Mediating for Reduction (On Minimizing Alternating Büchi Automata)[☆]

Parosh Aziz Abdulla

Dept. Information Technology, Uppsala University, Box 337, 751 05 Uppsala, Sweden

Yu-Fang Chen

Institute of Information Science, Academia Sinica, 128 Academia Road, Section 2,
Nankang, Taipei 115, Taiwan

Lukáš Holík

FIT, Brno University of Technology, IT4Innovations Centre of Excellence, Božetěchova 2,
612 66 Brno, Czech Republic

Tomáš Vojnar

FIT, Brno University of Technology, IT4Innovations Centre of Excellence, Božetěchova 2,
612 66 Brno, Czech Republic

Abstract

We propose a new approach for minimizing alternating Büchi automata (ABA). The approach is based on the mediated equivalence on states of an ABA, which is the maximal equivalence contained in the mediated preorder. Two states p and q are related by the mediated preorder if there is a mediator (mediating state) which forward simulates p and backward simulates q. Under further conditions, letting a computation on some word jump from q to p preserves the language as the automaton can anyway already accept the word without jumps by runs through the mediator. We further show how the mediated equivalence can be computed efficiently. Finally, we show that, compared to the standard forward simulation equivalence, the mediated equivalence can yield much larger reductions when applied within the process of complementing Büchi automata where ABA are used as an intermediate model.

Keywords: alternating Büchi automata, simulation, reduction, quotient

[☆] This is an extended version of the paper [1]. This work was supported by the Czech Science Foundation (projects 13-37876P, 14-11384S), the EU/Czech IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070, the internal project of Brno University of Technology FIT-S-14-2486, the National Science Council of Taiwan project no. NSC-102-2221-E-001-018, and the Uppsala Programming for Multicore Architectures Research Center (UPMARC).

Email addresses: `parosh@it.uu.se` (Parosh Aziz Abdulla), `yfc@iis.sinica.edu.tw` (Yu-Fang Chen), `holik@fit.vutbr.cz` (Lukáš Holík), `vojnar@fit.vutbr.cz` (Tomáš Vojnar)

1. Introduction

Alternating Büchi automata (ABA) are succinct state-machine representations of ω-regular languages (regular sets of infinite sequences). They are widely used in the area of formal specification and verification of non-terminating systems. One of the most prominent examples of the use of ABA is the complementation of nondeterministic Büchi automata [2]. The complementation is an essential step of the automata-theoretic approach to model checking when the specification is given as a positive Büchi automaton [3] and also of learning-based model checking for liveness properties [4]. ABA also play an important role as an intermediate data structure for translating a linear temporal logic (LTL) specification to a nondeterministic Büchi automaton (NBA) [5].

However, due to the compactness of ABA[1], the algorithms that work on them are usually of high complexity. For example, both the complementation and the LTL translation algorithm transforms an intermediate ABA to an equivalent NBA. The transformation is exponential in the size of the input ABA. Hence, one may prefer to reduce the size of the ABA (with some faster algorithm) before giving it to the exponential procedure.

In the study of Fritz and Wilke [6], simulation-based minimization is proven as a very effective tool for reducing the size of ABA. However, they considered only *forward* simulation relations. Inspired by previous work [7], we believe that *backward* simulation can be used for reducing the size of ABA as well. Unfortunately, as will be explained in Section 3, the quotient wrt. *backward* simulation (i.e., the automaton that arises by collapsing backward simulation equivalent states) may have different language.

In this paper, we develop an approach that uses backward simulation for simplifying ABA indirectly. Instead of looking for a suitable fragment of backward simulation that can be used to reduce the number of states of an ABA, we combine backward and forward simulation to form an even coarser relation called the *mediated preorder* that can be used for minimization. The efficiency of minimizing ABA using the *mediated preorder* is evaluated on a large set of experiments. In the experiments, we apply different simulation-based minimization approaches to improve the LTL to Büchi automata translation algorithm and complementation algorithm of nondeterministic Büchi automata. The experimental results show that the minimization using mediated preorder outperforms the minimization using forward simulation. For example, for 100 randomly generated automata of alphabet size 2, forward minimization on average reduced ABA from 11.8 states and 39.8 transitions to 7 states and 26.9 transitions while mediated minization reduced the numbers to 5.66 states and 20.49 transitions, on average.

[1] ABA are exponentially more succinct than nondeterministic Büchi automata.

Outline. The next section contains basic definitions of trees and of alternating Büchi automata. Section 3 presents notions of simulations on ABA and their important properties. In Section 4, we define the notion of mediated equivalence and prove that the quotient ABA wrt. the mediated equivalence has the same language as the original ABA. Algorithms for computing the proposed relations are given in Section 5. The experimental evaluation of the presented technique for reducing size of ABA by computing the quotint wrt. mediated equivalence is discussed in Section 6 and Section 7 concludes the paper.

2. Basic Definitions

Given a finite set X, we use X^* to denote the set of all finite words over X and X^ω for the set of all infinite words over X. The empty word is denoted by ϵ and $X^+ = X^* \setminus \{\epsilon\}$. The concatenation of a finite word $u \in X^*$ and a finite or infinite word $v \in X^* \cup X^\omega$ is denoted by uv. For a word $w \in X^* \cup X^\omega$, $|w|$ is the length of w ($|w| = \infty$ if $w \in X^\omega$), w_i is the ith letter of w and w^i the ith prefix of w (the word u with $w = uv$ and $|u| = i$). Note that $w^0 = \epsilon$. The concatenation of a finite word u and a set $S \subseteq X^* \cup X^\omega$ is defined as $uS = \{uv \mid v \in S\}$.

An *alternating Büchi automaton* is a tuple $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$ where Σ is a finite alphabet, Q is a finite set of states, $\iota \in Q$ is an initial state, $\alpha \subseteq Q$ is a set of accepting states, and $\delta : Q \times \Sigma \to 2^{2^Q}$ is a total transition function. A *transition* of $\mathcal{A}$ is of the form $q \xrightarrow{a} P$ where $P \in \delta(q, a)$.

A *tree T over Q* is a subset of Q^+ that contains all nonempty prefixes of each of its elements (i.e., $T \cup \{\epsilon\}$ is prefix-closed). Furthermore, we require that T contains exactly one $r \in Q$, the *root of T*, denoted *root*(T). We call the elements of Q^+ *paths.* For a path πq, we use *leaf*(πq) to denote its last element q. We define the set *branches*$(T) \subseteq Q^+ \cup Q^\omega$ such that $\pi \in$ *branches*(T) iff T contains all prefixes of π and π is not a proper prefix of any path in T. In other words, a *branch of T* is either a maximal path of T, or it is a word from Q^ω such that T contains all its nonempty prefixes. We use $succ_T(\pi) = \{r \mid \pi r \in T\}$ to denote the set of successors of a path π in T, and *height*(T) to denote the length of the longest branch of T. A tree U over Q is a *prefix of T* iff $U \subseteq T$ and for every $\pi \in U$, $succ_U(\pi) = succ_T(\pi)$ or $succ_U(\pi) = \emptyset$. The *suffix of T* defined by a path πq is the tree $T(\pi q) = \{q\psi \mid \pi q \psi \in T\}$.

Given a word $w \in \Sigma^\omega$, a tree T over Q is a *run of $\mathcal{A}$ on w*, if for every $\pi \in T$, *leaf*$(\pi) \xrightarrow{w_{|\pi|}} succ_T(\pi)$ is a transition of $\mathcal{A}$. Finite prefixes of T are called *partial runs of $\mathcal{A}$ on w*. A run T of $\mathcal{A}$ on w is *accepting* iff every infinite branch of T contains infinitely many accepting states. A word w is *accepted* by $\mathcal{A}$ from a state $q \in Q$ iff there exists an accepting run T of $\mathcal{A}$ on w with *root*$(T) = q$. The *language of a state $q \in Q$ in $\mathcal{A}$*, denoted $\mathcal{L}_\mathcal{A}(q)$, is the set of all words accepted by $\mathcal{A}$ from q. Then $\mathcal{L}(\mathcal{A}) = \mathcal{L}_\mathcal{A}(\iota)$ is the *language of $\mathcal{A}$*. For simplicity of presentation, we assume in the rest of the paper that δ never allows a transition of the form $p \xrightarrow{a} \emptyset$. This means that no run can contain a finite branch. Any automaton can be easily transformed into one without such transitions by adding a new accepting state q with $\delta(q, a) = \{\{q\}\}$ for every $a \in \Sigma$ and replacing every transition $p \xrightarrow{a} \emptyset$ by $p \xrightarrow{a} \{q\}$.

We note that for technical reasons, we use a simpler definition of a tree and a run of an alternating automaton than the usual one (e.g., [2]). A tree is usually defined as a prefix-closed subset of $\mathbb{N}^*$ and a run is then a map r that assigns a state to every element (node) of a tree. This definition allows nodes with more than one immediate successor labelled by the same state and successors of a node are ordered. However, order as well as the number of occurrences of a state as a successor of a parent state have no relevance for the semantics of an ABA. From this point of view, it is more convenient to define runs simply as unordered trees.

3. Simulation Relations

In this section, we give the definitions of forward and backward simulation over ABA and discuss some of their properties. The notion of backward simulation is inspired by a similar tree automata notion studied in [7, 8]—namely, the upward simulation parametrised by a downward simulation (the connection between tree automata and ABA follows from the fact that the runs of ABA are in fact trees).

For the rest of the section, we fix an ABA $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$. We define relations $\preceq_\alpha$ and $\preceq_\iota$ on Q s.t. $q \preceq_\alpha r$ iff $q \in \alpha \implies r \in \alpha$ and $q \preceq_\iota r$ iff $q = \iota \implies r = \iota$. For a binary relation $\preceq$ on a set X, the relation $\preceq^{\forall\exists}$ on subsets of X is defined as $Y \preceq^{\forall\exists} Z$ iff $\forall z \in Z.\ \exists y \in Y.\ y \preceq z$, i.e., iff the upward closure of Z wrt. $\preceq$ is a subset of the upward closure of Y wrt. $\preceq$.

Forward Simulation. A *forward simulation* on $\mathcal{A}$ is a relation $\preceq_F \subseteq Q \times Q$ such that $p \preceq_F r$ implies that (i) $p \preceq_\alpha r$ and (ii) for all $p \xrightarrow{a} P$, there exists a $r \xrightarrow{a} R$ such that $P \preceq_F^{\forall\exists} R$.

For the basic properties of forward simulation, we rely on the work [9] by Gurumurthy et al. In particular, (i) there exists a unique maximal forward simulation $\preceq_F$ on $\mathcal{A}$ called *forward simulation preorder* and it is reflexive and transitive, (ii) for any $q, r \in Q$ such that $q \preceq_F r$, it holds that $\mathcal{L}_\mathcal{A}(q) \subseteq \mathcal{L}_\mathcal{A}(r)$, and (iii) the quotient of $\mathcal{A}$ wrt. $\preceq_F \cap \preceq_F^{-1}$ has the same language as $\mathcal{A}$.

Backward Simulation. Let $\preceq_F$ be a forward simulation on $\mathcal{A}$. A *backward simulation* on $\mathcal{A}$ parametrised by $\preceq_F$ is a relation $\preceq_B \subseteq Q \times Q$ such that $p \preceq_B r$ implies that (i) $p \preceq_\iota r$, (ii) $p \preceq_\alpha r$, and (iii) for all $q \xrightarrow{a} P \cup \{p\}, p \notin P$, there exists a $s \xrightarrow{a} R \cup \{r\}, r \notin R$ such that $q \preceq_B s$ and $P \preceq_F^{\forall\exists} R$. The lemma below describes basic properties of backward simulation.

Lemma 3.1. *For any reflexive and transitive forward simulation $\preceq_F$ on $\mathcal{A}$, there exists a unique maximal backward simulation $\preceq_B$ on $\mathcal{A}$ parametrised by $\preceq_F$ that is reflexive and transitive.*

Proof. Union: Given two backward simulations $\preceq_B^1$ and $\preceq_B^2$ parametrised by $\preceq_F$, we want to prove that $\preceq_B = \preceq_B^1 \cup \preceq_B^2$ is also a backward simulation parametrised by $\preceq_F$. Let $p \preceq_B r$ for some $p, r \in Q$, then either $p \preceq_B^1 r$ or $p \preceq_B^2 r$. Assume without loss of generality that $p \preceq_B^1 r$. Then, from the definition of backward simulation, whenever $p' \xrightarrow{a} P \cup \{p\}, p \notin P$, then there is a rule $r' \xrightarrow{a} R \cup \{r\}, r \notin R$, $p' \preceq_B^1 r'$, and $P \preceq_F^{\forall\exists} R$. As $\preceq_B^1 \subseteq \preceq_B$ gives $p' \preceq_B r'$, $\preceq_B$ fulfills the definition of backward simulation parametrised by $\preceq_F$.

Reflexive closure: It can be seen from the definition that the identity is a backward simulation parametrised by $\preceq_F$ for any forward simulation $\preceq_F$. Therefore, from the closure under union, the union of the identity and any backward simulation parametrised by $\preceq_F$ is a backward simulation parametrised by $\preceq_F$.

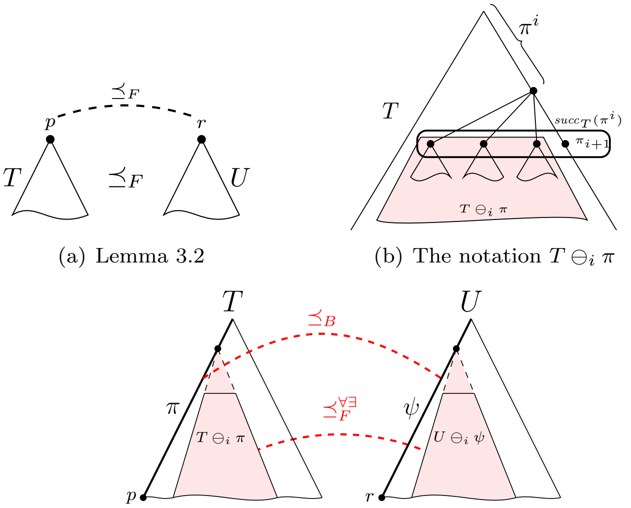
Transitive closure: Let $\preceq_B$ be a backward simulation parametrised by $\preceq_F$ and let $\preceq_B^T$ be its transitive closure. Let $p^1 \preceq_B^T p^m$ and $r^1 \xrightarrow{a} P^1 \cup \{p^1\}, p^1 \notin P^1$. We have that $p^1 \preceq_\alpha p^m$ since $\preceq_B$ is a subset of $\preceq_\alpha$ and $\preceq_\alpha$ is transitive. From $p^1 \preceq_B^T p^m$, we have that there are states $p^1, \ldots, p^m$ such that $p^1 \preceq_B p^2 \preceq_B \cdots \preceq_B p^m$. Therefore, there are also rules $r^2 \xrightarrow{a} P^2 \cup \{p^2\}, \ldots, r^m \xrightarrow{a} P^m \cup \{p^m\}$ with $p^2 \notin P^2, \ldots, p^m \notin P^m$, $r^1 \preceq_B \cdots \preceq_B r^m$, and $P^1 \preceq_F^{\forall\exists} P^2 \preceq_F^{\forall\exists} \cdots \preceq_F^{\forall\exists} P^m$. By definition of $\preceq_B^T$, we have $r^1 \preceq_B^T r^m$. By definition of $\preceq_F^{\forall\exists}$ and since $\preceq_F$ is transitive, $\preceq_F^{\forall\exists}$ is transitive. Hence we have $P^1 \preceq_F^{\forall\exists} P^m$. Therefore, $\preceq_B^T$ fulfills the definition of a backward simulation parametrised by $\preceq_F$. □

By Lemma 3.1, for any reflexive and transitive forward simulation $\preceq_F$, there is a unique maximal backward simulation parametrised by $\preceq_F$ and it is a preorder. We call it the *backward simulation preorder* on $\mathcal{A}$ parametrised by $\preceq_F$. Our backward simulation is an analogy of upward simulation for tree automata. Similarly as upward simulation, backward simulation cannot be directly used for computing the quotient (below we give an example of an automaton such that its language differs from the language of its quotient wrt. backward simulation). However, in Section 4.1, we show that backward simulation can be combined with forward simulation into a mediated equivalence (in the same way as tree automata upward simulation can be combined with downward simulation) such that the quotient wrt. to this new relation preserves the language.

Example 1 (the language of the quotient wrt. backward simulation may differ)**.** *Let* $\mathcal{A} = (\{a, b\}, \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}, s_0, \delta, \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\})$ *be an ABA where*

$$\begin{array}{llll} s_0 \xrightarrow{a} \{s_4\}, & s_1 \xrightarrow{b} \{s_2, s_5\}, & s_2 \xrightarrow{b} \{s_2, s_3\}, & s_5 \xrightarrow{b} \{s_0\}, \\ s_0 \xrightarrow{a} \{s_1\}, & s_1 \xrightarrow{b} \{s_1, s_3\}, & s_3 \xrightarrow{a} \{s_0\}, & s_6 \xrightarrow{a} \{s_0\}, \\ s_0 \xrightarrow{b} \{s_0\}, & & s_4 \xrightarrow{b} \{s_4, s_6\} & \end{array}$$

are transitions of $\mathcal{A}$. The maximal forward simulation relation $\preceq_F$ in $\mathcal{A}$ is the smallest preorder such that $s_3 \equiv_F s_6$ and $s_4 \equiv_F s_2 \preceq_F s_1 \preceq_F s_5 \preceq_F s_0$. The maximal backward simulation relation $\preceq_B$ parametrised with $\preceq_F$ is the smallest preorder such that $s_1 \equiv_B s_4$ and $s_2 \succeq_B s_5 \equiv_B s_6 \preceq_B s_3$. If we collapse states of $\mathcal{A}$ wrt. $\preceq_B$ (i.e., the two sets of states $\{s_1, s_4\}, \{s_5, s_6\}$ are collapsed), we will get the ABA $\mathcal{A}' = (\{a, b\}, \{s_0, \{s_1, s_4\}, s_2, s_3, \{s_5, s_6\}\}, s_0, \delta, \{s_0, \{s_1, s_4\},$



(a) Lemma 3.2

(b) The notation $T \ominus_i \pi$

(c) Lemma 3.3

Figure 1: An illustration of the lemmas

$s_2, s_3, \{s_5, s_6\}\})$ *with transitions:*

$$\begin{array}{llll}
s_0 \xrightarrow{a} \{\{s_1, s_4\}\}, & \{s_1, s_4\} \xrightarrow{b} \{s_2, \{s_5, s_6\}\}, & s_2 \xrightarrow{b} \{s_2, s_3\}, & \{s_5, s_6\} \xrightarrow{b} \{s_0\}, \\
s_0 \xrightarrow{b} \{s_0\}, & \{s_1, s_4\} \xrightarrow{b} \{\{s_1, s_4\}, s_3\}, & s_3 \xrightarrow{a} \{s_0\}, & \\
 & & \{s_1, s_4\} \xrightarrow{b} \{\{s_1, s_4\}, \{s_5, s_6\}\}. &
\end{array}$$

Note that $\mathcal{A}'$ accepts the word ab^ω, but $\mathcal{A}$ does not. □

3.1. Runs and Simulations

We now formulate connections between simulations and runs of ABA that are fundamental for our further reasoning. Let $\preceq_F$ and $\preceq_B$ be forward and backward simulations on $\mathcal{A}$, which are both reflexive and transitive. For every $x \in \{B, F, \alpha\}$, we extend the relation $\preceq_x$ to $Q^+ \times Q^+$ such that for $\pi, \psi \in Q^+$, $\pi \preceq_x \psi$ iff $|\pi| = |\psi|$ and for all $1 \leq i \leq |\pi|$, $\pi_i \preceq_x \psi_i$. We say that ψ forward simulates π, ψ backward simulates π, or ψ is more accepting than π when $\pi \preceq_F \psi$, $\pi \preceq_B \psi$, or $\pi \preceq_\alpha \psi$, respectively. This notation is further extended to trees. For trees T, U over Q and for $x \in \{\alpha, F\}$, we write $T \preceq_x U$ if $\mathit{branches}(T) \preceq_x^{\forall\exists} \mathit{branches}(U)$. Similarly, we say that U forward simulates T, or U is more accepting than T when $T \preceq_F U$, or $T \preceq_\alpha U$, respectively. Note that $\preceq_x$ is reflexive and transitive for all the variants of $x \in \{F, B, \alpha\}$ defined over states, paths, or trees (this follows from the assumption that the original relations $\preceq_F$ and $\preceq_B$ on states are reflexive and transitive). Moreover, $\preceq_B \subseteq \preceq_\alpha$, $\preceq_B \subseteq \preceq_\iota$, and $\preceq_F \subseteq \preceq_\alpha$.

Lemma 3.2. *For any $p, r \in Q$ with $p \preceq_F r$ and a partial run T of $\mathcal{A}$ on $w \in \Sigma^\omega$ with the root p, there is a partial run U of $\mathcal{A}$ on w with the root r such that $T \preceq_F U$.*

Proof. We prove the lemma by induction on $height(T)$. In the base case when $T = \{p\}$, it is sufficient to take $U = \{r\}$. Suppose now that the lemma holds for every word u and for every partial run V of $\mathcal{A}$ on u such that $height(V) < height(T)$. From $p \preceq_F r$, there is a transition $r \xrightarrow{w_1} R$ of $\mathcal{A}$ where $succ_T(p) \preceq_F^{\forall\exists} R$. Observe that $T = \{p\} \cup \bigcup_{p' \in succ_T(p)} pT(p')$ where for each $p' \in succ_T(p)$, $T(p')$ is a partial run of $\mathcal{A}$ with the root p' on the word v such that $w = w_1 v$. Notice that $height(T(p')) < height(T)$. The induction hypothesis now can be applied to every triple $p' \in succ_T(p), r' \in R, T(p')$ with $p' \preceq_F r'$. It gives us a partial run $U_{r'}$ of $\mathcal{A}$ on v with $root(U_{r'}) = r'$ such that $T(p') \preceq_F U_{r'}$. The run U with the required properties is then constructed by plugging the runs $U_{r'}, r' \in R$, to r, i.e., $U = \{r\} \cup \bigcup_{r' \in R} rU_{r'}$. □

We will need to inspect the connection between runs and backward simulation in a relatively detailed way. For this, we introduce the following notation. Given a tree T over Q, $\pi \in T$, and $1 \leq i \leq |\pi|$, the set $T \ominus_i \pi$ is the union of branches of suffix trees $T(\pi^i q), q \in succ_T(\pi^i)$, with the branches of the suffix tree $T(\pi^{i+1})$ excluded. Formally, for $1 \leq i < |\pi|$, let $Q^i = succ_T(\pi^i) \setminus \{\pi_{i+1}\}$ be the set of all successors of π^i in T without the successor continuing in π. Then $T \ominus_i \pi = \bigcup_{q \in Q^i} branches(T(\pi^i q))$. For $i = |\pi|$, $T \ominus_i \pi = \emptyset$.

Lemma 3.3. *For any $p, r \in Q$ with $p \preceq_B r$, a partial run T of $\mathcal{A}$ on $w \in \Sigma^\omega$ and $\pi \in branches(T)$ with $leaf(\pi) = p$, there is a partial run U of $\mathcal{A}$ on w and $\psi \in branches(U)$ with $leaf(\psi) = r$ such that $\pi \preceq_B \psi$, and for all $1 \leq i \leq |\pi|$, $T \ominus_i \pi \preceq_F^{\forall\exists} U \ominus_i \psi$.*

Proof. We will prove the lemma by induction on the length of π. In the base case, when $\pi = p$ and $T = \{p\}$, it is sufficient to take $U = \{r\}$ and $\psi = r$. Suppose now that $\pi \neq p$ and that the lemma holds for every partial run T' of $\mathcal{A}$ on w, states $p', r' \in Q$ such that $p' \preceq_B r'$, and every $\pi' \in branches(T')$ with $leaf(\pi') = p'$ and $|\pi'| < |\pi|$.

For the induction step, let $\pi = \pi' p$ and let $succ_T(\pi') = P \cup \{p\}, p \notin P$. By the definition of $\preceq_B$, there is a transition $s \xrightarrow{w_{|\pi'|}} R \cup \{r\}, r \notin R$ of $\mathcal{A}$ such that $leaf(\pi') \preceq_B s$ and $P \preceq_F^{\forall\exists} R$. Let $T' = T \setminus \{\pi\} \setminus \bigcup_{p' \in P} \pi' T(\pi' p')$. Then T' is a partial run of $\mathcal{A}$ on w and $\pi' \in branches(T')$, $|\pi'| < |\pi|$, and therefore we can apply induction hypothesis to T', $leaf(\pi')$, s, and π'. This gives us a partial run U' of $\mathcal{A}$ on w with $\psi' \in branches(U')$ such that $leaf(\psi') = s$, $\pi' \preceq_B \psi'$ and for each $1 \leq j \leq |\pi'|$, $T' \ominus_j \pi' \preceq_F^{\forall\exists} U' \ominus_j \psi'$. For every $p' \in succ_T(\pi')$, $T(\pi' p')$ is a partial run of $\mathcal{A}$ with the root p' on the suffix v of w such that $w = uv, |u| = |\pi| - 1$. We can apply Lemma 3.2 to the triples $r' \in R, p' \in P, T(\pi' p')$ with $p' \preceq_F r'$. This gives us for each $r' \in R$ a run $U_{r'}$ of $\mathcal{A}$ on v with $root(U_{r'}) = r'$ such that there is some $p' \in P$ with $T(\pi' p') \preceq_F U_{r'}$. Now we construct a run U and a path ψ with the required properties by plugging r and runs $U_{r'}, r' \in R$ to the path ψ' in U', i.e., $\psi = \psi' r$

and $U = U' \cup \{\psi\} \cup \bigcup_{r' \in R} \psi' U_{r'}$. (To see that U really satisfies the required properties, observe the following: (i) As $U \ominus_{|\pi'|} \psi = \bigcup_{r' \in R} \mathit{branches}(U_{r'})$ and $T \ominus_{|\pi'|} \pi = \bigcup_{p' \in P} \mathit{branches}(T(\pi' p'))$, and because for each $r' \in R$, there is $p' \in P$ with $T(\pi' p') \preceq_F U_{r'}$, we have that $T \ominus_{|\pi'|} \pi \preceq_F^{\forall\exists} U \ominus_{|\pi'|} \psi$. (ii) For all $1 \leq j < |\pi'|$, $T \ominus_j \pi = T' \ominus_j \pi' \preceq_F^{\forall\exists} U' \ominus_j \psi' = U \ominus_j \psi$.). □

4. Quotient wrt. Mediated Equivalence

Here we discuss the possibility of an indirect use of backward simulation for simplifying ABA via computing its quotient. We do not look for a suitable fragment of backward simulation. Instead, we (1) combine backward and forward simulation to form an equivalence that subsumes both backward and forward simulation equivalence and (2) compute the quotient wrt. a certain fragment of this equivalence, called *mediated equivalence*.

4.1. The Notion and Intuition of Mediated Equivalence

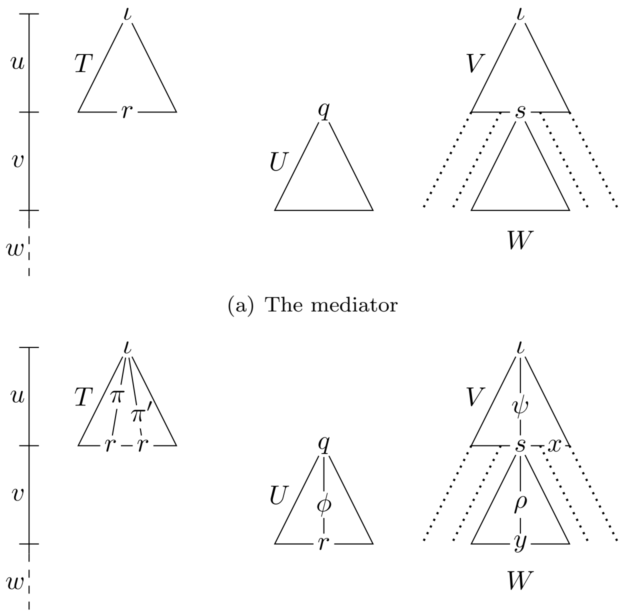
Collapsing states of an automaton wrt. some equivalence allows a run that arrives to some state to *jump* to another equivalent state and continue from there. Alternatively, this can be viewed as *extending* the source state of the jump by the outgoing transitions of the target state[2]. The equivalence must have the property that the language is not increased even when the jumps (or, alternatively, transition extensions) are allowed. This is what we aim at when introducing the *mediated equivalence* $\equiv_M$ based on a so called *mediated preorder* $\preceq_M$. The mediated preorder $\preceq_M$ will be defined as a suitable transitive fragment of $\preceq_F \circ \preceq_B^{-1}$ in the following.

The intuition behind allowing a run to jump from a state r to a state q such that $q \preceq_F \circ \preceq_B^{-1} r$ is the existence of the so called *mediator*, i.e., a state s such that $q \preceq_F s \preceq_B^{-1} r$ (cf. Fig. 2(a)). The state s can be reached in the same way and in the same context[3] as r, and, at the same time, the automaton can continue from s in the same way as from q. Hence, intuitively, the newly allowed run based on the jump from r to q does not add anything to the language because it can anyway be realized through s without jumps.

Unfortunately, jumping cannot be allowed between all pairs of states from $\preceq_F \circ \preceq_B^{-1}$. We will have to restrict ourselves only to its fragments $\preceq_M$ that are *preorders* and are also *forward extensible*, which means that if $q_1 \preceq_M q_2 \preceq_F q_3$, then $q_1 \preceq_M q_3$. The reason for this is that we were so far taking into account only one isolated jump, however, nothing prevents another jump from occurring in the context or below the marked occurrence of r. This is problematic since the relations $q \preceq_F s \preceq_B^{-1} r$ are guaranteed only when no further jumps are allowed. The forward extensibility is required to ensure the mechanism to work

[2] The first view is better when explaining the intuition whereas the other is easier to be used in proofs.

[3] If a state s is a leaf of a partial run, then by a *context* of s we mean all the other leaves of the partial run.



(a) The mediator

(b) Potential problems

Figure 2: The basic intuition behind mediated equivalence

with arbitrary many jumps. We describe the potential problems when $\preceq_M$ is not forward extensible (see Figure 2(b) for the illustration).

Problem (i): The first problem will arise if there is a branch ϕ of U with $\mathit{leaf}(\phi) = r$. Here, apart from interconnecting T and U, r can use its new transitions also at the end of $\pi\phi$ and connect another copy of U to the end of $\pi\phi$. Suppose that all leaves of T except r accept vvw and that all leaves of U except r accept vw. Then this enables a new accepting run on the word $uvvw$. In this case, the existence of the mediator s is not a guarantee that some accepting run on $uvvw$ was possible before adding transitions to r.

Problem (ii): Another problem may arise if there are two (or more) branches in T ending by r. Here we use the two branches π and π' in Figure 2(b) as an example. To construct an accepting run on uvw from T, r has to use the transitions of q at the end of π as well as at the end of π' to connect U to T in the both places. But the partial run V "covers" only one of the two occurrences of r. There may be a leaf x of V different from s for which r is the only leaf in T with $r \preceq_F x$. Therefore, x needs not accept vw as there is no guaranteed relation between q and x. In this case V is not a prefix of an accepting run on uvw and uvw need not be in $\mathcal{L}(\mathcal{A})$.

We will show how the two problems can be solved by requiring $\preceq_M$ to be a forward extensible preorder.

In the case of Problem (i), if y uses transitions of q to accept vw, then W

becomes a prefix of an accepting run on vvw and thus V becomes a prefix of a new accepting run on $uvvw$. We know that $r \preceq_F y$. By forwards extensibility, $q \preceq r \preceq_F y$ gives $q \preceq y$, which implies that there is a mediator for q and y. Observe that y used transitions of q just once. Therefore, by an analogous argument by which we derived that $\mathcal{A}$ accepts uvw in the first case when r used the new transitions only once, we can here derive that there is an accepting run of $\mathcal{A}$ on $uvvw$ which does not involve new transitions.

In the case of Problem (ii), if x uses the transitions of q to accept vw, V becomes a prefix of a new accepting run on uvw. We know that $r \preceq_F x$ and thus by forward extensibility $q \preceq r \preceq_F x$ gives $q \preceq x$, which means that there is a mediator for q and x. Similarly as in the previous case, since x used the transitions of q only once, we can derive that there exists an accepting run of $\mathcal{A}$ on uvw that does not involve new transitions.

The argumentation from the two above paragraphs can be used inductively for a run where r uses transitions of q arbitrarily many times.

Mediated Preorder and Equivalence. We now formally define mediated preorder and equivalence. Let $\preceq_F$ be a reflexive and transitive forward simulation on $\mathcal{A}$, and $\preceq_B$ a reflexive and transitive backward simulation on $\mathcal{A}$ parametrised by $\preceq_F$. A preorder $\preceq_M \subseteq \preceq_F \circ \preceq_B^{-1}$ such that for all $q, r, s \in Q$, $q \preceq_M r \preceq_F s$ implies $q \preceq_M s$, is a *mediated preorder* induced by $\preceq_F$ and $\preceq_B$. The relation $\equiv_M = \preceq_M \cap \preceq_M^{-1}$ is then a *mediated equivalence* induced by $\preceq_F$ and $\preceq_B$.

Lemma 4.1. *[8] There is a unique maximal mediated preorder $\preceq_M$ induced by $\preceq_F$ and $\preceq_B$.*

Ambiguity. To make the mediated equivalence applicable, we must pose one more requirement. Namely, we require that the transitions of the given ABA are not $\preceq_F$-*ambiguous*, meaning that no two states on the right hand side of a transition are forward equivalent. Intuitively, allowing such transitions goes against the spirit of the backward simulation. For a mediator p to backward simulate a state r wrt. rules $\rho_1 : p' \xrightarrow{a} P \cup \{p\}, p \notin P$, and $\rho_2 : r' \xrightarrow{a} R \cup \{r\}, r \notin R$, it must be the case that each state x in the context P of p within ρ_1 is less restrictive (i.e., forward bigger) than some state y in the context R of r within ρ_2. The state r itself is not taken into account when looking for y because we aim at extending its behaviour by collapsing (and it could then become less restrictive than the appropriate x). In the case of $\preceq_F$-ambiguity, the spirit of this restriction is in a sense broken since the forward behaviour of r may still be taken into account when checking that the context of p is less restrictive than that of r. This is because the behaviour of r appears in R as the behaviour of some other forward equivalent state r'' too. Consequently, r and r'' may back up each other in a circular way when checking the restrictiveness of the contexts within the construction of the backward simulation. Both of them can then seem extensible, but once their behaviour gets extended, the restriction of their context based on their own original behaviour is lost, which may then increase the language (an example of such a scenario is given below). However, in Section 5, we show that $\preceq_F$-ambiguity can be efficiently removed.

Example 2 (mediated minimization cannot be used on an ambiguous ABA)**.** *Consider the following ABA $\mathcal{A} = (\{a,b\}, \{s_0,s_1,s_2,s_3,s_4\}, s_0, \delta, \{s_4\})$ where*

$$\begin{array}{ll} s_0 \xrightarrow{a} \{s_1,s_2,s_3\}, & s_3 \xrightarrow{b} \{s_4\}, \\ s_1 \xrightarrow{b} \{s_4\}, & s_3 \xrightarrow{a} \{s_1,s_2,s_3\}, \\ s_2 \xrightarrow{b} \{s_4\}, & s_4 \xrightarrow{a} \{s_4\} \end{array}$$

are transitions of $\mathcal{A}$. The maximal forward simulation relation $\preceq_F$ in $\mathcal{A}$ is the smallest preorder such that $s_1 \equiv_F s_2 \preceq_F s_3 \succeq_F s_0$. From $s_1 \equiv_F s_2$ and the transition $s_0 \xrightarrow{a} \{s_1,s_2,s_3\}$ we can find that $\mathcal{A}$ is $\preceq_F$-ambiguous. The maximal backward simulation relation $\preceq_B$ parametrised with $\preceq_F$ is the smallest equivalence where $s_1 \equiv_B s_2 \equiv_B s_3$, and the mediated preorder $\preceq_M$ is the smallest preorder where $s_0 \preceq_M s_1 \equiv_M s_2 \equiv_M s_3$.

If we collapse states wrt. $\equiv_M = \preceq_M \cap \preceq_M^{-1}$ (i.e., merge the three states s_1, s_2, and s_3), we will get the ABA $\mathcal{A}' = (\{a,b\}, \{s_0, \{s_1,s_2,s_3\}, s_4\}, s_0, \delta, \{s_4\})$ where

$$\begin{array}{ll} s_0 \xrightarrow{a} \{\{s_1,s_2,s_3\}\}, & \{s_1,s_2,s_3\} \xrightarrow{b} \{s_4\}, \\ \{s_1,s_2,s_3\} \xrightarrow{b} \{s_4\}, & \{s_1,s_2,s_3\} \xrightarrow{a} \{\{s_1,s_2,s_3\}\}, \\ & s_4 \xrightarrow{a} \{s_4\}. \end{array}$$

are transitions of $\mathcal{A}'$. Note that $\mathcal{A}'$ accepts the word $aaba^\omega$, but $\mathcal{A}$ does not. □

4.2. Quotient wrt. Mediated Equivalence Has the Same Language

In this section, we give a formal proof that under the assumption that $\mathcal{A}$ is $\preceq_F$-unambiguous, the language of quotient wrt. mediated equivalence is the same as the language of $\mathcal{A}$.

Quotient Automata versus Extended Automata. As already mentioned, computing a quotient can be seen as a simpler operation of adding transitions and accepting states which simplifies the forthcoming reasoning. Let $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$ be an ABA and let $\equiv$ be an equivalence on Q such that $\equiv = \preceq \cap \preceq^{-1}$ for some preorder $\preceq$. We will use $\mathcal{A}/{\equiv}$ to denote the quotient of $\mathcal{A}$ wrt. $\equiv$ that arises by merging $\equiv$-equivalent states of $\mathcal{A}$, and $\mathcal{A}^+_\preceq$ will stand for the automaton *extended* according to $\preceq$, that is created as follows: for every two states q, r of $\mathcal{A}$ with $q \preceq r$, (i) add all outgoing transitions of q to r, (ii) if $q \equiv r$ and q is final, make r final.

Formally, the automata $\mathcal{A}/{\equiv}$ and $\mathcal{A}^+_\preceq$ are defined as follows. Let $Q/{\equiv}$ denote the partitioning of Q wrt. $\equiv$, and let $[q]$ denote the equivalence class of $\equiv$ containing q. Then $\mathcal{A}/{\equiv} = (\Sigma, Q/{\equiv}, [\iota], \delta/{\equiv}, \{[q] \mid q \in \alpha\})$ and $\mathcal{A}^+_\preceq = (\Sigma, Q, \iota, \delta^+_\preceq, \alpha^+_\preceq)$ where $\alpha^+_\preceq = \{p \mid \exists q \in \alpha.\ q \equiv p\}$ and, for each $a \in \Sigma$, $q \in Q$, $\delta/{\equiv}([q], a) = \bigcup_{p \in [q]} \{\{[p'] \mid p' \in P\} \mid P \in \delta(p,a)\}$ and $\delta^+_\preceq(q,a) = \bigcup_{p \in Q \wedge p \preceq q} \delta(p,a)$.

The following lemma implies that if adding transitions and accepting states according to $\preceq$ preserves the language, then the quotient wrt. $\equiv$ has the same language too.

Lemma 4.2. *$\mathcal{L}(\mathcal{A}/{\equiv}) \subseteq \mathcal{L}(\mathcal{A}^+_\preceq)$.*

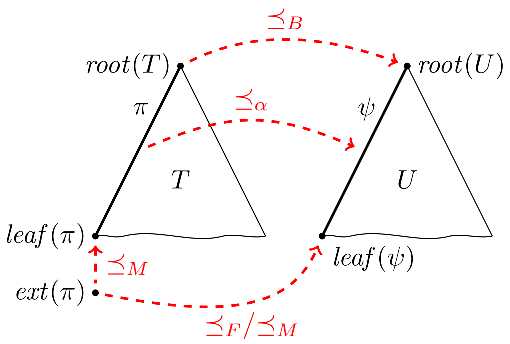


Figure 3: U strongly/weakly covers T wrt. ext

Proof. Let $\mathcal{A}^+_{\equiv} = (\Sigma, Q, \iota, \delta^+_{\equiv}, \alpha^+_{\equiv})$ be the automaton extended according to $\equiv$. Observe that states q and r with $q \equiv r$ are forward simulation equivalent in $\mathcal{A}^+_{\equiv}$. (q and r are in $\mathcal{A}^+_{\equiv}$ either both accepting or both nonaccepting, and for all $a \in \Sigma$, $\delta^+_{\equiv}(q, a) = \delta^+_{\equiv}(r, a)$). Gurumurthy et al. [9] prove that the quotient wrt. forward simulation has the same language as the original automaton. Therefore, $\mathcal{L}(\mathcal{A}/\equiv) = \mathcal{L}(\mathcal{A}^+_{\equiv})$. It is also easy to see that $\mathcal{L}(\mathcal{A}^+_{\equiv}) \subseteq \mathcal{L}(\mathcal{A}^+_{\preceq})$, as $\mathcal{A}^+_{\preceq}$ has a richer transition function than $\mathcal{A}^+_{\equiv}$ and $\alpha^+_{\preceq} = \alpha^+_{\equiv}$. Thus, $\mathcal{L}(\mathcal{A}/\equiv) = \mathcal{L}(\mathcal{A}^+_{\equiv}) \subseteq \mathcal{L}(\mathcal{A}^+_{\preceq})$. □

We now give the proof that extending automata according to the mediated preorder preserves the language. For the rest of the section, we fix an ABA $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$, a reflexive and transitive forward simulation $\preceq_F$ on $\mathcal{A}$ such that $\mathcal{A}$ is $\preceq_F$-unambiguous, and a reflexive and transitive backward simulation $\preceq_B$ on $\mathcal{A}$ parametrised by $\preceq_F$. Let $\preceq_M$ be the maximal mediated preorder induced by $\preceq_F$ and $\preceq_B$, and let $\mathcal{A}^+ = (\Sigma, Q, \iota, \delta^+, \alpha^+)$ be the automaton extended according to $\preceq_M$ (we omit the subscript $\preceq_M$ for the ease of notation). Let $\equiv_M = \preceq_M \cap \preceq_M^{-1}$.

We want to prove that $\mathcal{L}(\mathcal{A}^+) = \mathcal{L}(\mathcal{A})$. The nontrivial part is showing that $\mathcal{L}(\mathcal{A}^+) \subseteq \mathcal{L}(\mathcal{A})$—the converse is obvious. To prove $\mathcal{L}(\mathcal{A}^+) \subseteq \mathcal{L}(\mathcal{A})$, we need to show that, for every accepting run of $\mathcal{A}^+$ on a word w, there is an accepting run of $\mathcal{A}$ on w. We first prove Lemma 4.3, which shows how partial runs of $\mathcal{A}$ with an increased power of their leaves (wrt. $\preceq_M$) can be built incrementally from other runs of $\mathcal{A}$, bridging the gap between $\mathcal{A}$ and $\mathcal{A}^+$. Then we prove Lemma 4.6 saying that for every partial run on a word w of $\mathcal{A}^+$, there is a partial run of $\mathcal{A}$ on w that is more accepting (recall that partial runs are finite). By carrying this result over to infinite runs we get the proof that the automaton extended according to $\preceq_M$, and thus also the quotient wrt. $\equiv_M$, have the same language as the original.

Extension Function and Covering. Consider a partial run T of $\mathcal{A}$ on a word w, we choose for each leaf p of T an $\preceq_M$-smaller state p'. Suppose that we allow p to make one step using the transitions of p' or to become accepting if p' is accepting and $p' \equiv_M p$. (Thus, we give the leaves of T a part of the power they would have in $\mathcal{A}^+$). We will show that there exists a partial run U of $\mathcal{A}$ on w such that (1) it is more accepting than T, and (2) the leaves of U can mimic the next step of the leaves of T even if the leaves of T use their extended power.

The above is formalised in Lemma 4.3 using the following notation. For a partial run T of $\mathcal{A}$ on w, we define *ext* as an *extension function* that assigns to every branch π of T a state $\mathit{ext}(\pi)$ such that $\mathit{ext}(\pi) \preceq_M \mathit{leaf}(\pi)$.

Let U be a partial run of $\mathcal{A}$ on w. For two branches $\pi \in \mathit{branches}(T)$ and $\psi \in \mathit{branches}(U)$, we say that ψ *strongly covers* π wrt. *ext*, denoted $\pi \preceq_{\mathit{ext}} \psi$, iff $\pi \preceq_\alpha \psi$ and $\mathit{ext}(\pi) \preceq_F \mathit{leaf}(\psi)$. Similarly, we say that ψ *weakly covers* π wrt. *ext*, denoted $\pi \preceq_{\mathsf{w}\text{-}\mathit{ext}} \psi$, iff $\pi \preceq_\alpha \psi$ and $\mathit{ext}(\pi) \preceq_M \mathit{leaf}(\psi)$. We extend the concept of covering to partial runs as follows. We write $T \preceq_{\mathit{ext}} U$ (U *strongly covers* T wrt. *ext*) iff $\mathit{branches}(T) \preceq_{\mathit{ext}}^{\forall\exists} \mathit{branches}(U)$ and $\mathit{root}(T) \preceq_B \mathit{root}(U)$. Likewise, we write $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} U$ (U *weakly covers* T wrt. *ext*) iff $\mathit{branches}(T) \preceq_{\mathsf{w}\text{-}\mathit{ext}}^{\forall\exists} \mathit{branches}(U)$ and $\mathit{root}(T) \preceq_B \mathit{root}(U)$. See Figure 3 for an illustration. Note that we have $\preceq_{\mathit{ext}} \subseteq \preceq_{\mathsf{w}\text{-}\mathit{ext}}$ for branches as well for partial runs because $\preceq_F \subseteq \preceq_M$—the strong covering implies the weak one.

Lemma 4.3. *For any partial run T of $\mathcal{A}$ on a word w with an extension function ext, there is a partial run U of $\mathcal{A}$ on w with $T \preceq_{\mathit{ext}} U$.*

Proving Lemma 4.3 is the most intricate part of the proof of Theorem 1. We now introduce the concepts used within the proof, prove auxiliary Lemma 4.5, and subsequently present the proof of Lemma 4.3 itself.

Observe that $\mathit{root}(T) \preceq_B \mathit{root}(T)$, and every branch of T weakly covers itself, which means that $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} T$. Within the proof of Lemma 4.3, we will show how to reach U by a chain of partial runs derived from T. The partial runs within the chain will all weakly cover T. Runs further from T will in some sense cover T more strongly than the runs closer to T and the last partial run of the chain will cover T strongly. In the following paragraph, we formulate what it means that a partial run weakly covering T covers T more strongly than another partial run.

The Relation of Covering T More Strongly. To define the relation of covering T more strongly on partial runs that weakly cover T, we concentrate on those branches of a partial run that cause that the partial run does not cover T strongly. Let V be a partial run of $\mathcal{A}$ on w with $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} V$. We call a branch $\psi \in \mathit{branches}(V)$ *strict weakly covering* if there is no $\pi \in \mathit{branches}(T)$ with $\pi \preceq_{\mathit{ext}} \psi$ (there are only some $\pi \in \mathit{branches}(T)$ with $\pi \preceq_{\mathsf{w}\text{-}\mathit{ext}} \psi$). Let $\mathsf{sw}_T(V)$ denote the tree which is the subset of V containing all prefixes of strict weakly covering branches of V wrt. T. Note that $T \preceq_{\mathit{ext}} V$ iff V contains no strict weakly covering branches, which is equivalent to $\mathsf{sw}_T(V) = \emptyset$. Given a partial run W of $\mathcal{A}$ on w, we will define which of V and W cover T more strongly by comparing $\mathsf{sw}_T(V)$ and $\mathsf{sw}_T(W)$. For this, we need the following definitions.

Given a finite tree X over Q and $\tau \in Q^+$, we define the *tree decomposition* of X according to τ as the sequence of (finite) sets of paths $\langle \tau, X \rangle = X \ominus_1 \tau, X \ominus_2 \tau, \ldots, X \ominus_{|\tau|} \tau$. We also let $\langle \epsilon, X \rangle = \mathit{branches}(X)$ (it is a sequence of length 1). A substantial property of tree decompositions is that under the condition that $\tau \notin \mathit{branches}(X)$, $\langle \tau, X \rangle = \emptyset \ldots \emptyset$ implies that $X = \emptyset$. Notice that if $\tau \in \mathit{branches}(X)$, $\langle \tau, X \rangle = \emptyset \ldots \emptyset$ does not imply $X = \emptyset$ as τ could be the only branch of X. This is important as for a partial run Y and $\tau' \in Y$, if $\tau' \notin \mathit{branches}(Y)$, the implications $\langle \tau', \mathsf{sw}_T(Y) \rangle = \emptyset \ldots \emptyset \implies \mathsf{sw}_T(Y) = \emptyset \implies T \preceq_{ext} Y$ hold. However, the first implication does not hold if $\tau' \in \mathit{branches}(Y)$.

Let $\tau_V \in V \cup \{\epsilon\}$ and $\tau_W \in W \cup \{\epsilon\}$ be such that $\tau_V \notin \mathit{branches}(\mathsf{sw}_T(V))$ and $\tau_W \notin \mathit{branches}(\mathsf{sw}_T(W))$. We say that *$W$ covers T more strongly* than V wrt. τ_V and τ_W, denoted $V \prec^T_{\tau_V,\tau_W} W$, iff $\mathit{root}(V) \preceq_B \mathit{root}(W)$ and $\langle \tau_V, \mathsf{sw}_T(V) \rangle \sqsubset \langle \tau_W, \mathsf{sw}_T(W) \rangle$ where $\sqsubset$ is a binary relation on finite sequences of sets of paths defined as follows:

For two sets of paths P and P', we use $P \prec^{\forall\exists}_F P'$ to denote that $P \preceq^{\forall\exists}_F P'$ but not $P' \preceq^{\forall\exists}_F P$. In other words, the upward closure of P' wrt. $\preceq_F$ is a proper subset of the upward closure of P wrt. $\preceq_F$. Then, for two finite sequences $S, S' \in (2^{Q^+})^+$ of sets of paths, $S \sqsubset S'$ iff there is some $k \in \mathbb{N}, k \leq \min\{|S|, |S'|\}$, such that $S_k \prec^{\forall\exists}_F S'_k$ and for all $1 \leq j < k$, $S_j \preceq^{\forall\exists}_F S'_j$.

Given $c \in \mathbb{N}$, we say that a sequence S of sets of paths is *c-bounded* if $|S| \leq c$ and also the length of every path in every $S_i, 1 \leq i \leq |S|$ is at most c. Lemma 4.4 below shows that every maximal increasing chain of c-bounded sequences related by $\sqsubset$ eventually arrives to $\emptyset \ldots \emptyset$. This will allow us to show that every maximal sequence of partial runs that cover T more and more strongly must terminate by a partial run that covers T strongly.

Lemma 4.4. *Given a constant c, every maximal increasing chain of c-bounded sequences related by $\sqsubset$ eventually terminates by $\emptyset \ldots \emptyset$.*

Proof. First, observe that for every sequence S of sets of paths with $S \neq \emptyset \ldots \emptyset$, it holds that $S \sqsubset \emptyset \ldots \emptyset$. This is easy to see since $\emptyset \preceq^{\forall\exists}_F \emptyset$ and $X \prec^{\forall\exists}_F \emptyset$ for any nonempty $X \in 2^{Q^+}$. Therefore, to prove the lemma, it is sufficient to show that $\sqsubset$ does not allow infinite increasing chains of c-bounded sequences.

Let $S = S(1) \sqsubset S(2) \sqsubset S(3) \sqsubset \cdots$ be such a chain of c-bounded sequences. We will show that S must be finite. Observe that the domain of possible c-bounded $S(i)$s is finite since there are only finitely many paths whose length is bounded by c (Q is finite). Therefore, if S is an infinite chain, there has to be i and j with $i < j$ such that $S(i) = S(j)$. We will argue that this is not possible by showing that $\sqsubset$ is irreflexive and transitive, which means that it does not allow loops (if there was a loop $X \sqsubset \cdots \sqsubset X$, then by transitivity, $X \sqsubset X$ which contradicts irreflexifity).

Irreflexivity of $\sqsubset$ may be shown as follows. Let $S \sqsubset S$ for some c-bounded sequence S. By the definition of $\sqsubset$, there is $k \in \mathbb{N}$ such that $S_i \preceq^{\forall\exists}_F S_i$ for all $i \in \mathbb{N}$ smaller than k, and $S_k \prec^{\forall\exists}_F S_k$. However, this is clearly not possible since since the upward closure of S_k wrt. $\preceq_F$ would have to be a proper subset of itself.

Transitivity of $\sqsubset$ can be shown as follows. Let S, S', S'' be three c-bounded sequences with $S \sqsubset S' \sqsubset S''$. By the definition of $\sqsubset$, there is $k \in \mathbb{N}$ such that $S_i \preceq^{\forall\exists}_F S'_i$ for all $i \in \mathbb{N}$ smaller than k, and $S_k \prec^{\forall\exists}_F S'_k$; and there is $k' \in \mathbb{N}$

such that $S'_i \preceq^{\forall\exists}_F S''_i$ for all $i \in \mathbb{N}$ smaller than k', and $S'_{k'} \prec^{\forall\exists}_F S''_{k'}$. Let $l = \min\{k, k'\}$. By transitivity of $\preceq^{\forall\exists}_F$, we have that $S_i \preceq^{\forall\exists}_F S''_i$ for all $i \in \mathbb{N}$ smaller than l. Then, for the lth position, we have that $S_l \prec^{\forall\exists}_F S'_l \prec^{\forall\exists}_F S''_l$ or $S_l \preceq^{\forall\exists}_F S'_l \prec^{\forall\exists}_F S''_l$ or $S_l \prec^{\forall\exists}_F S'_l \preceq^{\forall\exists}_F S''_l$. All these three possibilities give $S_l \prec^{\forall\exists}_F S''_l$, and thus $S \sqsubset S''$. □

The last ingredient we need for the proof of Lemma 4.3 is to show that for every maximal sequence of partial runs that cover T more and more strongly, the underlying $\sqsubset$-related sequence is also maximal. Particularly, we need to show that for any partial run weakly (but not strongly) covering T, we are always able to construct a partial run covering T more strongly. This is stated by the following lemma.

Lemma 4.5. *Given a partial run V of $\mathcal{A}$ on w s.t. $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} V$, $T \not\preceq_{\mathit{ext}} V$, and $\tau_V \in V \cup \{\epsilon\}$ with $\tau_V \notin \mathit{branches}(\mathsf{sw}_T(V))$, we can construct a partial run W of $\mathcal{A}$ on w with $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} W$ and a path $\tau_W \in W$ with $\tau_W \notin \mathit{branches}(\mathsf{sw}_T(W))$ such that $V \prec^T_{\tau_V,\tau_W} W$.*

Proof. The proof relies on Lemma 3.3 and the definition of $\preceq_M$. We first choose a suitable branch π of $\mathsf{sw}_T(V)$ as follows. Let $1 \leq k \leq |\tau_V|$ be some index such that $\mathsf{sw}_T(V) \ominus_k \tau_V$ is nonempty. If $\tau_V = \epsilon$, then $k = 1$. We choose some $\pi' \in \mathsf{sw}_T(V) \ominus_k \tau_V$ which is minimal wrt. $\preceq_F$, meaning that there is no $\pi'' \in \mathsf{sw}_T(V) \ominus_k \tau_V$ different from π' such that $\pi'' \preceq_F \pi'$. We put $\pi = \tau_V^k \pi'$. We note that this is the place where we use the $\preceq_F$-unambiguity assumption. If $\mathcal{A}$ was $\preceq_F$-ambiguous, there need not be a k such that $\mathsf{sw}_T(V) \ominus_k \tau_V$ contains a minimal element wrt. $\preceq_F$.

As $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} V$, there is $\sigma \in \mathit{branches}(T)$ with $\sigma \preceq_{\mathsf{w}\text{-}\mathit{ext}} \pi$. ¿From $\mathit{ext}(\sigma) \preceq_M \mathit{leaf}(\pi)$, there is a mediator s with $\mathit{ext}(\sigma) \preceq_F s \succeq_B \mathit{leaf}(\pi)$. We can apply Lemma 3.3 to V, π, $\mathit{leaf}(\pi)$ and s, which gives us a partial run W and $\psi \in \mathit{branches}(W)$ with $\mathit{leaf}(\psi) = s$ such that $\pi \preceq_B \psi$, and for all $1 \leq i \leq |\pi|$, $V \ominus_i \pi \preceq^{\forall\exists}_F W \ominus_i \psi$. Let $\tau_W = \psi$. The proof will be concluded by showing that (i) $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} W$, (ii) $\tau_W \notin \mathit{branches}(\mathsf{sw}_T(W))$, and (iii) $\langle \tau_V, \mathsf{sw}_T(V)\rangle \sqsubset \langle \tau_W, \mathsf{sw}_T(W)\rangle$, which implies $V \prec^T_{\tau_V,\tau_W} W$.

(i) To show that $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} W$, we proceed as follows. Observe that for every $\phi \in \mathit{branches}(W) \setminus \{\psi\}$ there is a branch $\phi' \in \mathit{branches}(V) \setminus \{\pi\}$ such that $\mathit{leaf}(\phi') \preceq_F \mathit{leaf}(\phi)$ and $\phi' \preceq_\alpha \phi$. This holds because for all $1 \leq i \leq |\pi|$, $V \ominus_i \pi \preceq^{\forall\exists}_F W \ominus_i \psi$ and because $\pi \preceq_B \psi$ (To be more detailed, for every $\phi \in \mathit{branches}(W) \setminus \{\psi\}$, $\phi = \psi^i \rho$ for some i and $\rho \in W \ominus_i \psi$. There must be $\rho' \in V \ominus_i \pi$ with $\rho' \preceq_F \rho$. As $\pi \preceq_B \phi$, $\pi^i \preceq_B \phi^i$ which implies $\pi^i \preceq_\alpha \phi^i$. Similarly, $\rho' \preceq_F \rho$ implies $\rho' \preceq_\alpha \rho$ and also $\mathit{leaf}(\rho') \preceq_F \mathit{leaf}(\rho)$. Therefore, we can construct the branch $\phi' = \pi^i \rho' \in \mathit{branches}(V) \setminus \{\pi\}$ with $\pi^i \rho' \preceq_\alpha \psi^i \rho = \phi$ and $\mathit{leaf}(\pi^i \rho') \preceq_F \mathit{leaf}(\psi^i \rho)$). We also know that since $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} V$, $\mathit{branches}(T) \preceq^{\forall\exists}_{\mathsf{w}\text{-}\mathit{ext}} \mathit{branches}(V)$. Thus, by the definition of $\preceq_{\mathsf{w}\text{-}\mathit{ext}}$, we have that for every $\phi \in \mathit{branches}(W) \setminus \{\psi\}$, there are $\phi' \in \mathit{branches}(V)$ and $\phi'' \in \mathit{branches}(T)$ with $\phi'' \preceq_\alpha \phi' \preceq_\alpha \phi$ and $\mathit{ext}(\phi'') \preceq_M \mathit{leaf}(\phi') \preceq_F \mathit{leaf}(\phi)$. This by transitivity of α and the definition of $\preceq_M$ gives $\phi'' \preceq_\alpha \phi$ and $\mathit{ext}(\phi'') \preceq_M \mathit{leaf}(\phi)$, which means $\phi'' \preceq_{\mathsf{w}\text{-}\mathit{ext}} \phi$. To see that also ψ is weakly covering, observe

that since $\sigma \preceq_{\mathsf{w}\text{-}ext} \pi$, we have $\sigma \preceq_\alpha \pi \preceq_B \psi$ and $ext(\sigma) \preceq_F s = leaf(\psi)$, which by $\preceq_B \subseteq \preceq_\alpha$ and transitivity of $\preceq_\alpha$ gives even $\sigma \preceq_{ext} \psi$ (immediately implying $\sigma \preceq_{\mathsf{w}\text{-}ext} \psi$). Finally, from $root(T) \preceq_B root(V)$ (implied by $T \preceq_{\mathsf{w}\text{-}ext} V$), $\pi \preceq_B \psi$, and transitivity of $\preceq_B$, $root(T) \preceq_B root(W)$. We have shown that $T \preceq_{\mathsf{w}\text{-}ext} W$.

(ii) Showing that $\psi \notin branches(\mathsf{sw}_T(W))$ is easy. In the above paragraph we have just shown that $\sigma \preceq_{ext} \psi$, thus ψ is not a strict weakly covering branch.

(iii) To show that $\langle \tau_V, \mathsf{sw}_T(V)\rangle \sqsubset \langle \psi, \mathsf{sw}_T(W)\rangle$, we will argue that (a) for all $1 \leq i < k$, it holds that $\mathsf{sw}_T(V) \ominus_i \tau_V \preceq_F^{\forall\exists} \mathsf{sw}_T(W) \ominus_i \psi$ and that (b) $\mathsf{sw}_T(V) \ominus_k \tau_V \prec_F^{\forall\exists} \mathsf{sw}_T(W) \ominus_k \psi$. Notice first that for any partial run X of $\mathcal{A}$ and $\tau \in X$ with $\tau \notin branches(\mathsf{sw}_T(X))$, for all $1 \leq j \leq |\tau|$, $\mathsf{sw}_T(X) \ominus_j \tau \subseteq X \ominus_j \tau$. Recall that $\tau_V^k = \pi^k$, that $\mathsf{sw}_T(V) \ominus_k \tau_V$ is nonempty, and that for all $1 \leq i < |\pi|$, $V \ominus_i \pi \preceq_F^{\forall\exists} W \ominus_i \psi$.

We first show that for all $1 \leq i < |\pi|$, $\mathsf{sw}_T(V) \ominus_i \pi \preceq_F^{\forall\exists} \mathsf{sw}_T(W) \ominus_i \psi$. For every $\phi \in \mathsf{sw}_T(W) \ominus_i \psi$, there is at least one $\phi' \in V \ominus_i \pi$ with $\phi' \preceq_F \phi$ (because $V \ominus_i \pi \preceq_F^{\forall\exists} W \ominus_i \psi$ and $\mathsf{sw}_T(W) \ominus_i \psi \subseteq W \ominus_i \psi$). We will show by contradiction that $\phi' \in \mathsf{sw}_T(V) \ominus_i \pi$ which will imply $\mathsf{sw}_T(V) \ominus_i \pi \preceq_F^{\forall\exists} \mathsf{sw}_T(W) \ominus_i \psi$. Suppose that $\phi' \notin \mathsf{sw}_T(V) \ominus_i \pi$. Then the branch $\pi^i\phi'$ of V is not strict weakly covering, and as $T \preceq_{\mathsf{w}\text{-}ext} V$, we have that there is some $\phi'' \in branches(T)$ with $\phi'' \preceq_{ext} \pi^i\phi'$. As $\pi \preceq_B \psi$, we have that $\pi^i \preceq_\alpha \psi^i$. As $\phi' \preceq_F \phi$, we have that $\phi' \preceq_\alpha \phi$ and $leaf(\phi') \preceq_F leaf(\phi)$. This together with $\phi'' \preceq_{ext} \pi^i\phi'$ gives that $\phi'' \preceq_\alpha \pi^i\phi' \preceq_\alpha \psi^i\phi$ and $ext(\phi'') \preceq_F leaf(\pi^i\phi') \preceq_F leaf(\psi^i\phi)$. By transitivity of $\preceq_\alpha$ and $\preceq_F$ and by the definition of $\preceq_{ext}$, we obtain $\phi'' \preceq_{ext} \psi^i\phi$. This contradicts with the fact that $\psi^i\phi$ is strict weakly covering (as $\phi \in \mathsf{sw}_T(W) \ominus_i \psi$) and therefore it must be the case that $\phi' \in \mathsf{sw}_T(V) \ominus_i \pi$.

(a) The fact that for all $1 \leq i < k$, $\mathsf{sw}_T(V) \ominus_i \tau_V \preceq_F^{\forall\exists} \mathsf{sw}_T(W) \ominus_i \psi$ is implied by the result of the previous paragraph, because $\tau_V^k = \pi^k$ (thus $\mathsf{sw}_T(V) \ominus_i \tau_V = \mathsf{sw}_T(V) \ominus_i \pi$).

(b) It remains to show that $\mathsf{sw}_T(V) \ominus_k \tau_V \prec_F^{\forall\exists} \mathsf{sw}_T(W) \ominus_k \psi$. By the definitions of $\ominus_k$, π and τ_V, it holds that $\mathsf{sw}_T(V) \ominus_k \tau_V \supset \mathsf{sw}_T(V) \ominus_k \pi$. (To see this, recall that π is strict weakly covering, but τ_V is not. Therefore, $\mathsf{sw}_T(V) \ominus_k \pi = \mathsf{sw}_T(V) \ominus_k \tau_V \setminus branches(\mathsf{sw}_T(V)(\pi^{k+1})))$. Since $\supset$ implies $\preceq_F^{\forall\exists}$, we have that $\mathsf{sw}_T(V) \ominus_k \tau_V \preceq_F^{\forall\exists} \mathsf{sw}_T(V) \ominus_k \pi$. Moreover, as $\pi' \notin \mathsf{sw}_T(V) \ominus_k \pi$ and π' is a minimal element of $\mathsf{sw}_T(V) \ominus_k \tau_V$, we have that $\mathsf{sw}_T(V) \ominus_k \pi \preceq_F^{\forall\exists} \mathsf{sw}_T(V) \ominus_k \tau_V$ cannot hold (there is no $\pi'' \in \mathsf{sw}_T(V) \ominus_k \pi$ with $\pi'' \preceq_F \pi'$), and therefore we have $\mathsf{sw}_T(V) \ominus_k \tau_V \prec_F^{\forall\exists} \mathsf{sw}_T(V) \ominus_k \pi$. Finally, $\mathsf{sw}_T(V) \ominus_k \tau_V \prec_F^{\forall\exists} \mathsf{sw}_T(V) \ominus_k \pi \preceq_F^{\forall\exists} \mathsf{sw}_T(W) \ominus_k \psi$ gives $\mathsf{sw}_T(V) \ominus_k \tau_V \prec_F^{\forall\exists} \mathsf{sw}_T(W) \ominus_k \psi$. This completes the part (iii) of the proof and we can conclude that $V \prec_{\tau_V,\psi}^T W$. □

With Lemma 4.5 in hand, we are finally ready to prove Lemma 4.3.

Proof of Lemma 4.3. If $T \preceq_{ext} T$, we are done as in the statement of the lemma, we can take T to be U. So, suppose that $T \not\preceq_{ext} T$. Observe that $root(T) \preceq_B root(T)$, and every branch of T weakly covers itself, which means that $T \preceq_{\mathsf{w}\text{-}ext} T$. We construct a run U strongly covering T as follows. Starting from T and ϵ, we can construct a chain $T \prec_{\epsilon,\tau_1}^T T_1 \prec_{\tau_1,\tau_2}^T T_2 \prec_{\tau_2,\tau_3}^T T_3 \ldots$ of partial runs that more and more strongly cover T by successively applying Lemma 4.5 for each i, $\tau_i \in T_i$, $\tau_i \notin \mathit{branches}(\mathsf{sw}_T(T_i))$, and $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} T_i$. Observe that by the definition of stronger covering, we have that $\langle \epsilon, \mathsf{sw}_T(T) \rangle \sqsubset \langle \tau_1, \mathsf{sw}_T(T_1) \rangle \sqsubset \langle \tau_2, \mathsf{sw}_T(T_2) \rangle \sqsubset \langle \tau_3, \mathsf{sw}_T(T_3) \rangle \ldots$.

Notice now that for each i, since $T \preceq_{\mathsf{w}\text{-}\mathit{ext}} T_i$, $\mathit{height}(T_i) \leq \mathit{height}(T)$. Therefore, since length of τ_i is bounded by $\mathit{height}(T)$, the length of $\langle \tau_i, \mathsf{sw}_T(T_i) \rangle$ is bounded by $\mathit{height}(T)$ too. As lengths of all paths in the sets within $\langle \tau_i, \mathsf{sw}_T(T_i) \rangle$ are obviously bounded by $\mathit{height}(T)$ as well, $\langle \tau_i, \mathsf{sw}_T(T_i) \rangle$ is a $\mathit{height}(T)$-bounded sequence. Therefore, by Lemma 4.4, the chain must eventually arrive to its last T_k and τ_k with $\langle \tau_k, \mathsf{sw}_T(T_k) \rangle = \emptyset \ldots \emptyset$. As $\langle \tau_k, \mathsf{sw}_T(T_k) \rangle = \emptyset \ldots \emptyset$, $\mathsf{sw}_T(T_k)$ has to be empty, which implies that $T \preceq_{\mathit{ext}} T_k$. We can put $U = T_k$ and Lemma 4.3 is proven. □

We use Lemma 4.3 to prove Lemma 4.6. Informally, it says that even despite the poorer transition relation and smaller set of accepting states, $\mathcal{A}$ can answer to any partial run of $\mathcal{A}^+$ by a more accepting partial run. To express this formally, we need to define the following weaker version $\preceq_{\alpha^+ \Rightarrow \alpha}$ of the relation of being more accepting that takes into account α^+ on the left and α on the right. This is, for states q and r, $q \preceq_{\alpha^+ \Rightarrow \alpha} r$ iff $q \in \alpha^+ \implies r \in \alpha$. For two paths $\pi, \psi \in Q^+$, $\pi \preceq_{\alpha^+ \Rightarrow \alpha} \psi$ iff $|\pi| = |\psi|$ and for all $1 \leq i \leq |\pi|$, $\pi_i \in \alpha^+ \implies \psi_i \in \alpha$. Last, for finite trees T and U over Q, we use $T \preceq_{\alpha^+ \Rightarrow \alpha} U$ to denote that $\mathit{branches}(T) \preceq_{\alpha^+ \Rightarrow \alpha}^{\forall\exists} \mathit{branches}(U)$.

Lemma 4.6. *For any partial run T of $\mathcal{A}^+$ on $w \in \Sigma^\omega$, there exists a partial run U of $\mathcal{A}$ on w such that $\mathit{root}(T) \preceq_B \mathit{root}(U)$ and $T \preceq_{\alpha^+ \Rightarrow \alpha} U$.*

Proof. We will prove the lemma by induction on the structure of T, using Lemma 4.3 within the induction step. To make the induction argument pass, we will prove a stronger variant of the lemma. Particularly, we will replace the relation $\preceq_{\alpha^+ \Rightarrow \alpha}$ within the statement of the lemma by its stronger variant $\preceq_{\alpha^+ \Rightarrow \alpha}^M$ which is defined as follows. Given paths π and ψ, $\pi \preceq_{\alpha^+ \Rightarrow \alpha}^M \psi$ iff $\pi \preceq_{\alpha^+ \Rightarrow \alpha} \psi$ and $\mathit{leaf}(\pi) \preceq_M \mathit{leaf}(\psi)$. For two partial runs V and W, we use $V \preceq_{\alpha^+ \Rightarrow \alpha}^M W$ to denote that $\mathit{branches}(V)$ $(\preceq_{\alpha^+ \Rightarrow \alpha}^M)^{\forall\exists}$ $\mathit{branches}(W)$. Apparently, $\preceq_{\alpha^+ \Rightarrow \alpha}^M \subseteq \preceq_{\alpha^+ \Rightarrow \alpha}$ for paths as well as for partial runs.

A stronger variant of Lemma 4.6: For any partial run T of $\mathcal{A}^+$ on $w \in \Sigma^\omega$, there exists a partial run U of $\mathcal{A}$ on w such that $\mathit{root}(T) \preceq_B \mathit{root}(U)$ and $T \preceq_{\alpha^+ \Rightarrow \alpha}^M U$.

It is obvious that the above statement implies the statement of the lemma. We will prove it by induction to the structure of T. In the base case, $T = \{q\}$ for some $q \in Q$. If $q \notin \alpha^+$, we can put $U = \{q\}$ ($\preceq_M$ and $\preceq_B$ are reflexive). If $q \in \alpha^+$, then by the definition of α^+, there is $p \in \alpha$ such that $p \equiv_M q$. This means that $q \preceq_M p$ and $p \preceq_M q$. By the definition of $\preceq_M$, there exists a mediator s with $p \preceq_F s \succeq_B q$. As $\preceq_F \subseteq \preceq_\alpha$, $s \in \alpha$. Again by the definition of $\preceq_M$, $q \preceq_M p \preceq_F s \succeq_B q$ gives us $q \preceq_M s \succeq_B q$ and we can put $U = \{s\}$.

Suppose now that T is not only a root and that the stronger variant of the lemma holds for every partial run of $\mathcal{A}^+$ on w that is a proper subset of T.

We choose some $\pi \in T$ such that $\mathit{succ}_T(\pi) \neq \emptyset$ and for every $p \in \mathit{succ}_T(\pi)$, $\mathit{succ}_T(\pi p) = \emptyset$. Notice that since T is a finite tree, such π always exists. Denote $P = \mathit{succ}_T(\pi)$ and $q = \mathit{leaf}(\pi)$. Let $T' = T \setminus \{\pi p \mid p \in P\}$. T' is a partial run of $\mathcal{A}^+$ on w which is a proper subset of T, therefore we can apply the induction hypothesis on it. This gives us a partial run V of $\mathcal{A}$ on w such that $\mathit{root}(T') \preceq_B \mathit{root}(V)$ and $T' \preceq^M_{\alpha^+\Rightarrow\alpha} V$.

Let $\mathit{Bad}_V \subseteq \mathit{branches}(V)$ be the set such that $\psi \in \mathit{Bad}_V$ iff there is no $\phi \in \mathit{branches}(T)$ such that $\phi \preceq^M_{\alpha^+\Rightarrow\alpha} \psi$, and let $\mathit{Good}_V = \mathit{branches}(V) \setminus \mathit{Bad}_V$. Intuitively, Bad_V contains the problematic branches because of which $T \preceq^M_{\alpha^+\Rightarrow\alpha} V$ does not hold. If Bad_V is empty, then the relation holds and we can conclude the proof. We continue assuming that $\mathit{Bad}_V \neq \emptyset$.

By the definition of δ^+ and because $q \xrightarrow{w_{|\pi|}} P$ is a transition of $\mathcal{A}^+$, there must be some $s \in Q, s \preceq_M q$ where $s \xrightarrow{w_{|\pi|}} P$ is a transition of δ. We define an extension function ext_V such that $\mathit{ext}_V(\phi) = s$ for every $\phi \in \mathit{Bad}_V$ and $\mathit{ext}_V(\psi) = \mathit{leaf}(\psi)$ for every $\psi \in \mathit{Good}_V$. To see that ext_V conforms to the definition of extension function, one has to show that for every branch $\phi \in \mathit{Bad}_V$, $s \preceq_M \mathit{leaf}(\phi)$. We know that $T' \preceq^M_{\alpha^+\Rightarrow\alpha} V$ but not $T \preceq^M_{\alpha^+\Rightarrow\alpha} V$. Therefore, there is some branch $\phi' \in T'$ with $\phi' \preceq^M_{\alpha^+\Rightarrow\alpha} \phi$ such that $\phi' \notin \mathit{branches}(T)$ (if ϕ' was a branch of T, ϕ would not be in Bad_V). Notice that π is the only branch of T' which is not a branch of T, which means that it must be the case that $\phi' = \pi$. Therefore, since $s \preceq_M q \preceq_M \mathit{leaf}(\phi)$, we have that $s \preceq_M \mathit{leaf}(\phi)$ holds.

By applying Lemma 4.3 to V and ext_V, we get a partial run W of $\mathcal{A}$ on w with $V \preceq_{\mathit{ext}_V} W$. Now, for each $\psi \in \mathit{branches}(W)$, there is $\phi \in \mathit{branches}(V)$ with $\phi \preceq_{\mathit{ext}_V} \psi$. As $T' \preceq^M_{\alpha^+\Rightarrow\alpha} V$, $\rho \preceq^M_{\alpha^+\Rightarrow\alpha} \phi$ for some $\rho \in \mathit{branches}(T')$. There are two cases of how ρ and ψ may be related, depending on ϕ:

1. If $\phi \in \mathit{Good}_V$, then $\mathit{ext}(\phi) = \mathit{leaf}(\phi)$. In this case, by the definitions of $\preceq^M_{\alpha^+\Rightarrow\alpha}$ and $\preceq_{\mathit{ext}_V}$, we have $\rho \preceq_{\alpha^+\Rightarrow\alpha} \phi \preceq_\alpha \psi$ and $\mathit{leaf}(\rho) \preceq_M \mathit{leaf}(\phi) \preceq_F \mathit{leaf}(\psi)$, which gives $\rho \preceq_{\alpha^+\Rightarrow\alpha} \psi$ and $\mathit{leaf}(\rho) \preceq_M \mathit{leaf}(\psi)$ (since $\preceq_M$ is forward extensible), meaning that $\rho \preceq^M_{\alpha^+\Rightarrow\alpha} \psi$.

2. To analyse the case when $\phi \in \mathit{Bad}_V$, recall that π is the only branch of T' which is not a branch of T, and therefore π is also the only branch of T' with $\pi \preceq^M_{\alpha^+\Rightarrow\alpha} \phi$. Therefore, $\rho = \pi$. According to the definition of ext_V, $\mathit{ext}_V(\phi) = s$. Since $\phi \preceq_{\mathit{ext}_V} \psi$, we have $\pi \preceq^M_{\alpha^+\Rightarrow\alpha} \phi \preceq_\alpha \psi$ which gives $\pi \preceq_{\alpha^+\Rightarrow\alpha} \psi$. However, since (contrary to the previous case) $\mathit{ext}_v(\phi) \neq \mathit{leaf}(\phi)$, we cannot guarantee any further relation between $\mathit{leaf}(\phi)$ and $\mathit{leaf}(\psi)$, and we cannot derive that $\mathit{leaf}(\pi) \preceq_M \mathit{leaf}(\psi)$ and $\pi \preceq^M_{\alpha^+\Rightarrow\alpha} \psi$ need not hold.

We define the set $\mathit{Bad}_W \subseteq \mathit{branches}(W)$ such as $\psi \in \mathit{Bad}_W$ iff there is no $\rho \in T$ with $\rho \preceq^M_{\alpha^+\Rightarrow\alpha} \psi$ and we let $\mathit{Good}_W = \mathit{branches}(W) \setminus \mathit{Bad}_V$. This is, Bad_W contains the branches because of which $T \preceq^M_{\alpha^+\Rightarrow\alpha} W$ does not hold. Note that if $\psi \in \mathit{Bad}_V$, then all the $\phi \in \mathit{branches}(V)$ with $\phi \preceq_{\mathit{ext}_V} \psi$ are as in the case (2) above, i.e., π is the only branch of T' with $\pi \preceq^M_{\alpha^+\Rightarrow\alpha} \phi$. By the definition of $\preceq_{\mathit{ext}_V}$, $s = \mathit{ext}_V(\phi) \preceq_F \mathit{leaf}(\psi)$. Therefore, by the definition of $\preceq_F$ and since $s \xrightarrow{w_{|\pi|}} P$, there must be some transition $\mathit{leaf}(\psi) \xrightarrow{w_{|\pi|}} R_\psi$ of $\mathcal{A}$ where

$P \preceq_F^{\forall\exists} R_\psi$. We extend W by firing these transitions for every $\psi \in \mathit{Bad}_W$, in which way we obtain a run $X = W \cup \{\psi R_\psi \mid \psi \in \mathit{Bad}_W\}$ of $\mathcal{A}$ on w.

Let us use $\mathit{New}_X = \{\psi R_\psi \mid \psi \in \mathit{Bad}_W\}$ to denote the branches of X that arose by firing the transitions. Observe that $\mathit{branches}(X) = \mathit{Good}_W \cup \mathit{New}_X$. Recall that for all $\psi \in \mathit{Bad}_W$, $\pi \preceq_{\alpha^+\Rightarrow\alpha} \psi$ and that for every $\psi \in \mathit{New}_X$, there is some $p \in P$ such that $p \preceq_F \mathit{leaf}(\psi)$. We will define an extension function ext_X of X as follows:

1. If $\psi \in \mathit{Good}_W$, $\mathit{ext}_X(\psi) = \mathit{leaf}(\psi)$.

2. If $\psi \in \mathit{New}_X$ and there is $p \in P$ with $p \preceq_F \mathit{leaf}(\psi)$ and $p \preceq_{\alpha^+\Rightarrow\alpha} \mathit{leaf}(\psi)$, we let $\mathit{ext}_X(\psi) = \mathit{leaf}(\psi)$.

3. If $\psi \in \mathit{New}_X$ and there is no $p \in P$ with $p \preceq_F \mathit{leaf}(\psi)$ and $p \preceq_{\alpha^+\Rightarrow\alpha} \mathit{leaf}(\psi)$, we proceed as follows. By the definition of New_X, there is some $p' \in P$ such that $p' \preceq_F \mathit{leaf}(\psi)$. Since $\preceq_F \subseteq \preceq_\alpha$, $p' \preceq_F \mathit{leaf}(\psi)$, and not $p' \preceq_{\alpha^+\Rightarrow\alpha} \mathit{leaf}(\psi)$, it must be the case that $p' \notin \alpha$, $\mathit{leaf}(\psi) \notin \alpha$, and $p' \in \alpha^+$. This by the definition of α^+ means that there is some $v \in \alpha$ with $p' \equiv_M v$. We put $\mathit{ext}_X(\psi) = v$.

We apply Lemma 4.3 to X and ext_X, which gives us a partial run U of $\mathcal{A}$ on w with $X \preceq_{\mathit{ext}_X} U$. We will check that U satisfies the statement of the stronger variant of the lemma. We will first prove that $T \preceq^M_{\alpha^+\Rightarrow\alpha} U$. For each $\tau \in \mathit{branches}(U)$, there is $\psi \in \mathit{branches}(X)$ with $\psi \preceq_{\mathit{ext}_X} \tau$. We will derive that there is some $\rho \in \mathit{branches}(T)$ with $\rho \preceq^M_{\alpha^+\Rightarrow\alpha} \tau$. The argument depends on properties of ψ. Particularly, we have the following three cases.

1. If $\psi \in \mathit{Good}_W$, then there is some $\rho \in T$ with $\rho \preceq^M_{\alpha^+\Rightarrow\alpha} \psi$. Recall that $\mathit{ext}_X(\psi) = \mathit{leaf}(\psi)$ in this case. Thus, by the definitions of $\preceq^M_{\alpha^+\Rightarrow\alpha}$ and $\preceq_{\mathit{ext}_X}$, we have $\rho \preceq_{\alpha^+\Rightarrow\alpha} \psi \preceq_\alpha \tau$ and $\mathit{leaf}(\rho) \preceq_M \mathit{leaf}(\psi) \preceq_F \mathit{leaf}(\tau)$, which gives $\rho \preceq_{\alpha^+\Rightarrow\alpha} \tau$ and $\mathit{leaf}(\rho) \preceq_M \mathit{leaf}(\tau)$, i.e., $\rho \preceq^M_{\alpha^+\Rightarrow\alpha} \tau$.

2. If $\psi \in \mathit{New}_X$ and there is some $p \in P$ with $p \preceq_F \mathit{leaf}(\psi)$ and $p \preceq_{\alpha^+\Rightarrow\alpha} \mathit{leaf}(\psi)$, then by the definition of ext_X, $\mathit{ext}_X(\psi) = \mathit{leaf}(\psi)$. Recall that as $\psi^{|\psi|-1} \in \mathit{Bad}_W$, $\pi \preceq_{\alpha^+\Rightarrow\alpha} \psi^{|\psi|-1}$. Therefore, also $\pi p \preceq_{\alpha^+\Rightarrow\alpha} \psi$. By the definition of $\preceq_{\mathit{ext}_X}$, we have that $\psi \preceq_\alpha \tau$ and $\mathit{leaf}(\psi) \preceq_F \mathit{leaf}(\tau)$. Finally, $\pi p \preceq_{\alpha^+\Rightarrow\alpha} \psi \preceq_\alpha \tau$ and $p \preceq_F \mathit{leaf}(\psi) \preceq_F \mathit{leaf}(\tau)$ together imply that $\pi p \preceq^M_{\alpha^+\Rightarrow\alpha} \tau$.

3. If $\psi \in \mathit{New}_X$ and there is no $p \in P$ with $p \preceq_F \mathit{leaf}(\psi)$ and $p \preceq_{\alpha^+\Rightarrow\alpha} \mathit{leaf}(\psi)$, then by the definition of ext_X, there are $p' \in P$ with $p' \preceq_F \mathit{leaf}(\psi)$ and $v \in \alpha$ with $v \equiv_M p'$ such that $\mathit{ext}_X(\psi) = v$. By $\psi \preceq_{\mathit{ext}_X} \tau$, we have $\psi \preceq_\alpha \tau$ and $v \preceq_F \mathit{leaf}(\tau)$. Thus, since $\preceq_M$ is forward extensible, $p' \equiv_M v \preceq_F \mathit{leaf}(\tau)$ gives $p' \preceq_M \mathit{leaf}(\psi)$. As $\preceq_F \subseteq \preceq_\alpha$, we have that $\mathit{leaf}(\tau) \in \alpha$ and thus $p' \preceq_{\alpha^+\Rightarrow\alpha} \mathit{leaf}(\tau)$. As $\psi^{|\psi|-1} \in \mathit{Bad}_W$, we have that $\pi \preceq_{\alpha^+\Rightarrow\alpha} \psi^{|\psi|-1}$. Together with $\psi \preceq_\alpha \tau$, this gives $\pi p' \preceq_{\alpha^+\Rightarrow\alpha} \tau$. Therefore, $\pi p' \preceq^M_{\alpha^+\Rightarrow\alpha} \tau$.

Since the above three cases cover all possible variants of ψ and thus all branches of U, we have proven that $T \preceq^M_{\alpha^+ \Rightarrow \alpha} U$. Finally, it is easy to show that $root(T) \preceq_B root(U)$ since $\preceq_B$ is transitive and we know that $root(T) = root(T') \preceq_B root(V) \preceq_B root(W) = root(X) \preceq_B root(U)$. We have verified that the constructed partial run U satisfies the statement of the stronger variant of the lemma, which concludes the proof. □

With Lemma 4.6 in hand, we can prove that for each accepting run of $\mathcal{A}^+$ on a word w, there is an accepting run of $\mathcal{A}$ on w. This requires to carry Lemma 4.6 from finite partial runs to full infinite runs.

Lemma 4.7. *A run T of $\mathcal{A}$ with $root(T) = \iota$ is accepting if and only if for every $\pi \in T$, there exists a constant $k_\pi \in \mathbb{N}$ such that every ψ with $\pi\psi \in T$ and $|\psi| \geq k$ contains an accepting state.*

Proof. *(if)* For every $\pi \in branches(T)$, there is an infinite sequence of $k_0, k_1 \ldots$ such that:

- $k_0 = 0$ and
- for all $i \in \mathbb{N}$, $k_i = k_{i-1} + k_{\pi^n}$ where $n = k_{i-1} + 1$.

For all $i \in \mathbb{N}$, every segment of π between $k_{i-1} + 1$ and k_i contains an accepting state, therefore π contains infinitely many accepting states.

(only if) By contradiction. Suppose that there is $\pi \in T$ for which there is no k_π. We will show that in this case, there must be $\psi \in Q^\omega$ such that $\pi\psi \in branches(T)$ and ψ does not contain an accepting state (which contradicts the assumption that T is accepting).

We will give a procedure which returns ψ^i for each $i \in \mathbb{N}$ (based on the knowledge of ψ^{i-1}). For each $i \in \mathbb{N}^0$, we will keep the invariant that for $\pi\psi^i$, $k_{\pi\psi^i}$ does not exist and that ψ^i does not contain an accepting state. Since $\psi^0 = \epsilon$, the invariant holds for $i = 0$.

Let the invariant hold for $i-1, i \in \mathbb{N}$, and suppose that we have already constructed ψ^{i-1}. Denote P the subset of $succ_T(\pi\psi^{i-1})$ containing nonaccepting states. P must be nonempty, because if all the states from $succ_T(\pi\psi^{i-1})$ were accepting, $k_{\pi\psi^{i-1}}$ would equal 1, violating the invariant for $i-1$. Then, there must be a state $q \in P$ such that $k_{\pi\psi^{i-1}q}$ does not exist, since otherwise we could put $k_{\pi\psi^{i-1}} = \max\{k_{\pi\psi^{i-1}p} \mid p \in P\} + 1$, which would also violate the invariant for $i-1$. We choose q as the continuation and put $\psi^i = \psi^{i-1}q$. Observe that this choice satisfied the invariant for i.

We have shown that for every $i \in \mathbb{N}$, we can construct the ith prefix ψ^i of ψ that does not contain an accepting state. Therefore, the whole infinite path ψ does not contain an accepting state, and the branch $\pi\psi$ of T does not contain infinitely many accepting states. This contradicts the assumption that T is accepting. □

Lemma 4.8. *For every accepting run T of $\mathcal{A}^+$ a word $w \in \Sigma^\omega$, there exists an accepting run U of $\mathcal{A}$ on w.*

Proof. For a tree X over Q, let $X(i) = \{\pi \in X \mid |\pi| \leq i\}$ be the ith prefix of X ($X(0) = \emptyset$). ¿From Lemma 4.6, for each $i \in \mathbb{N}$, there is a partial run U_i of $\mathcal{A}$ on w such that $T(i) \preceq_{\alpha^+ \Rightarrow \alpha} U_i$ and $\mathit{root}(T(i)) \preceq_B \mathit{root}(U_i)$. As $\preceq_B \subseteq \preceq_\iota$, $\mathit{root}(U_i) = \iota$. Note that for all $\pi \in \mathit{branches}(U_i)$, $|\pi|$ equals i, because only paths of the same length can be related by $\preceq_{\alpha^+ \Rightarrow \alpha}$. Denote $\mathbb{U}^\infty = \{U_1, U_2, \ldots\}$. $\mathbb{U}^\infty$ is an infinite set that for each $k \in \mathbb{N}$ contains a partial run U_k of $\mathcal{A}$ with all the branches of the length k. We will use $\mathbb{U}^\infty$ to construct the infinite accepting run U.

Observe that for any infinite set $\mathbb{V}^\infty$ of partial runs of $\mathcal{A}$ and for any $i \in \mathbb{N}$, there has to be at least one partial run W of $\mathcal{A}$ such that for infinitely many $V \in \mathbb{V}^\infty$, $W = V(i)$. The reason is that for any $i \in \mathbb{N}$, there is obviously only finitely many of possible partial runs of the height i that $\mathcal{A}$ can generate.

We prove the existence of U by giving a procedure, which for every $k \in \mathbb{N}$ gives the kth prefix $U(k)$ of U.

- Let $\mathbb{U}_0^\infty = \mathbb{U}^\infty$ and let $U(0) = \emptyset$.
- For every $k \in \mathbb{N}$, $U(k)$ is derived from $U(k-1)$ as follows. Let $\mathbb{U}_k^\infty \subseteq \mathbb{U}^\infty$ be defined as the set such that for all $i \in \mathbb{N}$, $U_i \in \mathbb{U}_k^\infty$ iff $U(k-1) = U_i(k-1)$. In other words, $\mathbb{U}_k^\infty$ is the subset of $\mathbb{U}^\infty$ of the partial runs with the ith prefix equal to $U(k-1)$. Then, $U(k) = U_n(k)$ for some $n \geq k$ such that $U_n \in \mathbb{U}_k^\infty$ and there is infinitely many $m \in \mathbb{N}$ such that $U_m \in \mathbb{U}_k^\infty$ and $U_n(k) = U_m(k)$. In other words, $U(k)$ is a tree that appears as the kth prefix of infinitely many partial runs in $\mathbb{U}_k^\infty$.

To see that this construction is well defined, observe that:

- $\mathbb{U}_0^\infty$ is infinite, and
- for all $k \in \mathbb{N}$, if $\mathbb{U}_{k-1}^\infty$ is infinite, then $U(k-1)$ is defined and $\mathbb{U}_k^\infty$ is infinite.

Thus, $U(k)$ is well defined for every $k \in \mathbb{N}$ and U is a run of $\mathcal{A}$.

It remains prove that U is accepting. We will show that for every $\pi \in U$, there is $k_\pi \in \mathbb{N}$ such that every ψ with $\pi\psi \in T$ and $|\psi| \geq k$ contains an accepting state. By Lemma 4.7, it will follow that U is accepting.

Let us choose arbitrary $\pi \in U$. Let $n = |\pi|$. By Lemma 4.7, for every $\pi' \in \mathit{branches}(T_n)$, there is $k_{\pi'} \in \mathbb{N}$ such that every ψ' with $\pi'\psi' \in T$ and $|\psi'| \geq k_{\pi'}$ contains an accepting state. Let $k = \max\{k_{\pi'} \mid \pi' \in \mathit{branches}(T(n))\}$. By the construction of U, $T(n+k) \preceq_{\alpha^+ \Rightarrow \alpha} U(n+k)$. This implies that for every $\pi'' \in \mathit{branches}(U(n))$, every ψ'' with $\pi''\psi'' \in T$ and $|\psi''| \geq k$ contains an accepting state. As π in $\mathit{branches}(U(n))$, we can put $k_\pi = k$ and we are done. □

Theorem 1. $\mathcal{L}(\mathcal{A}^+) = \mathcal{L}(\mathcal{A})$.

Proof. The inclusion $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}^+)$ is obvious as $\mathcal{L}(\mathcal{A}^+)$ has richer both transition function and the set of accepting states. The inclusion $\mathcal{L}(\mathcal{A}^+) \subseteq \mathcal{L}(\mathcal{A})$ follows immediately from Lemma 4.8. □

Corollary 1. *Quotienting with mediated equivalence preserves the language.*

5. Computing the Relations

In this section, we describe algorithms for computing forward and backward simulation for ABA, and mediated preorder. For forward simulation, we use an algorithm from [10], for backward simulation, we present an algorithm based on a translation to an LTS simulation problem similar to the one from [7] for computing upward simulations over tree automata. The mediated preorder is then computed by a simple procedure that we also sketch below. For the mediated preorder to be useful for simplification via computing the quotient, we also need to remove ambiguity before we start computing the backward simulation. This can be done by a simple procedure presented in this section too. For the rest of the section, we fix an ABA $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$.

Forward Simulation. The algorithm for computing the maximal forward simulation $\preceq_F$ on $\mathcal{A}$ can be found in Fritz and Wilke's work [10] (it is called direct simulation in their paper). They reduce the problem of computing the maximal forward simulation to a simulation game. Although Fritz and Wilke use a slightly different definition of ABA, we can easily translate an ABA with n states and m transitions to their notion of ABA with $\mathcal{O}(n+m)$ states and $\mathcal{O}(nm)$ transitions, and then use their algorithm to compute $\preceq_F$. The time complexity of the above procedure is $\mathcal{O}(nm^2)$.

Removing Ambiguity. As we have argued in Section 4.1, $\mathcal{A}$ needs to be $\preceq_F$-unambiguous for mediated minimisation. Here, we describe how to modify $\mathcal{A}$ to make it $\preceq_F$-unambiguous. The modification neither changes the language of $\mathcal{A}$ nor the forward simulation relation $\preceq_F$, therefore we do not need to recompute the forward simulation again for the modified automaton.

The procedure for removing ambiguity is simple. For every transition $p \xrightarrow{a} P$ with $P = \{p_1, \ldots, p_k\}$ and for each $i \in \{1, \ldots, k\}$, we check if there exists some $i < j \leq k$ such that $p_j \preceq_F p_i$. If there is one, remove p_i from P. The time complexity of this procedure is obviously in $\mathcal{O}(n^2 m)$.

We note that an alternative way is to work with the quotient wrt. forward simulation equivalence. This approach also does not change forward simulation (two states of the quotient are related by forward simulation iff they are related by $\preceq_F^{\forall\exists}$ in the original ABA) and hence it does not have to be recomputed.

Mediated Preorder. Here we explain how to compute the mediated preorder $\preceq_M$ of $\mathcal{A}$ from $\preceq_F$ and $\preceq_B$. It is proved in [7] that $\preceq_M$ equals the maximal relation $R \subseteq \preceq_F \circ \preceq_B^{-1}$ satisfying $x \; R \; y \preceq_F z \implies x \; (\preceq_F \circ \preceq_B^{-1}) \; z$. Based on the result, we can obtain the mediated preorder by the following procedure. Initially, let $\preceq_M \; = \; \preceq_F \circ \preceq_B^{-1}$. For all $(p, q) \in \preceq_M$, if there exists some $(q, r) \in \preceq_F$ such that $(p, r) \notin \preceq_F \circ \preceq_B^{-1}$, remove (p, q) from $\preceq_M$. A naïve implementation of this simple procedure has time complexity $\mathcal{O}(n^3)$.

5.1. *Computing Backward Simulation*

Our algorithm for computing backward simulation is inspired by the algorithms for computing tree automata simulations from [7]—we translate the

A transition in $\mathcal{A}$ Transitions in $\mathcal{A}^{\odot}$

$$\begin{tikzcd}
 & & p_1^{\odot} \arrow[r, "a"] & (p,a,\{p_2,p_3\})^{\odot} \arrow[rd, "a"] & \\
p \xrightarrow{a} \{p_1,p_2,p_3\} & \Rightarrow & p_2^{\odot} \arrow[r, "a"] & (p,a,\{p_1,p_3\})^{\odot} \arrow[r, "a"] & p^{\odot} \\
 & & p_3^{\odot} \arrow[r, "a"] & (p,a,\{p_1,p_2\})^{\odot} \arrow[ru, "a"'] &
\end{tikzcd}$$

Figure 4: An example of the reduction from an ABA transition to LTS transitions

problem of computing the maximal backward simulation on $\mathcal{A}$ to a problem of computing the maximal simulation on a labelled transition system.

Computing Simulation on Labeled Transition Systems. Let $T = (S, \mathcal{L}, \rightarrow)$ be a finite *labeled transition system (LTS)*, where S is a finite set of states, $\mathcal{L}$ is a finite set of labels, and $\rightarrow \subseteq S \times \mathcal{L} \times S$ is a transition relation. A *simulation* on T is a binary relation $\preceq_L$ on S such that if $q \preceq_L r$ and $(q, a, q') \in \rightarrow$, then there is an r' with $(r, a, r') \in \rightarrow$ and $q' \preceq_L r'$.

An instance of the problem of computing the maximal LTS simulation is given by an LTS $T = (S, \mathcal{L}, \rightarrow)$ and an *initial* preorder $I \subseteq S \times S$. The task is to find the unique maximal simulation on T included in I. An algorithm for computing the maximal simulation $\preceq^I$ on the LTS T included in I with time complexity $\mathcal{O}(|\mathcal{L}| \cdot |S|^2 + |S| \cdot |\rightarrow|)$ and space complexity $\mathcal{O}(|\mathcal{L}| \cdot |S|^2)$ can be found in [7].

Computing Backward Simulation via a Reduction to LTS. We now describe the reduction of the problem of computing the maximal backward simulation on $\mathcal{A}$ to the problem of computing a simulation on an LTS. In order to simplify the explanation of the reduction, we first define the notion of an *environment*, which is a tuple of the form $(p, a, P \setminus \{p'\})$ obtained by removing a state $p' \in P$ from the transition $p \xrightarrow{a} P$ of $\mathcal{A}$. Intuitively, an environment records the neighbours of the removed state p' in the transition $p \xrightarrow{a} P$. We denote the set of all environments of $\mathcal{A}$ by $\mathit{Env}(\mathcal{A})$. Formally, we define the LTS $\mathcal{A}^{\odot} = (\Sigma, Q^{\odot}, \Delta^{\odot})$ as follows:

- $Q^{\odot} = \{q^{\odot} \mid q \in \ Q\} \cup \{(p, a, P)^{\odot} \mid (p, a, P) \in \mathit{Env}(\mathcal{A})\}$.
- $\Delta^{\odot} = \{(p, a, P \setminus \{p'\})^{\odot} \xrightarrow{a} p^{\odot}, p'^{\odot} \xrightarrow{a} (p, a, P \setminus \{p'\})^{\odot} \mid P \in \delta(p, a), p' \in P\}$.

An example of the reduction is given in Figure 4. The goal of this reduction is to obtain a simulation relation on $\mathcal{A}^{\odot}$ with the following property: $p^{\odot}$ is simulated by $q^{\odot}$ in $\mathcal{A}^{\odot}$ iff $p \preceq_B q$ in $\mathcal{A}$. However, the maximal simulation on $\mathcal{A}^{\odot}$ is not sufficient to achieve this goal. Some essential conditions for backward simulation (e.g., $p \preceq_B q \Longrightarrow p \preceq_\alpha q$) are missing in $A^{\odot}$. This can be fixed by defining a proper initial preorder I.

Formally, we let $I = \{(q_1^{\odot}, q_2^{\odot}) \mid q_1 \preceq_\iota q_2 \wedge q_1 \preceq_\alpha q_2\} \cup \{((p, a, P)^{\odot}, (r, a, R)^{\odot}) \mid P \preceq_F^{\forall\exists} R\}$. Observe that I is a preorder. Recall that according to the definition of the backward simulation, $p \preceq_B r$ implies that (1) $p \preceq_\iota r$, (2) $p \preceq_\alpha r$, and (3) for all transitions $q \xrightarrow{a} P\cup\{p\}, p \notin P$, there exists a transition $s \xrightarrow{a} R\cup\{r\}, r \notin R$ such that $q \preceq_B s$ and $P \preceq_F^{\forall\exists} R$. The set $\{(q_1^\odot, q_2^\odot) \mid q_1 \preceq_\iota q_2 \wedge q_1 \preceq_\alpha q_2\}$ encodes conditions (1) and (2) required by the backward simulation, while the set $\{((p,a,P)^\odot, (r,a,R)^\odot) \mid P \preceq_F^{\forall\exists} R\}$ encodes condition (3). A simulation relation $\preceq^I$ can be computed using the aforementioned procedure with LTS $\mathcal{A}^\odot$ and the *initial* preorder I. The following theorem shows the correctness of our approach to computing backward simulation.

Theorem 2. *For all $q, r \in Q$, we have $q \preceq_B r$ iff $q^\odot \preceq^I r^\odot$.*

Proof. *(if)* We define $\preceq$ to be a binary relation on Q such that $p \preceq r$ iff $p^\odot \preceq^I r^\odot$. We show that $\preceq$ is a backward simulation on Q which immediately implies the result.

Suppose that $p \preceq r$ and $p' \xrightarrow{a} \{p\} \cup P$ where $p \notin P$ is a transition of $\mathcal{A}$. Since $p \preceq r$, we know that $p^\odot \preceq^I r^\odot$; and since $p' \xrightarrow{a} \{p\} \cup P$ is a transition of $\mathcal{A}$, we know by definition of $\mathcal{A}^\odot$ that $p^\odot \xrightarrow{a} (p',a,P)^\odot$ and $(p',a,P)^\odot \xrightarrow{a} p'^\odot$ are transitions in $\mathcal{A}^\odot$. Since $\preceq^I$ is a simulation, we can find two transitions $r^\odot \xrightarrow{a} (r',a,R)^\odot$ and $(r',a,R)^\odot \xrightarrow{a} r'^\odot$ in $\mathcal{A}^\odot$ with $(p',a,P)^\odot \preceq^I (r',a,R)^\odot$ and $p'^\odot \preceq^I r'^\odot$. ¿From $p'^\odot \preceq^I r'^\odot$, $(p',a,P)^\odot \preceq^I (r',a,R)^\odot$, and the definition of the initial preorder I, we have $p' \preceq r'$ and $P \preceq_F^{\forall\exists} R$. It follows that $\preceq$ is in fact a backward simulation parametrised by $\preceq_F$.

(only if) Define $\preceq_\odot$ as a binary relation on $Q^\odot$ such that $p^\odot \preceq_\odot r^\odot$ iff $p \preceq_B r$ and $(p,a,P)^\odot \preceq_\odot (r,a,R)^\odot$ iff $P \preceq_F^{\forall\exists} R$ and $p \preceq_B r$. By definition, $\preceq_\odot \subseteq I$. We show that $\preceq_\odot$ is a simulation on $Q^\odot$ which immediately implies the result. In the proof, we consider two sorts of states in $\mathcal{A}^\odot$; namely those corresponding to states and those corresponding to "environments".

Suppose that $p^\odot \preceq_\odot r^\odot$ and the transition $p^\odot \xrightarrow{a} (p',a,P)^\odot$ is in $\mathcal{A}^\odot$. Since $p^\odot \preceq_\odot r^\odot$, we know that $p \preceq_B r$. From the transition $p^\odot \xrightarrow{a} (p',a,P)^\odot$ and by definition of $\mathcal{A}^\odot$, $p' \xrightarrow{a} P \cup \{p\}$ is a transition in $\mathcal{A}$. Since $p \preceq_B r$, there exists a transition $r' \xrightarrow{a} R \cup \{r\}$ in $\mathcal{A}$ such that $p' \preceq_B r'$ and $P \preceq_F^{\forall\exists} R$. It follows that there exists a transition $r^\odot \xrightarrow{a} (r',a,R)^\odot$ in $\mathcal{A}^\odot$ such that $(p',a,P)^\odot \preceq_\odot (r',a,R)^\odot$.

Suppose that $(p,a,P)^\odot \preceq_\odot (r,a,R)^\odot$ and the transition $(p,a,P)^\odot \xrightarrow{a} p^\odot$ is in $\mathcal{A}^\odot$. Since $(p,a,P)^\odot \preceq_\odot (r,a,R)^\odot$, we know that $P \preceq_F^{\forall\exists} R$ and $p \preceq_B r$. By definition of $\mathcal{A}^\odot$, the transition $(r,a,R)^\odot \xrightarrow{a} r^\odot$ is in $\mathcal{A}^\odot$. Since $p \preceq_B r$, we have $p^\odot \preceq_\odot r^\odot$. Together we have there exists a transition $(r,a,R)^\odot \xrightarrow{a} r^\odot$ in $\mathcal{A}^\odot$ such that $p^\odot \preceq_\odot r^\odot$. It follows that $\preceq_\odot$ is a simulation on $Q^\odot$. □

5.2. Complexity of Computing Backward Simulation

The complexity comes from three parts of the procedure: (1) compiling $\mathcal{A}$ into its corresponding LTS $\mathcal{A}^\odot$, (2) computing the initial preorder I, and running the algorithm from [7] for computing the LTS simulation relation.

Let now n and m be the number of states and transitions in $\mathcal{A}$, respectively. The LTS $\mathcal{A}^\odot$ has at most $nm+n$ states and $2nm$ transitions. It follows that

Part (3) has both time complexity and space complexity $\mathcal{O}(|\Sigma|n^2m^2)$. As we will show, among the three parts, Part (3) has the highest time and space complexity and therefore computing backward simulation also has time and space complexity $\mathcal{O}(|\Sigma|n^2m^2)$. Under our definition of ABA, every state has at least one outgoing transition for each symbol in Σ. It follows that $m \geq |\Sigma|n$. Therefore, we can also say that the procedure for computing the maximal backward simulation has time and space complexity $\mathcal{O}(nm^3)$.

Initial Preorder for Computing Backward Simulation. Let us recall that the preorder I is the union of two components: $\{(q_1^{\odot}, q_2^{\odot}) \mid q_1 \preceq_\iota q_2 \wedge q_1 \preceq_\alpha q_2\}$ and $\{((p,a,P)^{\odot},(r,a,R)^{\odot}) \mid \forall r_j \in R \exists p_i \in P : p_i \preceq_F r_j\}$. It is trivial that the first set can be computed by an algorithm with time complexity $\mathcal{O}(n^2)$. However, a naïve algorithm (pairwise comparison of all different environments in $\mathit{Env}(\mathcal{A})$) for computing the second set has time complexity $\mathcal{O}(n^4m^2)$. Here, we will describe a more efficient algorithm, which allows the computation of I in time $\mathcal{O}(n^2m^2)$ and space $\mathcal{O}(n)$.

The main idea of the algorithm is the following. For each pair of transitions of $\mathcal{A}$, it computes all the pairs of environments that arise from them (by deleting a right-hand side state) and adds them to I at once, reusing a lot of information that a naïve algorithm would compute repeatedly for each pair of environments. For a fixed pair of transitions, this procedure has time complexity $\mathcal{O}(n^2)$ and space complexity $\mathcal{O}(n)$. Because $\mathcal{A}$ has at most m^2 different pairs of transitions and the $\mathcal{O}(n)$ memory needed for the data structures for one pair of transitions can be reused for the other pairs, the second component of I can be computed in time $\mathcal{O}(n^2m^2)$ and space $\mathcal{O}(n)$.

We now explain how to efficiently compute all pairs of environments that arise from a given pair of transitions and that are related by I. Let us fix transitions $p \xrightarrow{a} P$ and $r \xrightarrow{a} R$. We will maintain a function $\beta : R \to \{T, F\} \cup P$ such that:

$$\beta(r') = \begin{cases} T & \text{if at least two states in } P \text{ are forward smaller than } r'. \\ F & \text{if no state in } P \text{ is forward smaller than } r'. \\ p' & \text{if } p' \text{ is the only state in } P \text{ such that } p' \preceq_F r'. \end{cases}$$

The function β can be computed by lines 1-4 of Algorithm 1 in time $\mathcal{O}(n^2)$ and space $\mathcal{O}(n)$. Let us consider a pair of states $((p,a,P\setminus\{p'\})^{\odot},(r,a,R\setminus\{r'\})^{\odot})$ in $\mathcal{A}^{\odot}$. This pair can be added to I if and only if the following two conditions hold:

1. $\forall \hat{r} \in (R \setminus \{r'\}).\beta(\hat{r}) \neq F$.

2. $\forall \hat{r} \in (R \setminus \{r'\}).\beta(\hat{r}) \neq p'$.

The algorithm first pre-processes $p \xrightarrow{a} P$ and $r \xrightarrow{a} R$, computing certain information that will allow us to check the two conditions in constant time for every pair of environments arising from the two transitions.

The pre-processing needed for efficient checking of Condition (1) is the following. We define $\hat{r} \in R$ as the `KeyState` if $\hat{r}$ is the only one state in R such

Algorithm 1: *Add Pairs of States to I*

```
Input: Two transitions $p \xrightarrow{a} P$ and $r \xrightarrow{a} R$ in $\mathcal{A}$.
/* Computing function $\beta$                                        */
forall $r' \in R$ do  $\beta(r') := F$;
forall $p' \in P, r' \in R$ do
    if $p' \preceq_F r'$ then
        if $\beta(r') = F$ then  $\beta(r') := p'$;
        else  $\beta(r') := T$;
/* Preprocessing for Condition (1) (computing KeyState)              */
forall $r' \in R$ do  if $\beta(r') = F$ then
    if there is no KeyState then Let $r'$ be the KeyState;
    else Terminate the algorithm;
/* Preprocessing for Condition (2) (computing function $\gamma$)     */
forall $p' \in P$ do  $\gamma(p') := F$;
forall $r' \in R$ do  if $\beta(r') \notin \{T, F\}$ then
    if $\gamma(\beta(r')) = F$ then  $\gamma(\beta(r')) := r'$;
    else  $\gamma(\beta(r')) := T$;
/* main loop                                                         */
forall $p' \in P, r' \in R$ do
    if there is no KeyState or $r'$ is the KeyState then
        if $\gamma(p') \in \{F, r'\}$ then add $((p, a, P \setminus \{p'\})^{\odot}, (r, a, R \setminus \{r'\})^{\odot})$ to $I$
```

that $\beta(\hat{r}) = F$. Given the function β, the `KeyState` can be found efficiently (with time complexity $\mathcal{O}(n)$ and space complexity $\mathcal{O}(1)$) by scanning through R and

- if there exist two distinct states $r_1, r_2 \in R$ such that $\beta(r_1) = \beta(r_2) = F$, the algorithm terminates immediately because it follows that none of the pairs of environments generated from the given pair of transitions satisfies the requirement of I;

- if there exists only one state such that β maps it to F, let it be the `KeyState`.

Then we have that Condition (1) is satisfied if there is no `KeyState` or r' is the `KeyState`.

For efficient checking of Condition (2), we maintain a function $\gamma : P \to \{T, F\} \cup R$ such that

$$\gamma(p') = \begin{cases} F & \text{if } \beta^{-1}(p') = \emptyset \\ r' & \text{if } \beta^{-1}(p') = \{r'\} \\ T & \text{otherwise.} \end{cases}$$

The function γ can be found in time $\mathcal{O}(n^2)$ and space $\mathcal{O}(n)$ by scanning once through β for each element of P. With the function γ, Condition (2) can easily be verified by checking if $\gamma(p') \in \{F, r'\}$, which means that for all the states $\hat{r}$ in $R\backslash\{r'\}$, there is some state $\hat{p}$ different from p' such that $\hat{p} \preceq_F \hat{r}$. In Algorithm 1, we first find out the `KeyState` if there is one and compute the function γ from β. Then in the main loop, for each pair of states $((p, a, P\setminus\{p'\})^{\odot}, (r, a, R\setminus\{r'\})^{\odot})$, we check if it belongs to I by verifying the Conditions (1) and (2). Since it is easy to see that Algorithm 1 has time complexity $\mathcal{O}(n^2)$ and space complexity $\mathcal{O}(n)$ (not taking into account the space needed for I itself), we can conclude that the initial preorder I can be computed in time $\mathcal{O}(n^2m^2)$ and space $\mathcal{O}(m^2)$ (encoding of I). This leads to the following theorem that summarises the complexity of computing the backward simulation.

Theorem 3. *Maximal backward simulation parametrised by a given transitive and reflexive forward simulation can be computed with both time and space complexity* $\mathcal{O}(|\Sigma|n^2m^2) \subseteq \mathcal{O}(nm^3)$.

6. Experimental Results

In this section, we evaluate the performance of ABA mediated minimisation by applying it to (1) the algorithm proposed by Gastin and Oddoux [5] for translating linear temporal logic (LTL) formulae to nondeterministic Büchi automata (NBA) and (2) the algorithm proposed by Vardi and Kupferman [2] for complementing NBA.

6.1. Translating LTL Formulae to Nondeterministic Büchi Automata

In the algorithm of [5] for translating LTL formulae to NBA, ABA are used as an intermediate representation. To be more specific, the translation consists of three steps: (1) an LTL formula is translated to an equivalent very weak ABA with co-Büchi acceptance conditions, (2) the ABA is subsequently translated to an equivalent generalised nondeterministic Büchi automaton (GBA), and then (3) the GBA is translated to an equivalent NBA. The resulting NBA is typically used as a formal specification against which some system is verified by model checking, and thus its size is crucial for the efficiency of the verification process. Below, we provide experimental evidence that our algorithm for reducing the size of ABA can be useful for reducing the size of NBA produced by the translation algorithm and hence for the efficiency of the entire verification process.

Our experiments were carried out as follows. Three sets of 200 random LTL formulae (with 1, 2, and 3 atomic propositions, respectively) were generated by the GOAL [11] tool and then used as inputs of the translation experiments. For each input LTL formula, we (1) translated it to a very weak ABA with co-Büchi acceptance conditions, (2) processed the ABA using one of the options mentioned below, and (3) translated it back to an equivalent NBA. In the translation, we considered the following three options: (1) **Original:** we kept the ABA as it was. (2) **Mediated:** we minimised the ABA with mediated

Table 1: Combining minimisation with LTL to Büchi translation

	$\lvert AP \rvert$	ABA		NBA		Time (ms)
		St.	Tr.	St.	Tr.	
Original	1	4.39	6.94	4.18	9.13	9.1
Mediated		3.81	6.3	3.94	8.26	42.9
Forward		4.26	6.52	3.94	7.75	31.5
Original	2	6.5	13.29	8.21	44.99	27.21
Mediated		5.09	11.52	6.76	26.63	101.4
Forward		6.16	12.19	7.11	29.54	67.6
Original	3	18.86	49.49	33	359.3	129.91
Mediated		12.34	42.4	22.9	159.8	12339.34
Forward		17.78	45.49	26.7	188.5	5814.78

equivalence. (3) **Forward:** we minimised the ABA with forward equivalence[4]. The results are given in Table 1.

In Table 1, the columns "ABA" and "NBA" give the average numbers of states and transitions of the intermediate ABA and the resulting NBA. The column "Time(ms)" is the average execution time of the translation in milliseconds. From the table, we can see that minimisation by mediated equivalence can result in smallest final NBA, with the cost of additional execution time. However, in most of the cases, the execution time of the translation takes a very small portion of the entire verification task only.

6.2. Complementing Nondeterministic Büchi Automata

In the algorithm [2] for complementing NBA, ABA are again used as an intermediate model. To be more specific, the complementation algorithm has two steps: (1) it translates an NBA to an ABA that recognises its complement language, and (2) it translates the ABA back to an equivalent NBA. The second step is an exponential procedure (exponential in the size of the ABA), hence reducing the size of the ABA before the second step usually pays off.

We have performed the following experiments. Three sets of 100 random NBA (of $|\Sigma| = 2$, 4, and 8, respectively) were generated by the GOAL [11] tool and then used as inputs of the complementation experiments. For each input NBA, we first translated it to an ABA that recognised its complement language. The ABA was (1) processed according to one of the options (Original, Mediated, or Forward) described above and then (2) translated back to an equivalent NBA using an exponential procedure[5]. The results are given in Table 2 and Table 3. Table 2 is an overall comparison between the three different options and Table 3 is a more detailed comparison between **Mediated** and **Forward** minimisation.

[4] We kept the ABA very weak by not merging a bigger state with a smaller state according to the partial order defined in the very weak ABA.

[5] For the option "Original", we also used the optimisation suggested in [2] that only takes

Table 2: Combining minimisation with complementation

	|Σ|	NBA		Complemented-NBA		Time (ms)	Timeout (10 min)
		St.	Tr.	St.	Tr.		
Original	2	2.5	3.3	13.7	52.34	4236	0
Mediated				6.73	32.89	3029	0
Forward				9.47	54.91	9548	0
Original	4	3.3	6.0	46.98	350.98	12926.87	5
Mediated				24.42	420.18	1426.84	6
Forward				25.7	310.17	917.76	8
Original	8	4.7	11.9	141.3	1809.29	40634.5	22
Mediated				75.85	2735.3	10267.6	23
Forward				94.12	2957.2	18099.2	26

Table 3: Comparison: *Mediated* vs. *Forward*. We only compare the cases that both approaches finished within the timeout period.

	|Σ|	ABA		Minimised-ABA		Complemented-NBA	
		St.	Tr.	St.	Tr.	St.	Tr.
Forward	2	11.8	39.8	7	26.9	9.47	54.91
	4	20.3	146.7	10.55	93.43	25.7	310.17
	8	36.4	517.4	15.245	275.73	94.12	2957.51
Mediated	2	11.8	39.8	5.66	20.49	6.73	33.89
	4	20.3	146.7	8.46	72.56	20.38	235.93
	8	36.4	517.4	13.59	238	70.54	2429.97

In Table 2, the columns "NBA" and "Complemented-NBA" give the average numbers of states and transitions of the input NBA and the complemented NBA. The column "Time(ms)" is the average execution time in milliseconds. "Timeout" is the number of cases that could not be finished within the timeout period (10 min). Note that in the table, the cases that could not be finished within the timeout period are excluded from the average number. From the table, we can see that minimisation by mediated equivalence can effectively speed up the complementation and also reduce the size of the complemented NBA.

In Table 3, we compare the performance between **Mediated** and **Forward** minimisation in detail. The columns "ABA", "Minimised-ABA", and "Complemented-NBA" give the average numbers of states and transitions of the ABA before minimisation, the ABA after minimisation, and the complemented BA. From the table, we observe that mediated minimisation consistently results in a better reduction than forward minimisation[6].

a consistent subset.

[6] The experimental results given here are different from the preliminary version of the

Some Notes on the Limitations of ABA Minimisation

We also tested mediated minimisation on complementing the hand-optimised Büchi automata in the Büchi store [12]. However, the result is not positive. The main reason is that the simulation-preorder on those hand-optimised automata is extremely sparse. Notice that all simulation equivalent states are already merged on these optimised automata. As a consequence, there is only a very limited space for further state/transition reduction in the produced ABA.

7. Conclusion and Future Work

We have introduced a novel notion of backward simulation for alternating automata. Inspired by our previous work simulation reduction for tree automata, we combined forward and backward simulation to form a coarser relation called mediated preorder and showed that the quotient wrt. mediated equivalence has the same language as the original ABA. Moreover, we developed an efficient algorithm for computing backward simulation and mediated equivalence. Experimental results show that the mediated reduction of ABA noticeably outperforms the reduction based on forward simulation.

In the future, we would like to extend the mediated equivalence by building it on top of even coarser forward simulation relations, e.g., *delayed* or *fair* forward simulation relations [6]. Also, we would like to study the possibility of using mediated preorder to remove redundant transitions (similar to the approaches described in [13]). We believe that the extensions described above can considerably improve the performance of mediated reduction.

References

[1] P. A. Abdulla, Y.-F. Chen, L. Holík, T. Vojnar, Mediating for Reduction (on Minimizing Alternating Büchi Automata), in: Proc. of FSTTCS'09, Vol. 4 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2009, pp. 1–12.

[2] O. Kupferman, M. Y. Vardi, Weak alternating automata are not that weak, ACM Transactions on Computational Logic 2 (3) (2001) 408–429.

[3] M. Y. Vardi, Automata-Theoretic Model Checking Revisited, in: Proc. of VMCAI'07, Vol. 4349 of LNCS, Springer, 2007, pp. 137–150.

paper [1] for two reasons. First, we re-ran the experiments on a different machine with a different set of random examples. Second, in Table 3, we compare only cases for which both approaches finished within the timeout period. If only one of them finished in the timeout period while the other did not, we dropped that case. This excluded some cases when one of the approaches finished with a huge final automaton while the other did not produce anything within the timeout period. The data presented in [1] was heavily dominated by such cases.

[4] A. Farzan, Y.-F. Chen, E. M. Clarke, Y.-K. Tsay, B.-Y. Wang, Extending Automated Compositional Verification to the Full Class of Omega-Regular Languages, in: Proc. of TACAS'08, Vol. 4963 of LNCS, Springer, 2008, pp. 2–17.

[5] P. Gastin, D. Oddoux, Fast LTL to Büchi Automata Translation, in: Proc. of CAV'01, Vol. 2102 of LNCS, Springer, 2001, pp. 53–65.

[6] C. Fritz, T. Wilke, Simulation relations for alternating Büchi automata, Theoretical Computer Science 338 (1-3) (2005) 275–314.

[7] P. A. Abdulla, A. Bouajjani, L. Holík, L. Kaati, T. Vojnar, Computing Simulations over Tree Automata: Efficient Techniques for Reducing Tree Automata, in: Proc. of TACAS'08, Vol. 4963 of LNCS, Springer, 2008, pp. 93–108.

[8] P. A. Abdulla, L. Holík, L. Kaati, T. Vojnar, A Uniform (Bi-)Simulation-Based Framework for Reducing Tree Automata, Electronic Notes in Theoretical Computer Science 251 (2009) 27–48.

[9] S. Gurumurthy, O. Kupferman, F. Somenzi, M. Y. Vardi, On Complementing Nondeterministic Büchi Automata, in: Proc. of CHARME'03, Vol. 2860 of LNCS, Springer, 2003, pp. 96–110.

[10] C. Fritz, T. Wilke, State Space Reductions for Alternating Büchi Automata, in: Proc. of FSTTCS'02, Springer, London, UK, 2002, pp. 157–168.

[11] Y.-K. Tsay, Y.-F. Chen, M.-H. Tsai, K.-N. Wu, W.-C. Chan, GOAL: A Graphical Tool for Manipulating Büchi Automata and Temporal Formulae, in: Proc. of TACAS'07, Vol. 4424 of LNCS, Springer, 2007, pp. 466–471.

[12] Y.-K. Tsay, M.-H. Tsai, J.-S. Chang, Y.-W. Chang, C.-S. Liu, Buchi Store: an open repository of ω-automata, STTT 15 (2) (2013) 109–123.

[13] F. Somenzi, R. Bloem, Efficient Büchi Automata from LTL Formulae, in: Proc. of CAV'00, Vol. 1855 of LNCS, Springer, 2000, pp. 248–263.