On the Succinctness of Nondeterminism

Benjamin Aminof and Orna Kupferman

Hebrew University, School of Engineering and Computer Science, Jerusalem 91904, Israel
Email: {benj,orna}@cs.huji.ac.il

Abstract. Much is known about the differences in expressiveness and succinctness between nondeterministic and deterministic automata on infinite words. Much less is known about the relative succinctness of the different classes of nondeterministic automata. For example, while the best translation from a nondeterministic Büchi automaton to a nondeterministic co-Büchi automaton is exponential, and involves determinization, no super-linear lower bound is known. This annoying situation, of not being able to use the power of nondeterminism, nor to show that it is powerless, is shared by more problems, with direct applications in formal verification.

In this paper we study a family of problems of this class. The problems originate from the study of the expressive power of deterministic Büchi automata: Landweber characterizes languages $L \subseteq \Sigma^\omega$ that are recognizable by deterministic Büchi automata as those for which there is a regular language $R \subseteq \Sigma^*$ such that L is the *limit* of R; that is, $w \in L$ iff w has infinitely many prefixes in R. Two other operators that induce a language of infinite words from a language of finite words are *co-limit*, where $w \in L$ iff w has only finitely many prefixes in R, and *persistent-limit*, where $w \in L$ iff almost all the prefixes of w are in R. Both co-limit and persistent-limit define languages that are recognizable by deterministic co-Büchi automata. They define them, however, by means of nondeterministic automata. While co-limit is associated with complementation, persistent-limit is associated with universality. For the three limit operators, the deterministic automata for R and L share the same structure. It is not clear, however, whether and how it is possible to relate nondeterministic automata for R and L, or to relate nondeterministic automata to which different limit operators are applied. In the paper, we show that the situation is involved: in some cases we are able to describe a polynomial translation, whereas in some we present an exponential lower bound. For example, going from a nondeterministic automaton for R to a nondeterministic automaton for its limit is polynomial, whereas going to a nondeterministic automaton for its persistent limit is exponential. Our results show that the contribution of nondeterminism to the succinctness of an automaton does depend upon its semantics.

1 Introduction

Finite *automata on infinite objects* were first introduced in the 60's, and were the key to the solution of several fundamental decision problems in mathematics and logic [5, 17, 21]. Today, automata on infinite objects are used for *specification* and *verification* of nonterminating systems. The automata-theoretic approach to verification reduces questions about systems and their specifications to questions about automata [13, 26]. Recent industrial-strength property-specification languages such as Sugar [3], ForSpec [2], and PSL 1.01 [7] include regular expressions and/or automata, making specification and verification tools that are based on automata even more essential and popular.

There are many ways to classify an automaton on infinite words. One is the class of its acceptance condition. For example, in *Büchi* automata, some of the states are designated as accepting states, and a run is accepting iff it visits states from the accepting set infinitely often [5]. Dually, in *co-Büchi* automata, a run is accepting iff it visits states from the accepting set only finitely often. Another way to classify an automaton is by the type of its branching mode. In a *deterministic* automaton, the transition function maps the current state and input letter to a single successor state. When the branching mode is *nondeterministic*, the transition function maps the current state and letter to a set of possible successor states. Thus, while a deterministic automaton has a single run on an input word, a nondeterministic automaton may have several runs on an input word, and the word is accepted by the automaton if at least one of the runs is accepting.

The different classes of automata have different *expressive power*. For example, unlike automata on finite words, where deterministic and nondeterministic automata have the same expressive power, deterministic Büchi automata (DBW) are strictly less expressive than nondeterministic Büchi automata (NBW). That is, there exists a language L over infinite words such that L can be recognized by a nondeterministic Büchi automaton but cannot be recognized by a deterministic Büchi automaton. It also turns out that some classes of automata may be more *succinct* than other classes. For example, translating a nondeterministic co-Büchi automaton (NCW) into a deterministic co-Büchi automaton (DCW) is possible [20], but involves an exponential blow up.

There has been extensive research on the expressiveness and succinctness of automata on infinite words [25]. In particular, since reasoning about deterministic automata is simpler than reasoning about nondeterministic ones, questions like deciding whether a nondeterministic automaton has an equivalent deterministic one, and the blow up involved in determinization, are of particular interest [8, 16, 12]. These questions get further motivation with the discovery that many natural specifications correspond to the deterministic fragments. In particular, it is shown in [12] that given a linear temporal logic (LTL) formula ψ, there is an alternation-free μ-calculus (AFMC) formula equivalent to $\forall\psi$ iff ψ can be recognized by a DBW. Evaluating specifications in the alternation-free fragment of μ-calculus can be done with linearly many symbolic steps, so coming up with an optimal translation of LTL to AFMC is a problem of great practical importance.

Let us elaborate on the LTL to AFMC example, as it highlights the open problems that have led to our research. Current translations translate the LTL formula ψ to a DBW, which can be linearly translated to an AFMC formula for $\forall\psi$. The translation of LTL to DBW, however, is doubly exponential, thus the overall translation is doubly-exponential, with only an exponential matching lower bound. A promising direction for tightening the upper bound was suggested in [12]: instead of translating an LTL formula ψ to a DBW, one can translate $\neg\psi$ to an NCW. Then, the NCW can be linearly translated to an AFMC formula for $\exists\neg\psi$, whose negation is equivalent to $\forall\psi$. The fact that the translation can go through a nondeterministic rather than a deterministic automaton is very promising, as nondeterministic automata are typically exponentially more succinct than deterministic ones. Nevertheless, the problem of translating LTL formulas to NCWs of exponential size[1] is still open. The best translation that is known today involves a doubly-exponential blow up, and it actually results in a DCW, giving up the idea that the translation of LTL to AFMC can be exponentially more efficient by using intermediate nondeterministic automata.

[1] As mentioned above, not all LTL formulas can be translated to NCWs. When we talk about the blow up in a translation, we refer to formulas for which a translation exists.

This unfortunate situation of not being able to use the power of nondeterminism is shared by more problems. One that is strongly related to the LTL to AFMC problem described above is the open problem of translating NBWs to NCWs (when possible). Despite continuous efforts, the best translation that is known first determinizes the NBW. Accordingly, starting with an NBW with n states, we end up with an NCW with $2^{O(n \log n)}$ states [22]. This is particularly annoying as even no super-linear lower bound is known, and in fact, only recently were we able to come up with an example that an NCW cannot be defined on top of the state space and transitions of the NBW [9]. The class of open problems of this nature expands also to the branching setting. For a language L of infinite words, let $der(L)$ be the language of infinite trees that contain exactly all trees all of whose paths are in L. It is known that $der(L)$ can be recognized by a nondeterministic Büchi tree automaton (NBT) iff L can be recognized by a DBW [10]. Given an NBT for $der(L)$, the most efficient construction that is known for generating from it an NBW for L is exponential, and it actually constructs a DBW for L. Also here, no super-linear lower bound is known, and yet it is not clear how nondeterminism, and its succinctness with respect to the deterministic model, can be used.

In this paper we study a family of problems in this class. Recall that DBWs are less expressive than NBWs. Landweber characterizes languages $L \subseteq \Sigma^\omega$ that can be recognized by a DBW as those for which there is a regular language $R \subseteq \Sigma^*$ such that L is the *limit* of R. Formally, w is in the limit of R iff w has infinitely many prefixes in R [14]. It is not hard to see that a DBW for L, when viewed as a deterministic finite automaton (DFW), recognizes a language whose limit is L, and vice versa – a DFW for R, when viewed as a DBW, recognizes the language that is the limit of R. What about the case in which R and L are given by nondeterministic automata? It is not hard to see that the simple transformation between the two formalisms no longer holds. For example, the NBW $\mathcal{A}$ in Figure 1 recognizes the language L of all words with infinitely many 1s, yet when viewed as a nondeteministic finite automaton (NFW), it recognizes $(0+1)^+$, whose limit is $(0+1)^\omega$. As another example, the language of the NBW $\mathcal{A}'$ is empty, yet when viewed as an NFW, it recognizes the language $(0+1)^* \cdot 1$, whose limit is L. As demonstrated by the examples, the difficulty of the nondeterministic case originates from the fact that different prefixes of the infinite word may follow different accepting runs of the NFW, and there is no guarantee that these runs can be merged into a single run of the NBW. Accordingly, the best translation that is known for going from an NFW to an NBW accepting its limit, or from an NBW to a limit NFW, is to first determinize the given automaton. This involves a $2^{O(n \log n)}$ blow up and gives up the potential succinctness of the nondeterministic model. On the other hand, no lower bound above $\Omega(n \log n)$ is known.

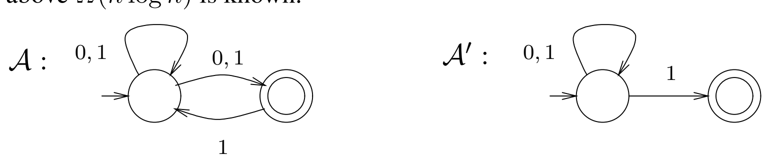

Fig. 1. Relating NBWs and limit NFWs.

In addition to the limit operator introduced by Landweber, we introduce and study two more ways to induce a language of infinite words from a language of finite words:

the *co-limit* of R is the set of all infinite words that have only finitely many prefixes in R. Thus, co-limit is dual to Landweber's limit. Also, the *persistent limit* of R is the set of all infinite words that have only finitely many prefixes not in R. Thus, eventually all the prefixes are in R.

We study the succinctness of NFWs for R with respect to DBWs, DCWs, NBWs, and NCWs recognizing languages induced by each of the three limit operators, and the succinctness of the Büchi and co-Büchi automata with respect to the NFWs. In particular, we prove that while the translation from an NFW to an NBW for its limit is cubic (thus, nondeterminism is helpful, and the traditional "determinize first" approach is beaten!), the translations from an NFW to an NCW for its co-limit or its persistent limit are exponential, thus determinization is legitimate. We also study succinctness among NFWs to which different limit operators are applied. For example, we prove that going from a persistent limit NFW to a limit NFW involves an exponential blow up. In other words, given an NFW $\mathcal{A}$ whose persistent limit is L, translating $\mathcal{A}$ to an NFW whose limit is L may involve an exponential blow up. Note that persistent limit and limit are very similar – both require the infinite word to have infinitely many prefixes in $L(\mathcal{A})$, only that the persistent limit requires, in addition, that only finitely many prefixes are not in $L(\mathcal{A})$. This difference, which is similar to the difference between NBW and NCW, makes persistent limit exponentially more succinct. Technically, it follows from the fact that persistent limit NFWs inherit the power of alternating automata. In a similar, though less surprising way, co-limit NFWs inherit the power of complementation, and are also exponentially more succinct. In cases where we are not able to describe a lower bound, we prove that the translations are not *type* [8, 9], namely that an equivalent NFW cannot be defined on top of the same transition structure.

The study of the limit operators checks behaviors in the limit. We examine how our results are affected by limiting attention to *safety*, *co-safety*, and *bounded* languages [1, 24, 11]. In these languages, the behavior in the limit is not restricted. In particular, in bounded languages, membership in the language depends on a bounded prefix of the word. We show that most of our lower bounds apply even in the restricted setting of the limited fragments, yet for some cases we are able to describe upper bounds that do not hold in the general case. Finally, recall that the difficulty of the nondeterministic case originates from the fact that the accepting runs on different prefixes of the infinite word may not be merged into one infinite accepting run of the NBW. We describe a sufficient structural condition on NFWs that guarantees that accepting runs can be merged. We call NFWs that satisfy this condition *continuous NFWs*. We show that while the limit of a continuous NFW $\mathcal{A}$ is the language of $\mathcal{A}$ when viewed as an NBW, continuous NFWs are exponentially more succinct than DBWs.

2 Preliminaries

2.1 Automata on finite and infinite words

Given an alphabet Σ, a *word* over Σ is a sequence $w = \sigma_1 \cdot \sigma_2 \cdot \sigma_3 \cdots$ of letters in Σ. A word may be either finite or infinite. An *automaton* is a tuple $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$, where Σ is the input alphabet, Q is a finite set of states, $\delta : Q \times \Sigma \to 2^Q$ is a transition function, $Q_0 \subseteq Q$ is a set of initial states, and $\alpha \subseteq Q$ is an acceptance condition. We define several acceptance conditions below. The automaton $\mathcal{A}$ may have several initial states and the transition function may specify many possible transitions for each state and letter, and hence we say that $\mathcal{A}$ is *nondeterministic*. In the case where $|Q_0| = 1$ and for every $q \in Q$ and $\sigma \in \Sigma$, we have that $|\delta(q, \sigma)| = 1$, we say that $\mathcal{A}$ is *deterministic*.

The automaton may run on finite or infinite words. A *run* of $\mathcal{A}$ on a finite word $w = \sigma_1 \cdot \sigma_2 \cdots \sigma_k \in \Sigma^*$ is a function $r : \{0, \ldots, k\} \to Q$ where $r(0) \in Q_0$, and for every $0 \leq i < k$, we have that $r(i+1) \in \delta(r(i), \sigma_{i+1})$. The run is *accepting* iff $r(k) \in \alpha$. Otherwise, it is rejecting. When the input word is infinite, and thus $w = \sigma_0 \cdot \sigma_1 \cdots \in \Sigma^\omega$, a run of $\mathcal{A}$ on w is a function $r : \mathbb{N} \to Q$ with $r(0) \in Q_0$, and for every $i \geq 0$, we have that $r(i+1) \in \delta(r(i), \sigma_{i+1})$. Acceptance is defined with respect to the set of states $inf(r)$ that the run r visits infinitely often. Formally, $inf(r) = \{q \in Q \, : \, \text{for infinitely many } i \in \mathbb{N}, \text{ we have } r(i) = q\}$. As Q is finite, it is guaranteed that $inf(r) \neq \emptyset$. The run r is *accepting* iff the set $inf(r)$ satisfies the acceptance condition α. We consider here the *Büchi* and the *co-büchi* acceptance conditions. A set S satisfies a Büchi acceptance condition $\alpha \subseteq Q$ if and only if $S \cap \alpha \neq \emptyset$. Dually, S satisfies a *co-Büchi* acceptance condition $\alpha \subseteq Q$ if and only if $S \cap \alpha = \emptyset$.

We sometimes view a run r as a (finite or infinite) word over the alphabet Q. For example, $r = q_0, q_5, q_5$ indicates that $r(0) = q_0$ whereas $r(1) = r(2) = q_5$. Note that while a deterministic automaton has a single run on an input word, a nondeterministic automaton may have several runs on w or none at all. An automaton accepts a word iff it has an accepting run on it. The language of an automaton $\mathcal{A}$, denoted $L(\mathcal{A})$, is the set of words that $\mathcal{A}$ accepts. For a language L, the complement of L, denoted $comp(L)$, is the set of words not in L. Thus, for $L \subseteq \Sigma^*$ we have $comp(L) = \Sigma^* \setminus L$, and for $L \subseteq \Sigma^\omega$ we have $comp(L) = \Sigma^\omega \setminus L$.

We denote the different classes of automata by three letter acronyms in $\{\mathrm{D}, \mathrm{N}\} \times$ {F, B, C} $\times$ {W}. The first letter stands for the branching mode of the automaton (deterministic or nondeterministic); the second letter stands for the acceptance-condition type (finite, Büchi, or co-Büchi). The third letter indicates that the automaton runs on words.

For two automata $\mathcal{A}$ and $\mathcal{A}'$, we say that $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent* if $L(\mathcal{A}) = L(\mathcal{A}')$. For a class γ of automata, we say that an automaton $\mathcal{A}$ is γ *realizable* iff $\mathcal{A}$ has an equivalent automaton in the class γ. Similarly, a language L is γ *realizable* iff there is an automaton $\mathcal{A}$ in the class γ whose language is L. In the case of finite words, NFWs can be determinized, thus all NFWs are DFW realizable. In the case of infinite words, different classes of automata have different expressive power. In particular, while NBWs recognize all ω-regular language [17], DBWs are strictly less expressive than NBW, and so are DCW [14]. In fact, a language L is DBW-realizable iff $comp(L)$ is DCW-realizable. Indeed, by viewing a DBW as a DCW, we get an automaton for the complementing language, and vice versa. The expressiveness superiority of the nondeterministic model with respect to the deterministic one does not apply to the co-Büchi acceptance condition. There, NCWs can be determinized[2], thus all NCWs are DCW realizable.

[2] When applied to universal Büchi automata, the translation in [20], of alternating Büchi automata into NBW, results in DBW. By dualizing it, one gets a translation of NCW to DCW.

2.2 Limits of Languages of Finite Words

Studying the expressive power of DBWs, Landweber characterizes languages $L \subseteq \Sigma^\omega$ that are DBW-realizable as those for which there is a regular language $R \subseteq \Sigma^*$ such that $w \in L$ iff w has infinitely many prefixes in R. Thus, each language of finite words induces a language of infinite words. In Definition 1 below, we introduce two additional ways to induce a language of infinite words from a language on finite words. Given a word $w = \sigma_1, \sigma_2, \cdots \in \Sigma^\omega$, we denote the i-th letter of w by $w[i]$, the sub-word $\sigma_i, \cdots, \sigma_j$ by $w[i, j]$ and the sub-word $\sigma_i, \cdots, \sigma_{j-1}$ by $w[i, j)$.

Definition 1. *Consider a language $R \subseteq \Sigma^*$. We define three languages of infinite words induced by R.*

1. **[limit]** *$lim(R) \subseteq \Sigma^\omega$ is the set of all words that have infinitely many prefixes in R. I.e., $lim(R) = \{w \mid w[1, i] \in R$ for infinitely many i's$\}$* [14].
2. **[co-limit]** *$co\text{-}lim(R) \subseteq \Sigma^\omega$ is the set of all words that have only finitely many prefixes in R. I.e., $co\text{-}lim(R) = \{w \mid w[1, i] \in R$ for finitely many i's$\}$.*
3. **[persistent limit]** *$plim(R) \subseteq \Sigma^\omega$ is the set of all words that have only finitely many prefixes not in R. I.e., $plim(R) = \{w \mid w[1, i] \in R$ for almost all i's$\}$.*

For example, for $R = (a+b)^*b$, the language $lim(R)$ consists of all words that have infinitely many b's, $co\text{-}lim(R)$ is the language of words that have finitely many b's, and $plim(R)$ is the language of words that have finitely many a's. For an NFW $\mathcal{A}$, we use $lim(\mathcal{A})$, $co\text{-}lim(\mathcal{A})$, and $plim(\mathcal{A})$, to denote $lim(L(\mathcal{A}))$, $co\text{-}lim(L(\mathcal{A}))$, and $plim(L(\mathcal{A}))$, respectively.

The three limit operators are dual in the following sense:

Lemma 1. *For every $R \subseteq \Sigma^*$, we have $comp(lim(R)) = co\text{-}lim(R) = plim(comp(R))$.*

Recall that a language $L \subseteq \Sigma^\omega$ is DBW realizable iff $L = lim(R)$ for some regular $R \subseteq \Sigma^*$ [14]. By Lemma 1 and the duality between DBW and DCW, it follows that L is DCW realizable iff $L = co\text{-}lim(R)$ for some regular $R \subseteq \Sigma^*$, or, equivalently, $L = plim(R)$ for some regular $R \subseteq \Sigma^*$. A direct way to prove the above expressiveness results is to consider the deterministic Büchi or co-Büchi automaton $\mathcal{A}$ for L. Let $\mathcal{A}_{fin}$ be $\mathcal{A}$ when viewed as a DFW, and let $\tilde{\mathcal{A}}_{fin}$ be $\mathcal{A}_{fin}$ with a dualized accepting set. In case $\mathcal{A}$ is a DBW, then $L(\mathcal{A}) = lim(\mathcal{A}_{fin})$. Similarly, if $\mathcal{A}$ is a DCW, then $L(\mathcal{A}) = co\text{-}lim(\mathcal{A}_{fin}) = plim(\tilde{\mathcal{A}}_{fin})$. Thus, in the deterministic setting, the transitions among the automata for L and R involve no blow up, and are even done on top of the same structure. Our goal in this paper is to study the blow up between the automata in the nondeterministic setting. In order to avoid lower bounds that are inherited directly from the exponential blow up of complementation, we study both co-limit and persistent-limit. Note that only the former has the flavor of complementation.

Finally, note that for all of the three limit operators, different regular languages may induce the same limit language. For example, if L is the language of all finite words of length at least 7, L' the language of all finite words of length at least 20, and L'' the language of all finite words of even length, then $L \neq L'$ and yet $lim(L) = lim(L') = plim(L) = plim(L') = \Sigma^\omega$, and $co\text{-}lim(L) = co\text{-}lim(L') = \emptyset$. Also, even though $L'' \neq comp(L)$, we still have that $co\text{-}lim(L) = plim(L'')$.

3 Succinctness of NFW with respect to Büchi and co-Büchi Automata

In this section we study the succinctness of the NFW for R with respect to the Büchi and co-Büchi automata that recognize the ω-regular languages induced by applying each of the three limit operators to R. We start with the case the Büchi and co-Büchi automata are deterministic, and show that then, the succinctness of the nondeterministic model in the case of finite words carries over to the limit setting. We then proceed to the case the Büchi and co-Büchi automata are nondeterministic and show that there, the situation is more involved. First, the exponential blow up in NFW complementation is carried over to an exponential blow up in a translation of co-limit NFW to an NCW. More surprising are the results for limit and persistent limit NFW: by analyzing the structure of an NFW, we are able to translate an NFW to an NBW for its limit with only a cubic blow up. On the other hand, while persistent limit involves no complementation, it enables a universal reference to the prefixes of the input word. Consequently, we are able to prove that the exponential succinctness of *alternating* automata on finite words with respect to NFW carries over to an exponential lower bound on the translation of an NFW to a NCW for its persistent limit.

We start with DBW and DCW. Recall that limit is associated with DBW whereas co-limit and persistent limit are associated with DCW. We are still able to describe a bound for the translation to both DBW and DCW.

Theorem 1. [lim NFW $\rightarrow$ DBW, plim NFW $\rightarrow$ DCW, clim NFW $\rightarrow$ DCW]

- *For every $n \geq 1$, there is $L_n \subseteq \Sigma^\omega$ such that there is an NFW $\mathcal{A}$ with $O(n)$ states such that lim$(\mathcal{A})$ = plim$(\mathcal{A}) = L_n$, but L_n cannot be recognized by a DBW or a DCW with less than 2^n states.*
- *For every $n \geq 1$, there is $L_n \subseteq \Sigma^\omega$ such that there is an NFW $\mathcal{A}$ with $O(n)$ states such that co-lim$(\mathcal{A}) = L_n$, but L_n cannot be recognized by a DBW or a DCW with less than 2^n states.*

Proof: We start with limit and persistent limit. For $n \geq 1$, let $R_n \subseteq \Sigma^*$ be such that an NFW for R_n has $O(n)$ states, whereas a DFW for it must have at least 2^n states. By [18], such R_n exist. Let $\#$ be a letter not in Σ, and let $L_n = R_n \cdot \#^\omega$. In the full version, we show that there is an NFW $\mathcal{A}$ with $O(n)$ states such that *lim*$(\mathcal{A})$ = *plim*$(\mathcal{A}) = L_n$, but a DBW or a DCW for L_n must have at least 2^n states.

We now turn to co-limit. For $n \geq 1$, let $R_n \subseteq \Sigma^*$ be such that an NFW for R_n has $O(n)$ states whereas a DFW for $comp(R_n)$ must have at least 2^n states. By [18], such R_n exist. In the full version, we show that there is an NFW $\mathcal{A}$ with $O(n)$ states such that *co-lim*$(\mathcal{A}) = L_n$, but there is no DBW or DCW for L_n with less than 2^n states. □

The results proved in Theorem 1 are not surprising, as they meet our expectation from the finite-word case. We now turn to study the succinctness of NFWs with respect to NBWs and NCWs. Here, we can no longer apply the known succinctness of NFWs.

We first show that in the case of the limit operator, it is possible to translate an NFW $\mathcal{A}$ to an NBW $\mathcal{A}'$ of cubic size such that *lim*$(\mathcal{A}) = L(\mathcal{A}')$. Thus, while the limit operator enables each prefix of the run to be accepted by following different nondeterministic choices, this flexibility does not lead to an exponential succinctness.

We first need some notations. Given an NFW $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$ and two sets of states $P, S \subseteq Q$, we denote by $L_{P,S}$ the language of $\mathcal{A}$ with initial set P and accepting set S. Formally, $L_{P,S}$ is the language accepted by the NFW $\langle \Sigma, Q, \delta, P, S \rangle$. If S or P are singletons we omit the curly braces; so instead of $L_{\{p\},S}$ we simply write $L_{p,S}$, etc.

Theorem 2. *For every NFW* $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$,

$$lim(\mathcal{A}) = \bigcup_{p \in Q} L_{Q_0,p} \cdot (L_{p,p} \cap L_{p,\alpha})^{\omega}.$$

Proof: Assume first that w can be partitioned into sub-words $w = u_0 \cdot u_1 \cdot u_2 \cdots$ such that for some $p \in Q$, we have $u_0 \in L_{Q_0,p}$, and for every $i \geq 1$, the word u_i is in $L_{p,p} \cap L_{p,\alpha}$. It follows that there is a run r_0 of $\mathcal{A}$ on u_0 that starts in Q_0 and ends in p, and that for every $i \geq 1$ there are runs r_i and r'_i of $\mathcal{A}$ on u_i such that r_i starts in p and ends in p while r'_i starts in p and ends in some state in α. Then, for every $i \geq 1$ the run $r_0 \cdot r_1 \cdots r_{i-1} \cdot r'_i$ is an accepting run of $\mathcal{A}$ on the prefix $u_0 \cdot u_1 \cdots u_i$ of w, thus $w \in lim(\mathcal{A})$.

For the other direction, assume that $w \in lim(\mathcal{A})$. For technical simplicity assume first that $\mathcal{A}$ has a single initial state q_0. We construct a tree T in which each node is labeled by a state in Q. For a node x of T, let $|x|$ denote the level of x in the tree, and let $state(x)$ be the state with which x is labeled. The tree T embodies all the possible accepting runs of $\mathcal{A}$ on prefixes of w. The root of T is labeled by q_0. Consider a node x in the tree. All the successors of x have different labels, and y is a successor of x iff $|y| = |x| + 1$, and there is an accepting run r of $\mathcal{A}$ on a prefix of w of length at least $|y|$ such that $r(|x|) = state(x)$ and $r(|y|) = state(y)$. Observe that every node in the tree has at most $|Q|$ successors and that the tree is infinite since $\mathcal{A}$ accepts infinitely many prefixes of w. Also note that every node in the tree is part of at least one accepting run of $\mathcal{A}$ on some prefix of w.

By König's Lemma the tree has an infinite path π. We associate with every node π_i on this path two nodes y_i and z_i such that y_i is some node labeled by an accepting state that is reachable in zero or more steps from π_i, and z_i is the node on π that is at the same level as y_i. Since Q is finite, there are two states $p \in Q, q \in \alpha$ such that $state(y_j) = q$ and $state(z_j) = p$ for infinitely many indices j. By taking a sub-sequence of these indices we can get an infinite set of indices $0 < j_0 < j_1 < \ldots$ such that for every $k \geq 0$ not only $state(y_{j_k}) = q$ and $state(z_{j_k}) = p$, but also $|z_{j_k}| < |\pi_{j_{k+1}}|$. It follows that $w[0, |z_{j_0}|)$ is a word in $L_{q_0,p}$. Also, for every $k \geq 0$ the tree has a path from z_{j_k} to $\pi_{j_{k+1}}$ and from there to $y_{j_{k+1}}$ implying that $w[|z_{j_k}|, |y_{j_{k+1}}|)$ is in $L_{p,\alpha}$. Similarly, the tree has a path from z_{j_k} to $z_{j_{k+1}}$ implying that $w[|z_{j_k}|, |z_{j_{k+1}}|)$ is in $L_{p,p}$. Recalling that $|y_{j_{k+1}}| = |z_{j_{k+1}}|$ we obtain that $w \in L_{q_0,p} \cdot (L_{p,p} \cap L_{p,\alpha})^{\omega}$.

When Q_0 is not a singleton, we may have instead of a single tree T, a forest of trees, with one tree for each state in Q_0. Since Q_0 is finite, one of the trees in the forest is infinite, and the proof proceeds with that tree. □

Given $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$ with n states, constructing an NBW accepting $\bigcup_{p \in Q} L_{Q_0,p} \cdot (L_{p,p} \cap L_{p,\alpha})^{\omega}$, involves n intersections of NFWs, n applications of the ω operation to an NFW, n concatenations of an NFW to an NBW, and finally, obtaining the union of the resulting n NBWs. Accordingly, the characterization in Theorem 2 implies the following upper bound.

Corollary 1. [lim NFW $\rightarrow$ NBW] *Given an NFW $\mathcal{A}$ with n states, there is an NBW $\mathcal{A}'$ with $O(n^3)$ states such that $L(\mathcal{A}') = lim(\mathcal{A})$.*

We now turn to study co-limit and persistent limit. In the first case, the exponential blow up in NFW complementation and the complementing nature of the co-limit operator hint that an exponential lower bound is likely to exist also for the translation of NFW to an NCW or an NBW for its co-limit. In the case of persistent limit, however, we expect the translation to be similar to the one for limit: the NFW enables the prefixes to be accepted each by following a different nondeterministic choice, and, as with the limit operator, an NCW that is polynomially larger should be able to merge these choices into a single nondeterministic choice. This expectation is refuted: the persistence of the plim operator adds to the NFW the power of universal branching, which makes it exponentially more succinct.

Theorem 3. [co-lim NFW $\rightarrow$ NCW / NBW, plim NFW $\rightarrow$ NCW/ NBW] *For every $n \geq 1$, there is a language $L_n \in \Sigma^\omega$ such that there are NFW $\mathcal{A}$ and $\mathcal{A}'$, both with $O(n)$ states, such that plim$(\mathcal{A})$ = co-lim$(\mathcal{A}') = L_n$, but L_n cannot be accepted by an NCW or an NBW with less than 2^n states.*

Proof: Let $\Sigma = \{0, 1\}$. Every word w over Σ can be viewed as a word in $(\Sigma^n)^\omega$, that is, as an infinite sequence of n-bit vectors. The language L_n is the language of sequences that are almost everywhere identical. Formally, $L_n = \{w \mid$ there is $u \in (\Sigma^n)^*$ and $v \in \Sigma^n$ such that $w = uv^\omega\}$.

In the full version, we describe an NFW $\mathcal{A}$ with $O(n)$ states such that *plim*$(\mathcal{A}) = L_n$. On the other hand, by [23], the language L_n cannot be accepted by a nondeterministic Streett automaton with less than 2^n states. Since NBWs and NCWs are a special case of nondeterministic Streett automata, we are done. $\square$

4 Succinctness Among the Different Limit Operators

In the previous section, we related NFWs with Büchi and co-Büchi automata. In this section we study the blow ups involved in translating an NFW that induces a language of infinite words by a limit operator (*lim*, *co-lim*, or *plim*) to an NFW that induces the same language by a different limit operator.

We first show that the exponential blow up in NFW complementation can be lifted to an exponential blow up in the translation of a *lim* or a *plim* NFW to a *co-lim* NFW.

Theorem 4. [lim NFW $\rightarrow$ co-lim NFW, plim NFW $\rightarrow$ co-lim NFW] *For every $n \geq 1$, there is $L_n \subseteq \Sigma^\omega$ such that there is an NFW $\mathcal{A}$ with $O(n)$ states such that lim$(\mathcal{A})$ = plim$(\mathcal{A}) = L_n$, but an NFW $\mathcal{A}'$ such that co-lim$(\mathcal{A}') = L_n$, must have at least 2^n states.*

Proof: For $n \geq 1$, let $R_n \subseteq \Sigma^*$ be such that an NFW for R_n has $O(n)$ states, whereas an NFW for $comp(R_n)$ must have at least 2^n states. By [18], such R_n exist. We can construct from an NFW $\mathcal{A}$ for R_n, an NFW $\mathcal{A}'$ with one extra state for $R_n \cdot \#^+$. Then, *lim*$(\mathcal{A}')$ = *plim*$(\mathcal{A}') = R_n \cdot \#^\omega$. In the full version, we prove that there is no NFW $\mathcal{A}$ with less than 2^n states, such that *co-lim*$(\mathcal{A}) = R_n \cdot \#^\omega$. $\square$

We note that similar arguments can be used to show that NCWs (and thus also NBWs, as NCWs are linearly translatable to NBWs) are exponentially more succinct than *co-lim* NFWs. To see this, note that the NCW obtained from the NFW $\mathcal{A}'$ for $R_n \cdot \#^+$ by letting all states but the $\#$-sink to be in α, accepts $R_n \cdot \#^\omega$.

Theorem 5. [co-lim NFW → lim NFW, plim NFW → lim NFW] *For every $n \geq 1$, there is a language $L_n \subseteq \Sigma^\omega$ such that there are NFWs $\mathcal{A}$ with $O(n)$ states, and $\mathcal{A}'$ with $O(n^2)$ states, such that co-lim$(\mathcal{A})$ = plim$(\mathcal{A}') = L_n$ but an NFW $\mathcal{A}''$ such that lim$(\mathcal{A}'') = L_n$ must have at least 2^n states.*

Proof: Consider the language $L_n \subseteq \{0,1\}^\omega$ of all words w such that $w = uuz$, with $|u| = n$. In the full version, we prove that an NFW $\mathcal{A}''$ such that *lim*$(\mathcal{A}'') = L_n$ must remember subsets of size n, and thus must have at least 2^n states.

In order to construct small NFW for the co-limit and persistent limit operators, we observe that a word w is in L_n iff $\bigwedge_{i=1}^{n}(w[i] = w[n+i])$. In the case of co-limit, we can check that only finitely many (in fact, 0) prefixes h of an input word are such that $h[i] \neq h[i+n]$ for some $1 \leq i \leq n$. This is done by letting $\mathcal{A}$ guess a position $1 \leq i \leq n$, remember $h[i]$, and accept a word iff the letter that comes n positions after it (that is, in $h[i+n]$) is different. It is easy to see that $\mathcal{A}$ requires only $O(n)$ states. A word w has finitely many prefixes in $L(\mathcal{A})$ iff $w \in L_n$. Hence, *co-lim*$(\mathcal{A}') = L_n$.

The case of persistent limit is much harder, as we cannot use the implicit complementation used in the co-limit case. Instead, we use the universal nature of persistence. We define the NFW $\mathcal{A}'$ as a union of n gadgets $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$. The gadget $\mathcal{A}'_i$ is responsible for checking that $w[i] = w[n+i]$. In order to make sure that the conjunction on all $1 \leq i \leq n$ is satisfied, we further limit $\mathcal{A}'_i$ to accept only words of length $i \bmod n$. Hence, $\mathcal{A}'_i$ accepts a word $u \in \Sigma^*$ iff $u[i] = u[n+i] \wedge |u| = i \bmod n$. Consequently, if $w[i] \neq w[n+i]$, then all the prefixes of w of length $i \bmod n$ are rejected by $\mathcal{A}'$. On the other hand, if only a finite number of prefixes of an infinite word are not accepted by $\mathcal{A}'$, then for all $1 \leq i \leq n$, only a finite number of prefixes of length $i \bmod n$ are not accepted by $\mathcal{A}'_i$. Thus, a word w is in *plim*$(\mathcal{A}')$ iff for every $1 \leq i \leq n$, almost all the prefixes of w of length $i \bmod n$ are accepted by $\mathcal{A}'_i$. Hence, $w \in$ *plim*$(\mathcal{A}')$ iff for all $1 \leq i \leq n$ we have that $w[i] = w[n+i]$, and we are done. Since each of the gadgets has $O(n)$ states, and $\mathcal{A}'$ needs n gadgets, it has $O(n^2)$ states. □

The notion of *typeness* for automata was introduced in [8] in the context of DBW. It is shown there that if a deterministic Rabin automaton $\mathcal{A}$ recognizes a language that is DBW realizable, then $\mathcal{A}$ has an equivalent DBW on the same structure. Typeness in general was studied in [9]. For example, it is shown in [9] that an NBW that is NCW realizable need not have an NCW on the same structure. Here, we study typeness for NFWs to which the limit operators are applied. For two limit operators β and γ (*lim*, *co-lim*, or *plim*) we say that an NFW $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$ is (β, γ)-*type* if there is $\alpha' \subseteq Q$ such that the NFW $\mathcal{A}' = \langle \Sigma, Q, \delta, Q_0, \alpha' \rangle$ satisfies $\gamma(\mathcal{A}') = \beta(\mathcal{A})$. Thus, we can apply the limit operator γ to an NFW obtained by only modifying the set of accepting states of $\mathcal{A}$, and get the same language obtained by applying to $\mathcal{A}$ the limit operator β. Finally, we say that β is γ-type if all NFWs $\mathcal{A}$ are (β, γ)-type.

The exponential lower bounds in Theorems 4 and 5 imply that *lim* and *plim* are not *co-lim*-type, and that *co-lim* and *plim* are not *lim*-type. Two lower bounds that we miss are from *co-lim* and *lim* to *plim*. Below we show that polynomial translations to a *plim* NFW, even if exist, cannot be done in general on the same structure.

Theorem 6. *lim and co-lim are not plim-type.*

Proof: We start with limit. Consider the NFW $\mathcal{A}$ in Figure 2. Note that $lim(A) = a^+b^\omega$. As such, $ab^\omega \in lim(\mathcal{A})$. It is not hard to see that if we change the set of accepting states in such a way that only a finite number of prefixes of ab^ω are rejected, then all prefixes of the word a^ω are accepted. Hence, no NFW $\mathcal{A}'$ with $plim(\mathcal{A}') = a^+b^\omega$ can be defined on the same structure as $\mathcal{A}$.

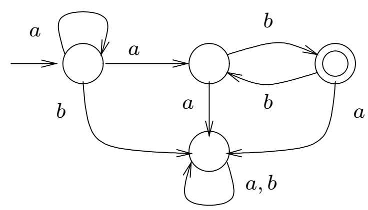

Fig. 2. An NFW $\mathcal{A}$ with $lim(\mathcal{A}) = a^+b^\omega$ and $co\text{-}lim(\mathcal{A}) = \Sigma^\omega \setminus a^+b^\omega$.

We now turn to co-limit. Consider again the NFW $\mathcal{A}$. Note that $co\text{-}lim(\mathcal{A}) = \Sigma^\omega \setminus a^+b^\omega$. As such, $b^\omega \in co\text{-}lim(\mathcal{A})$. Thus, an NFW $\mathcal{A}'$ with $plim(\mathcal{A}') = co\text{-}lim(\mathcal{A})$ should reject only finitely many prefixes of b^ω. The only way for $\mathcal{A}'$ with the same structure as $\mathcal{A}$ to do so, is to let the sink be accepting. Then, however, all but two prefixes of the word aab^ω are also accepted, contradicting the fact that $aab^\omega \notin co\text{-}lim(A)$. □

5 Succinctness in Safety, co-Safety, and Bounded Properties

The study of limit operators checks behaviors in the limit. In this section we restrict attention to properties that refer to a bounded prefix of the computation. We show that even though such properties can be recognized by automata of very restricted type, almost all the lower bounds that hold in the general case, hold also in this restricted case. We consider *safety*, *co-safety*, and *bounded* properties. We start with some definitions. Let L be a language of infinite words over Σ. A finite word $x \in \Sigma^*$ is a *bad prefix* for L if for all infinite words $y \in \Sigma^\omega$, the concatenation $x \cdot y$ of x and y is not in L. Thus, a bad prefix for L is a finite word that cannot be extended into an infinite word in L. In a similar fashion, a finite word $x \in \Sigma^*$ is a *good prefix* for L, if for all infinite words $y \in \Sigma^\omega$, the concatenation $x \cdot y$ of x and y is in L.

Definition 2. *A language L is*

- *a* safety language *if every word not in L has a bad prefix,*
- *a* co-safety language *if every word in L has a good prefix,*
- *a* bounded *language if it is both safety and co-safety.*

Note that a language L is bounded iff every word $w \in \Sigma^\omega$ has either a good or a bad prefix [11]. Accordingly, evaluation of bounded properties can be done by traversing a bounded prefix of the computation, making bounded properties suitable for bounded model checking [6].

From an automata-theoretic point of view [24, 11], safety properties correspond to looping automata (Büchi automata where all states are accepting), co-safety properties to co-looping automata (Büchi automata with a single accepting state that is a loop), and bounded properties to cycle-free automata (automata whose transition function contains no cycle, except possibly a self loop in an accepting sink). Accordingly, we expect the differences between the limit operators to vanish.

Examining the results in the previous sections, however, we see that most of the succinctness results established for the general case were actually proven with a bounded language, making them valid also for the bounded fragment. An exception is Theorem 3, which makes a heavy use of the unbounded nature of the language L_n. Nevertheless, the language we have used in Theorem 5 is bounded and cannot be recognized by a sub-exponential NCW or NBW. Hence, we also have exponential lower bounds for the co-lim NFW $\rightarrow$ NCW/NBW and plim NFW $\rightarrow$ NCW/NBW transformations in the bounded case.

In some cases, however, safetyness (and hence also boundedness) makes things simpler. We start with the *plim*-typeness of *lim* NFWs:

Lemma 2. *When restricted to safety properties, lim is plim-type.*

Proof: Consider an NFW $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$ such that $lim(\mathcal{A})$ is a safety language. We prove that there is $\mathcal{A}' = \langle \Sigma, Q, \delta, Q_0, \alpha' \rangle$ such that $plim(\mathcal{A}') = lim(\mathcal{A})$.

By Theorem 2, we have that $lim(\mathcal{A}) = \bigcup_{p \in Q} L_{Q_0,p} \cdot (L_{p,p} \cap L_{p,\alpha})^\omega$. Let $S \subseteq Q$ be the set of states in $\mathcal{A}$ that are not reachable from Q_0 or from which no state p such that $L_{p,p} \cap L_{p,\alpha} \neq \emptyset$ is reachable. The NFW $\mathcal{A}'$ is obtained from $\mathcal{A}$ by defining the accepting set to be $\alpha' = Q \setminus S$. In the full version, we prove that $lim(\mathcal{A}) = lim(\mathcal{A}') = plim(\mathcal{A}')$. □

Note that, in the construction above, removing all the states in S from $\mathcal{A}'$ does not change the language $lim(\mathcal{A}')$, and results in an NFW in which all the states are accepting. It is not hard to prove that if $\mathcal{A}$ is an NFW in which all states are accepting, then it is always the case that $lim(\mathcal{A}) = L(\mathcal{A}_{inf})$, where $\mathcal{A}_{inf}$ is $\mathcal{A}$ when viewed as a Büchi automaton. Thus, in the case of safety properties, the above simple linear construction gives a transformation from *lim* NFWs to NBWs, and the cubic construction in Section 3 can be circumvented. In addition, if $\mathcal{A}$ is a looping NBW, then it is always the case that $L(\mathcal{A}) = lim(\mathcal{A}_{fin})$, where $\mathcal{A}_{fin}$ is $\mathcal{A}$ when viewed as an NFW. Hence, we have the following.

Theorem 7. *When restricted to safety properties, the transformations from an NBW to a limit NFW and from a limit NFW to an NBW are linear.*

It is not hard to see that *co-lim* is not *plim* type also in the context of bounded properties. Indeed, the non-typeness there has to do with the non-typeness of NFW complementation (that is, the fact that NFW complementation cannot always be done on top of the same structure). More difficult is to show that *lim* is not *plim* type for co-safety properties:

Lemma 3. *lim is not plim-type, even for co-safety properties.*

Proof: Consider the NFW $\mathcal{A}$ in Figure 3. Note that $lim(\mathcal{A}) = \Sigma^\omega \setminus \{a^\omega, b^\omega\}$. Observe that there is no way to define the accepting states in such a way that only a finite number of prefixes of ab^ω are rejected, while maintaining the requirement that infinitely many prefixes of a^ω and b^ω are rejected. □

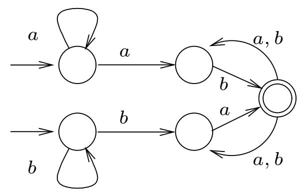

Fig. 3. An NFW $\mathcal{A}$ with $lim(\mathcal{A}) = \Sigma^\omega \setminus \{a^\omega, b^\omega\}$.

6 Discussion

In Figure 4, we summarize most of our results. All the lower bounds in the table, with the exception of *plim* NFW $\to$ *co-lim* NFW, are tight.

	lim NFW	*co-lim* NFW	*plim* NFW
DBW	$2^{\Omega(n)}$ [Theorem 1]		
DCW	$2^{\Omega(n)}$ [Theorem 1]		
NBW	$O(n^3)$ [Corollary 1]	$2^{\Omega(n)}$ [Theorem 3]	
NCW	?	$2^{\Omega(n)}$ [Theorem 3]	
lim NFW	-	$2^{\Omega(n)}$ [Theorem 5]	$2^{\Omega(\sqrt{n})}$ [Theorem 5]
co-lim NFW	$2^{\Omega(n)}$ Theorem 4	-	$2^{\Omega(n)}$ [Theorem 4]
plim	? (not type [Theorem 6], type for safety [Lemma 2])	? (not type [Theorem 6])	-

Fig. 4. Main Results Summary.

Below we discuss the cases that were left open and our efforts to solve them. In addition to the results described in Figure 4, Theorem 4 describes an exponential lower bound for the translation of NBW and NCW to *co-lim* NFW. A translation of an NBW to a *lim* NFW was left open (the considerations for the NCW to *plim* NFW case are similar). Recall that an NBW $\mathcal{A}$ can be transformed to an NFW $\mathcal{A}'$ with $L(\mathcal{A}) = lim(\mathcal{A}')$ iff $L(\mathcal{A})$ can be accepted by a DBW. As demonstrated in Section 1, even in cases where the transformation is possible, the NFW $\mathcal{A}'$ may not be defined on the same structure as $\mathcal{A}$. This follows from the fact that different prefixes of an infinite word may follow different accepting runs, and there is no guarantee that these runs can be merged into a single infinite accepting run. Since a deterministic automaton has a single run on every input, it does not suffer from this problem, and indeed the transformation from a DBW to a *lim* DFW can be done on the same structure. This suggests an exponential upper bound for the NBW to *lim* NFW transformation, and also hints that an exponential lower bound may follow from the exponential lower bound on determinization. On the other hand, similar considerations apply to the reverse transformation — of a *lim* NFW to an NBW, and there, as we have seen in Section 3, we are able to avoid determinization and have a polynomial transformation. Another related observation is that an exponential lower bound, if exists, cannot follow easily from the exponential lower bound on NFW determinization. Indeed, as we have noted in Section 5, the transformation from an NBW to a *lim* NFW is linear for safety languages (and hence also for ω-regular languages that are based on regular languages).

It follows that the most promising direction for obtaining an exponential lower bound in the NBW to *lim* NFW case is one that makes use of the combinatorial properties of the Büchi condition and relies on the $2^{O(n \log n)}$ lower bound for NBW determinization. A natural candidate for a family of languages with which a lower bound can be proved is therefore the family L_n defined by Michel in the context of NBW complementation and later used by Löding in the context of NBW determinization [19, 15]. As we show, however, in the full version, even though there is no DFW $\mathcal{A}$ with less than $2^{\Omega(n \log n)}$ states such that $lim(\mathcal{A}) = L_n$, there is an NFW $\mathcal{A}$ with only $O(n^2)$ states, such that $lim(\mathcal{A}) = L_n$. The NFW $\mathcal{A}$ belongs to a special class of NFWs we call *continuous* NFWs. The main property of continuous NFW is that the language they accept as NBWs coincides with their limit. I.e., for a continuous NFW, the different accepting runs over prefixes of an infinite word do merge into an accepting run on the infinite word. Formally, we have the following. Consider an NFW $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$. For sets $P, S \subseteq Q$, we use $L^{\neg\alpha}_{P,S}$ to denotes the language of all words that $\mathcal{A}$ can read along a run disjoint from α that start in P and ends in S (the runs may start and/or end in a state in α, but states of α are not allowed in the middle of the run).

Definition 3. *An NFW $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$ is* continuous *if the languages $lim(L^{\neg\alpha}_{Q_0,\alpha})$ and $lim(L^{\neg\alpha}_{\alpha,\alpha})$ are both empty.*

In the full version, we show that all DFWs are continuous and that if $\mathcal{A}$ is a continuous NFW, then $L(\mathcal{A}) = lim(\mathcal{A}_{inf})$, when $\mathcal{A}_{inf}$ is $\mathcal{A}$ viewed as an NBW. As detailed in the full version, the proof makes use of the characterization described for limit languages in Theorem 2 — the characterization that was the key to the polynomial *lim* NFW to NBW transformation. Our conjecture is that a polynomial translation from NBW to lim NFW is possible also in the general case.

We now discuss another problem that was left open: the transformation from a *lim* NFW to a *plim* NFW. Note that a "*lim* to *plim*" transformation is possible only for languages that are recognizable by both DBW and DCW, and hence are also recognizable by a deterministic *weak* automaton [4] (a similar challenge is the "*lim* to NCW" transformation, which was also left open). Our initial conjecture was that *lim* is *plim* type. The examples in Theorem 6 and Lemma 3 have made us realize that the fact a *lim* NFW does not have to eventually accept all prefixes enables it to classify states that are the only destination of some prefixes as rejecting ones. As demonstrated in the examples, this enables the NFW to use these states in cycles that are traversed along runs of words that are not in the limit. On the one hand, this points to an advantage of *lim* NFWs over *plim* NFWs. Note that a dual advantage enabled us to prove an exponential lower bound in the reverse "*plim* to *lim*" transformation. On the other hand, this advantage of *lim* seems to help it only with a bounded number of prefixes. Technically, it may be (and this is the case in both examples), that by unwinding the graph of the NFW some fixed number of times, we get a new NFW that is *plim* type. Thus, here too, our conjecture is that a polynomial transformation exists.

Acknowledgment We thank Annielo Murano for many helpful discussions.

References

1. B. Alpern and F.B. Schneider. Defining liveness. *IPL*, 21:181–185, 1985.
2. R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singerman, A. Tiemeyer, M.Y. Vardi, and Y. Zbar. The ForSpec temporal logic: A new temporal property-specification logic. In *8th TACAS*, LNCS 2280, pages 296–211, 2002.
3. I. Beer, S. Ben-David, C. Eisner, D. Fisman, A. Gringauze, and Y. Rodeh. The temporal logic Sugar. In *Proc. 13th CAV*, LNCS 2102, pages 363–367, 2001.
4. B. Boigelot, S. Jodogne, and P. Wolper. On the use of weak automata for deciding linear arithmetic with integer and real variables. In *IJCAR*, LNCS 2083, pages 611–625, 2001.
5. J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. International Congress on Logic, Method, and Philosophy of Science. 1960*, pages 1–12, Stanford, 1962.
6. E. M. Clarke, A. Bierea, R. Raimi, and Y. Zhu. Bounded model checking using satisfiability solving. *Formal Methods in System Design*, 19(1):7–34, 2001.
7. Accellera Organization Inc. http://www.accellera.org.
8. S.C. Krishnan, A. Puri, and R.K. Brayton. Deterministic ω-automata vis-a-vis deterministic Büchi automata. In *Algorithms and Computations*, LNCS 834, pages 378–386, 1994.
9. O. Kupferman, G. Morgenstern, and A. Murano. Typeness for ω-regular automata. In *Proc. 2nd ATVA*, LNCS 3299, pages 324–338, 2004.
10. O. Kupferman, S. Safra, and M.Y. Vardi. Relating word and tree automata. In *Proc. 11th LICS*, pages 322–333, DIMACS, June 1996.
11. O. Kupferman and M.Y. Vardi. On bounded specifications. In *Proc. 8th LPAR*, LNCS 2250, pages 24–38, 2001.
12. O. Kupferman and M.Y. Vardi. From linear time to branching time. *ACM TOCL*, 6(2):273–294, April 2005.
13. R.P. Kurshan. *Computer Aided Verification of Coordinating Processes*. Princeton Univ. Press, 1994.
14. L.H. Landweber. Decision problems for ω–automata. *Mathematical Systems Theory*, 3:376–384, 1969.
15. C. Löding. Optimal bounds for the transformation of omega-automata. In *Proc. 19th FSTTCS*, LNCS 1738, pages 97–109, 1999.
16. C. Löding. Efficient minimization of deterministic weak omega-automata. *IPL*, 79(3):105–109, 2001.
17. R. McNaughton. Testing and generating infinite sequences by a finite automaton. *I& C*, 9:521–530, 1966.
18. A.R. Meyer and M.J. Fischer. Economy of description by automata, grammars, and formal systems. In *Proc. 12th SSAT*, pages 188–191, 1971.
19. M. Michel. Complementation is more difficult with automata on infinite words. CNET, Paris, 1988.
20. S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *TCS*, 32:321–330, 1984.
21. M.O. Rabin. Decidability of second order theories and automata on infinite trees. *Transaction of the AMS*, 141:1–35, 1969.
22. S. Safra. On the complexity of ω-automata. In *Proc. 29th FOCS*, pages 319–327, 1988.
23. S. Safra and M.Y. Vardi. On ω-automata and temporal logic. In *Proc. 21st ACM STOC*, pages 127–137, 1989.
24. A.P. Sistla. Safety, liveness and fairness in temporal logic. *Formal Aspects of Computing*, 6:495–511, 1994.
25. W. Thomas. Automata on infinite objects. *Handbook of Theoretical Computer Science*, pages 133–191, 1990.
26. M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *I& C*, 115(1):1–37, 1994.