LIPIcs Leibniz International Proceedings in Informatics

Mediating for Reduction
(on Minimizing Alternating Büchi Automata)

Parosh A. Abdulla[1], Yu-Fang Chen[1], Lukáš Holík[2], Tomáš Vojnar[2]

[1] University of Uppsala, Sweden,
{parosh,yu-fang.chen}@it.uu.se

[2] FIT, Brno University of Technology, Czech Republic,
{holik,vojnar}@fit.vutbr.cz

ABSTRACT. We propose a new approach for minimizing alternating Büchi automata (ABA). The approach is based on the so called *mediated equivalence* on states of ABA, which is the maximal equivalence contained in the so called *mediated preorder*. Two states p and q can be related by the mediated preorder if there is a *mediator* (mediating state) which forward simulates p and backward simulates q. Under some further conditions, letting a computation on some word jump from q to p (due to they get collapsed) 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 more significant reductions when applied within the process of complementing Büchi automata where ABA are used as an intermediate model.

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 [9]. It is an essential step of the automata-theoretic approach to model checking when the specification is given as a positive Büchi automaton [12] and also learning based model checking for liveness properties [4]. The other important usage of ABA is as the intermediate data structure for translating a linear temporal logic (LTL) specification to an automaton [7].

However, because of the compactness of ABA*, usually the algorithms that work on them are of high complexity. For example, both the complementation and the LTL translation algorithms transform 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 relatively cheaper algorithm) before giving it to the exponential procedure.

In the study of Fritz and Wilke, simulation-based minimization is proven as a very effective tool for reducing the size of ABA [6]. However, they considered only *forward* simulation relations. Inspired by some previous works [1], we believe that *backward* simulation can be used for reducing the size of ABA as well. Unfortunately, as will be explained in Section 3, quotienting wrt. *backward* simulation (i.e., simplify the automaton by collapsing backward simulation equivalent states) does not preserve the language.

*ABA's are exponentially more succinct than the nondeterministic ones.

Foundations of Software Technology and Theoretical Computer Science (Kanpur) 2009.
Editors: Ravi Kannan and K. Narayan Kumar; pp 1–12
Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany.
Digital Object Identifier: 10.4230/LIPIcs/FSTTCS/2009/2302

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 *mediated preorder* that can be used for minimization. The performance of minimizing ABA with *mediated preorder* is evaluated on a large set of experiments. In the experiments, we apply different simulation-based minimization approaches to improve the complementation algorithm of nondeterministic Büchi automata. The experimental results show that the minimization using mediated preorder significantly outperforms the minimization using forward simulation. To be more specific, in average, mediated minimization results in a 30% better reduction in the number of states and 50% better reduction in the number of transitions than forward minimization on the intermediate ABA. Moreover, in the complemented nondeterministic Büchi automata, mediated minimization results in a 100% better reduction in the number of states and 300% better reduction in the number of transitions than forward minimization.

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 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 $|u| = i$). $w^0 = \epsilon$. 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 $p \xrightarrow{a} P$ where $P \in \delta(q, a)$.

A *tree T over Q* is a subset of Q^+ that contains all nonempty prefixes of each one 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 $leaf(\pi q)$ to denote its last element q. 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. The 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 on w*. A run T of $\mathcal{A}$ over 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 $\mathcal{A}$ over 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 $p \xrightarrow{a} \{q\}$.

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 [1, 3]—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 [8] by Gurumurthy et al. In particular, (i) there exists a unique maximal forward simulation $\preceq_F$ on $\mathcal{A}$ which 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) quotienting wrt. $\preceq_F \cap \preceq_F^{-1}$ preserves the language of $\mathcal{A}$.

Backward Simulation. Let $\preceq_F$ be a forward simulation on $\mathcal{A}$. A *backward simulation* on $\mathcal{A}$ parameterized 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 below lemma describes some properties of backward simulation.

LEMMA 1. *For any reflexive and transitive forward simulation $\preceq_F$ on $\mathcal{A}$, there exists a unique maximal backward simulation $\preceq_B$ on $\mathcal{A}$ parameterized by $\preceq_F$ that is reflexive and transitive.*

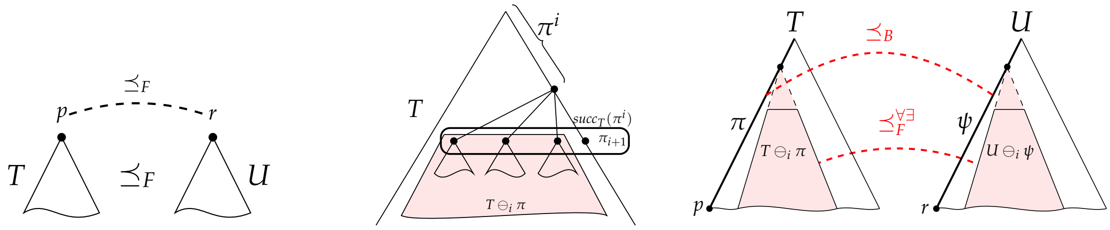
Backward simulation itself cannot be used for quotienting. In [2], we give an example of an automaton, where quotienting using backward simulation does not preserve language. However, in Section 4.1, we show how backward simulation can be used to define a new relation for reducing ABA.

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 $branches(T) \preceq_x^{\forall\exists} branches(U)$. 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 $\preceq_F \subseteq \preceq_\alpha$.

The following two lemmas formulate properties of the simulation relations that we will use in the rest of the paper.

LEMMA 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 $T \preceq_F U$.*

For 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.



(a) Lemma 2 (b) The notation $T \ominus_i \pi$ (c) Lemma 3

Figure 1: Illustration of the lemmas

Formally, 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 $T \ominus_i \pi = \bigcup_{q \in Q^i} branches(T(\pi^i q))$ (notice that if $i = 0$, then $T \ominus_i \pi = \emptyset$).

LEMMA 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 $1 \leq i \leq |\pi|$, $T \ominus_i \pi \preceq_F^{\forall\exists} U \ominus_i \psi$.*

4 Mediated Equivalence and Quotienting

Here we discuss the possibility of an indirect use of backward simulation for simplifying ABA via quotienting. We do not look for a suitable fragment of backward simulation only. Instead, we (1) combine backward and forward simulation to form an equivalence that subsumes both backward and forward simulation equivalence and (2) take a certain fragment of this equivalence, called *mediated equivalence*, that can be used for quotienting.

4.1 The Notion and Intuition of Mediated Equivalence

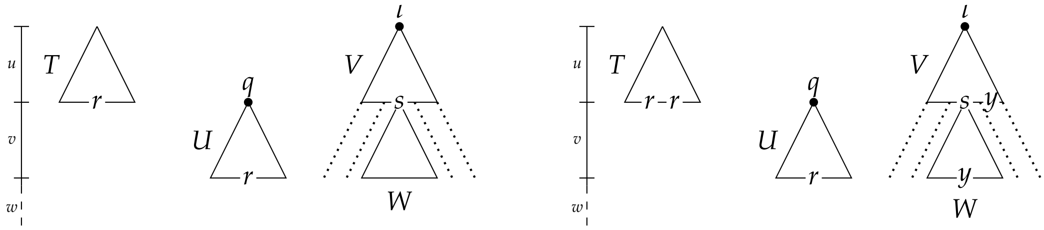
Collapsing states of an automaton wrt. some equivalence allows a run that arrives to some state to *jump* to another equivalent state and continue from there. Alternatively, this can be viewed as *extending* the source state of the jump by the outgoing transitions of the target state†. The equivalence 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 in particular 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‡ 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, the relation $\preceq_F \circ \preceq_B^{-1}$ cannot be directly used as it is not transitive, and taking its symmetric closure would thus not yield an equivalence. We thus have to take some of its *transitive fragments*. This is natural as if the automaton can safely jump from q_1 to q_2 and from q_2 to q_3, it should be able to safely jump from q_1 to q_3 too.

This is, however, still not enough. Not all of the transitive fragment of $\preceq_F \circ \preceq_B^{-1}$ can be used for quotienting. We can only take a fragment $\preceq_M$ that is *forward extensible*, meaning

†The first view is better when explaining the intuition whereas the other is easier to be used in proofs.
‡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: Basic Intuition Behind Mediated Equivalence

that if $q_1 \preceq_M q_2 \preceq_F q_3$, then $q_1 \preceq_M q_3$. The intuitive meaning of this requirement is the following. When a run jumps from r to q, it may be the case that r is again reached later on or it appears in the context of itself (cf. Fig. 2(b)). If r is reached in the continuation of the run from q, the mediated preorder assures that there is some state y in the run continuing from the mediator s that forward simulates r. Similarly, if the context of r contains another occurrence of r, there is some state y in the context of s that forward simulates r. However, this forward simulation is in general guaranteed to hold only when no further jumps are allowed. In order to guarantee a possibility of further simulation, we require that if the computation is allowed to jump from r to q, it is allowed to jump from y to q too.

Finally, 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 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 in [2]). However, in Section 5, we show that $\preceq_F$-ambiguity can be efficiently removed.

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 $\mathcal{A}$ parameterized 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.*[3] There is a unique maximal mediated preorder $\preceq_M$ induced by $\preceq_F$ and $\preceq_B$.*

4.2 Extending Automata According to Mediated Preorder Preserves Language

Quotient Automata versus Extended Automata. We first show that quotienting can be seen as a simpler operation of adding transitions and accepting states. 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 $\preceq$. Let the automaton $\mathcal{A}/\equiv$ be the quotient of $\mathcal{A}$ wrt. $\equiv$ that arises by merging $\equiv$-equivalent states of $\mathcal{A}$, and let $\mathcal{A}^+$ be 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.

The automata $\mathcal{A}/\equiv$ and $\mathcal{A}^+$ are formally defined as follows. Let $Q/\equiv$ denote the quotient 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 $\mathcal{A}^+ = (\Sigma, Q, \delta^+, \iota, \alpha^+)$, where $\alpha^+ = \{p \mid \exists q \in \alpha.\, q \equiv p\}$ 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 $\delta^+(q, a) = \bigcup_{p \in Q \wedge p \preceq q} \delta(p, a)$. It is not difficult to show that $\mathcal{L}(\mathcal{A}/\equiv) \subseteq \mathcal{L}(\mathcal{A}^+)$ [2] (Lemma 8 in [2]). Hence, if adding transitions and accepting states according to $\preceq$ preserves the language, then quotienting according to $\equiv$ preserves the language too.

Language Preservation by Mediated Equivalence. We now give a sketch of the proof that extending automata according to the mediated preorder preserves the language. The full proofs can be found in [2]. For the rest of the section, we fix an ABA $A = (\Sigma, Q, \iota, \delta, \alpha)$, 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 $\mathcal{A}$ parameterized by $\preceq_F$. Let $\preceq_M$ be a mediated preorder induced by $\preceq_F$ and $\preceq_B$, and let $\mathcal{A}^+$ be the automaton extended according to $\preceq_M$. 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 proceed as follows. We first prove Lemma 5, which shows how partial runs of $\mathcal{A}$ with an increased power of their leaves (wrt. $\preceq_F$) can be built incrementally from other runs of $\mathcal{A}$, bridging the gap between $\mathcal{A}$ and $\mathcal{A}^+$. Then we prove Lemma 7 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 carry this result over to infinite runs we get the proof of Theorem 8.

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 formalized in Lemma 5 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 $ext(\pi)$ such that $ext(\pi) \preceq_M leaf(\pi)$.

Let U be a partial run of $\mathcal{A}$ on w. For two branches $\pi \in branches(T)$ and $\psi \in branches(U)$, we say that ψ *strongly covers* π wrt. *ext*, denoted $\pi \preceq_{ext} \psi$, iff $\pi \preceq_\alpha \psi$ and $ext(\pi) \preceq_F leaf(\psi)$. Similarly, we say that ψ *weakly covers* π wrt. *ext*, denoted $\pi \preceq_{\text{w-}ext} \psi$, iff $\pi \preceq_\alpha \psi$ and $ext(\pi) \preceq_M leaf(\psi)$. We extend the concept of covering to partial runs as follows. We write $T \preceq_{ext} U$ (U *strongly covers* T wrt. *ext*) iff $branches(T) \preceq_{ext}^{\forall\exists} branches(U)$ and $root(T) \preceq_B$

$root(U)$. Likewise, we write $T \preceq_{\text{w-}ext} U$ (U *weakly covers* T wrt. ext) iff $branches(T) \preceq^{\forall\exists}_{\text{w-}ext} branches(U)$ and $root(T) \preceq_B root(U)$. Note that we have $\preceq_{ext} \subseteq \preceq_{\text{w-}ext}$ for branches as well for partial runs because $\preceq_F \subseteq \preceq_M$. So, the strong covering implies the weak one.

LEMMA 5. *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 $T \preceq_{ext} U$.*

Proving Lemma 5 is the most intricate part of the proof of Theorem 8. We introduce the concepts used within the proof of Lemma 5 and provide an overview of the proof.

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 $T \preceq_{\text{w-}ext} T$. 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. The last partial run of the chain will cover T strongly. To do this, we need a suitable measure that, for a partial run V of $\mathcal{A}$ on w with $T \preceq_{\text{w-}ext} V$, tells us how strongly V covers T.

To define the measure, we concentrate on branches of V that cause that V does not cover T strongly. These are branches $\psi \in branches(V)$ for which there is no $\pi \in branches(T)$ with $\pi \preceq_{ext} \psi$ (there are only some $\pi \in branches(T)$ with $\pi \preceq_{\text{w-}ext} \psi$). We call them *strict weakly covering branches*. Let $\mathsf{sw}_T(V)$ denote the tree which is the subset of V containing prefixes of strict weakly covering branches of V wrt. T. Note that $T \preceq_{ext} V$ iff V contains no strict weakly covering branches, which is equivalent to $\mathsf{sw}_T(V) = \emptyset$. For 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 $\langle \epsilon, X \rangle = branches(X)$, which is a sequence of length 1. Notice that under the condition that $\tau \notin branches(X)$, $\langle \tau, X \rangle = \emptyset \ldots \emptyset$ implies that $X = \emptyset$[§].

Let $\tau_V \in V \cup \{\epsilon\}$ and $\tau_W \in W \cup \{\epsilon\}$ be such that $\tau_V \notin branches(\mathsf{sw}_T(V))$ and $\tau_W \notin branches(\mathsf{sw}_T(W))$. 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 $root(V) \preceq_B root(W)$ 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 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 sequences of finite sets $S, S' \in (2^Q)^+$, $S \sqsubset S'$ 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 $1 \leq j < k$, $S_j \preceq^{\forall\exists}_F S'_j$. It is not hard to show that the relation $\sqsubset$ is a partial order. Observe that $\sqsubset$ does not allow infinite increasing chains of sequences where the length of the sequences is bounded by some constant (this follows from that $\preceq_F$ compares only paths of an equal length and therefore every increasing chain of finite sets of paths related by $\prec^{\forall\exists}_F$ is finite). Moreover, $S \sqsubset \emptyset \ldots \emptyset$ for every sequence of sets of paths $S \neq \emptyset \ldots \emptyset$.

[§]Note that if $\tau \in branches(X)$, $\langle \tau, X \rangle = \emptyset \ldots \emptyset$ 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 $\tau' \notin branches(Y)$, 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 $\tau' \in branches(Y)$.

LEMMA 6. *Given a partial run V of $\mathcal{A}$ on w s.t. $T \preceq_{\text{w-}ext} V$, $T \not\preceq_{ext} V$, and $\tau_V \in V \cup \{\epsilon\}$ with $\tau_V \notin branches(\mathsf{sw}_T(V))$, we can construct a partial run W of $\mathcal{A}$ on w with $T \preceq_{\text{w-}ext} W$ and a path $\tau_W \in W$ with $\tau_W \notin branches(\mathsf{sw}_T(W))$ such that $V \prec^T_{\tau_V,\tau_W} W$.*

PROOF. [Sketch] The proof of Lemma 6 relies on Lemma 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. $\preceq_F$.

From $ext(\pi) \preceq_M leaf(\pi)$, there is a mediator s with $ext(\pi) \preceq_F s \succeq_B leaf(\pi)$. We apply Lemma 3 to V, π, $leaf(\pi)$ and s, which give us a partial run W and $\psi \in branches(W)$ with $leaf(\psi) = s$ such that $\pi \preceq_B \psi$, and for all $1 \leq i \leq |\pi|$, $V \ominus_i \pi \preceq_F^{\forall\exists} W \ominus_i \psi$. Let $\tau_W = \psi$. The proof can be concluded by showing that (i) $T \preceq_{\text{w-}ext} W$, (ii) $\tau_W \notin branches(\mathsf{sw}_T(W))$, and (iii) $\langle \tau_V, \mathsf{sw}_T(V)\rangle \sqsubset \langle \tau_W, \mathsf{sw}_T(W)\rangle$, which implies $V \prec^T_{\tau_V,\tau_W} W$. ∎

Now we construct a run U strongly covering T as follows. Starting from T and ϵ, we can construct a chain $T \prec^T_{\epsilon,\tau_1} T_1 \prec^T_{\tau_1,\tau_2} T_2 \prec^T_{\tau_2,\tau_3} T_3 \ldots$ by successively applying Lemma 6 for each i, $\tau_i \in T_i$, $\tau_i \notin branches(\mathsf{sw}_T(T_i))$, and $T \preceq_{\text{w-}ext} T_i$. 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 that, for each i, as $T \preceq_{\text{w-}ext} T_i$, $height(T_i) = height(T)$. Therefore the length of τ_i as well as the length of $\langle \tau_i, \mathsf{sw}_T(T_i)\rangle$ are bounded by $height(T)$.

Recall that (i) the relation $\sqsubset$ is a partial order, (ii) that $\sqsubset$ does not allow infinite increasing chains of sequences where the length of the sequences is bounded by some constant, and (iii) that $S \sqsubset \emptyset \ldots \emptyset$ for every sequence $S \neq \emptyset \ldots \emptyset$. This means that after a finite number of steps, this chain must arrive to its last T_k and τ_k with $\langle \tau_k, \mathsf{sw}_T(T_k)\rangle = \emptyset \ldots \emptyset$. This means that $\mathsf{sw}_T(T_k) = \emptyset$, which implies that $T \preceq_{ext} T_k$. We can put $U = T_k$ and Lemma 5 is proven.

Now we can use Lemma 5 to prove Lemma 7. It relates partial runs of $\mathcal{A}^+$ with partial runs of $\mathcal{A}$ by the relation $\preceq_{\alpha^+ \Rightarrow \alpha}$ defined as follows. For two 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 $1 \leq i \leq |\pi|$, $\pi_i \in \alpha^+ \implies \psi_i \in \alpha$. Finally, for finite trees T and U over Q, we use $T \preceq_{\alpha^+ \Rightarrow \alpha} U$ to denote that $branches(T) \preceq^{\forall\exists}_{\alpha^+ \Rightarrow \alpha} branches(U)$.

LEMMA 7. *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 $root(T) \preceq_B root(U)$ and $T \preceq_{\alpha^+ \Rightarrow \alpha} U$.*

The proof of Lemma 7 is done by induction on the structure of T, where the induction step employs Lemma 5 (which bridges the gap between $\mathcal{A}^+$ and $\mathcal{A}$ by showing that there is a partial run of $\mathcal{A}$ strongly covering T even when the power of its leaves is extended by transitions of some $\preceq_M$-smaller states). With Lemma 7 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 7 from finite partial runs to full infinite runs¶. This results in Theorem 8, which together with the fact that $\mathcal{L}(\mathcal{A}/\equiv) \subseteq \mathcal{L}(\mathcal{A}^+)$ immediately gives Corollary 9.

¶For an accepting run T of $\mathcal{A}^+$ on a word w, Lemma 7 gives us for every $k \in \mathbb{N}$ and a prefix of T of the height k a partial run of U of the same height that is more accepting. From the infinite set of partial runs of $\mathcal{A}$ obtained this way, we can construct an accepting run of $\mathcal{A}$ on w. The details may be found in [2] and in [2].

THEOREM 8. $\mathcal{L}(\mathcal{A}^+) = \mathcal{L}(\mathcal{A})$.

COROLLARY 9. *Quotienting with mediated equivalence preserves the language.*

5 Algorithm for Computing Mediated Preorder

In this section, we describe an algorithm for computing mediated preorder on an ABA $\mathcal{A} = (\Sigma, Q, \iota, \delta, \alpha)$. We first explain how to compute the maximal forward simulation $\preceq_F$ and backward simulation $\preceq_B$ of $\mathcal{A}$. Both $\preceq_F$ and $\preceq_B$ will be used as the input parameters for computing the mediated preorder $\preceq_M$. In the rest of the section, we will fix $\mathcal{A}$ as the input ABA, use n for the number of states in $\mathcal{A}$, and use m for the number of transitions in $\mathcal{A}$.

Forward Simulation. The algorithm for computing maximal forward simulation $\preceq_F$ on $\mathcal{A}$ can be found in Fritz and Wilke's work [5] (it is called direct simulation in their paper). They reduce the problem of computing maximal forward simulation to a simulation game. Although Fritz and Wilke use a slightly different definition of ABA, it is easy to translate $\mathcal{A}$ to an ABA under their definition with $O(n+m)$ states and $O(nm)$ transitions and then use their algorithm to compute $\preceq_F$. The time complexity of the above procedure is $O(nm^2)$.

Removing Ambiguity. As shown in Section 4.1, $\mathcal{A}$ needs to be $\preceq_F$-unambiguous for mediated minimization. Here we describe how to modify $\mathcal{A}$ to make it not $\preceq_F$-ambiguous. The modification does not change the the language of $\mathcal{A}$ and also the forward simulation relation $\preceq_F$, therefore we do not need to recompute forward simulation again for the modified automaton.

Here we describe the ambiguity removal procedure. 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. This procedure has time complexity $O(n^2 m)$.

Backward Simulation. We now show how to translate the problem of computing maximal backward simulation to a problem of computing maximal simulation on a labeled 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 $q' \preceq_L r'$.

Here we describe the problem of computing the maximal simulation on an LTS. Given an LTS $T = (S, \mathcal{L}, \rightarrow)$ and an *initial* preorder $I \subseteq S \times S$, the task is to find out the unique maximal simulation on T included in I. An algorithm for computing maximal simulation $\preceq^I$ on the LTS T included in I with time complexity $O(|\mathcal{L}|.|S|^2 + |S|.|\rightarrow|)$ and space complexity $O(|\mathcal{L}|.|S|^2)$ can be found in [1].

Computing Backward Simulation via a Reduction to LTS. The problem of computing the maximal backward simulation on $\mathcal{A}$ can be reduced to the problem of computing simulation on an LTS. In order to simplify the explanation of the reduction, we first make the following definition. An *environment* 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 neighbors of the removed state p' in the transition $p \xrightarrow{a} P$. We denote the set of all environments of $\mathcal{A}$ by $Env(A)$. Formally, we define the LTS $A^{\odot} = (Q^{\odot}, \Sigma, \Delta^{\odot})$ as follows:

$$\begin{array}{ccccccc} \text{A transition in } \mathcal{A} & & & & \text{Transitions in } \mathcal{A}^{\odot} & & \\ & & p_1^{\odot} & \xrightarrow{a} & (p,a,\{p_2,p_3\})^{\odot} & \searrow^{a} & \\ p \xrightarrow{a} \{p_1,p_2,p_3\} & \Rightarrow & p_2^{\odot} & \xrightarrow{a} & (p,a,\{p_1,p_3\})^{\odot} & \xrightarrow{a} & p^{\odot} \\ & & p_3^{\odot} & \xrightarrow{a} & (p,a,\{p_1,p_2\})^{\odot} & \nearrow_{a} & \end{array}$$

Figure 3: An example of the reduction from an ABA transition to LTS transitions

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

An example of the reduction is given in Figure 3. The goal of this reduction is to obtain a simulation relation on $A^{\odot}$ with the following property: $p^{\odot}$ is simulated by $q^{\odot}$ in $A^{\odot}$ iff $p \preceq_B q$ in $\mathcal{A}$. However, the maximal simulation on $A^{\odot}$ is not sufficient to achieve this goal. Some essential conditions for backward simulation (e.g., $p \preceq_B q \implies p \preceq_\alpha q$) are missing in $A^{\odot}$. This can be fixed by defining a proper initial preorder I.

Formally, we define $I = \{(q_1^{\odot}, q_2^{\odot}) \mid q_1 \preceq_\iota q_2 \wedge q_1 \preceq_\alpha q_2\} \cup \{((p,a,P)^{\odot}, (r,a,R)^{\odot}) \mid P \preceq_F^{\forall\exists} R\}$. 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 the 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 the condition (3). A simulation relation $\preceq^I$ can be computed using the aforementioned procedure with LTS $A^{\odot}$ and the *initial* preorder I. The following theorems shows the correctness and complexity of computing backward simulation.

THEOREM 10. *For all $q, r \in Q$, we have $q \preceq_B r$ iff $q^{\odot} \preceq^I r^{\odot}$.*

THEOREM 11. *Computing backward simulation has both time and space complexity $O(nm^3)$.*

The complexity comes from three parts of the procedure: (1) compiling $\mathcal{A}$ into its corresponding LTS $A^{\odot}$, (2) computing the initial preorder I, and (3) running the algorithm for computing the LTS simulation relation. The LTS $A^{\odot}$ has at most nm+n states and $2nm$ transitions. It follows that Part (3) has time complexity $O(|\Sigma|n^2m^2)$ and space complexity $O(|\Sigma|n^2m^2)$. In [2], we show that among the three parts, Part (3) has the highest time‖ and space complexity and therefore computing backward simulation also has time complexity $O(|\Sigma|n^2m^2)$ and space complexity $O(|\Sigma|n^2m^2)$. 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 maximal backward simulation has time complexity $O(nm^3)$ and space complexity $O(nm^3)$.

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 [1] 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 naive implementation of this simple procedure has time complexity $O(n^3)$.

‖In [2] we will describe an efficient algorithm for computing I. It has time complexity $O(n^2m^2)$ and space complexity $O(n^2m^2)$.

6 Experimental Results

In this section, we evaluate the performance of mediated minimization by applying it to accelerate the algorithm proposed by Vardi and Kupferman [9] for complementing nondeterministic Büchi automata (NBA). In this algorithm, ABA's are used as intermediate notion for the complementation. To be more specific, the complementation algorithm has two steps: (1) it translates an NBA to an ABA that recognizes 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.

The experimentation is carried out as follows. Three sets of 100 random NBA's (of $|\Sigma| =$ 2,4, and 8, respectively) are generated by the GOAL [11] tool and then used as inputs of the complementation experiments. We compare results of experiments performed according to the following different options: (1) **Original:** keep the ABA as what it is, (2) **Mediated:** minimizing the ABA with mediated equivalence, and (3) **Forward:** minimizing the ABA with forward equivalence.

For each input NBA, we first translate it to an ABA that recognizes its complement language. The ABA is (1) processed according to one of the options described above and then (2) translated back to an equivalent NBA using an exponential procedure **. The results are given in Table 1 and Table 2. Table 1 is an overall comparison between the three different options and Table 2 is a more detailed comparison between **Mediated** and **Forward** minimization.

In Table 1, the columns "NBA" and "Complemented-NBA" are the average statistical data 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 cannot finish within the timeout period (10 min). Note that in the table, the cases that cannot finish within the timeout period are excluded from the average number. From this table, we can see that minimization by mediated equivalence can effectively speed up the complementation and also reduce the size of the complemented NBA's.

	$\|\Sigma\|$	NBA		Complemented-NBA		Time (ms)	Timeout
		St.	Tr.	St.	Tr.		(10 min)
Original				13.9	52.75	5500.9	0
Mediated	2	2.5	3.3	6.68	34.02	524.7	0
Forward				9.45	55.25	5443.7	1
Original				46.4	348.5	9298.6	6
Mediated	4	3.3	6.0	20.42	235.5	1985.4	6
Forward				26.88	325.6	1900.6	7
Original				127.1.3	1723.4	33429.4	24
Mediated	8	4.7	11.9	57.63	1738.3	12930.6	21
Forward				81.23	2349.2	22734.2	24

Table 1: Combining minimization with complementation.

In Table 2, we compare the performance between **Mediated** and **Forward** minimization in detail. The columns "Minimized-ABA" and "Complemented-NBA" are the average difference in the sizes of the ABA after minimization and the complemented BA. From the table, we observe that mediated minimization results in a much better reduction than forward minimization.

	$\|\Sigma\|$	Minimized-ABA		Complemented-NBA	
		St.	Tr.	St.	Tr.
Average Difference	2	33.54%	51.62%	63.3%	235.56%
	4	36.24%	51.44%	89.9%	298.99%
	8	27.94%	40.88%	152.3%	412.7%

Table 2: Comparison: **Mediated** v.s. **Forward**

**For the option "Original", we also use the optimization suggested in [9] that only takes consistent subset.

7 Conclusion and Future Work

We combined forward and backward simulation to form a coarser relation called mediated preorder and showed that quotienting wrt. mediated equivalence preserves the language of ABA. Moreover, we developed an efficient algorithm for computing mediated equivalence. Experimental results show that the mediated reduction of ABA significantly outperforms the reduction based on forward simulation. In the future, we would like to extend our experiments to other applications such as LTL to NBA translation. 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 [6]. Also, we intend to study possibilities of using mediated preorder to remove redundant transitions (in a similar way to [10]). We believe that the extensions described above can improve the performance of mediated reduction.

Acknowledgements. The work was supported by the CONNECT project, the UPMARC project, the Czech Science Foundation (projects 102/07/0322, 102/09/H042), the Barrande project MEB 020840, and the Czech institutional project MSM 0021630528.

References

[1] P. Abdulla, A. Bouajjani, L. Holík, L. Kaati, T. Vojnar. Computing Simulations over TA: Efficient Techniques for Reducing TA. In *Proc. of TACAS'08*, LNCS 4963, Springer, 2008.

[2] P. Abdulla, Y.-F. Chen, L. Holík, T. Vojnar. Mediating for Reduction (On Minimizing Alternating Büchi Automata). Tech. Rep. FIT-TR-2009-02, Brno Univ. of Technology, 2009.

[3] P. Abdulla, L. Holík, L. Kaati, and T. Vojnar. A Uniform (Bi-)simulation-based Framework for Reducing Tree Automata. *ENTCS*, 2009(251):27–48, 2009.

[4] A. Farzan, Y.-F. Chen, E. Clarke, Y.-K. Tsay, and B.-Y. Wang. Extending Automated Compositional Verification to the Full Class of Omega-regular Languages. In *Proc. of TACAS'08*, LNCS 4963, Springer, 2008.

[5] C. Fritz and T. Wilke. State Space Reductions for Alternating Büchi Automata: Quotienting by Simulation Equivalences. In *Proc. of FSTTCS'02*, LNCS 2556, Springer, 2002.

[6] C. Fritz and T. Wilke. Simulation Relations for Alternating Büchi Automata. *Theoretical Computer Science*, 338(1-3):275–314, 2005.

[7] P. Gastin and D. Oddoux. Fast LTL to Büchi Automata Translations. In *Proc. of CAV'01*, LNCS 2102, Springer, 2001.

[8] S. Gurumurthy, O. Kupferman, F. Somenzi, and M. Vardi. On Complementing Nondeterministic Büchi Automata. In *Proc. of CHARME'03*, LNCS 2860, Springer, 2003.

[9] O. Kupferman and M. Vardi. Weak Alternating Automata Are Not That Weak. *ACM Transactions on Computational Logic*, 2(3):408–429, 2001.

[10] F. Somenzi and R. Bloem. Efficient Büchi Automata from LTL Formulae. In *Proc. of CAV'00*, LNCS 1855, Springer, 2000.

[11] Y.K. Tsay, Y.F. Chen, M.H. Tsai, K.N. Wu, W.C. Chan. GOAL: A Graphical Tool for Manipulating Büchi Automata & Temp. Formulae. *TACAS'07*, LNCS 4424, Springer, 2007.

[12] M. Vardi. Automata-theoretic Model Checking Revisited. In *Proc. of VMCAI'07*, LNCS 4349, Springer, 2007.