First-order definable languages*

Volker Diekert[1]
Paul Gastin[2]

[1] Institut für Formale Methoden der Informatik
Universität Stuttgart
Universitätsstraße 38
70569 Stuttgart, Germany
`diekert@fmi.uni-stuttgart.de`

[2] Laboratoire Spécification et Vérification
École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex, France
`Paul.Gastin@lsv.ens-cachan.fr`

Abstract

We give an essentially self-contained presentation of some principal results for first-order definable languages over finite and infinite words. We introduce the notion of a *counter-free* Büchi automaton; and we relate counter-freeness to *aperiodicity* and to the notion of *very weak alternation.* We also show that aperiodicity of a regular ∞-language can be decided in polynomial space, if the language is specified by some Büchi automaton.

1 Introduction

The study of regular languages is one of the most important areas in formal language theory. It relates logic, combinatorics, and algebra to automata theory; and it is widely applied in all branches of computer sciences. Moreover it is the core for generalizations, e.g., to tree automata [26] or to partially ordered structures such as Mazurkiewicz traces [6].

In the present contribution we treat first-order languages over finite and infinite words. First-order definability leads to a subclass of regular languages and again: it relates logic, combinatorics, and algebra to automata theory; and it is also widely applied in all branches of computer sciences. Let us mention that first-order definability for Mazurkiewicz traces leads essentially to the same picture as for words (see, e.g., [5]), but nice characterizations for first-order definable sets of trees are still missing.

The investigation on first-order languages has been of continuous interest over the past decades and many important results are related to the efforts

* We would like to thank the anonymous referee for the detailed report.

Jörg Flum, Erich Grädel, Thomas Wilke (eds.). *Logic and Automata: History and Perspectives.* Texts in Logic and Games 2, Amsterdam University Press 2007, pp. 261–306.

of Wolfgang Thomas [31, 32, 33, 34, 35]. We also refer to his influential contributions in the handbooks of Theoretical Computer Science [36] and of Formal Languages [37].

We do not compete with these surveys. Our plan is more modest. We try to give a self-contained presentation of some of the principal characterizations of first-order definable languages in a single paper. This covers description with star-free expressions, recognizability by aperiodic monoids and definability in linear temporal logic. We also introduce the notion of a *counter-free* Büchi automaton which is somewhat missing in the literature so far. We relate counter-freeness to the aperiodicity of the transformation monoid. We also show that first-order definable languages can be characterized by very weak alternating automata using the concept of aperiodic automata. In some sense the main focus in our paper is the explanation of the following theorem.

Theorem 1.1. Let L be a language of finite or infinite words over a finite alphabet. Then the following assertions are equivalent:

1. L is first-order definable.
2. L is star-free.
3. L is aperiodic.
4. L is definable in the linear temporal logic LTL.
5. L is first-order definable with a sentence using at most 3 names for variables.
6. L is accepted by some counter-free Büchi automaton.
7. L is accepted by some aperiodic Büchi automaton.
8. L is accepted by some very weak alternating automaton.

Besides, the paper covers related results. The translation from first-order to LTL leads in fact to the pure future fragment of LTL, i.e., the fragment without any past tense operators. This leads to the separation theorem for first-order formulae in one free variable as we shall demonstrate in Section 9. We also show that aperiodicity (i.e., first-order definability) of a regular ∞-language can be decided in polynomial space, if the language is specified by some Büchi automaton.

Although the paper became much longer than expected, we know that much more could be said. We apologize if the reader's favorite theorem is not covered in our survey. In particular, we do not speak about varieties, and we gave up the project to cover principle results about the fragment of first-order logic which corresponds to unary temporal logic. These diamonds will continue to shine, but not here, and we refer to [30] for more background. As mentioned above, we use Büchi automata, but we do not discuss deterministic models such as deterministic Muller automata.

The history of Theorem 1.1 is related to some of the most influential scientists in computer science. The general scheme is that the equivalences above have been proved first for finite words. After that, techniques were developed to generalize these results to infinite words. Each time, the generalization to infinite words has been non-trivial and asked for new ideas. Perhaps, the underlying reason for this additional difficulty is due to the fact that the subset construction fails for infinite words. Other people may say that the difficulty arises from the fact that regular ω-languages are not closed in the Cantor topology. The truth is that combinatorics on infinite objects is more complicated.

The equivalence of first-order definability and star-freeness for finite words is due to McNaughton and Papert [19]. The generalization to infinite words is due to Ladner [15] and Thomas [31, 32]. These results have been refined, e.g. by Perrin and Pin in [24]. Based on the logical framework of Ehrenfeucht-Fraïssé-games, Thomas also related the quantifier depth to the so-called dot-depth hierarchy, [33, 35]. Taking not only the quantifier alternation into account, but also the length of quantifier blocks one gets even finer results as studied by Blanchet-Sadri in [2].

The equivalence of star-freeness and aperiodicity for finite words is due to Schützenberger [28]. The generalization to infinite words is due to Perrin [23] using the syntactic congruence of Arnold [1]. These results are the basis allowing to decide whether a regular language is first-order definable.

Putting these results together one sees that statements 1, 2, and 3 in Theorem 1.1 are equivalent. From the definition of LTL it is clear that linear temporal logic describes a fragment of FO^3, where the latter means the family of first-order definable languages where the defining sentence uses at most three names for variables. Thus, the implications from 4 to 5 and from 5 to 1 are trivial. The highly non-trivial step is to conclude from 1 (or 2 or 3) to 4. This is usually called Kamp's Theorem and is due to Kamp [13] and Gabbay, Pnueli, Shelah, and Stavi [9].

In this survey we follow the algebraic proof of Wilke which is in his habilitation thesis [38] and which is also published in [39]. Wilke gave the proof for finite words, only. In order to generalize it to infinite words we use the techniques from [5], which were developed to handle Mazurkiewicz traces. Cutting down this proof to the special case of finite or infinite words leads to the proof presented here. It is still the most complicated part in the paper, but again some of the technical difficulties lie in the combinatorics of infinite words which is subtle. Restricting the proof further to finite words, the reader might hopefully find the simplest way to pass from aperiodic languages to LTL. But this is also a matter of taste, of course.

Every first-order formula sentence can be translated to a formula in FO^3. This is sharp, because it is known that there are first-order properties which are not expressible in FO^2, which characterizes unary temporal logic [7] over infinite words.

The equivalence between definability in monadic second order logic, regular languages, and acceptance by Büchi automata is due to Büchi [3]. However, Büchi automata are inherently non-deterministic. In order to have deterministic automata one has to move to other acceptance conditions such as Muller or Rabin-Streett conditions. This important result is due to McNaughton, see [18]. Based on this, Thomas [32] extended the notion of deterministic counter-free automaton to deterministic counter-free automaton with Rabin-Streett condition and obtained thereby another characterization for first-order definable ω-languages. There is no canonical object for a minimal Büchi automaton, which might explain why a notion of counter-free Büchi automaton has not been introduced so far. On the other hand, there is a quite natural notion of counter-freeness as well as of aperiodicity for non-deterministic Büchi automata. (Aperiodic non-deterministic *finite* automata are defined in [16], too.) For non-deterministic automata, aperiodicity describes a larger class of automata, but both counter-freeness and aperiodicity can be used to characterize first-order definable ω-languages. This is shown in Section 11 and seems to be an original part in the paper.

We have also added a section about very weak alternating automata. The notion of *weak alternating automaton* is due to Muller, Saoudi, and Schupp [21]. A *very weak alternating automaton* is a special kind of weak alternating automaton and this notion has been introduced in the PhD thesis of Rhode [27] in a more general context of ordinals. (In the paper by Löding and Thomas [17] these automata are called *linear alternating*.) Section 13 shows that very weak alternating automata characterize first-order definability as well. More precisely, we have a cycle from 3 to 6 to 7 and back to 3, and we establish a bridge from 4 to 8 and from 8 to 7.

It was shown by Stern [29] that deciding whether a deterministic finite automaton accepts an aperiodic language over finite words can be done in polynomial space, i.e., in PSPACE. Later Cho and Huynh showed in [4] that this problem is actually PSPACE-complete. So, the PSPACE-hardness transfers to (non-deterministic) Büchi automata. It might belong to folklore that the PSPACE-upper bound holds for Büchi automata, too; but we did not find any reference. So we prove this result here, see Proposition 12.3.

As said above, our intention was to give simple proofs for existing results. But simplicity is not a simple notion. Therefore for some results, we present two proofs. The proofs are either based on a congruence lemma established for first-order logic in Section 10.1, or they are based on a splitting lemma established for star-free languages in Section 3.1. Depending on his background, the reader may wish to skip one approach.

2 Words, first-order logic, and basic notations

By P we denote a unary predicate taken from some finite set of *atomic propositions*, and $x, y, \ldots$ denote variables which represent positions in finite or infinite words. The syntax of first-order logic uses the symbol $\perp$ for *false* and has atomic formulae of type $P(x)$ and $x < y$. We allow Boolean connectives and first-order quantification. Thus, if φ and ψ are first-order formulae, then $\neg\varphi$, $\varphi \vee \psi$ and $\exists x \varphi$ are first-order formulae, too. As usual we have derived formulae such as $x \leq y$, $x = y$, $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$, $\forall x \varphi = \neg\exists x \neg\varphi$ and so on.

We let Σ be a finite *alphabet*. The relation between Σ and the set of unary predicates is that for each letter $a \in \Sigma$ and each predicate P the truth-value $P(a)$ must be well-defined. So, we always assume this. Whenever convenient we include for each letter a a predicate P_a such that $P_a(b)$ is *true* if and only if $a = b$. We could assume that all predicates are of the form P_a, but we feel more flexible of not making this assumption. If x is a position in a word with label $a \in \Sigma$, then $P(x)$ is defined by $P(a)$.

By Σ^* (resp. Σ^ω) we mean the set of finite (resp. infinite) words over Σ, and we let $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. The length of a word w is denoted by $|w|$, it is a natural number or ω. A *language* is a set of finite or infinite words.

Formulae without free variables are *sentences*. A first-order sentence defines a subset of Σ^∞ in a natural way. Let us consider a few examples. We can specify that the first position is labeled by a letter a using $\exists x \forall y\ P_a(x) \wedge x \leq y$. We can say that each occurrence of a is immediately followed by b with the sentence $\forall x\ \neg P_a(x) \vee \exists y\ x < y \wedge P_b(y) \wedge \forall z\ \neg(x < z \wedge z < y)$. We can also say that the direct successor of each b is the letter a. Hence the language $(ab)^\omega$ is first-order definable. We can also say that a last position in a word exists and this position is labeled b. For $a \neq b$ this leads almost directly to a definition of $(ab)^*$. But $(aa)^*$ cannot be defined with a first-order sentence. A formal proof for this statement is postponed, but at least it should be clear that we cannot define $(aa)^*$ the same way as we did for $(ab)^*$, because we have no control that the length of a word in a^* is even.

The set of positions $\mathrm{pos}(w)$ is defined by $\mathrm{pos}(w) = \{i \in \mathbb{N} \mid 0 \leq i < |w|\}$. We think of $\mathrm{pos}(w)$ as a linear order where each position i is labeled with $\lambda(i) \in \Sigma$, and $w = \lambda(0)\lambda(1)\cdots$.

A *k-structure* means here a pair $(w, \overline{p})$, where $w \in \Sigma^\infty$ is a finite or infinite word and $\overline{p} = (p_1, \ldots, p_k)$ is a k-tuple of positions in $\mathrm{pos}(w)$. The set of all k-structures is denoted by $\Sigma^\infty_{(k)}$, and the subset of finite structures is denoted by $\Sigma^*_{(k)}$. For simplicity we identify Σ^∞ with $\Sigma^\infty_{(0)}$.

Let $\overline{x}$ be a k-tuple $(x_1, \ldots, x_k)$ of variables and φ be a first-oder formula where all free variables are in the set $\{x_1, \ldots, x_k\}$. The semantics of

$$(w, (p_1, \ldots, p_k)) \models \varphi$$

is defined as usual: It is enough to give a semantics to atomic formulae, and $(w, (p_1, \ldots, p_k)) \models P(x_i)$ means that the label of position p_i satisfies P, and $(w, (p_1, \ldots, p_k)) \models x_i < x_j$ means that position p_i is before position p_j, i.e., $p_i < p_j$.

With every formula we can associate its language by

$$\mathcal{L}(\varphi) = \left\{ (w, \overline{p}) \in \Sigma^{\infty}_{(k)} \;\middle|\; (w, \overline{p}) \models \varphi \right\}.$$

In order to be precise we should write $\mathcal{L}_{\Sigma,k}(\varphi)$, but if the context is clear, we omit the subscript Σ, k.

Definition 2.1. By $\mathrm{FO}(\Sigma^*)$ (resp. $\mathrm{FO}(\Sigma^\infty)$) we denote the set of first-order definable languages in Σ^* (resp. Σ^∞), and by FO we denote the family of all first-order definable languages. Analogously, we define families $\mathrm{FO}^n(\Sigma^*)$, $\mathrm{FO}^n(\Sigma^\infty)$, and FO^n by allowing only those formulae which use at most n different names for variables.

3 Star-free sets

For languages $K, L \subseteq \Sigma^\infty$ we define the *concatenation* by

$$K \cdot L = \{uv \mid u \in K \cap \Sigma^*, v \in L\}.$$

The n-th power of L is defined inductively by $L^0 = \{\varepsilon\}$ and $L^{n+1} = L \cdot L^n$. The *Kleene-star* of L is defined by $L^* = \bigcup_{n \geq 0} L^n$. Finally, the ω-iteration of L is

$$L^\omega = \{u_0 u_1 u_2 \cdots \mid u_i \in L \cap \Sigma^* \text{ for all } i \geq 0\}.$$

We are interested here in families of *regular languages*, also called *rational languages*. In terms of expressions it is the smallest family of languages which contains all finite subsets, which is closed under finite union and concatenation, and which is closed under the *Kleene-star* (and ω-power). The relation to finite automata (Büchi automata resp.) is treated in Section 11. For the main results on first-order languages the notion of a Büchi automaton is actually not needed.

The Kleene-star and the ω-power do not preserve first-order definability, hence we consider subclasses of regular languages. A language is called *star-free*, if we do not allow the Kleene-star, but we allow complementation. Therefore we have all Boolean operations. In terms of expressions the *class of star-free languages* is the smallest family of languages in Σ^∞ (resp. Σ^*)

which contains Σ^*, all singletons $\{a\}$ for $a \in \Sigma$, and which is closed under finite union, complementation and concatenation. It is well-known that regular languages are closed under complement[1], hence star-free languages are regular.

As a first example we note that for every $A \subseteq \Sigma$ the set A^* (of finite words containing only letters from A) is also star-free. We have:

$$A^* = \Sigma^* \setminus (\Sigma^*(\Sigma \setminus A)\Sigma^\infty).$$

In particular, $\{\varepsilon\} = \emptyset^*$ is star-free. Some other expressions *with star* are also in fact star-free. For example, for $a \neq b$ we obtain:

$$(ab)^* = (a\Sigma^* \cap \Sigma^* b) \setminus \Sigma^*(\Sigma^2 \setminus \{ab, ba\})\Sigma^*.$$

The above equality does not hold, if $a = b$. Actually, $(aa)^*$ is not star-free. The probably best way to see that $(aa)^*$ is not star-free, is to show (by structural induction) that for all star-free languages L there is a constant $n \in \mathbb{N}$ such that for all words x we have $x^n \in L$ if and only if $x^{n+1} \in L$. The property is essentially *aperiodicity* and we shall prove the equivalence between star-free sets and aperiodic languages later. Since $(ab)^*$ is star-free (for $a \neq b$), but $(aa)^*$ is not, we see that a *projection* of a star-free set is not star-free, in general.

Definition 3.1. By $\mathbf{SF}(\Sigma^*)$ (resp. $\mathbf{SF}(\Sigma^\infty)$) we denote the set of star-free languages in Σ^* (resp. Σ^∞), and by $\mathbf{SF}$ we denote the family of all star-free languages.

An easy exercise (left to the interested reader) shows that

$$\mathbf{SF}(\Sigma^*) = \{L \subseteq \Sigma^* \mid L \in \mathbf{SF}(\Sigma^\infty)\} = \{L \cap \Sigma^* \mid L \in \mathbf{SF}(\Sigma^\infty)\}.$$

3.1 The splitting lemma

A star-free set admits a canonical decomposition given a partition of the alphabet. This will be shown here and it is used to prove that first-order languages are star-free in Section 4 and for the separation theorem in Section 9. The alternative to this section is explained in Section 10, where the standard way of using the *congruence lemma* is explained, see Lemma 10.2. Thus, there is an option to skip this section.

Lemma 3.2. Let $A, B \subseteq \Sigma$ be disjoint subalphabets. If $L \in \mathbf{SF}(\Sigma^\infty)$ then we can write

$$L \cap B^*AB^\infty = \bigcup_{1 \leq i \leq n} K_i a_i L_i$$

where $a_i \in A$, $K_i \in \mathbf{SF}(B^*)$ and $L_i \in \mathbf{SF}(B^\infty)$ for all $1 \leq i \leq n$.

[1] We do not need this standard result here.

Proof. Since $B^*AB^\infty = \bigcup_{a\in A} B^*aB^\infty$, it is enough to show the result when $A = \{a\}$. The proof is by induction on the star-free expression and also on the alphabet size. (Note that $|B| < |\Sigma|$.). The result holds for the basic star-free sets:

- If $L = \{a\}$ with $a \in A$ then $L \cap B^*AB^\infty = \{\varepsilon\}a\{\varepsilon\}$.
- If $L = \{a\}$ with $a \notin A$ then $L \cap B^*AB^\infty = \emptyset a \emptyset$ (or we let $n = 0$).
- If $L = \Sigma^*$ then $L \cap B^*AB^\infty = B^*AB^*$.

The inductive step is clear for union. For concatenation, the result follows from

$$(L \cdot L') \cap B^*AB^\infty = (L \cap B^*AB^\infty) \cdot (L' \cap B^\infty) \cup (L \cap B^*) \cdot (L' \cap B^*AB^\infty).$$

It remains to deal with the complement $\Sigma^\infty \setminus L$ of a star-free set. By induction, we have $L \cap B^*aB^\infty = \bigcup_{1\le i\le n} K_iaL_i$. If some K_i and K_j are not disjoint (for $i \neq j$), then we can rewrite

$$K_iaL_i \cup K_jaL_j = (K_i \setminus K_j)aL_i \cup (K_j \setminus K_i)aL_j \cup (K_i \cap K_j)a(L_i \cup L_j).$$

We can also add $(B^* \setminus \bigcup_i K_i)a\emptyset$ in case $\bigcup_i K_i$ is strictly contained in B^*. Therefore, we may assume that $\{K_i \mid 1 \le i \le n\}$ forms a partition of B^*. This yields:

$$(\Sigma^\infty \setminus L) \cap B^*aB^\infty = \bigcup_{1\le i\le n} K_ia(B^\infty \setminus L_i).$$

Q.E.D.

4 From first-order to star-free languages

This section shows that first-order definable languages are star-free languages. The transformation is involved in the sense that the resulting expressions are much larger than the size of the formula, in general. The proof presented here is based on the splitting lemma. The alternative is again in Section 10.

Remark 4.1. The converse that star-free languages are first-order definable can be proved directly. Although strictly speaking we do not use this fact, we give an indication how it works. It is enough to give a sentence for languages of type $L = \mathcal{L}(\varphi) \cdot a \cdot \mathcal{L}(\psi)$. We may assume that the sentences φ and ψ use different variable names. Then we can describe L as a language $\mathcal{L}(\xi)$ where

$$\xi = \exists z\, P_a(z) \wedge \varphi_{<z} \wedge \psi_{>z},$$

where $\varphi_{<z}$ and $\psi_{>z}$ relativize all variables with respect to the position of z. We do not go into more details, because, as said above, we do not need this fact.

We have to deal with formulae having free variables. We provide first another semantics of a formula with free variables in a set of words over an extended alphabet allowing to encode the assignment. This will also be useful to derive the separation theorem in Section 9.

Let V be a finite set of variables. We define $\Sigma_V = \Sigma \times \{0,1\}^V$. (Do not confuse Σ_V with $\Sigma_{(k)}$ from above.) Let $w \in \Sigma^\infty$ be a word and σ be an assignment from the variables in V to the positions in w, thus $0 \leq \sigma(x) < |w|$ for all $x \in V$. The pair (w, σ) can be encoded as a word $\overline{(w,\sigma)}$ over Σ_V. More precisely, if $w = a_0 a_1 a_2 \cdots$ then $\overline{(w,\sigma)} = (a_0, \tau_0)(a_1, \tau_1)(a_2, \tau_2) \cdots$ where for all $0 \leq i < |w|$ we have $\tau_i(x) = 1$ if and only if $\sigma(x) = i$. We let $\mathcal{N}_V \subseteq \Sigma_V^\infty$ be the set of words $\overline{(w,\sigma)}$ such that $w \in \Sigma^\infty$ and σ is an assignment from V to the positions in w. We show that $\mathcal{N}_V$ is star-free. For $x \in V$, let $\Sigma_V^{x=1}$ be the set of pairs (a, τ) with $\tau(x) = 1$ and let $\Sigma_V^{x=0} = \Sigma_V \setminus \Sigma_V^{x=1}$ be its complement. Then,

$$\mathcal{N}_V = \bigcap_{x \in V} (\Sigma_V^{x=0})^* \Sigma_V^{x=1} (\Sigma_V^{x=0})^\infty.$$

Given a first-order formula φ and a set V containing all free variables of φ, we define the semantics $[\![\varphi]\!]_V \subseteq \mathcal{N}_V$ inductively:

$$\begin{aligned}
[\![P_a(x)]\!]_V &= \{\overline{(w,\sigma)} \in \mathcal{N}_V \mid w = b_0 b_1 b_2 \cdots \in \Sigma^\infty \text{ and } b_{\sigma(x)} = a\} \\
[\![x < y]\!]_V &= \{\overline{(w,\sigma)} \in \mathcal{N}_V \mid \sigma(x) < \sigma(y)\} \\
[\![\exists x, \varphi]\!]_V &= \{\overline{(w,\sigma)} \in \mathcal{N}_V \mid \exists i, 0 \leq i < |w| \wedge \overline{(w, \sigma[x \to i])} \in [\![\varphi]\!]_{V \cup \{x\}}\} \\
[\![\varphi \vee \psi]\!]_V &= [\![\varphi]\!]_V \cup [\![\psi]\!]_V \\
[\![\neg\varphi]\!]_V &= \mathcal{N}_V \setminus [\![\varphi]\!]_V.
\end{aligned}$$

Proposition 4.2. Let φ be a first-order formula and V be a set of variables containing the free variables of φ. Then, $[\![\varphi]\!]_V \in \mathbf{SF}(\Sigma_V^\infty)$.

Proof. The proof is by induction on the formula. We have

$$\begin{aligned}
[\![P_a(x)]\!]_V &= \mathcal{N}_V \cap (\Sigma_V^* \cdot \{(a,\tau) \mid \tau(x) = 1\} \cdot \Sigma_V^\infty) \\
[\![x < y]\!]_V &= \mathcal{N}_V \cap (\Sigma_V^* \cdot \Sigma_V^{x=1} \cdot \Sigma_V^* \cdot \Sigma_V^{y=1} \cdot \Sigma_V^\infty).
\end{aligned}$$

The induction is trivial for disjunction and negation since the star-free sets form a Boolean algebra and $\mathcal{N}_V$ is star-free. The interesting case is existential quantification $[\![\exists x, \varphi]\!]_V$.

We assume first that $x \notin V$ and we let $V' = V \cup \{x\}$. By induction, $[\![\varphi]\!]_{V'}$ is star-free and we can apply Lemma 3.2 with the sets $A = \Sigma_{V'}^{x=1}$ and $B = \Sigma_{V'}^{x=0}$. Note that $\mathcal{N}_{V'} \subseteq B^* A B^\infty$. Hence, $[\![\varphi]\!]_{V'} = [\![\varphi]\!]_{V'} \cap B^* A B^\infty$ and we obtain $[\![\varphi]\!]_{V'} = \bigcup_{1 \leq i \leq n} K'_i a'_i L'_i$ where $a'_i \in A$, $K'_i \in \mathbf{SF}(B^*)$ and $L'_i \in \mathbf{SF}(B^\infty)$ for all i. Let $\pi : B^\infty \to \Sigma_V^\infty$ be the *bijective renaming* defined by $\pi(a,\tau) = (a, \tau_{\restriction V})$. Star-free sets are not preserved by projections but indeed they are preserved by bijective renamings. Hence, $K_i = \pi(K'_i) \in \mathbf{SF}(\Sigma_V^*)$ and $L_i = \pi(L'_i) \in \mathbf{SF}(\Sigma_V^\infty)$. We also rename $a'_i = (a,\tau)$ into $a_i = (a, \tau_{\restriction V})$. We have $[\![\exists x, \varphi]\!]_V = \bigcup_{1\leq i\leq n} K_i a_i L_i$ and we deduce that $[\![\exists x, \varphi]\!]_V \in \mathbf{SF}(\Sigma_V^\infty)$.

Finally, if $x \in V$ then we choose a new variable $y \notin V$ and we let $U = (V \setminus \{x\}) \cup \{y\}$. From the previous case, we get $[\![\exists x, \varphi]\!]_U \in \mathbf{SF}(\Sigma_U^\infty)$. To conclude, it remains to rename y to x. Q.E.D.

Corollary 4.3. We have:

$$\mathrm{FO}(\Sigma^*) \subseteq \mathbf{SF}(\Sigma^*) \text{ and } \mathrm{FO}(\Sigma^\infty) \subseteq \mathbf{SF}(\Sigma^\infty).$$

5 Aperiodic languages

Recall that a monoid $(M, \cdot)$ is a non-empty set M together with a binary operation $\cdot$ such that $((x \cdot y) \cdot z) = (x \cdot (y \cdot z))$ and with a neutral element $1 \in M$ such that $x \cdot 1 = 1 \cdot x = x$ for all x, y, z in M. Frequently we write xy instead of $x \cdot y$.

A *morphism* (or *homomorphism*) between monoids M and M' is a mapping $h : M \to M'$ such that $h(1) = 1$ and $h(x \cdot y) = h(x) \cdot h(y)$.

We use the algebraic notion of *recognizability* and the notion of *aperiodic* languages. Recognizability is defined as follows. Let $h : \Sigma^* \to M$ be a morphism to a finite monoid M. Two words $u, v \in \Sigma^\infty$ are said to be h-similar, denoted by $u \sim_h v$, if for some $n \in \mathbb{N} \cup \{\omega\}$ we can write $u = \prod_{0\leq i<n} u_i$ and $v = \prod_{0\leq i<n} v_i$ with u_i, $v_i \in \Sigma^+$ and $h(u_i) = h(v_i)$ for all $0 \leq i < n$. The notation $u = \prod_{0\leq i<n} u_i$ refers to an *ordered product*, it means a factorization $u = u_0 u_1 \cdots$. In other words, $u \sim_h v$ if either $u = v = \varepsilon$, or $u, v \in \Sigma^+$ and $h(u) = h(v)$ or $u, v \in \Sigma^\omega$ and there are factorizations $u = u_0 u_1 \cdots$, $v = v_0 v_1 \cdots$ with $u_i, v_i \in \Sigma^+$ and $h(u_i) = h(v_i)$ for all $i \geq 0$.

The transitive closure of $\sim_h$ is denoted by $\approx_h$; it is an equivalence relation. For $w \in \Sigma^\infty$, we denote by $[w]_h$ the equivalence class of w under $\approx_h$. Thus,

$$[w]_h = \{u \mid u \approx_h w\}.$$

In case that there is no ambiguity, we simply write $[w]$ instead of $[w]_h$. Note that there are three cases $[w] = \{\varepsilon\}$, $[w] \subseteq \Sigma^+$, and $[w] \subseteq \Sigma^\omega$.

Definition 5.1. We say that a morphism $h : \Sigma^* \to M$ *recognizes* L, if $w \in L$ implies $[w]_h \subseteq L$ for all $w \in \Sigma^\infty$.

Thus, a language $L \subseteq \Sigma^\infty$ is *recognized* by h if and only if L is saturated by $\approx_h$ (or equivalently by $\sim_h$). Note that we may assume that a recognizing morphism $h : \Sigma^* \to M$ is surjective, whenever convenient.

Since M is finite, the equivalence relation $\approx_h$ is of finite index. More precisely, there are at most $1+|M|+|M|^2$ classes. This fact can be derived by some standard Ramsey argument about infinite monochromatic subgraphs. We repeat the argument below in order to keep the article self-contained, see also [3, 12, 25]. It shows the existence of a so-called *Ramsey factorization.*

Lemma 5.2. Let $h : \Sigma^* \to M$ be a morphism to a finite monoid M and $w = u_0u_1u_2\cdots$ be an infinite word with $u_i \in \Sigma^+$ for $i \geq 0$. Then there exist $s, e \in M$, and an increasing sequence $0 < p_1 < p_2 < \cdots$ such that the following two properties hold:

1. $se = s$ and $e^2 = e$.

2. $h(u_0 \cdots u_{p_1-1}) = s$ and $h(u_{p_i} \cdots u_{p_j-1}) = e$ for all $0 < i < j$.

Proof. Let $E = \left\{(i,j) \in \mathbb{N}^2 \;\middle|\; i < j\right\}$. We consider the mapping $c : E \to M$ defined by $c(i,j) = h(u_i \cdots u_{j-1})$. We may think that the pairs (i,j) are (edges of an infinite complete graph and) *colored* by $c(i,j)$. Next we wish to color an infinite set of positions.

We define inductively a sequence of infinite sets $\mathbb{N} = N_0 \supset N_1 \supset N_2 \cdots$ and a sequence of natural numbers $n_0 < n_1 < n_2 < \cdots$ as follows. Assume that N_p is already defined and infinite. (This is true for $p = 0$.) Choose any $n_p \in N_p$, e.g., $n_p = \min N_p$. Since M is finite and N_p is infinite, there exists $c_p \in M$ and an infinite subset $N_{p+1} \subset N_p$ such that $c(n_p, m) = c_p$ for all $m \in N_{p+1}$. Thus, for all $p \in \mathbb{N}$ infinite sets N_p are defined and for every position n_p we may choose the color c_p. Again, because M is finite, one color must appear infinitely often. This color is called e and it is just the (idempotent) element of M we are looking for. Therefore we find a strictly increasing sequence $p_0 < p_1 < p_2 < \cdots$ such that $c_{p_i} = e$ and hence $e = h(u_{p_i} \cdots u_{p_j-1})$ for all $0 \leq i < j$. Note that $e = c(n_{p_0}, n_{p_2}) = c(n_{p_0}, n_{p_1})c(n_{p_1}, n_{p_2}) = e^2$. Moreover, if we set $s = h(u_0 \cdots u_{p_1-1})$, we obtain

$$s = c(0, n_{p_1}) = c(0, n_{p_0})c(n_{p_0}, n_{p_1}) = c(0, n_{p_0})c(n_{p_0}, n_{p_1})c(n_{p_1}, n_{p_2}) = se.$$

This is all we need. Q.E.D.

The lemma implies that for each (infinite) word w we may choose some $(s,e) \in M \times M$ with $s = se$ and $e = e^2$ such that $w \in h^{-1}(s)\,(h^{-1}(e))^\omega$. This establishes that $\approx_h$ has at most $|M|^2$ classes $[w]$ where w is infinite; and this in turn implies the given bound $1 + |M| + |M|^2$.

Pairs $(s,e) \in M \times M$ with $s = se$ and $e = e^2$ are also called *linked pair.*

Remark 5.3. The existence of a Ramsey factorization implies that a language $L \subseteq \Sigma^\omega$ recognized by a morphism h from Σ^* to some finite monoid M

can be written as a finite union of languages of type UV^ω, where $U, V \subseteq \Sigma^*$ are recognized by h and where moreover $U = h^{-1}(s)$ and $V = h^{-1}(e)$ for some $s, e \in M$ with $se = s$ and $e^2 = e$. In particular, we have $UV \subseteq U$ and $VV \subseteq V$. Since $\{\varepsilon\}^\omega = \{\varepsilon\}$, the statement holds for $L \subseteq \Sigma^*$ and $L \subseteq \Sigma^\infty$ as well.

A (finite) monoid M is called *aperiodic*, if for all $x \in M$ there is some $n \in \mathbb{N}$ such that $x^n = x^{n+1}$.

Definition 5.4. A language $L \subseteq \Sigma^\infty$ is called *aperiodic*, if it is recognized by some morphism to a finite and aperiodic monoid. By $\mathbf{AP}(\Sigma^*)$ (resp. $\mathbf{AP}(\Sigma^\infty)$) we denote the set of aperiodic languages in Σ^* (resp. Σ^∞), and by $\mathbf{AP}$ we denote the family of aperiodic languages.

6 From star-freeness to aperiodicity

Corollary 4.3 (as well as Proposition 10.3) tells us that all first-order definable languages are star-free. We want to show that all star-free languages are recognized by aperiodic monoids. Note that the trivial monoid recognizes the language Σ^*, actually it recognizes all eight Boolean combinations of $\{\varepsilon\}$ and Σ^ω.

Consider next a letter a. The smallest recognizing monoid of the singleton $\{a\}$ is aperiodic, it has just three elements $1, a, 0$ with $a \cdot a = 0$ and 0 is a *zero*, this means $x \cdot y = 0$ as soon as $0 \in \{x, y\}$.

Another very simple observation is that if L_i is recognized by a morphism $h_i : \Sigma^* \to M_i$ to some finite (aperiodic) monoid M_i, $i = 1, 2$, then (the direct product $M_1 \times M_2$ is aperiodic and) the morphism

$$h : \Sigma^* \to M_1 \times M_2,\ w \mapsto (h_1(w), h_2(w))$$

recognizes all Boolean combinations of L_1 and L_2.

The proof of the next lemma is rather technical. Its main part shows that the family of recognizable languages is closed under concatenation. Aperiodicity comes into the picture only at the very end in a few lines. There is alternative way to prove the following lemma. In Section 11 we introduce non-deterministic counter-free Büchi automata which can be used to show the closure under concatenation as well, see Lemma 11.3.

Lemma 6.1. Let $L \subseteq \Sigma^*$ and $K \subseteq \Sigma^\infty$ be aperiodic languages. Then $L \cdot K$ is aperiodic.

Proof. As said above, we may choose a single morphism $h : \Sigma^* \to M$ to some finite aperiodic monoid M, which recognizes both L and K.

The set of pairs $(h(u), h(v))$ with $u, v \in \Sigma^*$ is finite (bounded by $|M|^2$) and so its power set S is finite, too. We shall see that there is a monoid structure on some subset of S such that this monoid recognizes $L \cdot K$.

To begin with, let us associate with $w \in \Sigma^*$ the following set of pairs:

$$g(w) = \{(h(u), h(v)) \mid w = uv\}\,.$$

The finite set $g(\Sigma^*) \subseteq S$ is in our focus. We define a multiplication by:

$$\begin{aligned} g(w) \cdot g(w') &= g(ww') \\ &= \{(h(wu'), h(v')) \mid w' = u'v'\} \cup \{(h(u), h(vw')) \mid w = uv\}\,. \end{aligned}$$

The product is well-defined. To see this, observe first that $(h(u), h(v)) \in g(w)$ implies $h(w) = h(u)h(v)$ since h is a morphism. Thus, the set $g(w)$ *knows* the element $h(w)$. Second, $h(wu') = h(w)h(u')$ since h is a morphism. Hence, we can compute $\{(h(wu'), h(v')) \mid w' = u'v'\}$ from $g(w)$ and $g(w')$. The argument for the other component is symmetric.

By the very definition of g, we obtain a morphism

$$g : \Sigma^* \to g(\Sigma^*).$$

In order to see that g recognizes $L \cdot K$ consider $u \in L \cdot K$ and v such that we can write $u = \prod_{0 \le i < n} u_i$ and $v = \prod_{0 \le i < n} v_i$ with $u_i,\ v_i \in \Sigma^+$ and $g(u_i) = g(v_i)$ for all $0 \le i < n$. We have to show $v \in L \cdot K$. We have $u \in L \cdot K = (L \cap \Sigma^*) \cdot K$. Hence, for some index j we can write $u_j = u'_j u''_j$ with

$$\left(\prod_{0 \le i < j} u_i\right) u'_j \in L \quad \text{and} \quad u''_j \left(\prod_{j < i < n} u_i\right) \in K.$$

Now, $g(u_i) = g(v_i)$ implies $h(u_i) = h(v_i)$. Moreover, $u_j = u'_j u''_j$ implies $(h(u'_j), h(u''_j)) \in g(u_j) = g(v_j)$. Hence we can write $v_j = v'_j v''_j$ with $h(u'_j) = h(v'_j)$ and $h(u''_j) = h(v''_j)$. Therefore

$$\left(\prod_{0 \le i < j} v_i\right) v'_j \in L \quad \text{and} \quad v''_j \left(\prod_{j < i < n} v_i\right) \in K$$

and $v \in L \cdot K$, too.

It remains to show that the resulting monoid is indeed aperiodic. To see this choose some $n > 0$ such that $x^n = x^{n+1}$ for all $x \in M$. Consider any element $g(w) \in g(\Sigma^*)$. We show that $g(w)^{2n} = g(w)^{2n+1}$. This is straightforward:

$$g(w)^{2n} = g(w^{2n}) = \left\{(h(w^k u), h(v w^m)) \;\middle|\; w = uv,\ k + m = 2n - 1\right\}.$$

If $k+m = 2n-1$ then either $k \ge n$ or $m \ge n$. Hence, for each pair, we have either $(h(w^k u), h(vw^m)) = (h(w^{k+1} u), h(vw^m))$ or $(h(w^k u), h(vw^m)) = (h(w^k u), h(vw^{m+1}))$. The result follows. Q.E.D.

Proposition 6.2. We have $\mathbf{SF} \subseteq \mathbf{AP}$ or more explicitly:

$$\mathbf{SF}(\Sigma^*) \subseteq \mathbf{AP}(\Sigma^*) \text{ and } \mathbf{SF}(\Sigma^\infty) \subseteq \mathbf{AP}(\Sigma^\infty).$$

Proof. Aperiodic languages form a Boolean algebra. We have seen above that $\mathbf{AP}$ contains Σ^* and all singletons $\{a\}$, where a is a letter. Thus, star-free languages are aperiodic by Lemma 6.1. Q.E.D.

7 From LTL to FO^3

The syntax of $\mathrm{LTL}_\Sigma[\mathsf{XU}, \mathsf{YS}]$ is given by

$$\varphi ::= \bot \mid a \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \,\mathsf{XU}\, \varphi \mid \varphi \,\mathsf{YS}\, \varphi,$$

where a ranges over Σ. When there is no ambiguity, we simply write LTL for $\mathrm{LTL}_\Sigma[\mathsf{XU}, \mathsf{YS}]$. We also write $\mathrm{LTL}_\Sigma[\mathsf{XU}]$ for the pure future fragment where only the *next-until* modality XU is allowed.

In order to give a semantics to an LTL formula we identify each $\varphi \in$ LTL with some first-order formula $\varphi(x)$ in at most one free variable. The identification is done by structural induction. $\top$ and $\bot$ still denote the truth value *true* and *false*, the formula a becomes $a(x) = P_a(x)$. The formulae *neXt-Until* and *Yesterday-Since* are defined by:

$$(\varphi \,\mathsf{XU}\, \psi)(x) = \exists z :\; x < z \wedge \psi(z) \wedge \forall y :\; x < y < z \to \varphi(y).$$
$$(\varphi \,\mathsf{YS}\, \psi)(x) = \exists z :\; x > z \wedge \psi(z) \wedge \forall y :\; x > y > z \to \varphi(y).$$

It is clear that each LTL formula becomes under this identification a first-order formula which needs at most three different names for variables. For simplicity let us denote this fragment by FO^3, too. Thus, we can write $\mathrm{LTL} \subseteq \mathrm{FO}^3$.

As usual, we may use derived formulas such as $\mathsf{X}\,\varphi = \bot \,\mathsf{XU}\, \varphi$ (read *neXt* φ), $\varphi \,\mathsf{U}\, \psi = \psi \vee (\varphi \wedge (\varphi \,\mathsf{XU}\, \psi))$ (read φ *Until* ψ), $\mathsf{F}\,\varphi = \top \,\mathsf{U}\, \varphi$ (read *Future* φ), etc.

Since $\mathrm{LTL} \subseteq \mathrm{FO}^3$ a model of an LTL_Σ formula φ is a word $v = a_0 a_1 a_2 \cdots \in A^\infty \setminus \{\varepsilon\}$ together with a position $0 \le i < |v|$ (the alphabet A might be different from Σ).

For a formula $\varphi \in \mathrm{LTL}_\Sigma$ and an alphabet A, we let

$$\mathcal{L}_A(\varphi) = \{v \in A^\infty \setminus \{\varepsilon\} \mid v, 0 \models \varphi\}.$$

We say that a language $L \subseteq A^\infty$ is definable in LTL_Σ if $L \setminus \{\varepsilon\} = \mathcal{L}_A(\varphi)$ for some $\varphi \in \mathrm{LTL}_\Sigma$. Note that the empty word ε cannot be a model of a formula. To include the empty word, it will be convenient to consider for any letter c (not necessarily in A), the language

$$\mathcal{L}_{c,A}(\varphi) = \{v \in A^\infty \mid cv, 0 \models \varphi\}.$$

Remark 7.1. When we restrict to the pure future fragment $\mathrm{LTL}_\Sigma[\mathsf{XU}]$ the two approaches define almost the same class of languages. Indeed, for each formula $\varphi \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$, we have $\mathcal{L}_A(\varphi) = \mathcal{L}_{c,A}(\mathsf{X}\,\varphi) \setminus \{\varepsilon\}$. Conversely, for each formula φ there is a formula $\overline{\varphi}$ such that $\mathcal{L}_A(\overline{\varphi}) = \mathcal{L}_{c,A}(\varphi) \setminus \{\varepsilon\}$. The translation is simply $\overline{\varphi\,\mathsf{XU}\,\psi} = \overline{\varphi}\,\mathsf{U}\,\overline{\psi}$, $\overline{c} = \top$ and $\overline{a} = \bot$ if $a \neq c$, and as usual $\overline{\neg\varphi} = \neg\overline{\varphi}$ and $\overline{\varphi \vee \psi} = \overline{\varphi} \vee \overline{\psi}$.

8 From AP to LTL

8.1 A construction on monoids

The passage from **AP** to LTL is perhaps the most difficult step in completing the picture of first-order definable languages. We shall use an induction on the size of the monoid M, for this we recall first a construction due to [5].

For a moment let M be any monoid and $m \in M$ an element. Then $mM \cap Mm$ is obviously a subsemigroup, but it may not have a neutral element. Hence it is not a monoid, in general. Note that, if $m \neq 1_M$ and M is aperiodic, then $1_M \notin mM \cap Mm$. Indeed, assume that $1_M \in mM$ and write $1_M = mx$ with $x \in M$. Hence $1_M = m^n x^n$ for all n, and for some $n \geq 0$ we have $m^n = m^{n+1}$. Taking this n we see:

$$1_M = m^n x^n = m^{n+1} x^n = m(m^n x^n) = m1_M = m.$$

Therefore $|mM \cap Mm| < |M|$, if M is aperiodic and if $m \neq 1_M$.

It is possible to define a new product $\circ$ such that $mM \cap Mm$ becomes a monoid where m is a neutral element: We let

$$xm \circ my = xmy$$

for $xm, my \in mM \cap Mm$. This is well-defined since $xm = x'm$ and $my = my'$ imply $xmy = x'my'$. The operation is associative and $m \circ z = z \circ m = z$. Hence $(mM \cap Mm, \circ, m)$ is indeed a monoid. Actually it is a divisor of M. To see this consider the submonoid $N = \{x \in M \mid xm \in mM\}$. (Note that N is indeed a submonoid of M.) Clearly, the mapping $x \mapsto xm$ yields a surjective morphism from $(N, \cdot, 1_M)$ onto $(mM \cap Mm, \circ, m)$, which is therefore a homomorphic image of the submonoid N of M. In particular, if M is aperiodic, then $(mM \cap Mm, \circ, m)$ is aperiodic, too. The construction is very similar to a construction of what is known as *local algebra*, see [8, 20]. Therefore we call $(mM \cap Mm, \circ, m)$ the *local divisor* of M at the element m.

8.2 Closing the cycle

Proposition 8.1. We have $\mathbf{AP} \subseteq \mathbf{LTL}$. More precisely, let $L \subseteq \Sigma^\infty$ be a language recognized by an aperiodic monoid M.

(1) We can construct a formula $\varphi \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$ such that $L \setminus \{\varepsilon\} = \mathcal{L}_\Sigma(\varphi)$.

(2) For any letter c (not necessarily in Σ), we can construct a formula $\varphi \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$ such that $L = \mathcal{L}_{c,\Sigma}(\varphi)$.

Proof. Note first that (1) follows from (2) by Remark 7.1. The proof of (2) is by induction on $(|M|, |\Sigma|)$ (with lexicographic ordering). Let $h : \Sigma^* \to M$ be a morphism to the aperiodic monoid M. The assertion of Proposition 8.1 is almost trivial if $h(c) = 1_M$ for all $c \in \Sigma$. Indeed, in this case, the set L is a Boolean combination of the sets $\{\varepsilon\}$, Σ^+ and Σ^ω which are easily definable in $\mathrm{LTL}_\Sigma[\mathsf{XU}]$: we have $\{\varepsilon\} = \mathcal{L}_{c,\Sigma}(\neg \mathsf{X} \top)$, $\Sigma^+ = \mathcal{L}_{c,\Sigma}(\mathsf{X}\,\mathsf{F} \neg \mathsf{X} \top)$ and $\Sigma^\omega = \mathcal{L}_{c,\Sigma}(\neg \mathsf{F} \neg \mathsf{X} \top)$. Note that when $|M| = 1$ or $|\Sigma| = 0$ then we have $h(c) = 1_M$ for all $c \in \Sigma$ and this special case ensures the base of the induction.

In the following, we fix a letter $c \in \Sigma$ such that $h(c) \neq 1_M$ and we let $A = \Sigma \setminus \{c\}$. We define the c-factorization of a word $v \in \Sigma^\infty$. If $v \in (A^*c)^\omega$ then its c-factorization is $v = v_0cv_1cv_2c \cdots$ with $v_i \in A^*$ for all $i \geq 0$. If $v \in (A^*c)^*A^\infty$ then its c-factorization is $v = v_0cv_1c \cdots v_{k-1}cv_k$ where $k \geq 0$ and $v_i \in A^*$ for $0 \leq i < k$ and $v_k \in A^\infty$.

Consider two new disjoint alphabets $T_1 = \{h(u) \mid u \in A^*\}$ and $T_2 = \{[u]_h \mid u \in A^\infty\}$. Let $T = T_1 \uplus T_2$ and define the mapping $\sigma : \Sigma^\infty \to T^\infty$ by $\sigma(v) = h(v_0)h(v_1)h(v_2) \cdots \in T_1^\omega$ if $v \in (A^*c)^\omega$ and its c-factorization is $v = v_0cv_1cv_2c \cdots$, and $\sigma(v) = h(v_0)h(v_1) \cdots h(v_{k-1})[v_k]_h \in T_1^*T_2$ if $v \in (A^*c)^*A^\infty$ and its c-factorization is $v = v_0cv_1c \cdots v_{k-1}cv_k$.

Lemma 8.2. Let $L \subseteq \Sigma^\infty$ be a language recognized by h. There exists a language $K \subseteq T^\infty$ which is definable in $\mathrm{LTL}_T[\mathsf{XU}]$ and such that $L = \sigma^{-1}(K)$.

Proof. We have seen that the local divisor $M' = h(c)M \cap Mh(c)$ is an aperiodic monoid with composition $\circ$ and neutral element $h(c)$. Moreover, $|M'| < |M|$ since $h(c) \neq 1_M$. Let us define a morphism $g : T^* \to M'$ as follows. For $m = h(u) \in T_1$ we define $g(m) = h(c)mh(c) = h(cuc)$. For $m \in T_2$ we let $g(m) = h(c)$, which is the neutral element in M'.

Let $K_0 = \{[u]_h \mid u \in L \cap A^\infty\} \subseteq T_2$. We claim that $L \cap A^\infty = \sigma^{-1}(K_0)$. One inclusion is clear. Conversely, let $v \in \sigma^{-1}(K_0)$. There exists $u \in L \cap A^\infty$ such that $\sigma(v) = [u]_h \in T_2$. By definition of σ, this implies $v \in A^\infty$ and $v \approx_h u$. Since $u \in L$ and L is recognized by h, we get $v \in L$ as desired.

For $n \in T_1$ and $m \in T_2$, let $K_{n,m} = nT_1^*m \cap n[n^{-1}\sigma(L) \cap T_1^*m]_g$ and let $K_1 = \bigcup_{n \in T_1, m \in T_2} K_{n,m}$. We claim that $L \cap (A^*c)^+A^\infty = \sigma^{-1}(K_1)$. Let first $v \in L \cap (A^*c)^+A^\infty$ and write $v = v_0cv_1 \cdots cv_k$ its c-factorization. With $n = h(v_0)$ and $m = [v_k]_h$ we get $\sigma(v) \in K_{n,m}$. Conversely, let $v \in \sigma^{-1}(K_{n,m})$ with $n \in T_1$ and $m \in T_2$. We have $v \in (A^*c)^+A^\infty$ and its c-factorization is $v = v_0cv_1 \cdots cv_k$ with $k > 0$, $h(v_0) = n$ and $[v_k]_h = m$. Moreover, $x = h(v_1) \cdots h(v_{k-1})[v_k]_h \in [n^{-1}\sigma(L) \cap T_1^*m]_g$ hence we find $y \in T_1^*m$ with $g(x) = g(y)$ and $ny \in \sigma(L)$. Let $u \in L$ be such that $\sigma(u) = ny \in nT_1^* m$. Then $u \in (A^*c)^+ A^\infty$ and its c-factorization is $u = u_0 c u_1 \cdots c u_\ell$ with $\ell > 0$, $h(u_0) = n$ and $[u_\ell]_h = m$. By definition of g, we get $h(cv_1 c \cdots c v_{k-1} c) = g(x) = g(y) = h(cu_1 c \cdots c u_{\ell-1} c)$. Using $h(v_0) = n = h(u_0)$ and $[v_k]_h = m = [u_\ell]_h$, we deduce that $v \approx_h u$. Since $u \in L$ and L is recognized by h, we get $v \in L$ as desired.

For $n \in T_1$, let $K_{n,\omega} = nT_1^\omega \cap n[n^{-1}\sigma(L) \cap T_1^\omega]_g$ and let $K_2 = \bigcup_{n \in T_1} K_{n,\omega}$. As above, we shall show that $L \cap (A^*c)^\omega = \sigma^{-1}(K_2)$. So let $v \in L \cap (A^*c)^\omega$ and consider its c-factorization $v = v_0 c v_1 c v_2 \cdots$. With $n = h(v_0)$, we get $\sigma(v) \in K_{n,\omega}$. To prove the converse inclusion we need some auxiliary results.

First, if $x \sim_g y \sim_g z$ with $x \in T^\omega$ and $|y|_{T_1} < \omega$ then $x \sim_g z$. Indeed, in this case, we find factorizations $x = x_0 x_1 x_2 \cdots$ and $y = y_0 y_1 y_2 \cdots$ with $x_i \in T^+$, $y_0 \in T^+$ and $y_i \in T_2^+$ for $i > 0$ such that $g(x_i) = g(y_i)$ for all $i \geq 0$. Similarly, we find factorizations $z = z_0 z_1 z_2 \cdots$ and $y = y_0' y_1' y_2' \cdots$ with $z_i \in T^+$, $y_0' \in T^+$ and $y_i' \in T_2^+$ for $i > 0$ such that $g(z_i) = g(y_i')$ for all $i \geq 0$. Then, we have $g(x_i) = g(y_i) = h(c) = g(y_i') = g(z_i)$ for all $i > 0$ and $g(x_0) = g(y_0) = g(y_0') = g(z_0)$ since y_0 and y_0' contain all letters of y from T_1 and g maps all letters from T_2 to the neutral element of M'.

Second, if $x \sim_g y \sim_g z$ with $|y|_{T_1} = \omega$ then $x \sim_g y' \sim_g z$ for some $y' \in T_1^\omega$. Indeed, in this case, we find factorizations $x = x_0 x_1 x_2 \cdots$ and $y = y_0 y_1 y_2 \cdots$ with $x_i \in T^+$, and $y_i \in T^* T_1 T^*$ such that $g(x_i) = g(y_i)$ for all $i \geq 0$. Let y_i' be the projection of y_i to the subalphabet T_1 and let $y' = y_0' y_1' y_2' \cdots \in T_1^\omega$. We have $g(y_i) = g(y_i')$, hence $x \sim_g y'$. Similarly, we get $y' \sim_g z$.

Third, if $\sigma(u) \sim_g \sigma(v)$ with $u, v \in (A^*c)^\omega$ then $cu \approx_h cv$. Indeed, since $u, v \in (A^*c)^\omega$, the c-factorizations of u and v are of the form $u_1 c u_2 c \cdots$ and $v_1 c v_2 c \cdots$ with $u_i, v_i \in A^*$. Using $\sigma(u) \sim_g \sigma(v)$, we find new factorizations $u = u_1' c u_2' c \cdots$ and $v = v_1' c v_2' c \cdots$ with $u_i', v_i' \in (A^*c)^* A^*$ and $h(cu_i'c) = h(cv_i'c)$ for all $i > 0$. We deduce

$$\begin{aligned} cu = (cu_1'c)u_2'(cu_3'c)u_4' \cdots &\sim_h (cv_1'c)u_2'(cv_3'c)u_4' \cdots = cv_1'(cu_2'c)v_3'(cu_4'c) \cdots \\ &\sim_h cv_1'(cv_2'c)v_3'(cv_4'c) \cdots = cv. \end{aligned}$$

We come back to the proof of $\sigma^{-1}(K_{n,\omega}) \subseteq L \cap (A^*c)^\omega$. So let $u \in \sigma^{-1}(K_{n,\omega})$. We have $u \in (A^*c)^\omega$ and $\sigma(u) = nx \in nT_1^\omega$ with $x \in [n^{-1}\sigma(L) \cap T_1^\omega]_g$. Let $y \in T_1^\omega$ be such that $x \approx_g y$ and $ny \in \sigma(L)$. Let $v \in L$ with $\sigma(v) = ny$. We may write $u = u_0 c u'$ and $v = v_0 c v'$ with $u_0, v_0 \in A^*$, $h(u_0) = n = h(v_0)$, $u', v' \in (A^*c)^\omega$, $x = \sigma(u')$ and $y = \sigma(v')$. Since $x \approx_g y$, using the first two auxiliary results above and the fact that the mapping $\sigma : (A^*c)^\omega \to T_1^\omega$ is surjective, we get $\sigma(u') \sim_g \sigma(w_1) \sim_g \cdots \sim_g \sigma(w_k) \sim_g \sigma(v')$ for some $w_1, \ldots, w_k \in (A^*c)^\omega$. From the third auxiliary result, we get $cu' \approx_h cv'$. Hence, using $h(u_0) = h(v_0)$, we obtain $u = u_0 c u' \approx_h v_0 c v' = v$. Since $v \in L$ and L is recognized by h, we get $u \in L$ as desired.

Finally, let $K = K_0 \cup K_1 \cup K_2$. We have already seen that $L = \sigma^{-1}(K)$. It remains to show that K is definable in $\mathrm{LTL}_T[\mathsf{XU}]$. Let $N \subseteq T^\infty$, then, by definition, the language $[N]_g$ is recognized by g which is a morphism to the aperiodic monoid M' with $|M'| < |M|$. By induction on the size of the monoid, we deduce that for all $n \in T_1$ and $N \subseteq T^\infty$ there exists $\varphi \in \mathrm{LTL}_T[\mathsf{XU}]$ such that $[N]_g = \mathcal{L}_{n,T}(\varphi)$. We easily check that $n\mathcal{L}_{n,T}(\varphi) = \mathcal{L}_T(n \wedge \varphi)$. Therefore, the language $n[N]_g$ is definable in $\mathrm{LTL}_T[\mathsf{XU}]$. Moreover, K_0, nT_1^*m and nT_1^ω are obviously definable in $\mathrm{LTL}_T[\mathsf{XU}]$. Therefore, K is definable in $\mathrm{LTL}_T[\mathsf{XU}]$. Q.E.D. (Lemma 8.2)

Let $b \in \Sigma$ be a letter. For a nonempty word $v = a_0a_1a_2 \cdots \in \Sigma^\infty \setminus \{\varepsilon\}$ and a position $0 \leq i < |v|$, we denote by $\mu_b(v,i)$ the largest factor of v starting at position i and not containing the letter b except maybe a_i. Formally, $\mu_b(v,i) = a_ia_{i+1} \cdots a_\ell$ where $\ell = \max\{k \mid i \leq k < |v| \text{ and } a_j \neq b \text{ for all } i < j \leq k\}$.

Lemma 8.3 (Lifting)**.** For each formula $\varphi \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$, there exists a formula $\overline{\varphi}^b \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$ such that for each $v \in \Sigma^\infty \setminus \{\varepsilon\}$ and each $0 \leq i < |v|$, we have $v, i \models \overline{\varphi}^b$ if and only if $\mu_b(v,i), 0 \models \varphi$.

Proof. The construction is by structural induction on φ. We let $\overline{a}^b = a$. Then, we have $\overline{\neg\varphi}^b = \neg\overline{\varphi}^b$ and $\overline{\varphi \vee \psi}^b = \overline{\varphi}^b \vee \overline{\psi}^b$ as usual. For next-until, we define $\overline{\varphi \mathbin{\mathsf{XU}} \psi}^b = (\neg b \wedge \overline{\varphi}^b) \mathbin{\mathsf{XU}} (\neg b \wedge \overline{\psi}^b)$.

Assume that $v, i \models \overline{\varphi \mathbin{\mathsf{XU}} \psi}^b$. We find $i < k < |v|$ such that $v, k \models \neg b \wedge \overline{\psi}^b$ and $v, j \models \neg b \wedge \overline{\varphi}^b$ for all $i < j < k$. We deduce that $\mu_b(v,i) = a_ia_{i+1} \cdots a_\ell$ with $\ell > k$ and that $\mu_b(v,i), k-i \models \psi$ and $\mu_b(v,i), j-i \models \varphi$ for all $i < j < k$. Therefore, $\mu_b(v,i), 0 \models \varphi \mathbin{\mathsf{XU}} \psi$ as desired. The converse can be shown similarly. Q.E.D. (Lemma 8.3)

Lemma 8.4. For all $\xi \in \mathrm{LTL}_T[\mathsf{XU}]$, there exists a formula $\widetilde{\xi} \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$ such that for all $v \in \Sigma^\infty$ we have $cv, 0 \models \widetilde{\xi}$ if and only if $\sigma(v), 0 \models \xi$.

Proof. The proof is by structural induction on ξ. The difficult cases are for the constants $m \in T_1$ or $m \in T_2$.

Assume first that $\xi = m \in T_1$. We have $\sigma(v), 0 \models m$ if and only if $v = ucv'$ with $u \in A^* \cap h^{-1}(m)$. The language $A^* \cap h^{-1}(m)$ is recognized by the restriction $h_{\restriction A} : A^* \to M$. By induction on the size of the alphabet, we find a formula $\varphi_m \in \mathrm{LTL}_A[\mathsf{XU}]$ such that $\mathcal{L}_{c,A}(\varphi_m) = A^* \cap h^{-1}(m)$. We let $\widetilde{m} = \overline{\varphi_m}^c \wedge \mathsf{XF}\, c$. By Lemma 8.3, we have $cv, 0 \models \widetilde{m}$ if and only if $v = ucv'$ with $u \in A^*$ and $\mu_c(cv, 0), 0 \models \varphi_m$. Since $\mu_c(cv,0) = cu$, we deduce that $cv, 0 \models \widetilde{m}$ if and only if $v = ucv'$ with $u \in \mathcal{L}_{c,A}(\varphi_m) = A^* \cap h^{-1}(m)$.

Next, assume that $\xi = m \in T_2$. We have $\sigma(v) \models m$ if and only if $v \in A^\infty \cap m$ (note that letters from T_2 can also be seen as equivalence classes which are subsets of Σ^∞). The language $A^\infty \cap m$ is recognized by the restriction $h_{\restriction A}$. By induction on the size of the alphabet, we find a formula $\psi_m \in \mathrm{LTL}_A[\mathsf{XU}]$ such that $\mathcal{L}_{c,A}(\psi_m) = A^\infty \cap m$. Then, we let $\widetilde{m} = \overline{\psi_m}^c \wedge \neg\, \mathsf{XF}\, c$ and we conclude as above.

Finally, we let $\widetilde{\neg\xi} = \neg\widetilde{\xi}$, $\widetilde{\xi_1 \vee \xi_2} = \widetilde{\xi_1} \vee \widetilde{\xi_2}$ and for the modality next-until we define $\widetilde{\xi_1 \,\mathsf{XU}\, \xi_2} = (\neg c \vee \widetilde{\xi_1})\; \mathsf{U}\; (c \wedge \widetilde{\xi_2})$.

Assume that $\sigma(v), 0 \models \xi_1 \,\mathsf{XU}\, \xi_2$ and let $0 < k < |\sigma(v)|$ be such that $\sigma(v), k \models \xi_2$ and $\sigma(v), j \models \xi_1$ for all $0 < j < k$. Let $v_0 c v_1 c v_2 c \cdots$ be the c-factorization of v. Since the logics $\mathrm{LTL}_T[\mathsf{XU}]$ and $\mathrm{LTL}_\Sigma[\mathsf{XU}]$ are pure future, we have $\sigma(v), k \models \xi_2$ if and only if $\sigma(v_k c v_{k+1} \cdots), 0 \models \xi_2$ if and only if (by induction) $c v_k c v_{k+1} \cdots, 0 \models \widetilde{\xi_2}$ if and only if $cv, |cv_0 \cdots cv_{k-1}| \models \widetilde{\xi_2}$. Similarly, $\sigma(v), j \models \xi_1$ if and only if $cv, |cv_0 \cdots cv_{j-1}| \models \widetilde{\xi_1}$. Therefore, $cv, 0 \models \widetilde{\xi_1 \,\mathsf{XU}\, \xi_2}$. The converse can be shown similarly. Q.E.D. (Lemma 8.4)

We conclude now the proof of Proposition 8.1. We start with a language $L \subseteq \Sigma^\infty$ recognized by h. By Lemma 8.2, we find a formula $\xi \in \mathrm{LTL}_T[\mathsf{XU}]$ such that $L = \sigma^{-1}(\mathcal{L}_T(\xi))$. Let $\widetilde{\xi}$ be the formula given by Lemma 8.4. We claim that $L = \mathcal{L}_{c,\Sigma}(\widetilde{\xi})$. Indeed, for $v \in \Sigma^\infty$, we have $v \in \mathcal{L}_{c,\Sigma}(\widetilde{\xi})$ if and only if $cv, 0 \models \widetilde{\xi}$ if and only if (Lemma 8.4) $\sigma(v), 0 \models \xi$ if and only if $\sigma(v) \in \mathcal{L}_T(\xi)$ if and only if $v \in \sigma^{-1}(\mathcal{L}_T(\xi)) = L$. Q.E.D. (Proposition 8.1)

9 The separation theorem

As seen in Section 7, an $\mathrm{LTL}_\Sigma[\mathsf{YS}, \mathsf{XU}]$ formula φ can be viewed as a first-order formula with one free variable. The converse, in a stronger form, is established in this section.

Proposition 9.1. For all first-order formulae ξ in one free variable we find a finite list $(K_i, a_i, L_i)_{i=1,\ldots,n}$ where each $K_i \in \mathbf{SF}(\Sigma^*)$ and each $L_i \in \mathbf{SF}(\Sigma^\infty)$ and a_i is a letter such that for all $u \in \Sigma^*$, $a \in \Sigma$ and $v \in \Sigma^\infty$ we have

$$(uav, |u|) \models \xi \text{ if and only if } u \in K_i, a = a_i \text{ and } v \in L_i \text{ for some } 1 \le i \le n.$$

Proof. By Proposition 4.2, with $V = \{x\}$ we have $[\![\xi]\!]_V \in \mathbf{SF}(\Sigma_V^\infty)$. Hence, we can use Lemma 3.2 with $A = \Sigma_V^{x=1}$ and $B = \Sigma_V^{x=0}$. Note that $\mathcal{N}_V = B^* A B^\infty$. Hence, we obtain

$$[\![\xi]\!]_V = \bigcup_{i=1,\ldots,n} K_i' \cdot a_i' \cdot L_i'$$

with $a_i' \in A$, $K_i' \in \mathbf{SF}(B^*)$ and $L_i' \in \mathbf{SF}(B^\infty)$ for all i. Let $\pi : B^\infty \to \Sigma^\infty$ be the *bijective renaming* defined by $\pi(a, \tau) = a$. Star-free sets are preserved by injective renamings. Hence, we can choose $K_i = \pi(K_i') \in \mathbf{SF}(\Sigma^*)$ and $L_i = \pi(L_i') \in \mathbf{SF}(\Sigma^\infty)$. Note also that $a_i' = (a_i, 1)$ for some $a_i \in \Sigma$. Q.E.D.

Theorem 9.2 (Separation)**.** Let $\xi(x) \in \mathrm{FO}_\Sigma(<)$ be a first-order formula with one free variable x. Then, $\xi(x) = \zeta(x)$ for some LTL formula $\zeta \in \mathrm{LTL}_\Sigma[\mathsf{YS}, \mathsf{XU}]$. Moreover, we can choose for ζ a disjunction of conjunctions of pure past and pure future formulae:

$$\zeta = \bigvee_{1 \leq i \leq n} \psi_i \wedge a_i \wedge \varphi_i$$

where $\psi_i \in \mathrm{LTL}_\Sigma[\mathsf{YS}]$, $a_i \in \Sigma$ and $\varphi_i \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$. In particular, every first-order formula with one free variable is equivalent to some formula in FO^3.

Note that we have already established a weaker version which applies to first-order sentences. Indeed, if ξ is a first-order sentence, then $\mathcal{L}(\varphi)$ is star-free by Proposition 10.3, hence aperiodic by Proposition 6.2, and finally definable in LTL by Proposition 8.1. The extension to first-order formulae with one free variable will also use the previous results.

Proof. By Proposition 9.1 we find for each ξ a finite list $(K_i, a_i, L_i)_{i=1,\ldots,n}$ where each $K_i \in \mathbf{SF}(\Sigma^*)$ and each $L_i \in \mathbf{SF}(\Sigma^\infty)$ and a_i is a letter such that for all $u \in \Sigma^*$, $a \in \Sigma$ and $v \in \Sigma^\infty$ we have

$$(uav, |u|) \models \xi \text{ if and only if } u \in K_i, a = a_i \text{ and } v \in L_i \text{ for some } 1 \leq i \leq n.$$

For a finite word $b_0 \cdots b_m$ where b_j are letters we let $\overleftarrow{b_0 \cdots b_m} = b_m \cdots b_0$. This means we read words from right to left. For a language $K \subseteq \Sigma^*$ we let $\overleftarrow{K} = \{\overleftarrow{w} \mid w \in K\}$. Clearly, each $\overleftarrow{K_i}$ is star-free. Therefore, using Propositions 6.2 and 8.1, for each $1 \leq i \leq n$ we find $\widehat{\psi}_i$ and $\varphi_i \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$ such that $\mathcal{L}_{a_i}(\widehat{\psi}_i) = \overleftarrow{K_i}$ and $\mathcal{L}_{a_i}(\varphi_i) = L_i$. Replacing all operators XU by YS we can transform $\widehat{\psi}_i \in \mathrm{LTL}_\Sigma[\mathsf{XU}]$ into a formula $\psi_i \in \mathrm{LTL}_\Sigma[\mathsf{YS}]$ such that $(a\overleftarrow{w}, 0) \models \widehat{\psi}_i$ if and only if $(wa, |w|) \models \psi_i$ for all $wa \in \Sigma^+$. In particular, $K_i = \{w \in \Sigma^* \mid wa_i, |w| \models \psi_i\}$.

It remains to show that $\xi(x) = \zeta(x)$ where $\zeta = \bigvee_{1 \leq i \leq n} \psi_i \wedge a_i \wedge \varphi_i$. Let $w \in \Sigma^\infty \setminus \{\varepsilon\}$ and p be a position in w.

Assume first that $(w, p) \models \xi(x)$ and write $w = uav$ with $|u| = p$. We have $u \in K_i$, $a = a_i$ and $v \in L_i$ for some $1 \leq i \leq n$. We deduce that $ua_i, |u| \models \psi_i$ and $a_i v, 0 \models \varphi_i$. Since ψ_i is pure past and φ_i is pure future, we deduce that $ua_i v, |u| \models \psi_i \wedge a_i \wedge \varphi_i$. Hence we get $w, p \models \zeta$.

Conversely, assume that $w, p \models \psi_i \wedge a_i \wedge \varphi_i$ for some i. As above, we write $w = ua_i v$ with $|u| = p$. Since ψ_i is pure past and φ_i is pure future, we deduce that $ua_i, |u| \models \psi_i$ and $a_i v, 0 \models \varphi_i$. Therefore, $u \in K_i$ and $v \in L_i$. We deduce that $(w, p) \models \xi(x)$. Q.E.D.

10 Variations

This section provides an alternative way to establish the bridge from first-order to star freeness and an alternative proof for Theorem 9.2.

There is a powerful tool to reason about first-oder definable languages which we did not discuss: Ehrenfeucht-Fraïssé-games. These games lead to an immediate proof of a congruence lemma, which is given in Lemma 10.2 below. On the other hand, in our context, it would be the only place where we could use the power of Ehrenfeucht-Fraïssé-games, therefore we skip this notion and we use Lemma 10.1 instead.

Before we continue we introduce a few more notations. The *quantifier depth* $\mathrm{qd}(\varphi)$ of a formula φ is defined inductively. For the atomic formulae $\perp$, P, and $x < y$ it is zero, the use of the logical connectives does not increase it, it is the maximum over the operands, but adding a quantifier in front increases the quantifier depth by one. For example, the following formula in one free variable y has quantifier depth two:

$$\forall x\ (\exists y\, P(x) \wedge \neg P(y)) \vee (\exists z\, P(z) \wedge (x < z) \vee (z < y))$$

By $\mathrm{FO}_{m,k}$ we mean the set of all formulae of quantifier depth at most m and where the free variables are in the set $\{x_1, \ldots, x_k\}$, and FO_m is a short-hand of $\mathrm{FO}_{m,0}$; it is the set of sentences of quantifier-depth at most m.

We say that formulae $\varphi, \psi \in \mathrm{FO}_{m,k}$ are equivalent if $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$ (for all Σ). Since the set of unary predicates is finite, there are, up to equivalence, only finitely many formulae in $\mathrm{FO}_{m,k}$ as soon as k and m are fixed. This is true for $m = 0$, because over any finite set of formulae there are, up to equivalence, only finitely many Boolean combinations. For $m > 0$ we have, by induction, only finitely many formulae of type $\exists x_{k+1}\, \varphi$ where φ ranges over $\mathrm{FO}_{m-1,k+1}$. A formula in $\mathrm{FO}_{m,k}$ is a Boolean combination over such formulae, as argued for $m = 0$ there are only finitely many choices.

10.1 The congruence lemma

Recall that $\Sigma^\infty_{(k)}$ means the set of pairs $(w, \overline{p})$, where $w \in \Sigma^\infty$ is a finite or infinite word and $\overline{p} = (p_1, \ldots, p_k)$ is a k-tuple of positions in $\mathrm{pos}(w)$. If we have $(u, \overline{p}) \in \Sigma^*_{(k)}$ and $(v, \overline{q}) \in \Sigma^\infty_{(\ell)}$, then we can define the concatenation in the natural way by shifting $\overline{q}$:

$$(u, \overline{p}) \cdot (v, \overline{q}) = (uv, p_1, \ldots, p_k, |u| + q_1, \ldots, |u| + q_\ell) \in \Sigma^\infty_{(k+\ell)}.$$

For each k and m and $(w, \overline{p}) \in \Sigma^\infty_{(k)}$ we define classes as follows:

$$[(w, \overline{p})]_{m,k} = \bigcap_{\varphi \in \mathrm{FO}_{m,k} \mid (w,\overline{p}) \models \varphi} \mathcal{L}(\varphi).$$

For $k = 0$ we simply write $[w]_{m,0}$. Since $\mathrm{qd}(\varphi) = \mathrm{qd}(\neg\varphi)$ and $\mathcal{L}(\neg\varphi) = \Sigma^{\infty}_{(k)} \setminus \mathcal{L}(\varphi)$ we obtain

$$\begin{aligned}[(w,\overline{p})]_{m,k} &= \bigcap_{\varphi\in\mathrm{FO}_{m,k}\,|\,(w,\overline{p})\models\varphi} \mathcal{L}(\varphi) \\ &= \bigcap_{\varphi\in\mathrm{FO}_{m,k}\,|\,(w,\overline{p})\models\varphi} \mathcal{L}(\varphi) \quad\setminus \bigcup_{\varphi\in\mathrm{FO}_{m,k}\,|\,(w,\overline{p})\not\models\varphi} \mathcal{L}(\varphi).\end{aligned}$$

Note that $(u',\overline{p}') \in [(u,\overline{p})]_{m,k}$ if and only if $(u,\overline{p}) \models \varphi \iff (u',\overline{p}') \models \varphi$ for all $\varphi \in \mathrm{FO}_{m,k}$ if and only if $[(u',\overline{p}')]_{m,k} = [(u,\overline{p})]_{m,k}$.

Lemma 10.1. Let $[(u,\overline{p})]_{m,k} = [(u',\overline{p}')]_{m,k}$ with $m \geq 1$, $\overline{p} = (p_1,\ldots,p_k)$, and $\overline{p}' = ({p'}_1,\ldots,{p'}_k)$. Then for all positions $p_{k+1} \in \mathrm{pos}(u)$ there exists a position $p'_{k+1} \in \mathrm{pos}(u')$ such that

$$[(u,(p_1,\ldots,p_{k+1}))]_{m-1,k+1} = [(u',\big({p'}_1,\ldots,{p'}_{k+1}\big))]_{m-1,k+1}.$$

Proof. Choose some $p_{k+1} \in \mathrm{pos}(u)$. We are looking for a position $p'_{k+1} \in \mathrm{pos}(u')$ such that for all $\psi \in \mathrm{FO}_{m-1,k+1}$ we have $(u,(p_1,\ldots,p_{k+1})) \models \psi$ if and only if $(u',\big({p'}_1,\ldots,{p'}_{k+1}\big)) \models \psi$.

Consider the following finite (up to equivalence) conjunction:

$$\Psi = \bigwedge_{\psi\in\mathrm{FO}_{m-1,k+1}\,|\,(u,(p_1,\ldots,p_{k+1}))\models\psi} \psi.$$

We have $(u,(p_1,\ldots,p_{k+1})) \models \Psi$, $\mathrm{qd}(\exists x_{k+1}\Psi) \leq m$ and $(u,\overline{p}) \models \exists x_{k+1}\Psi$. Hence $(u',\overline{p}') \models \exists x_{k+1}\Psi$; and therefore there is some $p'_{k+1} \in \mathrm{pos}(u')$ such that $(u',\big({p'}_1,\ldots,{p'}_{k+1}\big)) \models \Psi$.

Finally, for each $\psi \in \mathrm{FO}_{m-1,k+1}$, either Ψ implies ψ or Ψ implies $\neg\psi$, because either $(u,(p_1,\ldots,p_{k+1})) \models \psi$ or $(u,(p_1,\ldots,p_{k+1})) \models \neg\psi$. Hence, if $(u,(p_1,\ldots,p_{k+1})) \models \psi$, then $(u',\big({p'}_1,\ldots,{p'}_{k+1}\big)) \models \psi$, too. If $(u,(p_1,\ldots,p_{k+1})) \models \neg\psi$, then $(u',\big({p'}_1,\ldots,{p'}_{k+1}\big)) \models \neg\psi$, too. The result follows. Q.E.D.

The next lemma is known as *congruence lemma.*

Lemma 10.2. Let $[(u,\overline{p})]_{m,k} = [(u',\overline{p}')]_{m,k}$ and $[(v,\overline{q})]_{m,\ell} = [(v',\overline{q}')]_{m,\ell}$, where u and u' are finite words. Then we have

$$[(u,\overline{p})\cdot(v,\overline{q})]_{m,k+\ell} = [(u',\overline{p}')\cdot(v',\overline{q}')]_{m,k+\ell}.$$

Proof. We have to show that for all $\varphi \in \mathrm{FO}_{m,k}$ we have $(u,\overline{p})\cdot(v,\overline{q}) \models \varphi$ if and only if $(u',\overline{p}')\cdot(v',\overline{q}') \models \varphi$. Since we get Boolean combinations for free, we may assume that φ is of the form $\exists x_{k+1}\psi$ or an atomic formula.

If $\varphi = P(x_i)$ and $i \leq k$, then we have $(u, \overline{p}) \cdot (v, \overline{q}) \models P(x_i)$ if and only if $(u, \overline{p}) \models P(x_i)$ and the result follows. The case $i > k$ is symmetric.

If $\varphi = x_i < x_j$, assume first $i \leq k$. If, in addition, $j > k$, then $(u, \overline{p}) \cdot (v, \overline{q}) \models x_i < x_j$ is true, otherwise $i, j \leq k$ and we see that $(u, \overline{p}) \cdot (v, \overline{q}) \models x_i < x_j$ if and only if $(u, \overline{p}) \models x_i < x_j$. The case $i > k$ is similar.

It remains to deal with $\varphi = \exists x_{k+1} \psi$. Assume $(u, \overline{p}) \cdot (v, \overline{q}) \models \varphi$. We have to show that $(u', \overline{p}') \cdot (v', \overline{q}') \models \varphi$. Assume first that there is some position $p_{k+1} \in \mathrm{pos}(u)$ such that

$$(u, (p_1, \ldots, p_{k+1})) \cdot (v, \overline{q}) \models \psi.$$

By Lemma 10.1 there is some position $p'_{k+1} \in \mathrm{pos}(u')$ such that

$$[(u, (p_1, \ldots, p_{k+1}))]_{m-1,k+1} = [(u', (p'{}_1, \ldots, p'{}_{k+1}))]_{m-1,k+1}.$$

We have $\mathrm{qd}(\psi) \leq m - 1$, hence by induction on m we deduce

$$(u', (p'{}_1, \ldots, p'{}_{k+1})) \cdot (v', \overline{q}') \models \psi$$

This in turn implies

$$(u', \overline{p}') \cdot (v', \overline{q}') \models \exists x_{k+1} \psi.$$

The case where $(u, \overline{p}) \cdot (v, (q_1, \ldots, q_{\ell+1})) \models \psi$ for some position $q_{\ell+1}$ in v is similar. Q.E.D.

10.2 From FO to SF and separation via the congruence lemma

It is convenient to define a *dot-depth hierarchy*. The Boolean combinations of Σ^* are of dot-depth zero. In order to define the m-th level of the dot-depth hierarchy, $m \geq 1$, one forms the Boolean closure of the languages $K \cdot a \cdot L$, where $a \in \Sigma$ and K, L are of level at most $m - 1$. Note that there are only finitely many languages of level m.

Proposition 10.3. Let $m \geq 0$ and $\varphi \in \mathrm{FO}_m$ be a sentence with quantifier-depth at most m. Then we find a star-free language L of level at most m in the dot-depth hierarchy such that $\mathcal{L}(\varphi) = L$.

Proof. We perform an induction on m. The case $m = 0$ is trivial since the only sentences are $\top$ and $\bot$. Hence let $m > 0$. By definition,

$$[w]_{m-1,0} = \bigcap_{\psi \in \mathrm{FO}_{m-1} \mid w \models \psi} \mathcal{L}(\psi).$$

By induction on m we may assume that $[w]_{m-1,0}$ is star-free of dot-depth $m - 1$. Consider next a sentence $\varphi \in \mathrm{FO}_m$. We want to show that $\mathcal{L}(\varphi)$ is of dot-depth m. Languages of dot-depth m form a Boolean algebra, thus by structural induction it is enough to consider a sentence $\varphi = \exists x\psi$. Consider the following union:

$$T = \bigcup_{(uav,|u|)\models\psi} [u]_{m-1,0} \cdot a \cdot [v]_{m-1,0}.$$

Since $[u]_{m-1,0}$ and $[v]_{m-1,0}$ are star-free sets of dot-depth $m-1$, there are finitely many sets $[u]_{m-1,0} \cdot a \cdot [v]_{m-1,0}$ in the union above. In fact, it is a star-free expression of dot-depth m.

It remains to show that $\mathcal{L}(\varphi) = T$. Let $w \in \mathcal{L}(\varphi) = \mathcal{L}(\exists x\psi)$. We find a position in w and a factorization $w = uav$ such that $(uav, |u|) \models \psi$. Since $u \in [u]_{m-1,0}$ and $v \in [v]_{m-1,0}$, we have $uav \in T$, hence $\mathcal{L}(\varphi) \subseteq T$.

The converse follows by a twofold application of the congruence lemma (Lemma 10.2): Indeed, let $u' \in [u]_{m-1,0}$ and $v' \in [v]_{m-1,0}$ then

$$\begin{aligned}
[(u'a, |u'|)]_{m-1,1} &= [(u') \cdot (a, 0)]_{m-1,1} \\
&= [(u) \cdot (a, 0)]_{m-1,1} = [(ua, |u|)]_{m-1,1} \\
[(u'av', |u'|)]_{m-1,1} &= [(u'a, |u'|) \cdot (v')]_{m-1,1} \\
&= [(ua, |u|) \cdot (v)]_{m-1,1} = [(uav, |u|)]_{m-1,1}.
\end{aligned}$$

Therefore, $(uav, |u|) \models \psi$ implies $(u'av', |u'|) \models \psi$ and this implies $u'av' \models \exists x\psi$. Thus, $T \subseteq \mathcal{L}(\Psi)$. Q.E.D.

The congruence lemma yields an alternative way to show Proposition 9.1 (and hence the separation theorem, Theorem 9.2) too.

Proof of Proposition 9.1 based on Lemma 10.2. Let $\mathrm{qd}(\xi) = m$ for some $m \geq 0$. As in the proof of Proposition 10.3 define a language:

$$T = \bigcup_{(uav,|u|)\models\xi} [u]_{m,0} \cdot a \cdot [v]_{m,0}.$$

The union is finite and the classes $[u]_{m,0} \cap \Sigma^*$ and $[v]_{m,0}$ are first-order definable. First-order definable languages are star-free by Proposition 10.3. Thus, we can rewrite T as desired:

$$T = \bigcup_{i=1,\ldots,n} K_i \cdot a_i \cdot L_i.$$

Moreover, the proof of Proposition 10.3 has actually shown that $(uav, |u|) \models \xi$ if and only if $u \in K_i$, $a = a_i$ and $v \in L_i$ for some $1 \leq i \leq n$.

For convenience, let us repeat the argument. If $(uav, |u|) \models \xi$, then we find an index i such that $u \in K_i$, $a = a_i$, and $v \in L_i$. For the converse, let

$u' \in K_i$, $a' = a_i$, and $v' \in L_i$ for some i. We have to show $(u'a'v, |u'|) \models \xi$. By definition of T, we have $u' \in K_i = [u]_{m,0} \cap \Sigma^*$, $a' = a$, and $v' \in L_i = [v]_{m,0}$ for some $(uav, |u|) \models \xi$. The congruence lemma (Lemma 10.2) applied twice yields:

$$[(a'v', 0)]_{m,1} = [(a', 0) \cdot (v')]_{m,1} = [(a, 0) \cdot (v)]_{m,1} = [(av, 0)]_{m,1}.$$
$$[(u'a'v', |u'|)]_{m,1} = [(u') \cdot (a'v', 0)]_{m,1} = [(u) \cdot (av, 0)]_{m,1} = [(uav, |u|)]_{m,1}.$$

We deduce $(u'a'v, |u'|) \models \xi$. Q.E.D.

11 Counter-free and aperiodic Büchi automata

There is a standard way to introduce recognizable languages with finite automata. Since we deal with finite and infinite words we use Büchi automata with two acceptance conditions, one for finite words and the other for infinite words. A *Büchi automaton* is given as a tuple

$$\mathcal{A} = (Q, \Sigma, \delta, I, F, R),$$

where Q is a finite set of *states* and δ is a relation:

$$\delta \subseteq Q \times \Sigma \times Q.$$

The set $I \subseteq Q$ is called the set of *initial states*, the sets $F, R \subseteq Q$ consist of *final* and *repeated* states respectively.

If δ is the graph of a partially defined function from $Q \times \Sigma$ to Q and if in addition $|I| \leq 1$, then the automaton is called *deterministic*.

A *path* means in this section a finite or infinite sequence

$$\pi = p_0, a_0, p_1, a_1, p_2, a_2, \ldots$$

such that $(p_i, a_i, p_{i+1}) \in \delta$ for all $i \geq 0$. We say that the path is *accepting*, if it starts in an initial state $p_0 \in I$ and either it is finite and ends in a final state from F or it is infinite and visits infinitely many repeated states from R. The label of the above path π is the word $u = a_0a_1a_2 \cdots \in \Sigma^\infty$. The language *accepted* by $\mathcal{A}$ is denoted by $\mathcal{L}(\mathcal{A})$ and is defined as the set of words which appear as label of an accepting path. We have $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^\infty$. Languages of the form $\mathcal{L}(\mathcal{A})$ are called *regular* or *regular ω-languages*, if $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^\omega$

McNaughton and Papert have introduced the classical notion of a *counter-free* deterministic finite automaton, [19]. They showed that counter-freeness captures star-freeness (hence aperiodicity) for languages over finite words. Our aim is to give a natural notion of counter-freeness for non deterministic (Büchi) automata such that a language $L \subseteq \Sigma^\infty$ is aperiodic if and only if it can be accepted by a counter-free Büchi automaton. To the best of our knowledge, all previous extensions to infinite words used deterministic automata.

If $p, q \in Q$ are states of $\mathcal{A}$, then we let $L_{p,q}$ be the set of labels of finite paths from p to q.

Definition 11.1. A Büchi automaton $\mathcal{A} = (Q, \Sigma, \delta, I, F, R)$ is called *counter-free*, if $u^m \in L_{p,p}$ implies $u \in L_{p,p}$ for all states $p \in Q$, words $u \in \Sigma^*$, and $m \geq 1$.

Note that the definition is taking only the underlying transition relation δ into account, but does not depend on the sets I, F, or R. For deterministic automata counter-freeness coincides with the standard notion as introduced in [19]. We start with the classical result of [19] on finite words.

Lemma 11.2. Let $L \subseteq \Sigma^*$ be a language of finite words recognized by a morphism h from Σ^* to some finite aperiodic monoid M. Then the minimal deterministic automaton recognizing L is counter-free.

Proof. The states of the minimal deterministic automaton recognizing L can be written as

$$L(u) = u^{-1}L = \{w \in \Sigma^* \mid uw \in L\}$$

with $u \in \Sigma^*$ and all transitions have the form $(L(u), a, L(ua))$. Assume that $L(uv^m) = L(u)$ for some $m \geq 1$. Then we can take m as large as we wish and since M is aperiodic we may assume that $x^{m+1} = x^m$ for all $x \in M$. Since h recognizes L, we deduce that $uv^m w \in L$ if and only if $uv^{m+1}w \in L$ for all $w \in \Sigma^*$, i.e., $L(uv^m) = L(uv^{m+1})$. Using $L(uv^m) = L(u)$ we obtain,

$$L(u) = L(uv^m) = L(uv^{m+1}) = L((uv^m)v) = L(uv).$$

Hence, the automaton is counter-free. Q.E.D.

Lemma 11.3. Let $L \subseteq \Sigma^*$ and $L' \subseteq \Sigma^\infty$ be accepted by counter-free automata. Then $L \cdot L'$ can be accepted by some counter-free automaton.

Proof. Trivial, just consider a usual construction showing that regular languages are closed under concatenation. Essentially, the new automaton is the disjoint union of the two automata with additional transitions allowing to switch from the first one to the second one. Therefore, a loop in the new automaton is either a loop in the first one or a loop in the second one. Thus, we have no *new* loops and hence the result. Q.E.D.

Proposition 11.4. Let $L \subseteq \Sigma^\infty$ be recognized by a morphism $h : \Sigma^* \to M$ to some finite aperiodic monoid M. Then we find a counter-free Büchi automaton $\mathcal{A}$ with $L = \mathcal{L}(\mathcal{A})$.

Proof. By Remark 5.3 we can write L as a finite union of languages of type UV^ω, where U and V are aperiodic languages of finite words and where moreover $V = h^{-1}(e)$ for some idempotent $e \in M$. By a simple construction on monoids we may actually assume that $h^{-1}(1) = \{\varepsilon\}$ and then in turn that $e \neq 1$. Hence without restriction we have $V \subseteq \Sigma^+$. The *union* of two counter-free Büchi automata is counter-free and recognizes the union of the accepted languages. Therefore we content to construct a counter-free Büchi automaton for the language UV^ω. By Lemmata 11.2 and 11.3 it is enough to find a counter-free automaton for V^ω. The trick is that V^ω can be accepted by some deterministic Büchi automaton. Define the *witness* W by

$$W = V \cdot (V \setminus V\Sigma^+).$$

The language W is aperiodic. By Lemma 11.2, its minimal automaton $\mathcal{A} = (Q, \Sigma, \delta, I, F, \varnothing)$ is counter-free. View this automaton as a deterministic Büchi automaton $\mathcal{A}' = (Q, \Sigma, \delta, I, \varnothing, F)$ where final states are now repeated states. (It is also counter-free according to Definition 11.1, because it is deterministic.)

The automaton $\mathcal{A}'$ accepts those infinite strings where infinitely many prefixes are in W. We want to show that this coincides with V^ω. Clearly, $w \in V^\omega$ implies that w has infinitely many prefixes in W. We show that the converse holds, too. Let $w \in \Sigma^\omega$ and w_i be a list of infinitely many prefixes in W. For each w_i choose some factorization $w_i = u_i v_i$ with $u_i \in V$ and $v_i \in V \setminus V\Sigma^+$. Note there might be several such factorizations. However, if $w_i \neq w_j$, then we cannot have $u_i = u_j$, because otherwise v_i is a strict prefix of v_j or vice versa. Thus, we find infinitely many u_i and by switching to some infinite subsequence we may assume

$$u_1 < u_1 v_1 < u_2 < u_2 v_2 < u_3 < u_3 v_3 \cdots$$

where $\leq$ means the prefix relation. For all i we can write $u_{i+1} = u_i v_i v_i'$. We have

$$e = h(u_{i+1}) = h(u_i v_i v_i') = e \cdot e \cdot h(v_i') = e \cdot h(v_i') = h(v_i) \cdot h(v_i') = h(v_i v_i').$$

Hence

$$w = u_1 (v_1 v_1')(v_2 v_2')(v_3 v_3') \cdots \in V^\omega.$$

Therefore, V^ω is accepted by the counter-free Büchi automaton $\mathcal{A}'$. Q.E.D.

To prove that conversely, a language accepted by a counter-free Büchi automaton is aperiodic, we shall use a weaker notion. The following definition coincides with the one given in [16, Definition 3.1] for non-deterministic finite automata in the context of finite transducers.

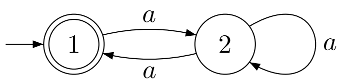

FIGURE 1. The non-deterministic Büchi automaton $\mathcal{A}_1$

Definition 11.5. A Büchi automaton $\mathcal{A} = (Q, \Sigma, \delta, I, F, R)$ is called *aperiodic*, if for some $m \geq 1$ we have:

$$u^m \in L_{p,q} \iff u^{m+1} \in L_{p,q}$$

for all states $p, q \in Q$ and words $u \in \Sigma^*$.

Lemma 11.6. Let $\mathcal{A}$ be a Büchi automaton.

1. If $\mathcal{A}$ is counter-free, then $\mathcal{A}$ is aperiodic.

2. If $\mathcal{A}$ is deterministic and aperiodic, then $\mathcal{A}$ is counter-free.

Proof. 1. Let $u^{m+1} \in L_{p,q}$. If m is large enough, we find $m+1 = k_1 + \ell + k_2$ with $\ell \geq 2$ and a state s such that $u^{k_1} \in L_{p,s}$, $u^\ell \in L_{s,s}$, and $u^{k_2} \in L_{s,q}$. Since the automaton is counter-free, we obtain $u \in L_{s,s}$ and therefore $u^m \in L_{p,q}$. Similarly, we can show that $u^m \in L_{p,q}$ implies $u^{m+1} \in L_{p,q}$.

2. Let $u^m \in L_{p,p}$ for some $m \geq 1$. Then $u^m \in L_{p,p}$ for m as large as we wish. Since the automaton is aperiodic we have $u^m, u^{m+1} \in L_{p,p}$ for some m large enough. Since the automaton is deterministic, we deduce that $u \in L_{p,p}$, too. Q.E.D.

Remark 11.7. Consider the non-deterministic Büchi automaton $\mathcal{A}_1$ of Figure 1 which accepts $\{a^\omega\}$. The automaton $\mathcal{A}_1$ is aperiodic, but not counter-free.

The *transformation monoid* $T(\mathcal{A})$ of $\mathcal{A}$ is realized as a submonoid of Boolean matrices. More precisely, let $\mathcal{A}$ have n states. We consider the monoid $\mathbb{B}^{n \times n}$ of $n \times n$ matrices over the finite commutative semiring $\mathbb{B} = \{0, 1\}$ with max as addition and the natural multiplication as product. For every word u we define a matrix $t(u) \in \mathbb{B}^{n \times n}$ by:

$$t(u)[p,q] = 1 \iff u \in L_{p,q}.$$

The mapping $t : \Sigma^* \to \mathbb{B}^{n \times n}$ is a monoid morphism, because $t(\varepsilon)$ is the identity matrix and we have for all $u, v \in \Sigma^*$:

$$t(u \cdot v)[p,q] = \sum_{r \in Q} t(u)[p,r] \cdot t(v)[r,q].$$

The transition monoid of $\mathcal{A}$ is $T(\mathcal{A}) = t(\Sigma^*) \subseteq \mathbb{B}^{n \times n}$.

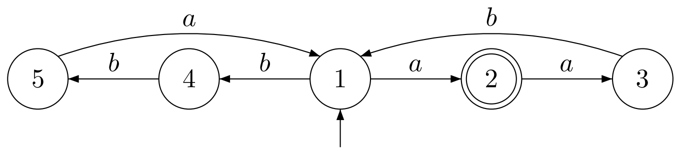

FIGURE 2. The deterministic and counter-free Büchi automaton $\mathcal{A}_2$

Remark 11.8. In terms of the transition monoid, Definition 11.5 says that a Büchi automaton $\mathcal{A}$ is aperiodic if and only if the monoid $T(\mathcal{A})$ is aperiodic.

The problem is that the morphism t to the transition monoid of $\mathcal{A}$ does not recognize $\mathcal{L}(\mathcal{A})$, in general. Indeed consider the deterministic automaton $\mathcal{A}_2$ on Figure 2 where the only repeated state is 2. The automaton accepts the language

$$L = \{w \in \{aab, bba\}^\omega \mid \text{the factor } aa \text{ appears infinitely often}\}.$$

Consider the matrix $t(aab)$ for which all entries are 0 except $t(aab)[1,1] = 1$. We have $t(aab) = t(bba)$, but $(aab)^\omega \in L$ and $(bba)^\omega \notin L$. Thus t does not recognize L.

It is therefore somewhat surprising that aperiodicity of $T(\mathcal{A})$ implies that $\mathcal{L}(\mathcal{A})$ is an aperiodic language. This is proved in Proposition 11.11, below.

We still need another concept. In Büchi's original proof that regular ω-languages are closed under complementation (see [3]) he used a finer congruence than given by the morphism t. To reflect this, we switch from the Boolean semiring $\mathbb{B}$ to the finite commutative semiring $K = \{0, 1, \infty\}$. The semiring structure of K is given by $x + y = \max\{x, y\}$ and the natural multiplication with the convention $0 \cdot \infty = 0$.

In order to take repeated states into account we let $R_{p,q} \subseteq L_{p,q}$ be the set of labels of *nonempty* and finite paths from p to q, *which use a repeated state at least once*. For every word u we define a matrix $h(u) \in K^{n \times n}$ by:

$$h(u)[p,q] = \begin{cases} 0 & \text{if } u \notin L_{p,q}, \\ 1 & \text{if } u \in L_{p,q} \setminus R_{p,q}, \\ \infty & \text{if } u \in R_{p,q}. \end{cases}$$

For the Büchi automaton $\mathcal{A}_2$ in Figure 2 we have $h(aab)[1,1] = \infty$, whereas $h(bba)[1,1] = 1$. For all other entries we have $h(aab)[p,q] = h(bba)[p,q] = 0$.

Note that $h(\varepsilon)$ is the identity matrix. In the semiring $K^{n \times n}$ we have as usual:

$$h(u \cdot v)[p,q] = \sum_{r \in Q} h(u)[p,r] \cdot h(v)[r,q].$$

Hence, $h : \Sigma^* \to K^{n\times n}$ is a monoid morphism and we can check easily that h recognizes $\mathcal{L}(\mathcal{A})$. The submonoid $BT(\mathcal{A}) = h(\Sigma^*) \subseteq K^{n\times n}$ is called either *Büchi's transition monoid* of $\mathcal{A}$ or the *ω-transition monoid* of $\mathcal{A}$. We obtain Büchi's result [3]:

Proposition 11.9. For every Büchi automaton $\mathcal{A}$ the morphism $h : \Sigma^* \to BT(\mathcal{A})$ onto the ω-transition monoid of $\mathcal{A}$ recognizes $\mathcal{L}(\mathcal{A})$.

Corollary 11.10. A language in $L \subseteq \Sigma^\infty$ can be accepted by some Büchi automaton if and only if it can be recognized by some morphism to some finite monoid.

Proof. Proposition 11.9 gives one direction. Conversely, assume that L is recognized by a morphism h from Σ^* to some finite monoid M. By Remark 5.3, L is a finite union of languages of type UV^ω, where $U, V \subseteq \Sigma^*$ are recognized by h. These sets are accepted by finite deterministic automata with M as set of states. Standard constructions on Büchi automata for union, concatenation, and ω-power yield the result. Q.E.D.

It also follows that regular ω-languages are closed under complementation, since recognizable languages are closed under complementation by definition (as they are unions of equivalence classes).

Proposition 11.11. Let $L \subseteq \Sigma^\infty$ a language. The following are equivalent.

1. There is a counter-free Büchi automaton $\mathcal{A}$ with $L = \mathcal{L}(\mathcal{A})$.
2. There is an aperiodic Büchi automaton $\mathcal{A}$ with $L = \mathcal{L}(\mathcal{A})$.
3. The language L is aperiodic.

Proof. $1 \Rightarrow 2$: Trivial by Lemma 11.6.1.

$2 \Rightarrow 3$: Let $\mathcal{A}$ have n states and consider Büchi's morphism $h : \Sigma^* \to K^{n\times n}$ as above. We show that the submonoid $BT(\mathcal{A}) = h(\Sigma^*) \subseteq K^{n\times n}$ is aperiodic. More precisely, we show for all states p, q and words u that $h(u^{2m})[p, q] = h(u^{2m+1})[p, q]$ as soon as m large enough.

Since the automaton is aperiodic we find a suitable m with $u^m \in L_{p,q}$ if and only if $u^{m+1} \in L_{p,q}$ for all states p, q and words u. We immediately get

$$h(u^{2m})[p, q] \geq 1 \iff h(u^{2m+1})[p, q] \geq 1.$$

Assume now that $h(u^{2m})[p, q] = \infty$. Then for some r we have $h(u^{2m})[p, q] = h(u^m)[p, r] \cdot h(u^m)[r, q]$ and by symmetry we may assume $h(u^m)[r, q] = \infty$ and $h(u^m)[p, r] \neq 0$. This implies $h(u^{m+1})[p, r] \neq 0$ and therefore $h(u^{2m+1})[p, q] = h(u^{m+1})[p, r] \cdot h(u^m)[r, q] = \infty$. Similarly, we can show that $h(u^{2m+1})[p, q] = \infty$ implies $h(u^{2m})[p, q] = \infty$.

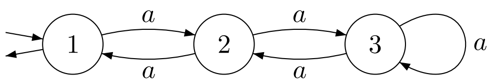

FIGURE 3. Aperiodicity does not imply counter-freeness for minimal size NFA.

Thus we have seen that $h(u^{2m})[p,q] = h(u^{2m+1})[p,q]$ for all $u \in \Sigma^*$ and all states p, q. This shows that L is recognized by some aperiodic monoid (of size at most 3^{n^2}).

$3 \Rightarrow 1$: This is the contents of Proposition 11.4. Q.E.D.

The automaton $\mathcal{A}_2$ above is counter-free, and this notion does not depend on final or repeated states. In particular, the languages $\{aab, bba\}^\omega$ and $\{aab, bba\}^*$ are further examples of aperiodic languages.

We conclude this section with several remarks concerning counter-freeness for Büchi automata.

Remark 11.12. If $L \subseteq \Sigma^\infty$ is aperiodic, then we actually find some Büchi automaton $\mathcal{A}$ with $L = \mathcal{L}(\mathcal{A})$, where for all states $p \in Q$, words $u \in \Sigma^*$, and $m \geq 1$ the following two conditions hold:

1. If $u^m \in L_{p,p}$, then $u \in L_{p,p}$.
2. If $u^m \in R_{p,p}$, then $u \in R_{p,p}$.

This is true, because all crucial constructions in the proof of Proposition 11.4 were done for deterministic automata. If an automaton is deterministic, then Condition 1 implies Condition 2, because if $u^m \in R_{p,p}$ and $u \in L_{p,p}$, then the path labeled by u^m from p to p visits the same states as the path labeled by u from p to p. For non-deterministic automata the second condition is a further restriction of counter-free automata.

Remark 11.13. For finite words, counter-freeness of the minimal automaton of a language $L \subseteq \Sigma^*$ characterizes aperiodicity of L. There is no canonical minimal Büchi automaton for languages of infinite words, but we may ask whether counter-freeness of a non-deterministic automaton of minimal size also characterizes aperiodicity. The answer is negative. Indeed, consider the language $L = \left\{\varepsilon,\, a^2\right\} \cup a^4a^*$ which is aperiodic and accepted by the 3-state automaton in Figure 3. This automaton is not counter-free since $a^2 \in L_{1,1}$ but $a \notin L_{1,1}$. We can check that L cannot be accepted by a 2-state automaton.

Remark 11.14. Let $\mathcal{A} = (Q, \Sigma, \delta, I)$ be a non-deterministic automaton and let $\mathcal{B} = (2^Q, \Sigma, \delta_{\mathcal{B}}, \{I\})$ be its (deterministic) subset automaton. Note

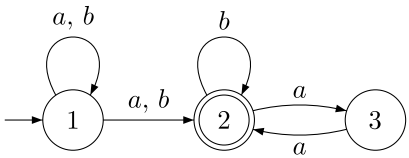

FIGURE 4. The Büchi automaton $\mathcal{A}$ accepting $\Sigma^+\{a^2, b\}^\omega$.

that, in this definition, we do not restrict to the accessible subsets from I. First, we prove that if $\mathcal{A}$ is counter-free, then so is $\mathcal{B}$. Assume that $\delta(X, u^m) = X$ for some $X \subseteq Q$, $u \in \Sigma^+$ and $m > 0$. Then, for each $p \in X$ we find some $p' \in X$ with $p \in \delta(p', u^m)$. Iterating these backward paths, we find $q \in X$ such that

$$q \xrightarrow{u^{jm}} q \xrightarrow{u^{km}} p$$

Since $\mathcal{A}$ is counter-free, it follows $q \xrightarrow{u} q$. Hence, $p \in \delta(X, u^{1+km}) = \delta(X, u)$. We have proved $X \subseteq \delta(X, u)$. It follows by induction that $\delta(X, u) \subseteq \delta(X, u^m) = X$. Therefore, $\mathcal{B}$ is counter-free.

Next, we show that if $\mathcal{B}$ is counter-free then $\mathcal{A}$ is aperiodic. Let $x \in T(\mathcal{A})$ be in the transition monoid of $\mathcal{A}$: $x = t(u)$ for some $u \in \Sigma^*$. We have $x^m = x^{m+k}$ for some $m, k > 0$. Let $X = x^m(Q) = \delta(Q, u^m)$. Since $x^m = x^{m+k}$ we have $\delta(X, u^k) = X$ and we deduce $\delta(X, u) = X$ since $\mathcal{B}$ is counter-free. Therefore, $x^m = x^{m+1}$ and we have shown that $T(\mathcal{A})$ is aperiodic.

Therefore, counter-freeness of the *full* subset automaton is another sufficient condition for aperiodicity. But, for this to hold over infinite words, it is important not to restrict to the subsets accessible from I. Indeed, let $\Sigma = \{a, b\}$ with $a \neq b$ and consider the language:

$$L = \Sigma^+\{a^2, b\}^\omega.$$

The non-deterministic 3-state Büchi automaton $\mathcal{A}$ in Figure 4 accepts L with $I = \{1\}$, $F = \emptyset$ and $R = \{2\}$ (an easy exercise shows that there is no deterministic Büchi automaton accepting L). The subset automaton restricted to the subsets reachable from $\{1\}$ is depicted in Figure 5. This automaton is counter-free, but L is not aperiodic.

12 Deciding aperiodicity in polynomial space

This section is devoted to a construction which shows that aperiodicity is decidable (in polynomial space) for recognizable languages. Thus, all properties mentioned in Theorem 1.1 are decidable for a regular ∞-languages.

Our aim is an optimal algorithm in a complexity theoretical meaning, and the best we can do is to find a polynomial space bounded algorithm.

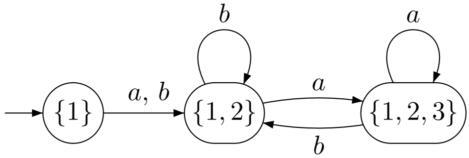

FIGURE 5. The subset automaton $\mathcal{B}$ of $\mathcal{A}$ restricted to reachable states.

This is indeed optimal, because PSPACE-hardness is known by [4]. It should be noted that our PSPACE-upper bound is not a formal consequence of [29] or any other reference we are aware of, because [29] deals only with deterministic automata over finite words. Moreover, our approach is not based on the syntactic congruence of Arnold [1]. Instead we start with any recognizing morphism and we consider its maximal aperiodic quotient. We check whether this monoid still recognizes the same language. This is possible in polynomial space, as we shall demonstrate below. We need an algebraic construction first.

Proposition 12.1. Let $h_1 : \Sigma^* \to M_1$ be a surjective morphism onto a finite monoid M_1 which recognizes L and let $m \geq |M_1|$. Let M_1' be the quotient of the monoid M_1 by the congruence generated by $\{x^m = x^{m+1} \mid x \in M_1\}$ and let $h_1' : \Sigma^* \to M_1'$ be the canonical morphism induced by h_1. Then L is aperiodic if and only if h_1' recognizes L.

Proof. First, If h_1' recognizes L, then L is aperiodic since M_1' is aperiodic by construction.

Conversely, if L is aperiodic, then there is some surjective morphism $h_2 : \Sigma^* \to M_2$ which recognizes L and where M_2 is aperiodic. We first show that L is also recognized by a *quotient* monoid M of both M_1 and M_2. This means that M is a homomorphic image of M_1 as well as of M_2.

$$\begin{tikzcd} & \Sigma^* \arrow[dl, "h_1"'] \arrow[dd, "h"] \arrow[dr, "h_2"] & \\ M_1 \arrow[dr, "\bar{h}_1"'] & & M_2 \arrow[dl, "\bar{h}_2"] \\ & M & \end{tikzcd}$$

We define the relation $H \subseteq \Sigma^* \times \Sigma^*$ by:

$$H = \{(u, v) \mid h_1(u) = h_1(v) \vee h_2(u) = h_2(v)\} .$$

The transitive closure H^+ of H is an equivalence relation, and easily seen to be a congruence. Thus, we can define the quotient monoid M of Σ^* by H^+. We have a canonical morphism $h : \Sigma^* \to M$ and $|M| \leq \min\{|M_1|, |M_2|\}$.

Since $h_i(u) = h_i(v)$ implies $h(u) = h(v)$ for all $u, v \in \Sigma^*$, the morphism h factorizes through M_1 and M_2 as shown in the diagram above: $h = \bar{h}_i \circ h_i$ for $i = 1, 2$.

We show that h recognizes L, too. First, we note that $H^+ = H^\ell$ where $\ell = \min\{|M_1|, |M_2|\}$. Indeed, if $u_0 \; H \; u_1 \cdots H \; u_k$ with $k \geq |M_1|$ then we find $0 \leq i < j \leq k$ with $h_1(u_i) = h_1(u_j)$ and we obtain $(u_0, u_k) \in H^{k-(j-i)}$. Now, consider some $u = \prod_{0 \leq i < n} u_i$ and $v = \prod_{0 \leq i < n} v_i$ with u_i, $v_i \in \Sigma^+$ such that $(u_i, v_i) \in H$ for all $0 \leq i < n$. Since $H^+ = H^\ell$ it is enough to see that $u \in L$ implies $v \in L$. Now, for all $0 \leq i < n$ there is $w_i \in \{u_i, v_i\}$ with $h_1(u_i) = h_1(w_i)$ and $h_2(w_i) = h_2(v_i)$. Since h_1 recognizes L, we have $u \in L$ implies $\prod_{0 \leq i < n} w_i \in L$, and this implies $v \in L$ since h_2 recognizes L.

The monoid M as constructed above is aperiodic, because it is a quotient monoid of M_2. But $|M| \leq |M_1| \leq m$, hence $x^m = x^{m+1}$ for all $x \in M$. By definition, M_1' is the quotient of the monoid M_1 by the congruence generated by $\{x^m = x^{m+1} \mid x \in M_1\}$. Since M satisfies all equations $x^m = x^{m+1}$, the morphism $\bar{h}_1 : M_1 \to M$ factorizes through M_1': $\bar{h}_1 = \bar{h}_1' \circ g$ where g is the canonical morphism from M_1 to M_1'.

$$\begin{tikzcd}
 & & \Sigma^* \arrow[dll, bend right, "h_1'"'] \arrow[dl, "h_1"'] \arrow[dd, "h"] \arrow[dr, "h_2"] & \\
M_1' \arrow[drr, bend right, "\bar{h}_1'"'] & M_1 \arrow[l, "g"'] \arrow[dr, "\bar{h}_1"'] & & M_2 \arrow[dl, "\bar{h}_2"] \\
 & & M &
\end{tikzcd}$$

By definition, $h_1' = g \circ h_1$ and we deduce that $h = \bar{h}_1' \circ h_1'$. Hence, $h_1'(u) = h_1'(v)$ implies $h(u) = h(v)$ for all $u, v \in \Sigma^*$. Since h recognizes L, this implies that h_1' recognizes L, too. Q.E.D.

From Proposition 12.1, we can derive easily a pure decidability result. Indeed, if we start with a language L recognized by a Büchi automaton $\mathcal{A}$ with n states, we know that L is aperiodic if and only if it is recognized by some aperiodic monoid with at most 3^{n^2} elements. Hence, we can guess a recognizing morphism h from Σ^* to an aperiodic monoid M of size at most 3^{n^2}, guess a set P of linked pairs, compute a Büchi automaton $\mathcal{A}'$ recognizing $L' = \bigcup_{(s,e) \in P} h^{-1}(s) h^{-1}(e)^\omega$ using Corollary 11.10, and finally check whether $L = L'$ starting from $\mathcal{A}, \mathcal{A}'$ and using complementations, intersections and an emptiness tests.

The complexity of this algorithm is not optimal. In order to derive a PSPACE algorithm, we first establish the following characterization.

Proposition 12.2. Let $h : \Sigma^* \to M$ be a surjective morphism that recognizes $L \subseteq \Sigma^\infty$. Let $g : M \to M'$ be a surjective morphism. Then, $h' = g \circ h$

recognizes L if and only if for all $s, e, s', e' \in M$ such that $g(s) = g(s')$ and $g(e) = g(e')$ we have

$$h^{-1}(s)h^{-1}(e)^\omega \subseteq L \quad \Longleftrightarrow \quad h^{-1}(s')h^{-1}(e')^\omega \subseteq L$$

Intuitively, this means that the set of linked pairs associated with L is saturated by g.

Proof. Assume first that h' recognizes L. Let $s, e, s', e' \in M$ with $g(s) = g(s')$ and $g(e) = g(e')$ and assume that $h^{-1}(s)h^{-1}(e)^\omega \subseteq L$. Since h is surjective, we find $u, v, u', v' \in \Sigma^*$ such that $h(u) = s$, $h(v) = e$, $h(u') = s'$ and $h(v') = e'$. From the hypothesis, we get $h'(u) = h'(u')$ and $h'(v) = h'(v')$. Now, $uv^\omega \in h^{-1}(s)h^{-1}(e)^\omega \subseteq L$. Since h' recognizes L we deduce $u'v'^\omega \in h^{-1}(s')h^{-1}(e')^\omega \cap L$. Since h recognizes L we obtain $h^{-1}(s')h^{-1}(e')^\omega \subseteq L$.

Conversely, let $u = u_0u_1u_2 \cdots \in L$ and $v = v_0v_1v_2 \cdots$ with $u_i, v_i \in \Sigma^+$ and $h'(u_i) = h'(v_i)$ for all $i \geq 0$. We have to show that $v \in L$. Grouping factors u_i and v_i using Lemma 5.2, we find new factorizations $u = u'_0u'_1u'_2 \cdots$ and $v = v'_0v'_1v'_2 \cdots$ which satisfy in addition $h(u'_i) = e$ and $h(v'_i) = e'$ for all $i > 0$. Let $s = h(u'_0)$ and $s' = h(v'_0)$. We have $g(s) = h'(u'_0) = h'(v'_0) = g(s')$ and similarly $g(e) = g(e')$. Now, $u \in h^{-1}(s)h^{-1}(e)^\omega \cap L \neq \varnothing$ and since h recognizes L we get $h^{-1}(s)h^{-1}(e)^\omega \subseteq L$. We deduce that $v \in h^{-1}(s')h^{-1}(e')^\omega \subseteq L$. Q.E.D.

Proposition 12.3. We can decide in PSPACE whether the accepted language $L \subseteq \Sigma^\infty$ of a given Büchi automaton $\mathcal{A}$ is aperiodic.

Proof. Let $h : \Sigma^* \to K^{n\times n}$ be Büchi's morphism and let $M = BT(\mathcal{A}) = h(\Sigma^*)$ so that $h : \Sigma^* \to M$ is surjective and recognizes $L = \mathcal{L}(\mathcal{A})$. Let g be the canonical morphism from M to the quotient M' of M by the congruence generated by $\{x^m = x^{m+1} \mid x \in M\}$ with $m = 3^{n^2} \geq |M|$.

It is enough to design a non-deterministic polynomial space algorithm which finds out that L is not aperiodic. By Propositions 12.1 and 12.2, we have to check whether there exist four elements $s, e, s', e' \in M$ such that $g(s) = g(s')$, $g(e) = g(e')$, $h^{-1}(s)h^{-1}(e)^\omega \subseteq L$ and $h^{-1}(s')h^{-1}(e')^\omega \nsubseteq L$. By definition of M', this is equivalent to the existence of $u, v, w, x, y, z \in M$ and $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 \in \{0, 1\}$ with $h^{-1}(s)h^{-1}(e)^\omega \subseteq L$ and $h^{-1}(s')h^{-1}(e')^\omega \nsubseteq L$ where $s = uv^{m+\varepsilon_1}w$, $e = xy^{m+\varepsilon_2}z$, $s' = uv^{m+\varepsilon_3}w$ and $e' = xy^{m+\varepsilon_4}z$.

We have $h^{-1}(s)h^{-1}(e)^\omega \subseteq L$ if and only if there are $p \in I$, $q \in Q$ such that $(se^k)[p, q] \geq 1$ and $e^\ell[q, q] = \infty$ for some $k, \ell \leq n$. Indeed, if the right hand side holds then we find an accepting run in $\mathcal{A}$ for some word $u \in h^{-1}(s)h^{-1}(e)^\omega$. Hence, we have $h^{-1}(s)h^{-1}(e)^\omega \cap L \neq \varnothing$ and since h recognizes L we deduce that $h^{-1}(s)h^{-1}(e)^\omega \subseteq L$. Conversely, let $u = u_0u_1u_2 \ldots \in L$ with $h(u_0) = s$ and $h(u_i) = e$ for $i > 0$. Consider an accepting run for u:

$$p \xrightarrow{u_0} q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} q_2 \cdots$$

Since this run is accepting, we find k such that a repeated state is visited in the path $q_k \xrightarrow{u_{k+1}} q_{k+1}$ and $q_k = q_{k+\ell}$ for some $\ell > 0$. Removing loops we may assume that $k < n$ and $\ell \leq n$. We get the result with $q = q_k$.

Therefore, we have the following algorithm.

1. Guess six matrices $u, v, w, x, y, z \in M$ and guess four values ε_1, ε_2, ε_3, ε_4 in $\{0, 1\}$ (with, if one wishes, $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 = 1$).

2. Compute $s = uv^{m+\varepsilon_1}w$, $e = xy^{m+\varepsilon_2}z$, $s' = uv^{m+\varepsilon_3}w$ and $e' = xy^{m+\varepsilon_4}z$.

3. Check that $h^{-1}(s)h^{-1}(e)^\omega \subseteq L$ and $h^{-1}(s')h^{-1}(e')^\omega \not\subseteq L$.

Computing x^m with $x \in M$ can be done with $\mathcal{O}(\log m) = \mathcal{O}(n^2)$ products of $n \times n$ matrices. Hence, steps 2 and 3 can be done in deterministic polynomial time, once the matrices $u, v, w, x, y, z \in M$ are known. It remains to explain how to guess in PSPACE an element $x \in M = h(\Sigma^*)$. As a matter of fact, it is here[2] where we need the full computational power of PSPACE. To do this, we guess a sequence $a_1, a_2, \ldots a_i \in \Sigma$ letter after letters and simultaneously we compute the sequence

$$h(a_1), h(a_1a_2), \ldots, h(a_1a_2 \cdots a_i).$$

We remember only the last element $h(a_1a_2 \cdots a_j)$ before we guess the next letter a_{j+1} and compute the next matrix. We stop with some $i \leq 3^{n^2}$ and we let $x = h(a_1a_2 \cdots a_i)$ be the last computed matrix. Q.E.D.

In some cases it is extremely easy to see that a language is not aperiodic. For example, $(aa)^*$ is recognized by the cyclic group $\mathbb{Z}/2\mathbb{Z}$ of two elements. Every aperiodic quotient of a group is trivial. But the trivial monoid cannot recognize $(aa)^*$.

13 Very weak alternating automata

For a finite set Q we mean by $\mathbb{B}^+(Q)$ the *non-empty positive* Boolean combinations of elements of Q, e.g., $p \wedge (q \vee r)$. We write $P \models \xi$, if a subset $P \subseteq Q$ *satisfies* a formula $\xi \in \mathbb{B}^+(Q)$. By definition, $P \models p$ if and only if $p \in P$. As a consequence, we have for instance $\{p, r\} \models p \wedge (q \vee r)$ and $\{p, r, s\} \models p \wedge (q \vee r)$, but $\{q, r\} \not\models p \wedge (q \vee r)$. Note that $\varnothing \not\models \xi$ since we use non-empty positive Boolean combinations, only. The *satisfiability relation* is monotone. This means, if $P \subseteq P'$ and $P \models \xi$, then $P' \models \xi$, too.

An *alternating* automaton is a tuple $\mathcal{A} = (Q, \Sigma, \delta, I, F, R)$ where

[2] For the interested reader, the test $x \in h(\Sigma^*)$ is PSPACE-hard, in general [10, Problem MS5]. This problem is closely related to the intersection problem of regular languages, where the PSPACE–hardness is due to Kozen [14].

- Q is a finite set of states,
- Σ is a finite alphabet,
- $I \in \mathbb{B}^+(Q)$ is the (alternating) initial condition,
- $\delta : Q \times \Sigma \to \mathbb{B}^+(Q)$ is the (alternating) transition function (for instance, $\delta(p, a) = (p \wedge (q \vee r)) \vee (q \wedge s)$ is a possible transition),
- $F \subseteq Q$ is the subset of *final* states,
- and $R \subseteq Q$ is the subset of *repeated* states.

A run of $\mathcal{A}$ over some word $w = a_0 a_1 a_2 \cdots \in \Sigma^\infty$ is a Q-labeled forest (V, E, ρ) with $E \subseteq V \times V$ and $\rho : V \to Q$ such that

- the set of roots $\{z \mid E^{-1}(z) = \varnothing\}$ satisfy the initial condition:
$$\rho(\{z \mid E^{-1}(z) = \varnothing\}) \models I,$$
- each node satisfies the transition relation: for all $x \in V$ of depth n, i.e., such that $x \in E^n(z)$ where $z \in V$ is the root ancestor of x, we have $n \leq |w|$ and if $n < |w|$ then x is not a leaf and $\rho(E(x)) \models \delta(\rho(x), a_n)$.

If the word w is finite then the run is *accepting*, if each leaf x satisfies $\rho(x) \in F$. If the word w is infinite then the run is *accepting*, if every infinite branch visits R infinitely often. Since we use *nonempty* boolean combinations of states for the transition function, if w is finite then each leaf must be of depth $|w|$ and if w is infinite then each maximal branch must be infinite. We denote by $\mathcal{L}(\mathcal{A})$ the set of words $w \in \Sigma^\infty$ for which there is some accepting run of $\mathcal{A}$.

An alternating automaton $\mathcal{A}$ is called *very weak*, if there is a partial order relation $\leq$ on Q such that the transition function is *non-increasing*, i.e., for each $p, q \in Q$ and $a \in \Sigma$, if q occurs in $\delta(p, a)$ then $q \leq p$. Clearly, we can transform the partial ordering into a linear ordering without changing the condition of being very weak[3]. The next proposition shows that every first-order definable language can be accepted by some very weak automaton. The converse is shown in Proposition 13.3.

Proposition 13.1. For any formula $\xi \in \mathrm{LTL}_\Sigma(\mathsf{XU})$, we can construct a very weak alternating automaton $\mathcal{A}$ over Σ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\xi)$.

[3] In [17] a very weak automaton is therefore called a *linear alternating automaton*.

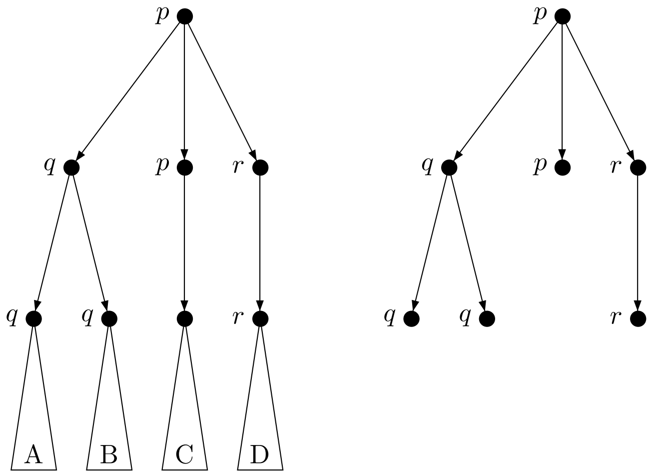

FIGURE 6. A run on the left and on the right the new tree with fresh leaves.

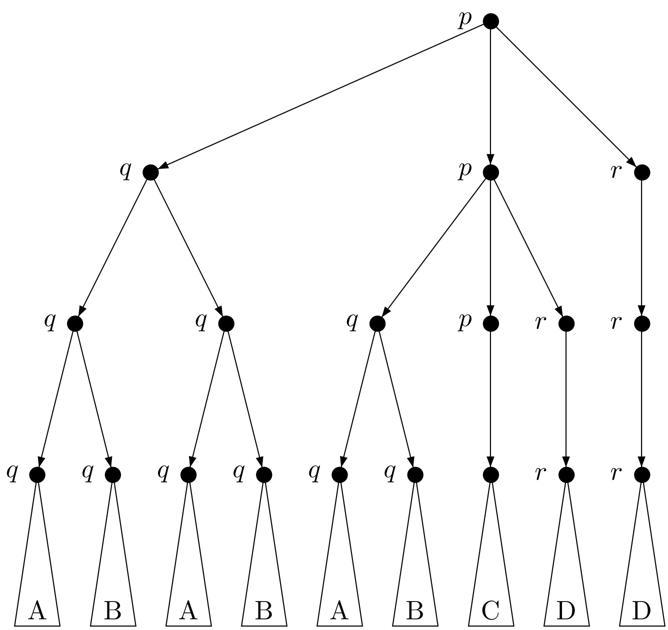

FIGURE 7. The new run with leaves on level $m + 1$.

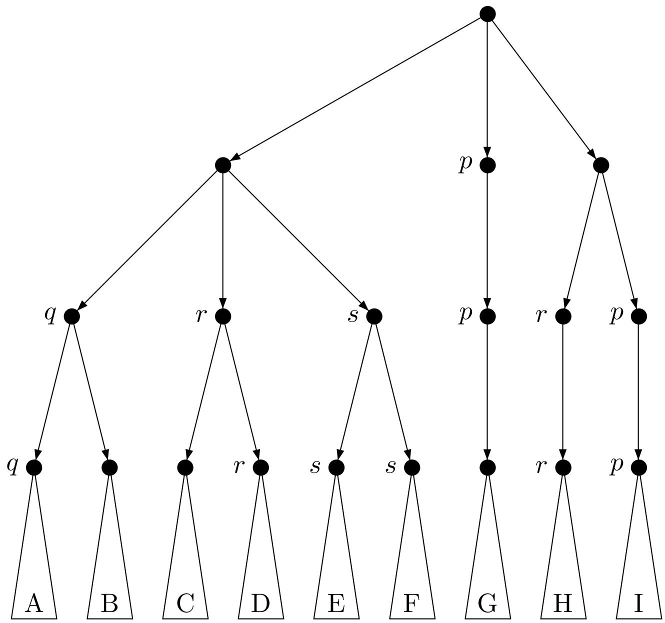

FIGURE 8. Another run with leaves on level $m+1$.

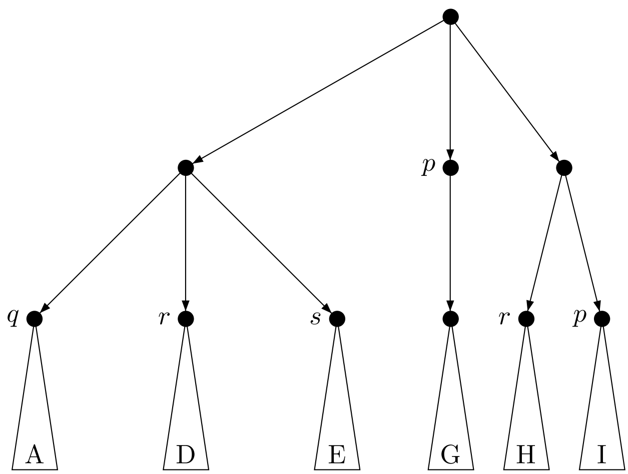

FIGURE 9. The new run with fewer labels at the leaves on level m.

Proof. First, we push the negations down to the constants. For this we need a dual for each operator. Clearly, $\vee$ and $\wedge$ are dual to each other. The dual of next-until is *next-release* which is defined by

$$\varphi \,\mathsf{XR}\, \psi = \neg(\neg\varphi \,\mathsf{XU}\, \neg\psi).$$

Hence, the semantics of *next-release* is given by

$$(\varphi \,\mathsf{XR}\, \psi)(x) = \forall z:\; x < z \rightarrow \psi(z) \vee \exists y:\; x < y < z \wedge \varphi(y).$$

Note that this is always true at the last position of a finite word: for all $v \in \Sigma^+$, we have $v, |v| - 1 \models \varphi \,\mathsf{XR}\, \psi$ for all formulae φ and ψ. One may also notice that

$$\varphi \,\mathsf{XR}\, \psi = \mathsf{X}\,\mathsf{G}\,\psi \vee (\psi \,\mathsf{XU}\, (\varphi \wedge \psi)).$$

All $\mathrm{LTL}_\Sigma(\mathsf{XU})$ formulae can be rewritten in *positive normal form* following the syntax

$$\varphi ::= \bot \mid \top \mid a \mid \neg a \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \,\mathsf{XU}\, \varphi \mid \varphi \,\mathsf{XR}\, \varphi.$$

Transforming a formula into positive normal form does not increase its size, and the number of temporal operators remains unchanged.

So, let ξ be an LTL formula in positive normal form. We define the alternating automaton $\mathcal{A} = (Q, \Sigma, \delta, I, F, R)$ as follows:

- The set Q of states consists of $\bot$, $\top$, END and the sub-formulae of ξ of the form a, $\neg a$, $\varphi \,\mathsf{XU}\, \psi$ or $\varphi \,\mathsf{XR}\, \psi$. Here, END means that we have reached the end of a finite word. Note that each sub-formula of ξ is in $\mathbb{B}^+(Q)$.
- The initial condition is $I = \xi$ itself.
- The transition function is defined by

$$\begin{aligned}
\delta(a, b) &= \begin{cases} \top & \text{if } b = a \\ \bot & \text{otherwise} \end{cases} \\
\delta(\neg a, b) &= \begin{cases} \bot & \text{if } b = a \\ \top & \text{otherwise} \end{cases} \\
\delta(\bot, a) &= \bot \\
\delta(\top, a) &= \top \\
\delta(\varphi \,\mathsf{XU}\, \psi, a) &= \psi \vee (\varphi \wedge \varphi \,\mathsf{XU}\, \psi) \\
\delta(\varphi \,\mathsf{XR}\, \psi, a) &= \mathrm{END} \vee (\psi \wedge (\varphi \vee \varphi \,\mathsf{XR}\, \psi)) \\
\delta(\mathrm{END}, a) &= \bot
\end{aligned}$$

- The set of final states is $F = \{\top, \text{END}\}$.
- The repeated states are the next-release sub-formulae of ξ together with $\top$.

Using the sub-formula partial ordering, we see that the alternating automaton $\mathcal{A}$ is very weak. We can also easily check that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\xi)$. Note that in a run over an infinite word, each infinite branch is ultimately labeled $\top$ or $\bot$ or with a XU or XR formula. A state $\varphi \mathbin{\mathsf{XU}} \psi$ is rejecting since if a branch is ultimately labeled with this state, this means that the eventuality ψ was not checked. On the other hand, $\varphi \mathbin{\mathsf{XR}} \psi$ is accepting since if a branch is ultimately labeled with this state then ψ is ultimately true for this word. Q.E.D.

As we see below, it is easy to transform a very weak alternating automaton into a Büchi automaton. We follow the construction of [11]. However, for this purpose it is convenient to generalize the acceptance conditions. A *generalized Büchi automaton* is a tuple

$$\mathcal{A} = (Q, \Sigma, \delta, I, F, T_1, \ldots, T_r)$$

where Q is a finite set of states, Σ is a finite alphabet,

$$\delta \subseteq Q \times \Sigma \times Q$$

is the non deterministic transition relation, $I \subseteq Q$ is the subset of initial states, $F \subseteq Q$ is the subset of final states, and $T_1, \ldots, T_r \subseteq \delta$ defines the accepting conditions. An infinite run $q_0, a_1, q_1, a_2, q_2, \cdots$ is accepted by $\mathcal{A}$ if for each $1 \leq i \leq r$, some transition in T_i occurs infinitely often in the run. Hence, the acceptance condition is generalized in two respects. First, it uses accepting *transitions* instead of accepting *states*. Second it allows a conjunction of Büchi's conditions. Obviously, each generalized Büchi automaton can be transformed into an equivalent *classical* Büchi automaton.

From a very weak alternating automaton, we construct an equivalent generalized Büchi automaton as follows. Let $\mathcal{A} = (Q, \Sigma, \delta, I, F, R)$ be a very weak alternating automaton. We define $\mathcal{A}' = (Q', \Sigma, \delta', I', F', (T_f)_{f \notin R})$ by

- $Q' = 2^Q$,
- $I' = \{P \subseteq Q \mid P \models I\}$,
- $(P, a, P') \in \delta'$ if and only if $P' \models \bigwedge_{p \in P} \delta(p, a)$,
- $F' = 2^F$ is the set of final states,

- for each $p \notin R$ we have an accepting condition

$$T_p = \{(P, a, P') \mid p \notin P \text{ or } P' \setminus \{p\} \models \delta(p, a)\}.$$

Proposition 13.2. The automata $\mathcal{A}$ and $\mathcal{A}'$ accept the same language.

The proof that $\mathcal{A}$ and $\mathcal{A}'$ accept the same language is a little bit technical, but not very hard. Details are left to the reader or can be found in [22].

We now state and prove the converse of Proposition 13.1.

Proposition 13.3. Let $L \subseteq \Sigma^\infty$ be accepted by some very weak alternating automaton. Then L is aperiodic.

Proof. Let $\mathcal{A} = (Q, \Sigma, \delta, I, F, R)$ be a very weak alternating automaton. For a word u and subsets P and P' of Q we write

$$P \stackrel{u}{\Longrightarrow} P',$$

if $\mathcal{A}$ has a run (V, E, ρ) over u, where P is the set of labels of the roots and P' is the set of labels of the leaves on level $|u|$. This means that in the corresponding generalized Büchi automaton $\mathcal{A}'$ there is path from state P to state P', which is labeled by the word u.

Let $m = |Q|$, we want to show that $P \stackrel{u^m}{\Longrightarrow} P'$ if and only if $P \stackrel{u^{m+1}}{\Longrightarrow} P'$ for all words u and subsets P and P'. This implies that the transformation monoid of $\mathcal{A}'$ is aperiodic. Then, we conclude that languages accepted by very weak alternating automata are always aperiodic in a similar way as in the proof of Proposition 11.11, (because the generalized accepting condition can be easily incorporated in that proof).

First, assume that $P \stackrel{u^m}{\Longrightarrow} P'$ and let us see that $P \stackrel{u^{m+1}}{\Longrightarrow} P'$, too. This is true if u is the empty word. Hence we may assume that $|u| \geq 1$. Let (V, E, ρ) be the forest which corresponds to this run. We assume that $P = \{p\}$ and that (V, E, ρ) is tree. This is not essential, but it simplifies the picture a little bit. To simplify the picture further, we assume that $u = a$ is in fact a letter. Formally, we replace E by $E^{|u|}$ and we restrict the new forest to the tree which has the same root as (V, E, ρ). Note that the set of leaves which were on level $|u^m|$ before are now exactly the leaves on level $|m|$. Hence the assumption $u = a$ is justified.

Since $m = |Q|$ we find on each branch from the root to leaves a first node which has the same label as its parent node. This happens because the automaton is very weak and therefore the ordering on the way down never increases. We cut the tree at these nodes and these nodes are called *fresh leaves*. See Figure 6, where the fresh leaves have labels q, q, p, and r from left-to-right.

Now, at each fresh leaf we glue the original sub tree of its parent node. We obtain a new tree of height $m+1$ which has as the set of labels at level $m+1$ exactly the same labels as before the labels at level m in the original tree. (See Figure 7.) It is clear that the new tree is a run over u^{m+1} and thus, $P \overset{u^{m+1}}{\Longrightarrow} P'$ as desired.

For the other direction, assume that $P \overset{u^{m+1}}{\Longrightarrow} P'$ and let (V, E, ρ) be a forest which corresponds to this run. Just as above we may assume that (V, E, ρ) is a tree and that u is a letter. This time we go down from the root to leaves and we cut at the first node, where the node has the same label as one of its children. See Figure 8. Now, we glue at these new leaves the original sub tree of one of its children which has the same label.

We obtain a new tree of height m such that each label at the leaves on level m appeared before as a label on some leaf of the original tree (V, E, ρ) at level $m+1$, see Figure 9.

Thus, $P \overset{u^{m}}{\Longrightarrow} P''$ for some subset $P'' \subseteq P'$. But the satisfiability relation is monotone; therefore $P \overset{u^{m}}{\Longrightarrow} P'$, too. Thus, indeed $P \overset{u^{m}}{\Longrightarrow} P'$ if and only if $P \overset{u^{m+1}}{\Longrightarrow} P'$ for $m = |Q|$. Q.E.D.

References

[1] A. Arnold. A syntactic congruence for rational omega-language. *Theor. Comput. Sci.*, 39:333–335, 1985.

[2] F. Blanchet-Sadri. Some logical characterizations of the dot-depth hierarchy and applications. *J. Comput. System Sci.*, 51(2):324–337, 1995.

[3] J. R. Büchi. On a decision method in restricted second order arithmetic. In *Logic, Methodology and Philosophy of Science (Proc. 1960 Internat. Congr .)*, pages 1–11, Stanford, Calif., 1962. Stanford Univ. Press.

[4] S. Cho and D. T. Huynh. Finite-automaton aperiodicity is pspace-complete. *Theor. Comput. Sci.*, 88(1):99–116, 1991.

[5] V. Diekert and P. Gastin. Pure future local temporal logics are expressively complete for mazurkiewicz traces. *Inf. Comput.*, 204(11):1597–1619, 2006.

[6] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1995.

[7] K. Etessami, M. Y. Vardi, and Th. Wilke. First-order logic with two variables and unary temporal logic. *Inf. Comput.*, 179(2):279–295, 2002.

[8] A. Fernández López and M. Tocón Barroso. The local algebras of an associative algebra and their applications. In J. Misra, editor, *Applicable Mathematics in the Golden Age*, pages 254–275, New Delhi, India, 2002. Narosa.

[9] D. M. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal basis of fairness. In *POPL*, pages 163–173, 1980.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[11] P. Gastin and D. Oddoux. Fast ltl to büchi automata translation. In G. Berry, H. Comon, and A. Finkel, editors, *CAV*, volume 2102 of *Lecture Notes in Computer Science*, pages 53–65. Springer, 2001.

[12] P. Gastin and A. Petit. Infinite traces. In *The book of traces*, pages 393–486. World Sci. Publ., River Edge, NJ, 1995.

[13] J. A. W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles (California), 1968.

[14] D. Kozen. Lower bounds for natural proof systems. In *FOCS*, pages 254–266. IEEE, 1977.

[15] R. E. Ladner. Application of model theoretic games to discrete linear orders and finite automata. *Information and Control*, 33(4):281–303, 1977.

[16] C. Lautemann, P. McKenzie, T. Schwentick, and H. Vollmer. The descriptive complexity approach to logcfl. *J. Comput. Syst. Sci.*, 62(4):629–652, 2001.

[17] C. Löding and W. Thomas. Alternating automata and logics over infinite words. In J. van Leeuwen, O. Watanabe, M. Hagiya, P. D. Mosses, and T. Ito, editors, *IFIP TCS*, volume 1872 of *Lecture Notes in Computer Science*, pages 521–535. Springer, 2000.

[18] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966.

[19] R. McNaughton and S. Papert. *Counter-free automata.* The M.I.T. Press, Cambridge, Mass.-London, 1971. With an appendix by William Henneman, M.I.T. Research Monograph, No. 65.

[20] K. Meyberg. Lectures on algebras and triple systems, 1972.

[21] D. E. Muller, A. Saoudi, and P. E. Schupp. Alternating automata. the weak monadic theory of the tree, and its complexity. In L. Kott, editor, *ICALP*, volume 226 of *Lecture Notes in Computer Science*, pages 275–283. Springer, 1986.

[22] D. Oddoux. *Utilisation des automates alternants pour un model-checking efficace des logiques temporelles linaires.* PhD thesis, Université Paris 7 (France), 2003.

[23] D. Perrin. Recent results on automata and infinite words. In M. Chytil and V. Koubek, editors, *MFCS*, volume 176 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 1984.

[24] D. Perrin and J.-E. Pin. First-order logic and star-free sets. *J. Comput. Syst. Sci.*, 32(3):393–406, 1986.

[25] D. Perrin and J.-E. Pin. *Infinite Words. Automata, Semigroups, Logic and Games*, volume 141 of *Pure and Applied Mathematics.* Elsevier, Amsterdam, 2004.

[26] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

[27] S. Rhode. *Alternating automata and the temporal logic of ordinals.* PhD Thesis, University of Illinois, Urbana Campaign II, 1997.

[28] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.

[29] J. Stern. Complexity of some problems from the theory of automata. *Information and Control*, 66(3):163–176, 1985.

[30] P. Tesson and D. Thérien. Diamonds are forever: the variety DA. In *Semigroups, algorithms, automata and languages (Coimbra, 2001)*, pages 475–499. World Sci. Publ., River Edge, NJ, 2002.

[31] W. Thomas. Star-free regular sets of omega-sequences. *Information and Control*, 42:148–156, 1979.

[32] W. Thomas. A combinatorial approach to the theory of omega-automata. *Information and Control*, 48(3):261–283, 1981.

[33] W. Thomas. Classifying regular events in symbolic logic. *J. Comput. Syst. Sci.*, 25(3):360–376, 1982.

[34] W. Thomas. An application of the Ehrenfeucht-Fraïssé game in formal language theory. *Mém. Soc. Math. France (N.S.)*, 16:11–21, 1984. Logic (Paris, 1983).

[35] W. Thomas. A concatenation game and the dot-depth hierarchy. In E. Börger, editor, *Computation Theory and Logic*, volume 270 of *Lecture Notes in Computer Science*, pages 415–426. Springer, 1987.

[36] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 133–192. Elsevier, Amsterdam, 1990.

[37] W. Thomas. Languages, automata, and logic. In *Handbook of formal languages, Vol. 3*, pages 389–455. Springer, Berlin, 1997.

[38] Th. Wilke. *Classifying Discrete Temporal Properties*. Habilitationsschrift, Universität Kiel, Apr. 1998.

[39] Th. Wilke. Classifying discrete temporal properties. In *STACS 99 (Trier)*, volume 1563 of *Lecture Notes in Comput. Sci.*, pages 32–46, Berlin, 1999. Springer.