On Repetition Languages

Orna Kupferman
School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel
orna@cs.huji.ac.il

Ofer Leshkowitz
School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel
ofer.leshkowitz@mail.huji.ac.il

Abstract

A regular language R of finite words induces three *repetition languages* of infinite words: the language $\lim(R)$, which contains words with infinitely many prefixes in R, the language ∞R, which contains words with infinitely many disjoint subwords in R, and the language R^ω, which contains infinite concatenations of words in R. Specifying behaviors, the three repetition languages provide three different ways of turning a specification of a finite behavior into an infinite one. We study the expressive power required for recognizing repetition languages, in particular whether they can always be recognized by a deterministic Büchi word automaton (DBW), the blow up in going from an automaton for R to automata for the repetition languages, and the complexity of related decision problems. For $\lim R$ and ∞R, most of these problems have already been studied or are easy. We focus on R^ω. Its study involves some new and interesting results about additional repetition languages, in particular $R^\#$, which contains exactly all words with unboundedly many concatenations of words in R. We show that R^ω is DBW-recognizable iff $R^\#$ is ω-regular iff $R^\# = R^\omega$, and there are languages for which these criteria do not hold. Thus, R^ω need not be DBW-recognizable. In addition, when exists, the construction of a DBW for R^ω may involve a $2^{O(n \log n)}$ blow-up, and deciding whether R^ω is DBW-recognizable, for R given by a nondeterministic automaton, is PSPACE-complete. Finally, we lift the difference between $R^\#$ and R^ω to automata on finite words and study a variant of Büchi automata where a word is accepted if (possibly different) runs on it visit accepting states unboundedly many times.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects

Keywords and phrases Büchi automata, Expressive power, Succinctness

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.59

Related Version A full version of the paper is available at `http://www.cs.huji.ac.il/~ornak/publications/mfcs20a.pdf`.

Acknowledgements We thank Michael Kaminski for asking us whether R^ω is DBW-recognizable for every regular language R.

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 [6, 14, 17]. Today, automata on infinite objects are used for specification, verification, and synthesis of nonterminating systems. The automata-theoretic approach reduces questions about systems and their specifications to questions about automata [11, 22], and is at the heart of many algorithms and tools. Industrial-strength property-specification languages such as the IEEE 1850 Standard for Property Specification Language (PSL) [7] include regular expressions and/or automata, making specification and verification tools that are based on automata even more essential and popular.

45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020).
Editors: Javier Esparza and Daniel Král'; Article No. 59; pp. 59:1–59:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One way to classify an automaton is by the type of its branching mode, namely whether it is *deterministic*, in which case it has a single run on each input word, or *nondeterministic*, in which case it may have several runs, and the input word is accepted if at least one of them is accepting. A run of an automaton on finite words is accepting if it ends in an accepting state. A run of an automaton on infinite words does not have a final state, and acceptance is determined with respect to the set of states visited infinitely often during the run. Another way to classify an automaton on infinite words 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 [6].

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 (DBWs) are strictly less expressive than nondeterministic Büchi automata (NBWs). That is, there exists a language L over infinite words such that L can be recognized by an NBW but cannot be recognized by a DBW. The different classes also differ in their *succinctness*. For example, while translating a nondeterministic automaton on finite words (NFW) into a deterministic one (DFW) is always possible, the translation may involve an exponential blow-up [19].

There has been extensive research on expressiveness and succinctness of automata on infinite words [21, 9]. Beyond the theoretical interest, the research has received further motivation with the realization that many algorithms, like synthesis and probabilistic model checking, need to operate on deterministic automata [5, 4], as well as the discovery that many natural specifications correspond to DBWs. In particular, it is shown in [10] 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. Since AFMC is as expressive as weak alternating automata and the weak monadic second-order theory of trees [18, 16, 3], this relates DBWs also with them.

Proving that NBWs are more expressive than DBWs, Landweber characterized languages that are DBW-recognizable as these that are the *limit* of some regular language on finite words. Formally, for an alphabet Σ and a language $R \subseteq \Sigma^*$, we define $\lim(R)$ as the set of infinite words in Σ^ω that have infinitely many prefixes in R. For example, if $R = (0+1)^* \cdot 0$, namely the set of finite words over $\{0, 1\}$ that end with a 0, then $\lim(R) = ((0+1)^* \cdot 0)^\omega$, namely the set of words with infinitely many 0's. On the other hand, we cannot point to a language R such that $\lim(R)$ is the set of all words with only finitely many 0's. Landweber proved that a language $L \subseteq \Sigma^\omega$ is DBW-recognizable iff there is a regular language R such that $L = \lim(R)$ [12].

Beyond the limit operator, another natural way to obtain a language of infinite words from a language R of finite words is to require the words in R to repeat infinitely often. This actually induces two "repetition languages". The first is ∞R, where $w \in \infty R$ iff w contains infinitely many disjoint subwords in R. Formally, $\infty R = \{\Sigma^* \cdot w_1 \cdot \Sigma^* \cdot w_2 \cdot \Sigma^* \cdot w_3 \cdots : w_i \in R \text{ for all } i \geq 1\}$. The second is R^ω, where $w \in R^\omega$ iff w is an infinite concatenation of words in R. Formally, $R^\omega = \{w_1 \cdot w_2 \cdot w_3 \cdots : w_i \in R \text{ for all } i \geq 1\}$. For example, for the language $R = (0+1)^* \cdot 0$ above, we have $\lim(R) = \infty R = R^\omega = ((0+1)^* \cdot 0)^\omega$. In order to see that the three repetition languages may be different, consider the language $R = 0 \cdot (0+1)^* \cdot 0$, namely of all words that start and end with 0. Now, $\lim(R) = 0 \cdot ((0+1)^* \cdot 0)^\omega$, $\infty R = ((0+1)^* \cdot 0)^\omega$, and $R^\omega = 0 \cdot ((0+1)^* \cdot 0 \cdot 0)^\omega$. When specifying on-going behaviors, the three repetition languages induce three different ways for turning a finite behavior into an infinite one. For example, if $R = \mathit{call} \cdot \mathit{true}^* \cdot \mathit{return}$ describes a sequence of events that starts with a call and ends with a return, then $\lim R$ describes behaviors that start with a call followed by infinitely many returns, ∞R behaviors with infinitely many calls and returns, and R^ω behaviors with infinitely many successive calls and returns.

In this paper we study expressiveness, succinctness, and complexity of repetition languages. We start with expressiveness, where we examine which of the repetition languages are ω-regular, and for those that are ω-regular, whether they are also DBW-recognizable. By [12], for $\lim(R)$ the answer is positive – it is DBW-recognizable for all regular languages R. For a *finite* regular language R, we show that $R^\omega = \lim(R^*)$, implying a positive answer too. Our main result is a negative answer in the general R^ω case: we point to a regular language R such that R^ω is not DBW-recognizable. In order to find such a language, we study repetition languages in general, and introduce the language $R^\# = \{w \in \Sigma^\omega : \text{for all } i \geq 1 \text{ there exists a prefix of } w \text{ in } R^i\}$, namely the language of exactly all words with unboundedly many concatenations of words in R. As detailed below, $R^\#$ is strongly related to R^ω and turns out to be also strongly related to our question. We show that when $R^\#$ is ω-regular, then $R^\# = R^\omega$, in which case, by Landweber's characterization of DBW-recognizable languages as countable intersections of open sets in the product topology over Σ^ω, both are DBW-recognizable. In other words, we show (Theorem 5) that $R^\#$ is ω-regular iff $R^\# = R^\omega$ iff R^ω is DBW-recognizable.

The above characterization enables us to point to a language R that does not satisfy the three criteria (Theorem 9). In short, $R = \$ + (0 \cdot \{0, 1, \$\}^* \cdot 1)$. It is easy to see that for every word $w \in R^\omega$, if w contains infinitely many 1's, then w contains infinitely many 0's. Hence, the word $w = 011\$1\$\$1\$\$\$1\$\$\$\$ \cdots = 0 \cdot \prod_{k=0}^{\infty} 1\$^k$ is not in R^ω, yet for all $i \geq 1$, its prefix $0 \cdot \prod_{k=0}^{i} 1\$^k = (0 \cdot (\prod_{k=0}^{i-1} 1\$^k) \cdot 1) \cdot \$^i$ is in R^i, and so $w \in R^\#$. It follows that $w \in R^\# \setminus R^\omega$, which by our characterization implies that R^ω is not DBW-recognizable. We also study the problem of deciding, given an NFW $\mathcal{A}$, whether $L(\mathcal{A})^\omega$ is DBW-recognizable, and show that it is PSPACE-complete. We lift the difference between $R^\#$ and R^ω to automata on finite words and define the *#-language* of a Büchi automaton $\mathcal{A}$ as the set of words w such that for all $i \geq 1$, there is a run of $\mathcal{A}$ on w that visits the set of accepting states at least i times. We show that the #-language of $\mathcal{A}$ is ω-regular iff the #-language of $\mathcal{A}$ coincides with its ω-regular language, iff $L(\mathcal{A})$ is DBW-recognizable.

We continue and study the size of automata for the repetition languages. We consider the cases R is given by a DFW or an NFW, and the automaton for the repetition language is a DBW or an NBW. By [12], going from a DFW for R to a DBW for $\lim(R)$ involves no blow up – we only have to view the DFW as a Büchi automaton. We show that the cases of ∞R and R^ω are more complicated, and involve a $2^{O(n)}$ and a $2^{O(n \log n)}$ blow-up, respectively. Beyond the relevancy to our study, the family of languages we use is a new witness to the known lower bound for NBW determinization [13]. The succinctness analysis for the cases the automata for the repetition languages are nondeterministic are much easier, as we show that, except for the case of $\lim(R)$, simple constructions with no blow-ups are possible, even when we start with an NFW for R. For the case of $\lim(R)$, going from an NFW for R to an NBW for $\lim(R)$ is not trivial and the best known upper bound is $O(n^3)$ [2]. Our results are summarized in Section 7.

Due to lack of space, some proofs are omitted and can be found in the full version, in the authors' URLs.

2 Preliminaries

2.1 Automata

An *alphabet* Σ is a finite set of letters. A *word* over Σ is a finite or infinite sequence $w = \sigma_1, \sigma_2, \sigma_3, \cdots$ of letters from Σ. We use $|w|$ to denote the length of w, with $|w| = \infty$ for an infinite word w. For $1 \leq i \leq |w|$, we use $w[i]$ to denote σ_i, that is, the i-th letter in w, and for $1 \leq i \leq j \leq |w|$, we use $w[i,j]$ to denote the *infix* $\sigma_i, \sigma_{i+1}, \cdots, \sigma_j$ of w. We use Σ^* and Σ^ω to denote the set of all finite and infinite words over Σ, respectively. For two words $x \in \Sigma^*$ and $y \in \Sigma^* \cup \Sigma^\omega$, we use $x \cdot y$ to denote the *concatenation* of x and y. We say that x is a *prefix* of a w, denoted $x \prec w$, if there is $1 \leq i \leq |w|$ such that $x = w[1,i]$. Equivalently, if $x \neq \varepsilon$, and there is $y \in \Sigma^* \cup \Sigma^\omega$, such that $x \cdot y = w$. Thus, $y = [i+1, |w|]$, and we call it a *suffix* of w. Note that we do not consider the empty word ε as a prefix of a word.

A *nondeterministic automaton* is $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$, where Σ is a finite 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. Intuitively, $\delta(q, \sigma)$ is the set of states $\mathcal{A}$ may move to when reading the letter σ from state q. Formally, a *run* of $\mathcal{A}$ on a word w is the function $r : \{i \in \mathbb{N}_0 : 0 \leq i \leq |w|\} \to Q$, such that $r(0) \in Q_0$, i.e., the run starts from an initial state, and for all $i \geq 0$, we have that $r(i+1) \in \delta(r(i), \sigma_{i+1})$, i.e., the run obeys the transition function. Note that as $\mathcal{A}$ may have several initial states and the transition function may specify several possible successor states, the automaton $\mathcal{A}$ may have several runs on w. If $|Q_0| = 1$ and for all $q \in Q$ and $\sigma \in \Sigma$, it holds that $|\delta(q, \sigma)| = 1$, then $\mathcal{A}$ has a single run on w, and we say that $\mathcal{A}$ is *deterministic*. We sometimes refer to a run r also as a sequence of states; that is, $r = r(0), r(1), \ldots \in Q^{|w|+1}$.

When $\mathcal{A}$ runs on finite words, the run r is finite, and it is *accepting* iff it ends in an accepting state, thus $r(|w|) \in \alpha$. When $\mathcal{A}$ runs on infinite words, acceptance depends on the set $\inf(r)$, of the states that r visits infinitely often. Formally $\inf(r) = \{q \in Q :$ for infinitely many $i \in \mathbb{N}$, we have that $r(i) = q\}$. As Q is finite, the set $\inf(r)$ is guaranteed not to be empty. In *Büchi* automata, the run r is *accepting* iff $\inf(r) \cap \alpha \neq \varnothing$. Otherwise, r is *rejecting*. The automaton $\mathcal{A}$ accepts a word w if there exists an accepting run r of $\mathcal{A}$ on w. The *language* of $\mathcal{A}$, denoted $L(\mathcal{A})$, is the set of words that $\mathcal{A}$ accepts. We also say that $\mathcal{A}$ *recognizes* $L(\mathcal{A})$.

We use three letter acronyms in {D,N} × {F,B} × {W} to denote classes of word automata. The first letter indicates whether the automaton is deterministic or nondeterministic, and the second letter indicates whether it is an automaton on finite words or a Büchi automaton on infinite words. For example, DBW is a deterministic Büchi automaton.

Throughout the paper, we use R and L to represent languages of finite and infinite words, respectively. A language $R \subseteq \Sigma^*$ is *finite* if $|R| < \omega$, where $|R|$ is the cardinality of R as a set. A language $R \subseteq \Sigma^*$ is *regular* if there is an NFW that recognizes R. Likewise, a language $L \subseteq \Sigma^\omega$ is *ω-regular* if there is an NBW that recognizes L. We sometimes refer to the three-letter acronyms as describing sets of languages, thus NBW is also the set of ω-regular languages, and DBW is its subset of languages recognizable by DBW.

2.2 Repetition languages

Consider a language $R \subseteq \Sigma^*$, and assume $\varepsilon \notin R$. We refer to R as the *base language* and define the following *repetition languages* of words induced by R. We start with languages of finite words:

1. For $i \geq 0$, we define $R^i = \{w_1 \cdot w_2 \cdots w_i : w_j \in R \text{ for all } 1 \leq j \leq i\}$.
2. $R^* = \bigcup_{i \geq 0} R^i$.
3. $R^+ = \bigcup_{i \geq 1} R^i$.

We continue with languages of infinite words:

3. $\lim(R) = \{w \in \Sigma^\omega : \ w[1,i] \in R \text{ for infinitely many } i\text{'s}\}$.
4. $\infty R = \{\Sigma^* \cdot w_1 \cdot \Sigma^* \cdot w_2 \cdot \Sigma^* \cdot w_3 \cdots : \ w_i \in R \text{ for all } i \geq 1 \ \}$.
5. $R^\omega = \{w_1 \cdot w_2 \cdot w_3 \cdots : \ w_i \in R \text{ for all } i \geq 1\}$.
6. $R^\# = \{w \in \Sigma^\omega : \text{for all } i \geq 1 \text{ there exists } j \geq 1 \text{ such that } w[1,j] \in R^i\}$.

Thus, R^i, R^*, R^+, and R^ω are the standard bounded, finite, finite and positive, and infinite concatenation operators. Then, $\lim(R)$ contains exactly all infinite words with infinitely many prefixes in R, and ∞R contains exactly all infinite words with infinitely many disjoint infixes in R. Finally, $R^\#$ contains exactly all words with prefixes with unboundedly many concatenations of words in R. The language $R^\#$ may seem equivalent to R^ω, and the difference between R^ω and $R^\#$ is in fact one of our main results.

▶ **Example 1.** Let $R = (0+1)^* \cdot 0$. Then, $\lim(R) = \infty R = R^\omega = R^\# = \infty 0$.

Now, let $R = \{0^n \cdot 1^m, 1^n \cdot 0^m : 0 \leq m \leq n\}$. While R is not regular, we have that $\lim(R) = \{0^\omega, 1^\omega\}$ and $\infty(R) = R^\omega = R^\# = \{0,1\}^\omega$ are in DBW.

Finally, for all $R \subseteq \Sigma^*$, we have $R^\omega \subseteq \lim(R^*)$ and $R^\omega \subseteq \infty R$. Thus, $R^\omega \subseteq \lim(R^*) \cap \infty R$. One may suspect that $R^\omega = \lim(R^*) \cap \infty R$. As a counterexample, consider $R = 0 \cdot (0+1)^* \cdot 0$. Then, $R^\omega = 0 \cdot ((0+1)^* \cdot 0 \cdot 0)^\omega$, $\lim(R^*) = 0 \cdot ((0+1)^* \cdot 0)^\omega$, and $\infty R = ((0+1)^* \cdot 0)^\omega = \infty 0$. Thus, the word $0 \cdot (1 \cdot 0)^\omega$ is in $\lim(R^*) \cap (\infty R)$ but is not in R^ω. ◀

As another warm up, we state the following lemma, which would be helpful in the sequel.

▶ **Lemma 2.** *Consider languages $R \subseteq \Sigma^*$ and $P \subset \Sigma^\omega$ such that $\varepsilon \notin R$. If $P \subseteq R \cdot P$, then $P \subseteq R^\omega$.*

Proof. Consider a word $w_0 \in P$. Since $P \subseteq R \cdot P$, then $w_0 = x_1 \cdot w_1$, for some $x_1 \in R$ and $w_1 \in P$. Have defined $x_1, \ldots, x_i \in R$ and $w_i \in P$, such that $w_0 = x_1 \cdots x_i \cdot w_i$, we can continue and define $x_{i+1} \in R$ and $w_{i+1} \in P$ such that $w_i = x_{i+1} \cdot w_{i+1}$. Overall, we have defined $\{x_i\}_{i=1}^\infty \subseteq R$ such that $w_0 = x_1 \cdot x_2 \cdot x_3 \cdots$. Hence, $w_0 \in R^\omega$, and we are done. ◀

Note that if $\varepsilon \in R$, then $P \subseteq R \cdot P$ trivially holds for all $P \subseteq \Sigma^\omega$, whereas possibly $P \nsubseteq R^\omega$. Also, if $\varepsilon \in R$, then ∞R, R^ω, and $R^\#$ as defined above include also finite words, in particular ε is a member of all of those languages. In order to circumvent the technical issues that the above entails, for $R \subseteq \Sigma^*$ such that $\varepsilon \in R$, we define $\infty R = \infty(R \setminus \{\varepsilon\})$, $R^\omega = (R \setminus \{\varepsilon\})^\omega$, and $R^\# = (R \setminus \{\varepsilon\})^\#$, and accordingly assume, throughout the paper, that $\varepsilon \notin R$.

We conclude the preliminaries with the case the base language R is finite. As we shall see, then $R^\omega = R^\# = \lim(R^*)$, implying that they are all in DBW.

▶ **Theorem 3.** *For every finite language $R \subseteq \Sigma^*$, we have that $R^\omega = R^\# = \lim(R^*)$.*

Proof. Consider a finite language $R \subseteq \Sigma^*$. We prove that $R^\omega \subseteq \lim(R^*) \subseteq R^\# \subseteq R^\omega$. First, it is easy to see, regardless of R being finite, that $R^\omega \subseteq \lim(R^*)$.

We prove that $\lim(R^*) \subseteq R^\#$. Clearly $R^{n+1} \cdot \Sigma^\omega \subseteq R^n \cdot \Sigma^\omega$, and thus we only need to show that for all $w \in \lim(R^*)$, we have that $w \in R^n \cdot \Sigma^\omega$ for infinitely many n's. Since R is finite, there exists some $k \geq 1$ such that for all $x \in R$, we have that $|x| \leq k$. It follows that for all $x \in R^*$, if $|x| \geq m \cdot k$, then $x \in R^n$ for some $n \geq m$. Consider some word $w \in \lim(R^*)$. By definition, w has infinitely many prefixes in R^*, thus for all $m \geq 1$, there exists a prefix $x \in R^*$ of w such that $|x| \geq m \cdot k$. Hence, $x \in R^n$ for some $n \geq m$, implying that $w \in R^n \cdot \Sigma^\omega$ for infinitely many n's, and we are done.

It is left to prove that $R^\# \subseteq R^\omega$. Consider a word $w \in R^\#$. Intending to use König's Lemma, we build a tree with set of nodes $V = \{(x,i) : x \prec w \text{ and } x \in R^i\}$. Since $w \in R^\#$, the set V is infinite. As the parent of a node $(x, i+1) \in V$, we set some $(y,i) \in V$ that satisfies $x = y \cdot z$ for some $z \in R$. Since $x \in R^{i+1}$, such a prefix y exists. Note that there might be several y's and only a single (y,i) is chosen to be the parent of $(x, i+1)$. Observe that all nodes (x,i) are connected to $(\varepsilon, 0)$ by a single path of length i, and thus we have defined an infinite tree above V. The out degree of each node is bounded by $|R| < \infty$. Hence, by König's Lemma, the tree has an infinite path $\pi = \langle(\varepsilon, 0), (x_1, 1), (x_2, 2), \ldots\rangle$. By construction, for all $i \geq 0$ there exists some y_i such that $x_{i+1} = x_i \cdot y_i$ and $y_i \in R$. It follows that $w = y_1 \cdot y_2 \cdots$, and hence $w \in R^\omega$, and we are done. ◀

For every regular language $R \subseteq \Sigma^*$, the language R^* is regular. Hence, by [12], the language $\lim(R^*)$ is in DBW, and so Theorem 3 implies the following.

▶ **Corollary 4.** *For every finite language $R \subseteq \Sigma^*$, we have that R^ω and $R^\#$ are in DBW.*

As we shall see in Section 3, the case of an infinite base language R is much more difficult.

3 Expressiveness

In this section we examine which of the repetition languages are ω-regular, and for these that are ω-regular, whether they are also DBW-recognizable. Note that going in the other direction need not be possible. For example, the language $L = 0 \cdot 1^\omega$ is DBW-recognizable, but there is no regular language R such that $L = \infty R$, $L = R^\#$, or $L = R^\omega$. By [12], a language $L \subseteq \Sigma^\omega$ is in DBW iff there exists a regular language $R \subseteq \Sigma^*$ such that $L = \lim(R)$. In particular, this means that for every $R \subseteq \Sigma^*$ regular, we have that $\lim(R) \in$ DBW. We study this question for ∞R, R^ω, and $R^\#$.

It is well known that for every regular language R, the language R^ω is ω-regular. This follows, for example, from the translation of ω-regular expressions to NBWs. Studying whether R^ω is always DBW-recognizable is much harder, and is our main result:

▶ **Theorem 5.** *For all regular languages $R \subseteq \Sigma^*$, the following are equivalent.*
(1) $R^\omega = R^\#$.
(2) R^ω *is in DBW.*
(3) $R^\#$ *is ω-regular.*

The proof of Theorem 5 is partitioned into Lemmas 6, 7, and 8.

▶ **Lemma 6.** [**(1)** $\to$ **(2)** **and** **(3)**] *If $R^\omega = R^\#$, then R^ω is in DBW and $R^\#$ is ω-regular.*

Proof. By Landweber's Theorem [12], an ω-regular language L is in DBW iff L is a countable intersection of open sets in the product topology over Σ^ω, induced by the discrete topology over Σ. Specifically, the topology that is induced by the basis $\mathcal{B} = \{N_x : x \in \Sigma^*\}$, where $N_x = x \cdot \Sigma^\omega$. That is, $A \subseteq \Sigma^\omega$ is an open set in the product topology if there is a $B \subseteq \Sigma^*$ such that $A = \cup_{x \in B} N_x = B \cdot \Sigma^\omega$. Equivalently, the topology induced by the metric $d : \Sigma^\omega \times \Sigma^\omega \to \mathbb{R}_{\geq 0}$, defined $d(x, y) = \frac{1}{2^n}$, where n is the first position that x and y differ, and $d(x, y) = 0$, if $x = y$. That is, $A \subseteq \Sigma^\omega$ is an open set if for all $x \in A$ there exists $\gamma > 0$ such that $\{y : d(x, y) < \gamma\} \subseteq A$.

As discussed above, an open set is a set of the form $K \cdot \Sigma^\omega$ for some $K \subseteq \Sigma^*$. Thus, Landweber's Theorem states that an ω-regular language L is in DBW iff there exists $\{K_i\}_{i \in \mathbb{N}}$, $K_i \subseteq \Sigma^*$, such that $L = \bigcap_i K_i \cdot \Sigma^\omega$. By definition, the language $R^\#$ fulfills the topological condition in Landweber's Theorem. Hence, if $R^\#$ is ω-regular, then $R^\#$ is in DBW.

Since R is regular, the language R^ω is ω-regular. Thus, $R^\# = R^\omega$ is ω-regular, and by the above, both are also in DBW. ◀

▶ **Lemma 7.** [**(2)** $\to$ **(1)**] *If R^ω is in DBW, then $R^\omega = R^\#$.*

Proof. Since, by definition, $R^\omega \subseteq R^\#$, we only have to prove that $R^\# \subseteq R^\omega$. Assume that $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$ is a DBW for R^ω. Let $n = |Q|$. Consider a word $w \in R^\#$, and let $r : \mathbb{N} \to Q$ be the run of $\mathcal{A}$ on w. Let t be a position from which r is contained in $\inf(r)$, i.e., for all $t' \geq t$, we have that $r(t') \in \inf(r)$. Let $w = w_1 \cdot w_2 \cdots w_{t+n} \cdot y$ be a partition of w to words such that for all $1 \leq j \leq t+n$, we have that $w_j \in R$. Since $w \in R^\#$, such a partition exists. Let $q_j = r(|w_1 \cdots w_j|)$, i.e., q_j is the state $\mathcal{A}$ reaches when reading the prefix $w_1 \cdots w_j$. Observe that since there are only n states, there must exist indices j_1 and j_2 such that $t \leq j_1 < j_2 \leq t+n$ and $q_{j_1} = q_{j_2}$.

Consider the word $w' = w_1 \cdot w_2 \cdots w_{j_1} \cdot (w_{j_1+1} \cdots w_{j_2})^\omega$, and let r' be the run of $\mathcal{A}$ on w'. Since all the words w_j are in R, then $w' \in R^\omega$, and so $\inf(r') \cap \alpha \neq \emptyset$. Moreover, since $|w_1 \cdots w_{j_1}| \geq t$ and $w_{j_1+1} \cdots w_{j_2}$ closes a cycle from q_{j_1}, then $\inf(r') \subseteq \inf(r)$. Hence, $\inf(r) \cap \alpha \neq \emptyset$. Thus, the run of $\mathcal{A}$ on w is accepting, and so $w \in R^\omega$. ◀

▶ **Lemma 8.** [(3) → (1)] *If $R^\#$ is ω-regular, then $R^\omega = R^\#$.*

Proof. Since, by definition, $R^\omega \subseteq R^\#$, we only have to prove that $R^\# \subseteq R^\omega$. Let $R \subseteq \Sigma^*$ be such that $R^\#$ is ω-regular. Then, as R^ω is ω-regular, so is $K = R^\# \setminus R^\omega$. Let $\mathcal{A}$ be an NBW for K. Assume by way of contradiction that $L(\mathcal{A}) \neq \varnothing$. There exist some accepting state q that is reachable from both an initial state by a path labeled with some $u \in \Sigma^*$, and from itself by a cycle labeled with some $v \in \Sigma^*$. Thus, the word $w = u.v^\omega$ is a lasso-shaped word in $L(\mathcal{A})$. Let x be a prefix of w with $x \in R^{|u|+|v|}$, and let $x = y_0.y_1...y_{|v|}$ be a partition of x such that $y_0 \in R^{|u|}$ and $y_i \in R$ for all $i > 0$. Note that $|y_0| \geq |u|$, thus for $i > 0$, the y_i's are nonempty subwords of $\{v\}^+$. For $0 \leq i \leq |v|$, let k_i be the position in v that is reached after reading $y_0.y_1...y_i$. I.e., $k_i = j$, for $0 \leq j \leq |v|-1$, such that $y_0...y_i = u.v^t.v[1,j]$ for some $t \geq 0$. For example, if $y_0 = u.v$, then $k_0 = 0$, and if $y_0.y_1 = u.v.v.v[1,2]$, then $k_1 = 2$. Since $0 \leq k_i \leq |v|-1$ for all $0 \leq i \leq |v|$, there are indices i and j such that $i < j$, and $k_i = k_j$. Therefore, there exist $t_1, t_2 \geq 0$ such that the following hold:

1. $z_1 = y_0...y_i = u.v^{t_1}.v[1,k_i] \in R^+$, and
2. $z_2 = y_{i+1}...y_j = v[k_i+1,|v|].v^{t_2}.v[1,k_j] = v[k_i+1,|v|].v^{t_2}.v[1,k_i] \in R^+$.

Clearly, $z_1.(z_2)^\omega \in R^\omega$. Also, $(z_2)^\omega = v[k_i+1,|v|].v^\omega$, thus $z_1.(z_2)^\omega = u.v^\omega = w$. Thus, $w \in R^\omega$, contradicting the assumption that $L(\mathcal{A}) = R^\# \setminus R^\omega$. Hence, $L(\mathcal{A}) = \varnothing$; thus $R^\# \subseteq R^\omega$. ◀

This completes the proof of Theorem 5. We now show that the theorem is not trivial, thus there is a language R that does not satisfy the three criteria in the theorem, in particular the criteria about DBW, which is our main interest.

▶ **Theorem 9.** *There exists a regular language $R \subseteq \Sigma^*$, such that R^ω is not in DBW.*

Proof. We define the regular language $R \subseteq \{0,1,\$\}^*$ by the regular expression $R = (\$ + 0 \cdot \{0,1,\$\}^* \cdot 1)$. It is easy to see that for every word $w \in R^\omega$, if w contains infinitely many 1's, then w contains infinitely many 0's. Indeed, the only way to have only finitely many 1's in a word in R^ω is to have an infinite tail of \$'s. Hence, the word

$$w = 011\$1\$\$1\$\$\$1\$\$\$\$1\$\$\$\$\$\ldots = 0 \cdot \prod_{i=0}^{\infty} 1\$^i$$

is not in R^ω. We prove that $w \in R^\#$. For $n \in \mathbb{N}$, consider the word $w_n = 0 \cdot \prod_{i=0}^{n} 1\$^i = (0 \cdot (\prod_{i=0}^{n-1} 1\$^i) \cdot 1) \cdot \$^n$. It is easy to see that $w_n \in R^{n+1}$. Since all of the w_n's are prefixes of w, it follows that $w \in R^\#$.

Thus, $w \in R^\# \setminus R^\omega$, implying that $R^\# \neq R^\omega$. Then, by Theorem 5, we have that R^ω is not in DBW, and we are done. ◀

▶ **Corollary 10.** *For every regular language $R \subseteq \Sigma^*$, we have that R^ω is ω-regular. Yet, R^ω need not be in DBW, and $R^\#$ need not be ω-regular.*

We continue to ∞R, showing it is an easy special case of R^ω. Given a regular language $R \subseteq \Sigma^*$, let $P = \Sigma^* \cdot R$. It is easy to see that $\infty R = P^\omega$. As we argue below, the special form of P implies it satisfies all the three criteria in Theorem 5:

▶ **Theorem 11.** *For every regular language $R \subseteq \Sigma^*$, we have that $(\Sigma^* \cdot R)^\# = (\Sigma^* \cdot R)^\omega$.*

Proof. Let $P = \Sigma^* \cdot R$. We prove that $P^\# \subseteq P \cdot P^\#$. By Lemma 2, the latter implies that $P^\# = P^\omega$. Consider a word $w \in P^\#$, and let $x_0 \prec w$ be a word of minimal length such that $x_0 \in P$. Let $w' \in \Sigma^\omega$ be such that $w = x_0 \cdot w'$. We prove that $w' \in P^\#$, implying that $w \in P \cdot P^\#$. For all $i \geq 1$, let $x_i \cdot y_i \prec w$, with $x_i \in P$ and $y_i \in P^i$. Note that by the minimality of x_0, it holds that $x_0 \prec x_i$ for all $i \geq 1$. Now, for all $i \geq 1$, let $z_i \in \Sigma^*$ be the suffix of x_i, with $x_i = x_0 \cdot z_i$, and consider $u_i = z_i \cdot y_i \prec w'$. Observe that for all $i \geq 1$ we have $u_i \in \Sigma^* \cdot P^i = (\Sigma^* \cdot R) \cdot P^{i-1} = P^i$. Hence, $w' \in P^\#$. ◀

▶ **Corollary 12.** *For every regular language $R \subseteq \Sigma^*$, the language ∞R is in DBW.*

4 Complexity

In this section we study the complexity of deciding, given an NFW $\mathcal{A}$, whether $L(\mathcal{A})^\omega$ is DBW-recognizable. We first describe a simple linear translation of an NFW for R to an NBW for R^ω.

▶ **Theorem 13.** *For every NFA $\mathcal{A}$ with n states, there exists an NBW $\mathcal{A}'$ with $O(n)$ states such that $L(\mathcal{A}') = L(\mathcal{A})^\omega$.*

Proof. First observe that if $\mathcal{A}$ is *simple*, thus it has a single initial state that is also the only accepting state, then the language of $\mathcal{A}$, when viewed as an NBW, is $L(\mathcal{A})^\omega$. Given a NFW $\mathcal{A}$ with n states, we can construct a simple NFW $\mathcal{A}'$ with $n+1$ states that recognizes $L(\mathcal{A})^*$. Thus, the language of $\mathcal{A}'$ when viewed as an NBW is $(L(\mathcal{A})^*)^\omega = L(\mathcal{A})^\omega$. In the full version, we give the complete construction. ◀

▶ **Theorem 14.** *Consider an NFA $\mathcal{A}$. Deciding whether $L(\mathcal{A})^\omega$ is DBW-recognizable is PSPACE-complete.*

Proof. We start with the upper bound. As described in the proof of Theorem 13, given an NFW $\mathcal{A}$ with n states, we can construct an NBW for $L(\mathcal{A})^\omega$ with $n+1$ states. By [10], deciding whether the language of a given NBW is DBW-recognizable can be done in PSPACE. Hence, membership in PSPACE for our result follows.

For the lower bound, we describe a logspace reduction from the universality problem for NFWs, proved to be PSPACE-hard in [15]. For two alphabets Σ_1 and Σ_2, and two words $w_1 \in \Sigma_1^\omega$ and $w_2 \in \Sigma_2^\omega$, let $w_1 \oplus w_2 \in (\Sigma_1 \times \Sigma_2)^\omega$ be the word obtained by merging w_1 and w_2. Formally, if $w_1 = \sigma_1^1 \cdot \sigma_2^1 \cdot \sigma_3^1 \cdots$ and $w_2 = \sigma_1^2 \cdot \sigma_2^2 \cdot \sigma_3^2 \cdots$, then $w_1 \oplus w_2 = \langle \sigma_1^1, \sigma_1^2 \rangle \cdot \langle \sigma_2^1, \sigma_2^2 \rangle \cdot \langle \sigma_3^1, \sigma_3^2 \rangle \cdots$. We use the operator $\oplus$ also for merging two finite words $w_1 \in \Sigma_1^*$ and $w_2 \in \Sigma_1^*$ of the same length. Note that then, $|w_1 \oplus w_2| = |w_1| = |w_2|$.

Consider an NFW $\mathcal{A}$ over some alphabet Σ, and assume $\perp \notin \Sigma$. Consider the language $R = \$^* + 0 \cdot \{0, 1, \$\}^* \cdot 1$. Note that R is similar to the language used in the proof of Theorem 9 – here we include in R words in $\$^*$. This does not change $R^\#$ or R^ω, and the word $0 \cdot \prod_{i=0}^\infty 1 \cdot \$^i$ is in $R^\# \setminus R^\omega$, witnessing that R^ω is not DBW-recognizable.

We define the language $R_{\mathcal{A}}$ over the alphabet $(\Sigma \cup \{\bot\}) \times \{0, 1, \$\}$ as follows.

$R_{\mathcal{A}} = \{(w_1 \cdot \bot) \oplus w_2 : w_1 \in L(\mathcal{A}) \text{ or } w_2 \in R\}$.

Note that since NFWs for R and for $(\Sigma \cup \{\bot\})^* \cdot \bot$ are of a fixed size, the size of an NFW for $R_{\mathcal{A}}$ is linear in the size of $\mathcal{A}$ and it can be constructed from $\mathcal{A}$ in logspace. We prove that $L(\mathcal{A}) = \Sigma^*$ iff $R_{\mathcal{A}}^\omega \in$ DBW. First, observe that if $L(\mathcal{A}) = \Sigma^*$, then $R_{\mathcal{A}}^\omega = (\infty\bot) \oplus \{0, 1, \$\}^\omega$, and so $R_{\mathcal{A}}^\omega \in$ DBW. For the other direction, assume that $L(\mathcal{A}) \neq \Sigma^*$, and consider a word $x \in \Sigma^* \setminus L(\mathcal{A})$. Let $w_x = (x \cdot \bot)^\omega$. Observe that for every partition $y_1 \cdot y_2 \cdot y_3 \cdots$ of w_x into subwords with $y_i \in (\Sigma \cup \{\bot\})^* \cdot \bot$, for all $i \geq 1$, it must be that $y_i \notin L(\mathcal{A}) \cdot \bot$ for all $i \geq 1$. It follows that for every $w \in \{0, 1, \$\}^\omega$, if $w_x \oplus w \in R_{\mathcal{A}}^\omega$, then $w \in R^\omega$.

Let $m = |x \cdot \bot|$, and consider the word $w = 0^m \cdot \prod_{i=0}^{\infty} 1^m \cdot \$^{im}$, obtained from $0 \cdot \prod_{i=0}^{\infty} 1 \cdot \$^i$ by replacing each letter $\sigma \in \{0, 1, \$\}$ by the word σ^m. Using the same arguments used in the proof of Theorem 9, we have that $w \notin R^\omega$. Hence, $w_x \oplus w \notin R_{\mathcal{A}}^\omega$.

We prove that $w_x \oplus w \in R_{\mathcal{A}}^\#$. Note that $w_x \oplus w = (x \cdot \bot)^\omega \oplus 0^m \cdot \prod_{i=0}^{\infty} 1^m \cdot \$^{im} = ((x \cdot \bot) \oplus 0^m) \cdot \prod_{i=0}^{\infty}((x \cdot \bot) \oplus 1^m) \cdot ((x \cdot \bot) \oplus \$^m)^i$. For all $j \geq 1$, we have $((x \cdot \bot) \oplus \$^m)^j) \in R_{\mathcal{A}}^j$, and $((x \cdot \bot) \oplus 0^m) \cdot (\prod_{i=0}^{j-1}((x \cdot \bot) \oplus 1^m) \cdot ((x \cdot \bot) \oplus \$^m)^i) \cdot ((x \cdot \bot) \oplus 1^m) \in R_{\mathcal{A}}$. Hence, $y^j = ((x \cdot \bot) \oplus 0^m) \cdot \prod_{i=0}^{j}((x \cdot \bot) \oplus 1^m) \cdot ((x \cdot \bot) \oplus \$^m)^i \in R_{\mathcal{A}}^{j+1}$. Since $y^j \prec w_x \oplus w$, for all $j \geq 1$, we conclude that $w_x \oplus w \in R_{\mathcal{A}}^\#$.

Thus, $w_x \oplus w \in R_{\mathcal{A}}^\# \setminus R_{\mathcal{A}}^\omega$, and so, by Theorem 5, we have that $R_{\mathcal{A}}^\omega \notin$ DBW. ◀

5 Succinctness

In this section we study the blow-up in going from an automaton for R to automata for $\lim(R)$, ∞R, and R^ω. Note that, by Theorem 5, a DBW for R^ω is also a DBW for $R^\#$, and thus we do not consider $R^\#$ explicitly.

Studying succinctness, we also refer to the *Rabin* acceptance condition. There, $\alpha = \{\langle G_1, B_1\rangle, \ldots, \langle G_k, B_k\rangle\} \subseteq 2^Q \times 2^Q$, and a run r is accepting iff there is a pair $\langle G, B\rangle \in \alpha$ such that $\inf(r) \cap G \neq \emptyset$ and $\inf(r) \cap B = \emptyset$. We use DRW to denote deterministic Rabin word automata. By [8], DRWs are *Büchi type*: if a DRW $\mathcal{A}$ recognizes a DBW-recognizable language, then a DBW for $L(\mathcal{A})$ can be defined on top of $\mathcal{A}$. In other words, if $L(\mathcal{A})$ is in DBW, then we can obtain a DBW for $L(\mathcal{A})$ by redefining the acceptance condition of $\mathcal{A}$.

Our study of succinctness considers the cases R is given by a DFW or an NFW, and the automaton for the repetition language is DBW, DRW, or NBW. We start with the case both automata are deterministic. Then, the case of $\lim(R)$ is easy and well known: Given a DFW $\mathcal{A}$ for R, viewing $\mathcal{A}$ as a DBW results in an automaton for $\lim(R)$ [12]. Hence, there is no blow-up in going from a DFW for R to a DBW for $\lim(R)$. We continue to the case of ∞R. We first consider the case we are given an NFW or DFW for $\Sigma^* \cdot R$.

▶ **Theorem 15.** *For every regular language $R \subseteq \Sigma^*$, there is no blow-up in going from an NFW (DFW) for $\Sigma^* \cdot R$ to an NBW (resp. DBW) for ∞R.*

Proof. Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, \alpha\rangle$ be an NFW with a single initial state that recognizes $\Sigma^* \cdot R$. We define an NBW $\mathcal{A}'$ for $(\Sigma^* \cdot R)^\omega = \infty R$ as follows. Intuitively, $\mathcal{A}'$ simulates a run of $\mathcal{A}$, each time the simulation reaches a state in α it "restarts" the simulation, and it accepts an infinite word iff simulation has been restarted infinitely often. The partition to successful simulations also partitions accepted words to infixes in $L(\mathcal{A})^\omega$, thus accepted words are in ∞R. In addition, if a word is in ∞R, then a word in $\Sigma^* \cdot R$ start in all positions, implying that a successful simulation is always eventually completed. Formally, $\mathcal{A}' = \langle Q, \Sigma, \delta', q_0, \alpha\rangle$, where for all $\sigma \in \Sigma$ and $q \notin \alpha$, $\delta'(q, \sigma) = \delta(q, \sigma)$, and otherwise $\delta'(q, \sigma) = \delta(q_0, \sigma)$. In the full version, we prove that $L(\mathcal{A}') = (\Sigma^* \cdot R)^\omega = \infty R$. Note that since $\varepsilon \notin R$, then $q_0 \notin \alpha$. Also, note that when $\mathcal{A}$ is deterministic, so is $\mathcal{A}'$. ◀

Going from a DFW for R to a DFW for $\Sigma^* \cdot R$ may involve an exponential blow-up. To see this, consider for example the language $R = 0 \cdot (0+1)^n$. While it can be recognized by a DFW with $n+2$ states, a DFA for $(0+1)^* \cdot 0 \cdot (0+1)^n$ needs at least 2^n states. Theorem 16 shows that this blow-up is inherited to the construction of a DBW for ∞R.

▶ **Theorem 16.** *The blow-up in going from a DFW for R to a DBW for ∞R is $2^{O(n)}$.*

Proof. For the upper bound, starting with a DFW with n states for R, one can construct an NFW with $n+1$ states for $\Sigma^* \cdot R$. Its determinization results in a DFW with 2^{n+1} states for $\Sigma^* \cdot R$. Then, by Theorem 15, we end up with a DBW with 2^{n+1} states for ∞R.

For the lower bound, we describe a family of languages $R_1, R_2, \ldots$ of finite words, such that for all $n \geq 1$, the language R_n can be recognized by a DFW with $O(n)$ states, yet a DBW for ∞R_n needs at least $\frac{2^{n-1}}{n}$ states.

Let $\Sigma = \{0, 1\}$. For $n \geq 1$, we define $R_n \subseteq \Sigma^*$ as the set of words of length $n+1$ that start and end with the same letter. That is, $R_n = \{\sigma \cdot w \cdot \sigma : \text{for } \sigma \in \Sigma \text{ and } w \in \Sigma^{n-1}\}$. Equivalently, $R_n = 0 \cdot (0+1)^{n-1} \cdot 0 + 1 \cdot (0+1)^{n-1} \cdot 1$. It is easy to see that R_n can be recognized by a DFW with $2n+3$ states. In the full version, we prove that a DBW for ∞R_n needs at least $\frac{2^{n-1}}{n}$ states. ◀

We continue to R^ω. While it is easy, given a DFW for R, to construct an NBW for R^ω (see Theorem 13), staying in the deterministic model is complicated, and not only in terms of expressive power. Formally, we have the following.

▶ **Theorem 17.** *The blow-up in going from a DFW for R to a DBW for R^ω, when exists, is $2^{O(n \log n)}$.*

Proof. For the upper bound, one can determinize the NBW for R^ω. Thus, starting with a DFW with n states for R, we construct an NBW with $n+1$ states for R^ω, and determinize it to a DRW with $2^{O(n \log n)}$ states [20]. Since DRWs are Büchi type, the result follows.

For the lower bound, we describe a family of languages $R_1, R_2, \ldots$ of finite words, such that for all $n \geq 1$, the language R_n can be recognized by a DFW with $\mathrm{O}(n)$ states, R_n^ω is in DBW, yet a DBW for R_n^ω needs at least $n!$ states.

Given $n \geq 1$, let $\Sigma_n = [n] \cup \{\#\}$, where $[n] = \{1, \ldots, n\}$. We define the language $R_n \subseteq \Sigma_n^*$ as the set of all finite words that start and end with the same letter from $[n]$. That is, $R_n = \{\sigma \cdot x \cdot \sigma : \text{for } x \in \Sigma_n^* \text{ and } \sigma \in [n]\}$. It is easy to see that R_n is regular and a DFW for R_n needs $2n+1$ states. In the full verison, we prove that R_n^ω is in DBW, and that a DBW for R_n^ω needs at least $n!$ states. ◀

Since DRWs are Büchi type, Theorems 16 and 17 imply the following.

▶ **Theorem 18.** *The blow-ups in going from a DFW with n states for R to DRWs for ∞R and R^ω are $2^{O(n)}$ and $2^{O(n \log n)}$, respectively.*

The succinctness analysis for case the automaton for the repetition languages is non-deterministic is much easier, as the constructions described above involve no blow-up, and except for the case of $\lim(R)$, they are valid also when R is given by an NFW. The case of $\lim(R)$ is more complicated and is studied in [2]. It is easy to see that just viewing an NFW for R as a Büchi automaton does not result in an NBW for $\lim(R)$. For example, an NFW for $(0+1) \cdot 0$ that guesses whether each 0 is the last letter, in which case it moves to an accepting state with no successors, is empty when viewed as an NBW. The best known construction of an NBW for $\lim(R)$ from an NFW $\mathcal{A}$ for R is based on a characterization of the limit of $L(\mathcal{A})$ as the union of languages, each associated with a state q of $\mathcal{A}$ and containing words that have infinitely many prefixes whose accepting run reaches q. Following this characterization, it is possible to construct, starting with $\mathcal{A}$ with n states, an NBW with $O(n^3)$ states for $\lim(R)$ [2].

6 On Unboundedly Many vs. Infinitely Many

Essentially, the definition of $R^\#$ replaces the "infinite" nature of R^ω by an "unbound" one. In this section we examine an analogous change in the definition of acceptance in Büchi automata. Consider a nondeterministic automaton $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$. When we view $\mathcal{A}$ as a *#-automaton*, it accepts a word $w \in \Sigma^\omega$ if for all $i \geq 0$, there is a run of $\mathcal{A}$ on w that visits α at least i times. Formally, for for all $i \geq 0$, there is a run $r_i = q_0^i, q_1^i, q_2^i, \ldots$ of $\mathcal{A}$ on w such that $r_j^i \in \alpha$ for at least i indices of $j \geq 0$. The *#-language* of $\mathcal{A}$, denoted $L_\#(\mathcal{A})$, is the set of words that $\mathcal{A}$ accepts as above. We use the notations $L_F(\mathcal{A})$ and $L_B(\mathcal{A})$ to refer the languages of $\mathcal{A}$ when viewed as an automaton on finite words and a Büchi automaton, respectively. It is not hard to see that when $\mathcal{A}$ is deterministic, then $L_B(\mathcal{A}) = L_\#(\mathcal{A})$. Indeed, in both cases, $\mathcal{A}$ accepts a word w if its single run on w visits α infinitely often. When, however, $\mathcal{A}$ is nondeterministic, its #-language may contain words accepted via infinitely many different runs, none of which visits α infinitely often.

▶ **Example 19.** Consider the automaton $\mathcal{A}_1$ in Figure 1. Note that $L_F(\mathcal{A}_1) = R$, for $R = (\$ + 0 \cdot \{0, 1, \$\}^* \cdot 1)$, namely the language used in Theorem 9 for demonstrating a language with $R^\# \neq R^\omega$. Here, we have that $L_\#(\mathcal{A}_1) \neq L_B(\mathcal{A}_1)$. For example, $w = 011\$1\$\$1\$\$\$1\$\$\$\$1\$\$\$\$\$ \cdots = 0 \cdot \prod_{i=0}^{\infty} 1\$^i \in L_\#(\mathcal{A}_1) \setminus L_B(\mathcal{A}_1)$.

Consider now the automaton $\mathcal{A}_2$. Here, $L_B(\mathcal{A}_2) = (0+1)^* \cdot 1^\omega$. On the other hand, for every $i \geq 1$, there is a run of $\mathcal{A}_2$ on $w = 01011011101111 \cdots = \prod_{i=0}^{\infty} 01^i$ that visits α at least i times. Thus, $w \in L_\#(\mathcal{A}_2)$ even though it has infinitely many 0's and is not in $L_B(\mathcal{A}_2)$. Note that the word w is also used to differentiate the Büchi and *prompt-Büchi* acceptance conditions. A prompt-Buch automaton $\mathcal{A}$ accepts a word w iff there is $i \geq 1$ and a run r of $\mathcal{A}$ on w, such that r visits α at least once in every i successive states [1]. It is not hard to see that w is not accepted by all DPWs for $(1^* \cdot 0)^\omega$. ◀

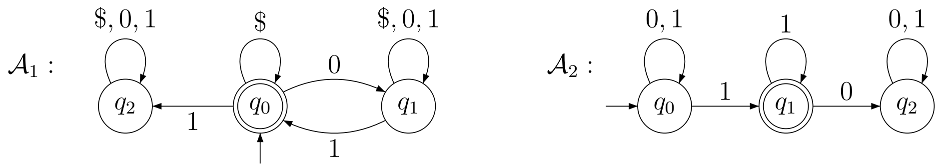

Figure 1 Automata with a non-regular #-language.

▶ Remark 20. Defining $L_\#(\mathcal{A})$, we require the transition function δ of $\mathcal{A}$ to be defined for all states and letters. Indeed, a rejecting sink in a #-automaton may support acceptance. To see this, consider $\mathcal{A}_2$ from Example 19, and assume that rather than going with the letter 0 to the rejecting sink q_2, the state q_1 would have no outgoing transitions labeled 0. Then, no run of $\mathcal{A}_2$ on the word w from the example can visit q_1 even once without getting stuck. Note that rather than requiring δ to be total, we could also define $L_\#(\mathcal{A})$ as these words for which, for all $i \geq 0$, there is a run of $\mathcal{A}$ on a prefix of w that visits α at least i times. ◀

Interestingly, the relation between $L_\#(\mathcal{A})$ and $L_B(\mathcal{A})$ is similar to the one obtained for $R^\#$ and R^ω. Formally, we have the following.

▶ **Theorem 21.** *For all finite automata $\mathcal{A}$, the following are equivalent.*

(1) $L_\#(\mathcal{A})$ *is ω-regular.*

(2) $L_B(\mathcal{A}) = L_\#(\mathcal{A})$.

(3) $L_\#(\mathcal{A})$ *is in DBW.*

Proof. Clearly, both (2) → (1) and (3) → (1). We prove that (1) → (2) and (1) → (3).

We start with (1) → (2). First, clearly, for all automata $\mathcal{A}$, we have that $L_B(\mathcal{A}) \subseteq L_\#(\mathcal{A})$. We prove that $L_\#(\mathcal{A}) \subseteq L_B(\mathcal{A})$. Since $L_\#(\mathcal{A})$ is ω-regular, then, as ω-regular languages are closed under complementation, there is an NBW $\mathcal{B}$ for $L_\#(\mathcal{A}) \setminus L_B(\mathcal{A})$. If $L_\#(\mathcal{A}) \not\subseteq L_B(\mathcal{A})$, then $L_B(\mathcal{B})$ is not empty, which implies $\mathcal{B}$ accepts a lasso-shaped word, namely a word of the form $u \cdot v^\omega$ for $u, v \in \Sigma^* \setminus \{\varepsilon\}$. But $L_\#(\mathcal{A})$ and $L_B(\mathcal{A})$ agree on all lasso-shaped words. Indeed, $u \cdot v^\omega \in L_\#(\mathcal{A})$ iff $\mathcal{A}$ has a cycle that visits α and is traversed when the v^ω suffix is read, iff $u \cdot v^\omega \in L_B(\mathcal{A})$. Hence, $\mathcal{B}$ is empty, $L_\#(\mathcal{A}) \subseteq L_B(\mathcal{A})$, and we are done.

We continue to (1) → (3). For all $i \geq 0$, let L_i be the set of words $w \in \Sigma^*$ such that there exists a run of $\mathcal{A}$ on w that visits α exactly i times. Observe that $L_\#(\mathcal{A}) = \bigcap_{i \geq 0} L_i \cdot \Sigma^\omega$. Thus, $L_\#(\mathcal{A})$ is a countable intersection of open sets. Hence, by Landweber, $L_\#(\mathcal{A})$ being ω-regular implies that $L_\#(\mathcal{A})$ is in DBW. ◀

7 Discussion

The expressiveness and succinctness of different classes of automata on infinite words have been studied extensively in the early days of the automata-theoretic approach to formal verification [21]. Specification formalisms that combine regular expressions or automata with temporal-logic modalities have been the subject of extensive research too [23, 22]. Quite surprisingly, the expressiveness and succinctness of repetition languages, which are at the heart of this study, have been left open. The research described in this paper started following a question asked by Michael Kaminski about R^ω being DBW-recognizable for every regular language R. We had two conjectures about this question. First, that the answer is positive, and second, that this must have been studied already. We were not able to prove either conjectures, and in fact refuted the first. In the process, we developed the full theory of repetition languages, their expressiveness, and succinctness, as well the notion of #-languages which goes beyond ω-regular languages. Our results are summarized in Table 1 below. The $\surd$ and $\times$ symbols indicate whether a translation always exists. All blow-ups except for the one from [2] are tight. The blow-ups in translations to DBWs apply also to DRWs (Th. 18). Finally, for $R^\#$, translations exist whenever R^ω is DBW-recognizable (Th. 5), in which case the blow-ups agree with the one described for R^ω.

Table 1 Translations from an automaton for R to automata for its repetition languages.

	$\lim(R)$	∞R	R^ω
DFW to DBW	$\surd$ $O(n)$ [12]	$\surd$ $2^{O(n)}$ Ths. 15 and 16	$\times$ $2^{O(n \log n)}$ Ths. 9 and 17
DFW to NBW	$\surd$ $O(n)$ [12]	$\surd$ $O(n)$ Th. 15	$\surd$ $O(n)$ Th. 13
NFW to NBW	$\surd$ $O(n^3)$ [2]	$\surd$ $O(n)$ Th. 15	$\surd$ $O(n)$ Th. 13

References

1 S. Almagor, Y. Hirshfeld, and O. Kupferman. Promptness in omega-regular automata. In *8th Int. Symp. on Automated Technology for Verification and Analysis*, volume 6252, pages 22–36, 2010.

2 B. Aminof and O. Kupferman. On the succinctness of nondeterminizm. In *4th Int. Symp. on Automated Technology for Verification and Analysis*, volume 4218 of *Lecture Notes in Computer Science*, pages 125–140. Springer, 2006.

3 A. Arnold and D. Niwiński. Fixed point characterization of weak monadic logic definable sets of trees. In M. Nivat and A. Podelski, editors, *Tree Automata and Languages*, pages 159–188. Elsevier, 1992.

4 C. Baier, L. de Alfaro, V. Forejt, and M. Kwiatkowska. Model checking probabilistic systems. In *Handbook of Model Checking.*, pages 963–999. Springer, 2018.

5 R. Bloem, K. Chatterjee, and B. Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking.*, pages 921–962. Springer, 2018.

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

7 C. Eisner and D. Fisman. *A Practical Introduction to PSL.* Springer, 2006.

8 S.C. Krishnan, A. Puri, and R.K. Brayton. Deterministic ω-automata vis-a-vis deterministic Büchi automata. In *Algorithms and Computations*, volume 834 of *Lecture Notes in Computer Science*, pages 378–386. Springer, 1994.

9 O. Kupferman. Automata theory and model checking. In *Handbook of Model Checking*, pages 107–151. Springer, 2018.

10 O. Kupferman and M.Y. Vardi. From linear time to branching time. *ACM Transactions on Computational Logic*, 6(2):273–294, 2005.

11 R.P. Kurshan. *Computer Aided Verification of Coordinating Processes.* Princeton Univ. Press, 1994.

12 L.H. Landweber. Decision problems for ω–automata. *Mathematical Systems Theory*, 3:376–384, 1969.

13 C. Löding. Optimal bounds for the transformation of ω-automata. In *Proc. 19th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 1738 of *Lecture Notes in Computer Science*, pages 97–109, 1999.

14 R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521–530, 1966.

15 A.R. Meyer and L.J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential time. In *Proc. 13th IEEE Symp. on Switching and Automata Theory*, pages 125–129, 1972.

16 D.E. Muller, A. Saoudi, and P.E. Schupp. Alternating automata, the weak monadic theory of the tree and its complexity. In *Proc. 13th Int. Colloq. on Automata, Languages, and Programming*, volume 226 of *Lecture Notes in Computer Science*, pages 275–283. Springer, 1986.

17 M.O. Rabin. Decidability of second order theories and automata on infinite trees. *Transaction of the AMS*, 141:1–35, 1969.

18 M.O. Rabin. Weakly definable relations and special automata. In *Proc. Symp. Math. Logic and Foundations of Set Theory*, pages 1–23. North Holland, 1970.

19 M.O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:115–125, 1959.

20 S. Safra. On the complexity of ω-automata. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 319–327, 1988.

21 W. Thomas. Automata on infinite objects. *Handbook of Theoretical Computer Science*, pages 133–191, 1990.

22 M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.

23 P. Wolper. Temporal logic can be more expressive. In *Proc. 22nd IEEE Symp. on Foundations of Computer Science*, pages 340–348, 1981.