preorders in the next macrostate. Further, by Lemma 9, we have that $d_i(q) = d_i(q') \Leftrightarrow q \simeq^i_C q'$ and $d_i(q) < d_i(q') \Leftrightarrow q \prec^i_C q'$ for all $i \in \mathbb{N}$. So, by Definition 5 and Definition 7, one can observe that, if $D^i \neq \emptyset$, $\sup\{d_i(q) \mid q \in D^i\} = \sup\{d_{i+1}(q) \mid q \in D^{i+1}\} + 1$ (choosing $\sup \emptyset = 0$), where D^i is the D-component of macrostate $\rho[i]$ with $i \in \mathbb{N}$. Since for every nonempty D^i, $\sup\{d_i(q) \mid q \in D^i\}$ is finite and the maximal value in D^i is always decreasing, the value will eventually become 0, i.e., D always becomes empty eventually. That is, ρ must be accepting. Hence, Theorem 12 follows; See [19] for more details.

▶ **Theorem 12.** *Let $\mathcal{B}_u$ be defined as in Definition 11. Then (1) $\mathcal{L}(\mathcal{B}_u) = \mathcal{L}(\mathcal{A})$, and (2) $\mathcal{B}_u$ is unambiguous.*

▶ **Example 13.** Consider again the AWW $\mathcal{A}$ depicted in Figure 1. Recall that, in Figure 1, the macrostate $(Q, \{q, s, t\})$ has two successors over b because of the nondeterminism in developing breakpoints. We now apply Definition 11 to construct a UBA $\mathcal{B}_u$ from $\mathcal{A}$. There are three SCCs in $\mathcal{A}$, namely $C_0 = \{p\}, C_1 = \{q, s, t\}$ and $C_2 = \{r\}$. Since C_0 and C_2 both have only one state, the total preorders for them are fixed and thus ignored here. We only need to guess the preorder over C_1. Let us consider the constucted $\mathcal{B}_u$ over b^ω starting from the macrostate $m_0 = (Q, \{\}, \preceq^0_{C_1}, C_1, C_1)$ where $\preceq^0_{C_1}$ is defined as $\{s \prec^0_{C_1} q \prec^0_{C_1} t\}$.

First, recall that $\mathtt{R}_{b^\omega} = Q^\omega$. Obviously, $m_{1a} = (Q, \{\}, \{s \prec^1_{C_1} q \prec^1_{C_1} t\}, C_1, \{q, s\})$, which corresponds to $(Q, \{q, s\})$ in Figure 1, is a valid successor of m_0 over b, while $m_{1b} = (Q, \{\}, \{s \prec^1_{C_1} q \prec^1_{C_1} t\}, C_1, \{q, t\})$, which corresponds to $(Q, \{q, t\})$ in Figure 1, is not. The reason is that $\{q, t\}$ is *not* a downwards closed set with respect to $\preceq^1_{C_1}$, since we have $s \prec^1_{C_1} t$ but s is missing in the breakpoint set. One may wonder whether we can change the preorder $\preceq^1_{C_1}$ and consider the candidate successor $m_{1c} = (Q, \{\}, \{q \prec^2_{C_1} t \prec^2_{C_1} s\}, \{q, t\})$. Indeed, $\{q, t\}$ is now a downwards closed set with respect to $\preceq^2_{C_1}$. However, $(Q, \preceq^0_{C_1})$ and $(Q, \preceq^2_{C_1})$ do not satisfy the local consistency as required by Definition 7. First, we have that $Q \setminus C_1 \cup \{\} \models \delta(s, b)$. So, there do not exist r-states in $C_1 \cap Q$ that witness $q \prec^2_{C_1} s$ and $t \prec^2_{C_1} s$, as required by R1' of Definition 7. In fact, one can verify that $s \prec_{C_1} q \prec_{C_1} t$ is the only valid preorder over C_1 when the input word is b^ω. This is due to the fact that when reading b, the distance to escape C_1 is 1 from s, 2 from q, and 3 from t. Hence, m_{1c} must not be a valid successor of m_0. The accepting macrorun of $\mathcal{B}_u$ (from Definition 11) over b^ω is $(Q, \{\}, \{s \prec_{C_1} q \prec_{C_1} t\}, C_0, \{\}) \xrightarrow{b} (Q, \{\}, \{s \prec_{C_1} q \prec_{C_1} t\}, C_1, \{q, s, t\}) \xrightarrow{b} (Q, \{\}, \{s \prec_{C_1} q \prec_{C_1} t\}, C_1, \{q, s\}) \xrightarrow{b} (Q, \{\}, \{s \prec_{C_1} q \prec_{C_1} t\}, C_1, \{s\}) \xrightarrow{b} (Q, \{\}, \{s \prec_{C_1} q \prec_{C_1} t\}, C_1, \{\}) \xrightarrow{b} (Q, \{\}, \{s \prec_{C_1} q \prec_{C_1} t\}, C_2, \{\}) \xrightarrow{b} (Q, \{\}, \{s \prec_{C_1} q \prec_{C_1} t\}, C_0, \{\}) \cdots$.

4 Improvements and Complexity

When revisiting the construction in search for improvements, it seems wasteful to keep total preorders for all SCCs in the graph of $\mathcal{A}$, given that they are not interacting with each other. Can we focus on just one at a time? It proves to be possible to optimise the automaton from Definition 11 in this way, with re-establishing uniqueness proving the greatest obstacle. The resulting automaton is smaller in practice, mainly because it only keeps track of a total preorder over only one SCC.

We provide this construction only as an improvement over the principle construction from Definition 11 for two reasons. First, while this provides quite a significant advantage where there are many small SCCs rather than one big SCC, this has little effect on the worst case (which occurs when there is one SCC, cf. Theorem 16). Second, it loosens the connection that the total preorders from Definition 11 need to be the natural abstraction of the unique distance function from Definition 5.

▶ **Definition 14.** *Let $\mathcal{A} = (\Sigma, Q, \iota, \delta, F)$ be an AWA. We define an NBA $\mathcal{U} = (\Sigma, Q_u, I_u, \delta_u, F_u)$ as follows.*

- *The macrostates of Q_u are tuples $(Q_1, Q_2, \preceq_C, C, D)$ such that*
  - *Q_1 and Q_2 partition Q*
  - *C is an SCC in the graph of $\mathcal{A}$ and*
    - *if $C \subseteq R$ then $\preceq_C$ is a total preorder of $Q_1 \cap C$*
    - *if $C \subseteq A$ then $\preceq_C$ is a total preorder of $Q_2 \cap C$*
  - *let M be the set of maximal elements of the total preorder $\preceq_C$, and let $H = C \cap Q_1$ if $C \subseteq R$ resp. $H = C \cap Q_2$ if $C \subseteq A$; then $D = H$ or $D = H \setminus M$*
- *$I_u = \{\, (Q_1, Q_2, \preceq_C, C, D) \in Q_u \mid \iota \in Q_1, C = C_0, D = \emptyset \,\}$,*
- *Let $(Q_1, Q_2, \preceq_C, C, D)$ be a macrostate in Q_u and $\sigma \in \Sigma$. Then we have that $(Q'_1, Q'_2, \preceq'_{C'}, C', D') \in \delta_u\big((Q_1, Q_2, \preceq_C, C, D), \sigma\big)$ if*
  - *$Q'_1 \models \wedge_{s \in Q_1} \delta(s, \sigma)$ and $Q'_2 \models \wedge_{s \in Q_2} \widehat{\delta}(s, \sigma)$ (local consistency)*
  - *if $D = \emptyset$, then $C' = \mathsf{next}(C)$ and $D' = Q'_1 \cap C'$ if $C' \subseteq R$ resp. $D' = Q'_2 \cap C'$ if $C' \subseteq A$,*
  - *if $D \neq \emptyset$ then $C' = C$,*

  - ∗ $(Q_1, \preceq_C)$ *and* $(Q'_1, \preceq'_C)$ *must satisfy the requirements of Rule R1' (if* $C \subseteq R$*) resp. Rule R2' (if* $C \subseteq A$*) and*
  - ∗ D' *is the smallest downward closed set w.r.t.* $\preceq'_C$ *such that*[5] $D' \cup (Q'_1 \setminus C) \models \wedge_{s \in D}\delta(s,\sigma)$ *if* $C \subseteq R$ *resp.* $D' \cup (Q'_2 \setminus C) \models \wedge_{s \in D}\delta(s,\sigma)$ *if* $C \subseteq A$*,*
- $F_u = \{ (Q_1, Q_2, \preceq_C, C, D) \in Q_u \mid D = \emptyset \}$.

The nondeterminism of the construction again lies in choosing Q'_1 (which entails Q'_2) and in updating the total preorder. One can also construct from an accepting macrorun of $\mathcal{U}$ over w an accepting run DAG $\mathcal{G}_w$ of $\mathcal{A}$, using the same way as we did for Theorem 12. So, $\mathcal{L}(\mathcal{U}) \subseteq \mathcal{L}(\mathcal{A})$.

For the other direction, we first observe that the preorders of *every* accepting macrorun $(Q_1^0, Q_2^0, \preceq_0, S^0, D^0)(Q_1^1, Q_2^1, \preceq_1, S^1, D^1)\cdots$ of $\mathcal{U}$ over w can be tightly related with the distance values of states defined in $\mathbf{d}$. More precisely, let $D^{i'} = D^i = \emptyset$ with $i' < i$ being two consecutive accepting positions. Then for all $j \in (i', i]$, we have that:

1. for all $q \in D^j$ and all $q' \in C^i \cap Q_1^j$. $d_j(q) \leq d_j(q') \Leftrightarrow q \preceq_j q'$, and $d_j(q) \leq i - j$ hold,
2. for all $q \in C^i \cap Q_1^j$ and all $q' \in M^j = (C^i \cap Q_1^j) \setminus D^j$. $q \preceq_j q'$ and $d_j(q') > i - j$ hold, and
3. $m_j = \sup\{d_j(q) \mid q \in D^j\} = i - j$, using $\sup\emptyset = 0$,

where $C^i \subseteq R$ is a rejecting SCC of $\mathcal{A}$. Note that $C^j = C^i$ for all $i' < j \leq i$. The case for $C^i \subseteq A$ can be defined similarly. Let $m_j = \sup\{d_j(q) \mid q \in D^j\}$. The intuition is that all states in $M^j = (C^i \cap Q_1^j) \setminus D^j = \{ s \in C^i \cap Q_1^j \mid d_j(s) > m_i \}$ are aggregated by construction as the maximal elements w.r.t. $\preceq_j$, while $\preceq_j$ orders all states in $D^j = \{ s \in C^i \cap Q_1^j \mid d_j(s) \leq m_j \}$ exactly as in the preorders of Corollary 10. So, the correspondence between d_j and $\preceq_j$ in the three items then follows naturally. For technical reasons, if $q \in D^j$ or $q' \in (C^i \cap Q_1^j) \setminus D^j$ do not exist in above items, we say the item above still holds. See [19] for proof details.

In fact, one can construct such an accepting macrorun satisfying the three items above for $\mathcal{U}$ by simulating $\mathcal{B}_u$ as follows. If $\rho = (Q_1^0, Q_2^0, \{\preceq_C^0\}_{C \in \mathcal{S}}, S^0, D^0)(Q_1^1, Q_2^1, \{\preceq_C^1\}_{C \in \mathcal{S}}, S^1, D^1)(Q_1^2, Q_2^2, \{\preceq_C^2\}_{C \in \mathcal{S}}, S^2, D^2)\cdots$ is the accepting macrorun of $\mathcal{B}_u$ on a word w, then $\mathcal{U}$ has an accepting macrorun $\widehat{\rho} = (Q_1^0, Q_2^0, \preceq_0, S^0, D^0)(Q_1^1, Q_2^1, \preceq_1, S^1, D^1)(Q_1^2, Q_2^2, \preceq_2, S^2, D^2)\cdots$ (that differs from ρ only in preorders), such that

- if $S^i \subseteq R$, then $\preceq_i$ is a total preorder on $S^i \cap Q_1^i$ where $\preceq_i = \preceq_{S^i}^i$ if $D^i = S^i \cap Q_1^i$ and otherwise, the maximal elements M^i of $\preceq_i$ are $(S^i \cap Q_1^i) \setminus D^i$, and the restriction of $\preceq_i$ to $D^i \times D^i$ agrees with the restriction of $\preceq_{S^i}^i$ to $D^i \times D^i$, and
- similarly, if $S^i \subseteq A$, then $\preceq_i$ is a total preorder on $S^i \cap Q_2^i$ where $\preceq_i = \preceq_{S^i}^i$ if $D^i = S^i \cap Q_2^i$ and otherwise, the maximal elements M^i of $\preceq_i$ are $(S^i \cap Q_2^i) \setminus D^i$, and the restriction of $\preceq_i$ to $D^i \times D^i$ agrees with the restriction of $\preceq_{S^i}^i$ to $D^i \times D^i$.

It is easy to verify that $\widehat{\rho}$ satisfies all local constraints for Rule R1' resp. R2'. Hence, $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B}_u) \subseteq \mathcal{L}(\mathcal{U})$, thus also $\mathcal{L}(\mathcal{U}) = \mathcal{L}(\mathcal{A})$.

One can show that $\widehat{\rho}$ is the sole accepting macrorun of $\mathcal{U}$ over w by the following facts. (i) There is only a single initial macrostate that fits $\mathtt{R}_w$, and when we take a transition from an accepting macrostate (including the first), the next SCC is deterministically selected; (ii) Moreover, all relevant states from this SCC are in the D^i component and $m_i = \sup\{d_i(q) \mid q \in D^i\}$ is the distance to the next breakpoint (by Item (3) above), and thus the $\preceq_i$ and D^i up to it. With a simple inductive argument we can thus conclude that $\widehat{\rho}$ is the only such accepting macrorun. Then, Theorem 15 follows.

---

[5] Note that this is a deterministic assignment that does not necessarily lead to a set D' that covers all of $\preceq'_C$ or all of $\preceq'_C$ except for the maximal elements; if it does not, then this transition is disallowed

▶ **Theorem 15.** *Let $\mathcal{U}$ be defined as in Definition 14. Then (1) $\mathcal{L}(\mathcal{U}) = \mathcal{L}(\mathcal{A})$ and (2) $\mathcal{U}$ is unambiguous.*

We now turn to the complexity of our constructions. Let $\mathsf{tpo}(n)$ denote the number of total preorders over a set with n states. By [3], $\mathsf{tpo}(n) \approx \frac{n!}{2(\ln 2)^{n+1}}$, so that we get $\lim_{n\to\infty} \frac{\sqrt[n]{\mathsf{tpo}(n)}}{n} = \lim_{n\to\infty} \frac{\sqrt[n]{n!}}{n} \cdot \frac{1}{\sqrt[n]{2\ln 2}} \cdot \frac{1}{\ln 2} = \frac{1}{e} \cdot 1 \cdot \frac{1}{\ln 2} = \frac{1}{e \ln 2} \approx 0.53$. Hence, $\mathsf{tpo}(n) \approx (0.53n)^n$, which is a better bound than the best known bound $(0.76n)^n$ for Büchi disambiguation [16] and complementation [24].

▶ **Theorem 16.** *If $\mathcal{A}$ has n states, then the numbers of states of $\mathcal{U}$ and $\mathcal{B}_u$ are $\mathcal{O}\big(\mathsf{tpo}(n)\big)$ and $\mathcal{O}\big(n \cdot \mathsf{tpo}(n)\big)$, respectively.*

Proof. For both automata, the worst case occurs when all states are in the same SCC C, say $C = R$. Starting with $\mathcal{U}$, each macrostate is a tuple $(Q_1, C \setminus Q_1, \preceq, C, D)$. There are four possibilities for the tuple, namely $C = Q_1 = D$, $C = Q_1 \supsetneq D$, $C \supsetneq Q_1 = D$, and $C \supsetneq Q_1 \supsetneq D$. For each of these four cases, we can produce an injection from the tuple (macrostate) onto a total preorder $\preceq'$ over C, so that we have at most $4 \cdot \mathsf{tpo}(n)$ macrostates. For $C = Q_1 = D$, for example, we can keep the $\preceq$ over C, i.e., we set $\preceq' = \preceq$. When there is strict inclusion, i.e., $C \supsetneq Q_1$, we extend the $\preceq$ on Q_1 to a total preorder $\preceq'$ over C by adding the states in $C \setminus Q_1$ resp. $Q_1 \setminus D$ as minimal resp. maximal elements (with their separate equivalence class). For each of the four cases, the respective mapping is injective.

As this covers all macrostates of $\mathcal{U}$, $\mathcal{U}$ has at most $4 \cdot \mathsf{tpo}(n)$ macrostates.

For $\mathcal{B}_u$, there are $\mathcal{O}(n)$ possible choices for D, since the maximal element in D with respect to the preorder $\preceq$ has at most n possibilities. This leads to $\mathcal{O}(n \cdot \mathsf{tpo}(n))$ macrostates. ◀

5 Discussion

We have given the *first* direct translation from AWAs to UBAs. The complexity of our translation is even *smaller* than that of the best known disambiguation algorithm for NBAs [16] (broadly $(0.53n)^n$ vs. $(0.76n)^n$). We can further optimise the construction of $\mathcal{U}$ slightly by moving to *transition-based* acceptance conditions. That is, an ω-word is now accepted by $\mathcal{U}$ if one of its corresponding runs visits accepting transitions for infinitely many times. Essentially, where $(Q_1', Q_2', \preceq', C, \emptyset) \in \delta_u\big((Q_1, Q_2, \preceq, C, D), \sigma\big)$, $(Q_1', Q_2', \preceq', C, \emptyset)$ would be replaced by $\delta_u\big((Q_1, Q_2, \equiv, C, \emptyset), \sigma\big)$. ($\equiv$ identifies all states it compares; it is the only total preorder acceptable for $D = \emptyset$.)

This is done recursively, until the only macrostates with $D = \emptyset$ left are those with $Q_1 \cap R = \emptyset = Q_2 \cap A$ and (arbitrarily) $C = C_0$. Note that the initial macrostate has to be changed for this, too.

Removing most macrostates with $D = \emptyset$, this reduces the statespace slightly. It is also the automaton obtained by de-generalising the standard LTL to transition-based unambiguous generalized Büchi automaton construction. We can also "re-generalise": every singleton SCC can be removed from the round-robin at the cost of including an individual Büchi condition that accepts when the state s is not in Q_1 or Q_2, respectively, or if $Q_1 \models \delta(s, \sigma)$ or $Q_2 \models \widehat{\delta}(s, \sigma)$, respectively, holds. If all components are singleton, we obtain the standard constuction for AVAs / LTL since the preorders of our construction given in Section 4 can be omitted. This way, the D set in a macrostate degenerates to a purely breakpoint construction. Then, the improved complexity for AVAs matches the current known bounds $n2^n$ for the LTL-to-UBA construction [14, 26].

References

1 Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

2 Christel Baier, Stefan Kiefer, Joachim Klein, Sascha Klüppelholz, David Müller, and James Worrell. Markov chains and unambiguous Büchi automata. In Swarat Chaudhuri and Azadeh Farzan, editors, *Computer Aided Verification – 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part I*, volume 9779 of *Lecture Notes in Computer Science*, pages 23–42. Springer, 2016. doi:10.1007/978-3-319-41528-4_2.

3 J.P. Barthelemy. An asymptotic equivalent for the number of total preorders on a finite set. *Discrete Mathematics*, 29(3):311–313, 1980. doi:10.1016/0012-365X(80)90159-4.

4 Michael Benedikt, Rastislav Lenhardt, and James Worrell. LTL model checking of interval markov chains. In Nir Piterman and Scott A. Smolka, editors, *Tools and Algorithms for the Construction and Analysis of Systems – 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings*, volume 7795 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2013. doi:10.1007/978-3-642-36742-7_3.

5 Frantisek Blahoudek, Juraj Major, and Jan Strejcek. LTL to smaller self-loop alternating automata and back. In Robert M. Hierons and Mohamed Mosbah, editors, *Theoretical Aspects of Computing – ICTAC 2019 – 16th International Colloquium, Hammamet, Tunisia, October 31 – November 4, 2019, Proceedings*, volume 11884 of *Lecture Notes in Computer Science*, pages 152–171. Springer, 2019. doi:10.1007/978-3-030-32505-3_10.

6 Udi Boker, Orna Kupferman, and Adin Rosenberg. Alternation removal in Büchi automata. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II*, volume 6199 of *Lecture Notes in Computer Science*, pages 76–87. Springer, 2010. doi:10.1007/978-3-642-14162-1_7.

7 Udi Boker, Karoliina Lehtinen, and Salomon Sickert. On the translation of automata to linear temporal logic. In Patricia Bouyer and Lutz Schröder, editors, *Foundations of Software Science and Computation Structures – 25th International Conference, FOSSACS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings*, volume 13242 of *Lecture Notes in Computer Science*, pages 140–160. Springer, 2022. doi:10.1007/978-3-030-99253-8_8.

8 J. Richard Büchi. On a decision method in restricted second order arithmetic. In *Proc. Int. Congress on Logic, Method, and Philosophy of Science. 1960*, pages 1–12. Stanford University Press, 1962.

9 Doron Bustan, Sasha Rubin, and Moshe Y. Vardi. Verifying omega-regular properties of Markov chains. In Rajeev Alur and Doron A. Peled, editors, *Computer Aided Verification, 16th International Conference, CAV 2004, Boston, MA, USA, July 13-17, 2004, Proceedings*, volume 3114 of *Lecture Notes in Computer Science*, pages 189–201. Springer, 2004. doi:10.1007/978-3-540-27813-9_15.

10 Olivier Carton and Max Michel. Unambiguous Büchi automata. *Theor. Comput. Sci.*, 297(1-3):37–81, 2003. doi:10.1016/S0304-3975(02)00618-7.

11 Paul Gastin and Denis Oddoux. Fast LTL to Büchi automata translation. In Gérard Berry, Hubert Comon, and Alain Finkel, editors, *Computer Aided Verification, 13th International Conference, CAV 2001, Paris, France, July 18-22, 2001, Proceedings*, volume 2102 of *Lecture Notes in Computer Science*, pages 53–65. Springer, 2001. doi:10.1007/3-540-44585-4_6.

12 Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 854–860. IJCAI/AAAI, 2013. URL: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6997.

13 Gerard J. Holzmann. The model checker SPIN. *IEEE Trans. Software Eng.*, 23(5):279–295, 1997. doi:10.1109/32.588521.

14 Simon Jantsch, David Müller, Christel Baier, and Joachim Klein. From LTL to unambiguous Büchi automata via disambiguation of alternating automata. *Formal Methods Syst. Des.*, 58(1-2):42–82, 2021. doi:10.1007/s10703-021-00379-z.

15 Detlef Kähler and Thomas Wilke. Complementation, disambiguation, and determinization of Büchi automata unified. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, volume 5125 of *Lecture Notes in Computer Science*, pages 724–735. Springer, 2008. doi:10.1007/978-3-540-70575-8_59.

16 Hrishikesh Karmarkar, Manas Joglekar, and Supratik Chakraborty. Improved upper and lower bounds for Büchi disambiguation. In Dang Van Hung and Mizuhito Ogawa, editors, *Automated Technology for Verification and Analysis – 11th International Symposium, ATVA 2013, Hanoi, Vietnam, October 15-18, 2013. Proceedings*, volume 8172 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2013. doi:10.1007/978-3-319-02444-8_5.

17 Orna Kupferman and Moshe Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. Comput. Log.*, 2(3):408–429, 2001. doi:10.1145/377978.377993.

18 Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification – 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011. doi:10.1007/978-3-642-22110-1_47.

19 Yong Li, Sven Schewe, and Moshe Y. Vardi. Singly exponential translation of alternating weak büchi automata to unambiguous büchi automata. *CoRR*, abs/2305.09966, 2023. doi:10.48550/arXiv.2305.09966.

20 Satoru Miyano and Takeshi Hayashi. Alternating finite automata on omega-words. *Theor. Comput. Sci.*, 32:321–330, 1984. doi:10.1016/0304-3975(84)90049-5.

21 David E. Muller, Ahmed Saoudi, and Paul E. Schupp. Alternating automata, the weak monadic theory of trees and its complexity. *Theor. Comput. Sci.*, 97(2):233–244, 1992. doi:10.1016/0304-3975(92)90076-R.

22 David E. Muller and Paul E. Schupp. Alternating automata on infinite objects, determinacy and rabin's theorem. In Maurice Nivat and Dominique Perrin, editors, *Automata on Infinite Words, Ecole de Printemps d'Informatique Théorique, Le Mont Dore, France, May 14-18, 1984*, volume 192 of *Lecture Notes in Computer Science*, pages 100–107. Springer, 1984. doi:10.1007/3-540-15641-0_27.

23 Gareth Scott Rohde. *Alternating automata and the temporal logic of ordinals*. PhD thesis, University of Illinois at Urbana-Champaign, 1997.

24 Sven Schewe. Büchi complementation made tight. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, volume 3 of *LIPIcs*, pages 661–672. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany, 2009. doi:10.4230/LIPIcs.STACS.2009.1854.

25 Moshe Y. Vardi. The rise and fall of LTL. In Giovanna D'Agostino and Salvatore La Torre, editors, *Proceedings of Second International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2011, Minori, Italy, 15-17th June 2011*, 2011. invited talk. URL: https://www.cs.rice.edu/~vardi/papers/gandalf11-myv.pdf.

26 Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *Proceedings of the Symposium on Logic in Computer Science (LICS '86), Cambridge, Massachusetts, USA, June 16-18, 1986*, pages 332–344. IEEE Computer Society, 1986.