SEPARATION PROPERTY FOR ωB- AND ωS-REGULAR LANGUAGES

MICHAŁ SKRZYPCZAK

Institute of Informatics, University of Warsaw, ul. Banacha 2, 02-097 Warsaw, Poland
e-mail address: mskrzypczak@mimuw.edu.pl

ABSTRACT. In this paper we show that ωB- and ωS-regular languages satisfy the following separation-type theorem

> If L_1, L_2 are disjoint languages of ω-words both recognised by ωB- (resp. ωS)-automata then there exists an ω-regular language L_{sep} that contains L_1, and whose complement contains L_2.

In particular, if a language and its complement are recognised by ωB- (resp. ωS)-automata then the language is ω-regular.

The result is especially interesting because, as shown by Bojańczyk and Colcombet, ωB-regular languages are complements of ωS-regular languages. Therefore, the above theorem shows that these are two mutually dual classes that both have the separation property. Usually (e.g. in descriptive set theory or recursion theory) exactly one class from a pair $\mathcal{C}, \mathcal{C}^c$ has the separation property.

The proof technique reduces the separation property for ω-word languages to profinite languages using Ramsey's theorem and topological methods. After that reduction, the analysis of the separation property in the profinite monoid is relatively simple. The whole construction is technically not complicated, moreover it seems to be quite extensible.

The paper uses a framework for the analysis of B- and S-regular languages in the context of the profinite monoid that was proposed by Toruńczyk.

1. INTRODUCTION

The classes of ωB- and ωS-regular languages are extensions of ω-regular languages proposed by Bojańczyk and Colcombet in [BC06]. The idea is to define *asymptotic* properties of ω-words. The standard example is the following ωB-regular language

$$\{a^{n_0} b a^{n_1} b a^{n_2} b \dots : \text{ the sequence } n_i \text{ is bounded}\} \subseteq \{a, b\}^{\omega}.$$

The main technical contribution of [BC06] is the following theorem.

Theorem 1.1 (Theorem 4.1 in [BC06])**.** The complement of an ωB-regular language is effectively ωS-regular and vice versa.

In this paper we show that both these classes admit the separation property. In general, a class of languages $\mathcal{C}$ has the *separation property* with respect to a class $\mathcal{D}$, if the following condition holds:

> For every pair of disjoint languages L_1, L_2 from $\mathcal{C}$ there exists a language $L_{\text{sep}} \in \mathcal{D}$ such that[1]

$$L_1 \subseteq L_{\text{sep}} \quad \text{and} \quad L_2 \subseteq L_{\text{sep}}^c.$$

In that case we say that L_{sep} *separates* L_1 and L_2. If not mentioned otherwise, the class $\mathcal{D}$ is taken as $\mathcal{C} \cap \mathcal{C}^c$.

Usually, one class from a pair of dual classes $\mathcal{C}, \mathcal{C}^c$ has the separation property and the other one does not. Below we recall some known separation-type theorems. The first one is a simple observation about Borel sets.

Theorem 1.2 (Theorem II 22.16 in [Kec95]). Let $\eta < \omega_1$. Every two disjoint $\mathbf{\Pi}_\eta^0$ languages can be separated by a language that belongs to $\mathbf{\Pi}_\eta^0 \cap \mathbf{\Sigma}_\eta^0$. On the other hand, there exists a pair of disjoint languages in $\mathbf{\Sigma}_\eta^0$ that cannot be separated as above.

The following theorem is an important extension to the projective hierarchy.

Theorem 1.3 (Lusin (see [Kec95])). If $L_1, L_2 \in \mathbf{\Sigma}_1^1$ are two disjoint analytic sets then there exists a Borel set separating them. There exists a pair of disjoint co-analytic (i.e. $\mathbf{\Pi}_1^1$) sets that cannot be separated by any Borel set.

The above theorem has its counterpart for regular languages of infinite trees. There is a correspondence between the parity index of a tree automaton and the topological complexity of the tree-language recognised by it. In particular, alternating $(1, 2)$-parity tree automata (ATA) recognise analytic languages, $(0, 1)$-parity ATA recognise co-analytic sets, while weak alternating automata recognise only Borel languages. The following two theorems are analogous to the above one in the tree-regular context.

Theorem 1.4 (Rabin [Rab70], also Kupferman, Vardi [KV99]). If L_1, L_2 are two disjoint regular tree languages recognised by $(1, 2)$-parity ATA then there exists a language separating them recognisable by a weak alternating automaton (thus Borel).

Theorem 1.5 (Hummel, Michalewski, Niwiński [HMN09]). There exists a pair of tree-languages recognised by $(0, 1)$-parity ATA that cannot be separated by any Borel set.

The above theorem is extended for higher levels of the alternating index hierarchy in [AMN12]. Theorem 1.4 relies on the fact that every alternating $(1, 2)$-parity tree automaton is equivalent to a nondeterministic $(1, 2)$-parity tree automaton.

In this work we show that both classes of ωB- and ωS-regular languages have the separation property with respect to ω-regular languages. The proposed constructions are effective. The result is especially interesting since these are two mutually dual classes (see Theorem 1.1 above). As a consequence of the separation properties we obtain the following corollary.

Corollary 1.6. If a given language of ω-words L and its complement L^c are both ωB-regular (resp. ωS-regular) then L is ω-regular.

---

[1]To distinguish the complement from the closure we denote the complement of a set X by X^c.

The above result in the case of ωB-regular languages was independently known by some researchers in the area. Nevertheless, to the best of the author's knowledge, it has never been published.

To prove the main result we reduce the separation property of ω-word languages to the case of profinite words. For this purpose we use B- and S-automata introduced in [Col09]. As shown in [Tor12] it is possible to define a language recognised by a B- or S-automaton as a subset of the profinite monoid $\widehat{A^*}$. An intermediate step in our reasoning is proving the separation property for B- and S-regular languages of profinite words.

The paper is organised as follows. In Section 2 we introduce basic notions. Section 3 defines the automata models we use. In Section 4 we prove separation results for languages of profinite words recognised by B- and S-automata. Section 5 contains the crucial technical tool, Theorem 5.1, that enables to transfer separation results for languages of profinite words to the case of ω-words. In Section 6 we use this theorem to show that ωB- and ωS-regular languages have the separation property. Section 7 contains a direct and simpler proof of the separation property for the ωB-regular case. This proof was proposed by Thomas Colcombet, we present it here with his kind permission. Finally, in Section 8 we give acknowledgements.

2. Basic notions

We work with two models (ωB and ωS) at the same time. Therefore, we introduce a notion ωT to denote one of the models: ωB or ωS. By T we denote the corresponding finite word automata (B or S). By A we denote a finite alphabet. Elements of A^* are called finite words while A^ω is the set of ω-words.

2.1. **Monoids.** We use monoids and Ramsey's theorem to decompose ω-words into finite ones.

Definition 2.1. A (finite) *monoid* is a (finite) algebraic structure M equipped with an operation $\cdot : M^2 \to M$ that is associative ($a \cdot (b \cdot c) = (a \cdot b) \cdot c$) and with a distinguished element $1 \in M$ that satisfies $1 \cdot a = a \cdot 1 = a$.

The operation $\cdot$ is called *product* and 1 is called the *neutral element*.

An element $e \in M$ is called *idempotent* if $e \cdot e = e$.

Observe that the set of all finite words A^* has a natural structure of monoid with the operation of concatenation and 1 defined as the empty word.

Definition 2.2. A function $h : M \to N$ is a *homomorphism* between monoids M, N if it preserves the product:
$$h(s \cdot t) = h(s) \cdot h(t).$$

Now we define the monoid representing possible *runs* of a nondeterministic automaton. It can be seen as an algebraic formalisation of the structure used by Büchi [Büc62] in his famous complementation lemma.

Definition 2.3. Let $\mathcal{A}$ be a nondeterministic automaton with states Q. Define $M_{\text{trans}}(\mathcal{A})$ as $\mathcal{P}(Q \times Q)$. Let the neutral element be $\{(q, q) : q \in Q\}$ and product:
$$s \cdot t = \{(p, r) : \exists_{q \in Q} \ (p, q) \in s \ \wedge \ (q, r) \in t\}.$$

Let $h_{\mathcal{A}}\colon A^* \to M_{\text{trans}}(\mathcal{A})$ map a given finite word w to the set of pairs (p, q) such that the automaton $\mathcal{A}$ has a run over w starting in p and ending in q.

It is easy to check that $M_{\text{trans}}(\mathcal{A})$ is a finite monoid and $h_{\mathcal{A}}$ is a homomorphism.

2.2. **Profinite monoid.** In this subsection we introduce the profinite monoid $\widehat{A^*}$. A formal introduction to profinite structures can be found in [Alm03] or [Pin09]. We refer to [Pin09].

First we provide a construction of the profinite monoid $\widehat{A^*}$. The idea is to enhance the set of all finite words by some *virtual* elements representing sequences of finite words that are more and more similar.

Let $K_0, K_1, \ldots$ be a list of all regular languages of finite words. Let $X = 2^\omega$. Each element $x \in X$ can be seen as a sequence of bits, the bit $x(n)$ indicates whether our *virtual* word belongs to the language K_n.

Define $\mu\colon A^* \to X$ by the following equation:

$$\mu(w)_n \ = \ \begin{cases} 1 & \text{if } w \in K_n, \\ 0 & \text{if } w \notin K_n. \end{cases}$$

The function μ defined above is an embedding of A^* into X. Let $\widehat{A^*} \subseteq X$ be the closure of $\mu(A^*)$ in X with respect to the product topology of X. Therefore, $\widehat{A^*}$ contains $\mu(A^*)$ and the limits of its elements. To simplify the notion we identify $w \in A^*$ with its image $\mu(w) \in \widehat{A^*}$.

Example 2.4 (Proposition 2.5 in [Pin09])**.** Let $w_n = a^{n!}$ for $n \in \mathbb{N}$. A simple automata-theoretic argument shows that for every regular language K, either almost all words $(w_n)_{n \in \mathbb{N}}$ belong to K or almost all do not belong to K. Therefore, the sequence $(\mu(w_n))_{n \in \mathbb{N}}$ is convergent coordinate-wise in X. The limit of this sequence is an element of $\widehat{A^*} - \mu(A^*)$.

The following fact summarises basic properties of $\widehat{A^*}$.

Fact 2.1 (Proposition 2.1, Proposition 2.4, and Theorem 2.7 in [Pin09])**.** $\widehat{A^*}$ is a compact metric space. A^* (formally $\mu(A^*)$) is a countable dense subset of $\widehat{A^*}$. $\widehat{A^*}$ has a structure of a monoid that extends the structure of A^* and the concatenation is continuous.

It turns out that the operation assigning to every regular language of finite words $K \subseteq A^*$ its topological closure $\overline{K} \subseteq \widehat{A^*}$ has good properties (see Theorem 2.6). Therefore, we introduce the following definition.

Definition 2.5. A *profinite-regular* language is a subset of $\widehat{A^*}$ of the form $\overline{K}$ for some regular language $K \subseteq A^*$.

Using this definition, we can denote a generic profinite-regular language as $\overline{K}$ for K ranging over regular languages. Using the definition of μ one can show the following easy fact.

Fact 2.2. A language of profinite words $M \subseteq \widehat{A^*}$ is profinite-regular if and only if it is of the form

$$M = \left\{ x \in X : \ x \in \widehat{A^*} \ \wedge \ x_n = 1 \right\}, \tag{2.1}$$

for some $n \in \mathbb{N}$. In that case $M = \overline{K_n}$.

The structures of profinite-regular and regular languages are in some sense identical. This is expressed by the following fact.

Theorem 2.6 (Theorem 2.4 in [Pin09])**.** The function $K \mapsto \overline{K} \subseteq \widehat{A^*}$ is an isomorphism of the Boolean algebra of regular languages and the Boolean algebra of profinite-regular languages. Its inverse is $M \mapsto \mu^{-1}(M) \subseteq A^*$ (when identifying A^* with $\mu(A^*)$ we can write $M \mapsto M \cap A^* \subseteq A^*$).

By the definition of $\widehat{A^*}$ and fact that regular languages are closed under finite intersection, we obtain the following important fact.

Fact 2.3. The family of regular languages of profinite words is a basis of the topology of $\widehat{A^*}$.

The topology of $\widehat{A^*}$ is the product topology. Therefore, a sequence of finite words $W = w_0, w_1, \ldots$ is convergent to $w \in \widehat{A^*}$ if and only if $(\mu(w_n))_{n \in \mathbb{N}} \subseteq X$ is convergent coordinate-wise to w. The following fact formulates this condition in a more intuitive way.

Fact 2.4. A sequence of finite words $W = w_0, w_1, \ldots$ is convergent to $w \in \widehat{A^*}$ if and only if for every profinite-regular language $\overline{K}$ either:
- $w \in \overline{K}$ and almost all words w_n belong to K,
- $w \notin \overline{K}$ and almost all words w_n do not belong to K.

The topology of $\widehat{A^*}$ is defined in such a way that it corresponds precisely to profinite-regular languages. The following fact summarises this correspondence.

Fact 2.5 (Proposition 4.2 in [Pin09])**.** A language $M \subseteq \widehat{A^*}$ is profinite-regular if and only if it is a closed and open (clopen) subset of $\widehat{A^*}$.

Proof. First assume that $M = \overline{K}$ is a regular language of profinite words. Equation (2.1) in Fact 2.2 defines a closed and open set.

Now assume that M is a closed and open subset of $\widehat{A^*}$. Recall that profinite-regular languages form a basis for the topology of $\widehat{A^*}$ (Fact 2.3). Since M is open so it is a union of base sets $\bigcup_{j \in J} \overline{K_j}$. Since M is a closed subset of a compact space $\widehat{A^*}$, M is compact. Therefore, only finitely many languages among $\left\{ \overline{K_j} \right\}_{j \in J}$ form a cover of M. But a finite union of profinite-regular languages is a profinite-regular language. Therefore, M is profinite-regular. $\square$

2.3. **Ramsey-type arguments.** In this subsection we recall Ramsey's theorem and show its application to finite monoids. This technique was used by Büchi in his complementation lemma [Büc62]. Additionally, we recall some extensions of Ramsey's theorem to compact spaces. In the following, by $[\mathbb{N}]^2$ we denote the set of all unordered pairs of natural numbers.

Theorem 2.7 (Ramsey)**.** Assume that $\alpha \colon [\mathbb{N}]^2 \to C$ is a function that assigns to every pair of numbers $\{n, m\} \in [\mathbb{N}]^2$ a *colour* $\alpha(\{n, m\}) \in C$. Additionally, assume that the set of colours C is finite. Then there exists an infinite *monochromatic* set $S \subseteq \mathbb{N}$: a set S such that for some colour $c \in C$ and every pair of numbers $\{n, m\} \subset S$ we have

$$\alpha(\{n, m\}) = c.$$

The following theorem shows an application of Ramsey's theorem to the ω-word case.

Theorem 2.8. Let M be a finite monoid and $h\colon A^* \to M$ be a homomorphism. Then for every ω-word $u \in A^\omega$ there exists a sequence of finite words $w_0, w_1, w_2, \ldots$ and two elements s, e of the monoid M such that:

(i) $u = w_0 w_1 w_2 \ldots$,
(ii) $h(w_0) = s$,
(iii) $h(w_n) = e$ for every $n > 0$,
(iv) $s \cdot e = s$ and $e \cdot e = e$.

A pair (s, e) satisfying the above constraints is often called a *linked pair*, see [PP04]. To simplify the properties in the above theorem we introduce the following definition.

Definition 2.9. For a given homomorphism $h\colon A^* \to M$ we say that the *type* (or *h-type*) of a decomposition $u = w_0 w_1 \ldots$ is $t = (s, e)$ if: $s \cdot e = s$, $e \cdot e = e$, $h(w_0) = s$, and $h(w_n) = e$ for all $n > 0$.

Using the above definition we can restate Theorem 2.8 as: for every ω-word u and homomorphism h there exists some decomposition of u of some type $t = (s, e)$. A priori there may be two decompositions of one ω-word of two distinct types.

There is an extension of finite-colour Ramsey's theorem to the case where colours form a compact metric space. To state it formally we use the following definitions.

Definition 2.10. Assume that $W = w_0, w_1, \ldots$ is a sequence of finite words. We say that $Z = z_0, z_1, \ldots$ is a *grouping* of W if there exists an increasing sequence of numbers $0 = i_0 < i_1 < \ldots$ such that for every $n \in \mathbb{N}$ we have

$$z_n = w_{i_n} w_{i_n+1} \ldots w_{i_{n+1}-1}.$$

Note that if W is a decomposition of an ω-word u and W is of h-type $t = (s, e)$ then every grouping of W is also a decomposition of u of h-type t. The notion of grouping introduces a stronger version of convergence.

Definition 2.11. We say that a sequence of finite words $W = w_0, w_1, \ldots$ is *strongly convergent* to a profinite word w if every grouping of W is convergent to w.

The following result can be seen as a simple extension of the Ramsey theorem to the case of the profinite monoid.

Theorem 2.12 (Bojańczyk, Kopczyński, Toruńczyk [BKT12])**.** Let $W = w_0, w_1, \ldots$ be an infinite sequence of finite words. There exists a grouping Z of W such that Z strongly converges in $\widehat{A^*}$.

For the sake of completeness we give a proof of this fact below. The theorem holds in general, where instead of $\widehat{A^*}$ is any compact metric monoid. Also, the notion of convergence can be strengthened in the thesis of the theorem: all the groupings of W converge in a *uniform way*. In this paper we use only the above, simplified form.

Proof. Let K be a regular language and $Z = z_0, z_1, \ldots$ be a sequence of finite words. Define a function $\alpha_{K,Z}\colon [\mathbb{N}]^2 \to \{0, 1\}$ that takes a pair of numbers $i < j$ and returns 1 if and only if $z_i z_{i+1} \ldots z_{j-1}$ belongs to K. By Theorem 2.7, there exists a monochromatic set $S \subseteq \mathbb{N}$ with colour $c \in \{0, 1\}$ such that for every pair $i < j \in S$ we have $\alpha_{K,Z}(\{i, j\}) = c$.

Now, take a sequence of finite words W. Let $K_0, K_1, \ldots$ be an enumeration of all regular languages and let $W^0 = W$. We proceed by induction for $i = 0, 1, \ldots$. Assume that

$W^i = w_0^i, w_1^i, \ldots$ is defined. First define z_i as w_0^i. Now, let $S = \{n_0, n_1, \ldots\}$ be an infinite monochromatic set with respect to α_{K_i, W^i}. Define

$$W^{i+1} = \left(w_{n_0}^i w_{n_0+1}^i \ldots w_{n_1-1}^i\right), \left(w_{n_1}^i w_{n_1+1}^i \ldots w_{n_2-1}^i\right), \left(w_{n_2}^i w_{n_2+1}^i \ldots w_{n_3-1}^i\right), \ldots$$

Note that W^{i+1} is a suffix of a grouping of W^i. Since S is monochromatic and by the definition of $\alpha_{K,Z}$, we know that:

($*$) For every grouping of W^{i+1} either all words in the grouping belong to K_i or all of them do not belong.

We claim that our sequence $Z = z_0, z_1, \ldots$ is strongly convergent. Let Y be a grouping of Z and let $K = K_i$ be a regular language. Observe that almost all words in Y (all except first at most i words) are obtained by grouping words in W^{i+1}. Therefore, by ($*$), either almost all words of Y belong to K or almost all of them do not belong to K. Fact 2.4 implies that Y is convergent in $\widehat{A^*}$.

Now observe that almost all words in Y belong to K_i if and only if almost all words in Z belong to K_i. Therefore, the limit of Y does not depend on the choice of Y. It means that Z is strongly convergent in $\widehat{A^*}$. $\square$

2.4. **Notation.** In this paper we deal with three types of languages: of finite words, of profinite words, and of ω-words. To simplify reading of the paper, we use the following conventions:

- finite and profinite words are denoted by w, z,
- sequences of finite words are denoted by W, Z, Y,
- ω-words are denoted by u, v,
- regular languages of finite words are denoted by K,
- profinite-regular languages are, using Theorem 2.6, denoted by $\overline{K}$,
- general languages of profinite words are denoted by M,
- languages of ω-words (both ω-regular and not) are denoted by L.

3. Automata

In this section we provide definitions of four kinds of automata: ω-word models ωB- and ωS-automata and their finite word variants B- and S-automata.

The ωB- and ωS-automata models were introduced in [BC06], we follow the definitions from this work. The B- and S-automata models were defined in [Col09]. For the sake of simplicity, we use only the operations $\{\mathbf{nil}, \mathbf{inc}, \mathbf{reset}\}$ (without the *check* operation). As noted in Remark 1 in [Col09], this restriction does not influence the expressive power.

The four automata models we study here are part of a more general theory of regular cost functions that is developed by Colcombet [Col09, Col13]. In particular, the theory of B- and S-automata has been extended to finite trees in [CL10].

All four automata models we deal with are built on the basis of a *counter automaton*. The difference is the acceptance condition that we introduce later.

Definition 3.1. A *counter automaton* is a tuple $\mathcal{A} = \langle A, Q, I, \Gamma, \delta \rangle$, where:
- A is an input alphabet,
- Q is a finite set of states,
- $I \subseteq Q$ is a set of initial states,
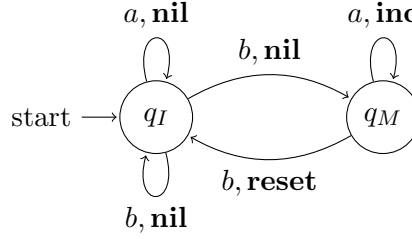
Figure 1: An example of an ωB-automaton $\mathcal{A}_{\omega\mathrm{B}}$.

- Γ is a finite set of counters,
- $\delta \subseteq Q \times A \times \{\mathbf{nil}, \mathbf{inc}, \mathbf{reset}\}^{\Gamma} \times Q$ is a transition relation.

All counters store natural numbers and cannot be read during a run. The values of the counters are only used in an acceptance condition.

In the initial configuration all counters equal 0. A transition $(p, a, o, q) \in \delta$ (sometimes denoted $p \xrightarrow{a,o} q$) means that if the automaton is in a state p and reads a letter a then it can perform counter operations o and go to the state q. For a counter $c \in \Gamma$ a counter operation $o(c)$ can:

$o(c) = \mathbf{nil}$	leave the counter value unchanged,
$o(c) = \mathbf{inc}$	increment the counter value by one,
$o(c) = \mathbf{reset}$	reset the counter value to 0.

A run ρ of the automaton $\mathcal{A}$ over a word (finite or infinite) is a sequence of transitions as for standard nondeterministic automata. Given a run ρ, a counter $c \in \Gamma$, and a position r_c of a word where the counter c is reset, we define $\mathrm{val}(c, \rho, r_c)$ as the value stored in the counter c at the moment before the reset r_c in ρ.

To simplify the constructions we allow ϵ-transitions in our automata. The only requirement is that there is no cycle consisting of ϵ-transitions only. ϵ-transitions can be removed using nondeterminism of an automaton and by combining a sequence of counter operations into one operation. Such a modification may change the exact values of counters, for instance when we replace $\mathbf{inc}, \mathbf{reset}$ by $\mathbf{reset}$. However, the limitary properties of the counters are preserved (the values may be disturbed only by a linear factor).

3.1. ωB- and ωS-automata. First we deal with automata for ω-words, following the definitions in [BC06]. An ωT-*automaton* (for $\omega\mathrm{T} \in \{\omega\mathrm{B}, \omega\mathrm{S}\}$) is just a counter automaton. A run ρ of an ωT-automaton over an ω-word u is accepting if it starts in an initial state in I, every counter is reset infinitely many times, and the following condition is satisfied:

ωB-**automaton:** the values of all counters are bounded during the run,
ωS-**automaton:** for every counter c the values of c during subsequent resets in ρ tend to
    infinity (i.e. the limit of the values of c is ∞).

An ωT-automaton $\mathcal{A}$ accepts an ω-word if it has an accepting run on it. The set of all ω-words accepted by $\mathcal{A}$ is denoted $\mathrm{L}(\mathcal{A})$.

Example 3.2. Consider the ωB-automaton $\mathcal{A}_{\omega\mathrm{B}}$ depicted on Figure 1. $\mathcal{A}_{\omega\mathrm{B}}$ guesses (by moving to the state q_M) to measure the length of some blocks of letters a. It accepts an

ω-word u if and only if it is of the form

$$u = a^{n_0} b a^{n_1} b \dots \quad \text{with} \quad \liminf_{i \to \infty} n_i < \infty.$$

We can also treat $\mathcal{A}_\omega$B as an ωS-automaton. In that case the language recognised by $\mathcal{A}_{\omega \mathrm{B}}$ is

$$\{u \in \{a,b\}^\omega : \; u = a^{n_0} b a^{n_1} b \dots \quad \text{and} \quad \limsup_{i \to \infty} n_i = \infty\}.$$

It is easy to check that a nondeterministic Büchi automaton can be transformed into an equivalent ωB- (resp. ωS)-automaton. Therefore, all ω-regular languages are both ωB- and ωS-regular.

3.2. B- and S-automata.

In the finite word models the situation is a little more complicated than in the ωB- and ωS-automata models. The automaton not only accepts or rejects a given word but also it assigns a *value* to a word.

Formally, a T-*automaton* (for $\mathrm{T} \in \{\mathrm{B}, \mathrm{S}\}$) is a counter automaton that is additionally equipped with a set of final states $F \subseteq Q$. An accepting run ρ of an automaton over a finite word w is a sequence of transitions starting in some initial state in I and ending in some final state in F.

The following equations define $\mathrm{val}(\mathcal{A}, w)$ — the value assigned to a given finite word by a given automaton. We use the convention that if a set of values is empty then the minimum of this set is ∞ and the maximum is 0. The variable ρ ranges over all accepting runs, c ranges over counters in Γ, while r_c ranges over positions where the counter c is reset in ρ. As noted at the beginning of this section, we do not allow explicit *check* operation, we only care about the values of the counters before resets.

B-automaton $\mathcal{A}_{\mathrm{B}}$:

$$\mathrm{val}(\mathcal{A}_{\mathrm{B}}, w) = \min_\rho \mathrm{val}(\rho) \quad \text{and} \quad \mathrm{val}(\rho) = \max_c \max_{r_c} \; \mathrm{val}(c, \rho, r_c),$$
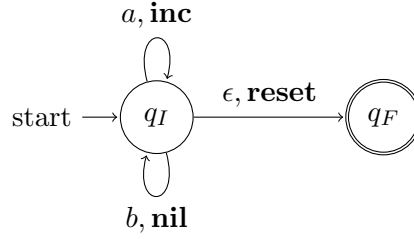
S-automaton $\mathcal{A}_{\mathrm{S}}$:

$$\mathrm{val}(\mathcal{A}_{\mathrm{S}}, w) = \max_\rho \mathrm{val}(\rho) \quad \text{and} \quad \mathrm{val}(\rho) = \min_c \min_{r_c} \; \mathrm{val}(c, \rho, r_c).$$

The following simple observation is crucial in the subsequent definitions.

Lemma 3.3. For a given number n, a B-automaton $\mathcal{A}_{\mathrm{B}}$, and an S-automaton $\mathcal{A}_{\mathrm{S}}$ the following languages of finite words are regular:

$$\begin{aligned} \mathrm{L}(\mathcal{A}_{\mathrm{B}} \leq n) &= \{w : \mathrm{val}(\mathcal{A}_{\mathrm{B}}, w) \leq n\}, \\ \mathrm{L}(\mathcal{A}_{\mathrm{S}} > n) &= \{w : \mathrm{val}(\mathcal{A}_{\mathrm{S}}, w) > n\}. \end{aligned}$$

Proof. We can encode a bounded valuation of the counters into a state of a finite automaton. $\qquad \square$

Figure 2: An example of an S-automaton $\mathcal{A}_S$.

3.3. Languages. The above definitions give a semantics of a T-automaton in terms of a function val$(\mathcal{A}, .)\colon A^* \to \mathbb{N} \cup \{\infty\}$. As noted in [Tor12], it is possible to define the language recognised by such an automaton as a subset of the profinite monoid $\widehat{A^*}$. We successively define it for B-automata and S-automata. In both cases the construction is justified by Lemma 3.3.

B **case:** Fix a B-automaton $\mathcal{A}_B$ and define

$$\mathrm{L}(\mathcal{A}_B) := \bigcup_{n \in \mathbb{N}} \overline{\mathrm{L}(\mathcal{A}_B \leq n)} \subseteq \widehat{A^*}. \tag{3.1}$$

S **case:** Fix an S-automaton $\mathcal{A}_S$ and define

$$\mathrm{L}(\mathcal{A}_S) := \bigcap_{n \in \mathbb{N}} \overline{\mathrm{L}(\mathcal{A}_S > n)} \subseteq \widehat{A^*}. \tag{3.2}$$

Note that the sequences of languages in the above equations are monotone: increasing in (3.1) and decreasing in (3.2).

There exists another, equivalent way of defining languages recognised by these automata [Tor12]. One can observe that the function val$(\mathcal{A}, .)$ assigning to every finite word its value has a unique continuous extension on $\widehat{A^*}$. The languages recognised by B- and S-automata can be defined as val$(\mathcal{A}, .)^{-1}(\mathbb{N})$ and val$(\mathcal{A}, .)^{-1}(\{\infty\})$ respectively. In this work we only refer to the definitions (3.1) and (3.2).

Example 3.4. Consider the S-automaton $\mathcal{A}_S$ depicted in Figure 2. The automaton measures the number of letters a in a given word. Then it guesses that the word is finished and moves to the accepting state. For every finite word w the value val$(\mathcal{A}_S, w)$ equals the number of letters a in w.

The language $\mathrm{L}(\mathcal{A}_S)$ does not contain any finite word. It contains a profinite word w if for every n the word w belongs to the profinite-regular language defined by the formula "the word contains more than n letters a" (i.e. $w \in \overline{\mathrm{L}(\mathcal{A}_S > n)}$). In particular, the limit of the sequence $(a^{n!})_{n \in \mathbb{N}}$ from Example 2.4 belongs to $\mathrm{L}(\mathcal{A}_S)$.

Lemma 3.5. Every B-regular language is an open subset of $\widehat{A^*}$ and dually every S-regular language is closed.

Proof. By equations (3.1) and (3.2), a B-regular language is a sum of profinite-regular languages and an S-regular language is an intersection of profinite-regular languages. By Fact 2.5, profinite-regular languages are closed and open, therefore their sum is open and the intersection is closed. □

The converse of Lemma 3.5 is false as there are uncountably many open subsets of $\widehat{A^*}$. We finish the definitions of automata models by recalling the following theorem.

Theorem 3.6 (Fact 2.6 and Corollary 3.4 in [BC06], Theorem 8 and paragraph *Closure properties* in [Tor12]). Let $T \in \{B, S, \omega B, \omega S\}$. The class of T-regular languages is effectively closed under union and intersection. The emptiness problem for T-regular languages is decidable.

Therefore, it is decidable whether given two T-regular languages are disjoint.

4. SEPARATION FOR PROFINITE LANGUAGES

In this section we show the following theorem.

Theorem 4.1. Let $T \in \{B, S\}$. Assume that the languages of profinite words $M_1, M_2 \subseteq \widehat{A^*}$ are recognised by T-automata and $M_1 \cap M_2 = \emptyset$. Then there exists a profinite-regular language $\overline{K_{\text{sep}}} \subseteq \widehat{A^*}$ such that

$$M_1 \subseteq \overline{K_{\text{sep}}} \quad \text{and} \quad M_2 \subseteq \overline{K_{\text{sep}}}^c.$$

The proof of the theorem consists of two parts, for the two cases of $T \in \{B, S\}$: Lemma 4.2 and Theorem 4.4.

First we prove the case when $T = S$. The presented proof uses a general topological fact: the separation property of closed (i.e. $\mathbf{\Pi}_1^0$) sets in a zero-dimensional Polish space.

Lemma 4.2. A pair of disjoint S-regular languages of profinite words can be separated by a profinite-regular language.

Proof. Take two S-regular languages $M_1, M_2 \subseteq \widehat{A^*}$.

Since $\widehat{A^*}$ is a zero-dimensional Polish space, the $\mathbf{\Pi}_1^0$-separation property holds for $\widehat{A^*}$ (see Theorem 22.16 in [Kec95]). By Lemma 3.5 every S-regular language is $\mathbf{\Pi}_1^0$ in $\widehat{A^*}$, therefore M_1, M_2 can be separated in $\widehat{A^*}$ by a set M_{sep} that is closed and open. By Fact 2.5, the language M_{sep} is profinite-regular. $\square$

Instead of using the $\mathbf{\Pi}_1^0$-separation property, one can provide the following straightforward argument that uses the compactness of $\widehat{A^*}$. We know that M_1 is a closed subset of a compact space $\widehat{A^*}$ so M_1 is itself compact. Assume that M_2 is recognised by an S-automaton $\mathcal{A}_S$. By (3.2) we obtain

$$M_2 = \bigcap_{n \in \mathbb{N}} \overline{L(\mathcal{A}_S > n)} \subseteq \widehat{A^*}.$$

For $n \in \mathbb{N}$ define $N_n := \overline{L(\mathcal{A}_S > n)}^c$ — the complement of the profinite-regular language $\overline{L(\mathcal{A}_S > n)}$. Clearly $M_1 \subseteq \bigcup_n N_n$ because M_1 and M_2 are disjoint. Fact 2.5 and Lemma 3.3 imply that the sets N_n are open subsets of $\widehat{A^*}$. Therefore, the family $(N_n)_{n \in \mathbb{N}}$ is an open cover of M_1. Since M_1 is compact, there is $n_0 \in \mathbb{N}$ such that

$$M_1 \subseteq N_0 \cup N_1 \cup \ldots \cup N_{n_0} = N_{n_0}.$$

Therefore, N_{n_0} is a profinite-regular language that separates M_1 and M_2.

Remark 4.3. The language N_{n_0} can be computed effectively.

Proof. It is enough to observe that n_0 can be taken as the minimal n such that M_1 does not intersect the profinite-regular language $\overline{L(\mathcal{A}_S > n)}$. Such n exists by the above argument. $\square$

Now we proceed with the separation property for B-regular languages. By Lemma 3.5 we know that B-regular languages are open sets in $\widehat{A^*}$. An easy exercise shows that in general open sets do not have the separation property. Thus, to show the following theorem we need an argument that is a bit more involved than in the case of S-regular languages.

Theorem 4.4. A pair of disjoint B-regular languages of profinite words can be separated by a profinite-regular language.

We obtain the above theorem by applying the following observation.

Lemma 4.5. For every B-regular language $M_{\mathrm{B}} \subseteq \widehat{A^*}$ there exists a profinite-regular language $\overline{K_R} \subseteq \widehat{A^*}$ such that

$$M_{\mathrm{B}} \subseteq \overline{K_R} \quad \text{and} \quad M_{\mathrm{B}} \cap A^* = \overline{K_R} \cap A^*.$$

Moreover, the language $\overline{K_R}$ can be computed effectively.

Proof. Take a B-automaton $\mathcal{A}_B$ recognising M_{B}. Define a new automaton $\mathcal{A}_R$ by removing from $\mathcal{A}_B$ all the counters and all the counter operations. What remains are transitions, initial states, and final states. Put $\overline{K_R} = \overline{\mathrm{L}(\mathcal{A}_R)} \subseteq \widehat{A^*}$. Of course $M_{\mathrm{B}} \subseteq \overline{\mathrm{L}(\mathcal{A}_R)}$ by the definition of M_{B}. Clearly $\overline{\mathrm{L}(\mathcal{A}_R)} \cap A^* = \mathrm{L}(\mathcal{A}_R)$ by Theorem 2.6. What remains to show is that $\mathrm{L}(\mathcal{A}_R) \subseteq M_{\mathrm{B}}$.

Take a finite word $w \in \mathrm{L}(\mathcal{A}_R)$. Observe that $\mathcal{A}_B$ has an accepting run on w because $w \in \mathrm{L}(\mathcal{A}_R)$. So $\mathrm{val}(\mathcal{A}_B, w) \leq |w|$ because $\mathcal{A}_B$ cannot do more increments than the number of positions of the word. Therefore $w \in M_{\mathrm{B}}$. $\square$

Proof of Theorem 4.4. Take two disjoint B-regular languages $M_1, M_2 \subseteq \widehat{A^*}$. Define $\overline{K_{\mathrm{sep}}}$ to be the language $\overline{K_R}$ from Lemma 4.5 for M_1. Thus we know that $M_1 \subseteq \overline{K_R}$. We only need to show that $M_2 \cap \overline{K_R} = \emptyset$. Assume the contrary, that $M_I := M_2 \cap \overline{K_R} \neq \emptyset$. Since B-regular languages are open sets in $\widehat{A^*}$, M_I is an open set. Since A^* is dense in $\widehat{A^*}$ so M_I contains a finite word $w \in A^*$. But by the definition of $\overline{K_R}$ in that case $w \in M_1$. So $w \in M_1 \cap M_2$ — a contradiction to the disjointness of M_1, M_2. $\square$

Remark 4.6. Both separation results for B- and S-regular languages are effective: there is an algorithm that inputs two counter automata, verifies that the intersection of the languages is empty, and outputs an automaton recognising a separating language.

Proof. By Theorem 3.6 it is decidable if two B- (resp. S-)regular languages are disjoint. As observed in Remark 4.3 and Lemma 4.5, both constructions can be performed effectively. $\square$

5. REDUCTION

This section contains a proof of our crucial technical tool — Theorem 5.1. It is inspired by the *reduction theorem* from [Tor12].

Intuitively, ωB- and ωS-automata are composed of two *orthogonal* parts, we can call them the ω-regular part and the asymptotic part. The ω-regular part corresponds to states and transitions of the automaton, while the asymptotic part represents quantitative conditions that can be measured by counters. In this section we show how to formally state this division. It can be seen as an extension of the technique presented in [BC06].

Theorem 5.1. Fix an ωT-automaton $\mathcal{A}$ and a type $t = (s, e)$ in the trace monoid $M_{\mathrm{trans}}(\mathcal{A})$. There exists a T-regular language of profinite words $M_t \subseteq \widehat{A^*}$ with the following property:

If u is an ω-word and $W = w_0, w_1, \ldots$ is a decomposition of u of type t then the following conditions are equivalent:

(1) $u \in \mathrm{L}(\mathcal{A})$,
(2) there exists a grouping Z of W that strongly converges to a profinite word $z \in M_t$,
(3) there exists a grouping Z of W that converges to a profinite word $z \in M_t$.

Additionally, one can ensure that $M_t \subseteq \overline{h_{\mathcal{A}}^{-1}(e)}$. The construction of a T-automaton recognising M_t is effective given $\mathcal{A}$ and t.

The rest of this section is devoted to showing the above theorem. We fix for the whole proof an ωT-automaton $\mathcal{A} = \langle A, Q, I, \Gamma, \delta \rangle$ and a type $t = (s, e)$ in $M_{\mathrm{trans}}(\mathcal{A})$.

Intuitively, the requirement for a decomposition W to be of the type t corresponds to the ω-regular part of $\mathcal{A}$ while the convergence of W to an element of M_t takes care of the asymptotic part of $\mathcal{A}$.

Let us put $K_e = h_{\mathcal{A}}^{-1}(e)$ and assume that $\mathcal{B}_e = \langle A, Q_e, \{q_{I,e}\}, \delta_e, F_e \rangle$ is a deterministic finite automaton recognising the regular language K_e. We will ensure that $M_t \subseteq \overline{K_e}$.

First we show how to construct a language M_t, later we prove its properties. The definition of M_t depends on whether $\mathrm{T} = \mathrm{B}$ or $\mathrm{T} = \mathrm{S}$. The first case is a bit simpler.

Case $\mathrm{T} = \mathrm{B}$ The language M_t is obtained as the union of finitely many B-regular languages indexed by states $q \in Q$:

$$M_t = \bigcup_{q \in Q} \mathrm{L}(\mathcal{A}_q),$$

for B-automata $\mathcal{A}_q$ that we describe below. Intuitively, an automaton $\mathcal{A}_q$ measures *loops* in $\mathcal{A}$ starting and ending in q.

If for no $q_0 \in I$ we have $(q_0, q) \in s$ or if $(q, q) \notin e$ then $\mathrm{L}(\mathcal{A}_q) = \emptyset$. Assume otherwise. First we give an informal definition of $\mathcal{A}_q$:

- it is obtained from $\mathcal{A}$ by interpreting it as a finite word B-automaton,
- it has initial and final state set to q,
- it checks that all the counters are reset in a given word,
- it checks that a given word belongs to K_e,
- it resets all the counters at the end of the word.

Now we give a precise definition of $\mathcal{A}_q = \langle A, Q_q, I_q, \Gamma_q, \delta_q, F_q \rangle$. Let:

- $Q_q = \{*\} \cup Q \times Q_e \times \{\bot, \top\}^{\Gamma}$,
- $I_q = \{(q, q_{I,e}, (\bot, \bot, \ldots, \bot))\}$,
- $\Gamma_q = \Gamma$,
- $F_q = \{*\}$,

and let δ_q contain the following transitions:

- $(p, r, b) \xrightarrow{a,o} (p', r', b')$ if $p \xrightarrow{a,o} p' \in \delta$, $r \xrightarrow{a} r' \in \delta_e$ and for every $c \in \Gamma$ we have $b'(c) = b(c) \vee (o(c) = \mathbf{reset})$,
- $(q, r, (\top, \top, \ldots, \top)) \xrightarrow{\epsilon,o} *$ for $o = (\mathbf{reset}, \mathbf{reset}, \ldots, \mathbf{reset})$ if $r \in F_e$.

The state $*$ is the only final state used to perform the reset at the end of a word. During a run, the automaton $\mathcal{A}_q$ simulates $\mathcal{A}$ and $\mathcal{B}_e$ in parallel, using Q and Q_e. Additionally, a vector in $\{\bot, \top\}^{\Gamma}$ denotes for every counter whether it was already reset in a word or not.

Case T = S In that case the language M_t is obtained as the union of finitely many S-regular languages indexed by pairs $(q, \tau) \in Q \times \{\leftarrow, \rightarrow\}^\Gamma$:

$$M_t = \bigcup_{(q,\tau)} \mathrm{L}(\mathcal{A}_{q,\tau}).$$

Intuitively, an automaton $\mathcal{A}_{q,\tau}$ recognises loops $q \rightarrow^* q$ as before. Additionally, the vector τ denotes whether a given counter $c \in \Gamma$ obtains bigger values before the first reset ($\tau(c) = \rightarrow$) or after the last reset ($\tau(c) = \leftarrow$) on a given finite word. The following definition formalises this property. A similar technique of assigning a *reset type* to a finite run can be found in [BC06].

Definition 5.2. Let ρ be a run of some counter automaton $\mathcal{A}$ over an ω-word u. Let $k \in \mathbb{N}$ be a position in u and let $c \in \Gamma$ be a counter of $\mathcal{A}$. Let:

- V_L be the number of increments of c between the last reset before k and k,
- V_R be the number of increments of c between k and the first reset after k.

If there is no reset of c at some side of k then the respective value is 0. Define the *end-type* of c on ρ in k (denoted as $\mathrm{Etp}(c, \rho, k)$) by the following equation:

$$\mathrm{Etp}(c, \rho, k) = \begin{cases} \rightarrow & \text{if } V_L < V_R, \\ \leftarrow & \text{if } V_L \geq V_R. \end{cases}$$

As before if for no $q_0 \in I$, we have $(q_0, q) \in s$ or if $(q, q) \notin e$ then $\mathrm{L}(\mathcal{A}_{q,\tau}) = \emptyset$. Assume otherwise. We start with an informal definition of $\mathcal{A}_{q,\tau}$:

- it is obtained from $\mathcal{A}$ by interpreting it as a finite word S-automaton,
- it has initial and final state set to q,
- it checks that all the counters are reset in a given word,
- it checks that a given word belongs to K_e,
- for every counter $c \in \Gamma$:
  - if $\tau(c) = \leftarrow$ then $A_{q,\tau}$ skips the first reset of c and all the previous increments of c but resets c at the end of a given word,
  - if $\tau(c) = \rightarrow$ then $A_{q,\tau}$ acts on c exactly as $\mathcal{A}$ (with no additional reset at the end of the word).

  Formally, let $\mathcal{A}_{q,\tau} = \langle A, Q_{q,\tau}, I_{q,\tau}, \Gamma_{q,\tau}, \delta_{q,\tau}, F_{q,\tau} \rangle$ such that

- $Q_{q,\tau} = \{*\} \cup Q \times Q_e \times \{\bot, \top\}^\Gamma$,
- $I_{q,\tau} = \{(q, q_{I,e}, (\bot, \bot, \ldots, \bot))\}$,
- $\Gamma_{q,\tau} = \Gamma$,
- $F_{q,\tau} = \{*\}$,

and $\delta_{q,\tau}$ contains the following transitions:

- $(p, r, b) \xrightarrow{a,o'} (p', r', b')$ if $p \xrightarrow{a,o} p' \in \delta$, $r \xrightarrow{a} r' \in \delta_e$, and for every $c \in \Gamma$ we have:
  - $b'(c) = b(c) \vee (o(c) = \mathbf{reset})$,
  - if $b(c) = \bot$ and $\tau(c) = \leftarrow$ then $o'(c) = \mathbf{nil}$, otherwise $o'(c) = o(c)$,
- $(q, r, (\top, \top, \ldots, \top)) \xrightarrow{\epsilon, o} *$ if $r \in F_e$ and for every $c \in \Gamma$ we have $o(c) = \mathbf{reset}$ if $\tau(c) = \leftarrow$ and $o(c) = \mathbf{nil}$ otherwise.

Now we proceed with the proof that the above constructions give us the desired language M_t. First note that in both cases the constructed automata explicitly verify that a given word belongs to K_e. Therefore, $M_t \subseteq \overline{K_e}$.

We start by taking an ω-word u and its decomposition $W = w_0, w_1, \ldots$ of the type t.

5.1. **Implication** $(1) \Rightarrow (2)$. Assume that there exists an accepting run ρ of $\mathcal{A}$ over u. We want to construct a grouping $Z = z_0, z_1, \ldots$ of W such that:

S.1 for $n > 0$ we have $z_n \in K_e$,

S.2 all counters in Γ are reset by ρ in every word z_n,

S.3 the state that occurs in the run ρ at the end-points of all the words z_n is some fixed state $q \in Q$,

S.4 there exists a vector $\tau \in \{\leftarrow, \rightarrow\}^\Gamma$ such that for every counter c and every position k between successive words z_n, z_{n+1} in u we have $\mathrm{Etp}(c, \rho, k) = \tau(c)$,

S.5 the sequence of words Z is strongly convergent to some profinite word z.

The grouping Z is obtained in steps. Observe that all the above properties are preserved when taking a grouping of a sequence. S.1 is already satisfied by the sequence W. First, we group words of W in such a way to satisfy S.2 using the fact that the run ρ is accepting. Then we further group the sequence to satisfy S.3 and S.4 — some state and value of Etp must appear in infinitely many end-points. Finally, we apply Theorem 2.12 to group the sequence into a strongly convergent one.

Now, it suffices to show that $z \in M_t$. First, observe that ρ is a witness that there is a path from I to q and from q to q in $\mathcal{A}$.

We consider two cases:

Case $T = B$: Since ρ is accepting, there exists a constant l such that the values of all counters during ρ are bounded by l. We show that for every $n > 0$ we have $z_n \in L(\mathcal{A}_q \leq l)$. It implies that $z \in \overline{L(\mathcal{A}_q \leq l)}$ and therefore $z \in L(\mathcal{A}_q) \subseteq M_t$.

Observe that ρ induces a run ρ_n of $\mathcal{A}_q$ on z_n. By S.1, S.2, and S.3 we know that ρ_n is an accepting run of $\mathcal{A}_q$ — it starts in the only initial state and ends in $*$. Since $\mathcal{A}_q$ simulates all the resets of $\mathcal{A}$, we know that $\mathrm{val}(\rho_n) \leq l$ and therefore $\mathrm{val}(\mathcal{A}_q, z_n) \leq l$.

Case $T = S$: We show that for every $l \in \mathbb{N}$ the sequence Z from some point on satisfies $\mathrm{val}(\mathcal{A}_{q,\tau}, z_n) > \frac{l}{2}$. It implies that for every l we have $z \in \overline{L(\mathcal{A}_{q,\tau} > l)}$ and therefore $z \in L(\mathcal{A}_{q,\tau})$.

Since ρ is accepting, for every constant l, from some point on, all the counters are reset with a value greater than l. Assume that the last reset with the value at most l occurs before the word z_N. We show that for $n \geq N$ we have $\mathrm{val}(\mathcal{A}_{q,\tau}, z_n) > \frac{l}{2}$. Let ρ'_n be the sequence of transitions of ρ on z_n. Observe that ρ'_n induces a run ρ_n of $\mathcal{A}_{q,\tau}$ on z_n. As before, ρ_n is accepting by S.1, S.3, and S.2. Take a counter $c \in \Gamma$ and a reset of this counter r_c in ρ_n. Consider the following cases, recalling Definition 5.2:

- r_c corresponds to the first reset of c in the run ρ'_n. Since $\mathcal{A}_{q,\tau}$ did not skip r_c, $\tau(c) = \rightarrow$. Therefore, c has more increments after the beginning of z_n than before it in ρ. Therefore $\mathrm{val}(c, \rho_n, r_c) > \frac{l}{2}$.

- r_c corresponds to a reset of c in the run ρ'_n but not the first one. In that case $\mathrm{val}(c, \rho_n, r_c) = \mathrm{val}(c, \rho'_n, r_c) > l$.

- r_c is the additional reset performed by $\mathcal{A}_{q,\tau}$ at the end of the word z_n. In that case $\tau(c) = \leftarrow$ so c has greater or equal number of increments before the end of the word z_n than after it in ρ. Therefore $\mathrm{val}(c, \rho_n, r_c) > \frac{l}{2}$.

In all three cases $\mathrm{val}(c, \rho_n, r_c) > \frac{l}{2}$. So we have shown that

$$\mathrm{val}(\mathcal{A}_{q,\tau}) \geq \mathrm{val}(\rho_n) > \frac{l}{2}.$$

5.2. **Implication** $(2) \Rightarrow (3)$. This implication is trivial since strong convergence entails convergence.

5.3. **Implication** $(3) \Rightarrow (1)$. Let Z be a grouping of W such that Z converges to a limit $z \in M_t$.

 We consider two cases:

Case T = B: Since $z \in M_t$, there exists a state $q \in Q$ such that $z \in \mathrm{L}(\mathcal{A}_q)$. Therefore, $z \in \overline{\mathrm{L}(\mathcal{A}_q \leq l)}$ for some l. Since $\overline{\mathrm{L}(\mathcal{A}_q \leq l)}$ is an open set and z is a limit of Z, almost all elements of Z belong to $\mathrm{L}(\mathcal{A}_q \leq l)$. Assume that for $n \geq N$ we have $z_n \in \mathrm{L}(\mathcal{A}_q \leq l)$. Let ρ_n be a run that witnesses this fact. By the construction of $\mathcal{A}_q$, the run ρ_n induces a run ρ'_n of $\mathcal{A}$ on z_n. Also, since ρ_n is accepting, ρ'_n resets all the counters at least once.

 By the assumption about t, there exists a run ρ'_0 of $\mathcal{A}$ on z_0 that starts in some state in I and ends in q, and a sequence of runs ρ'_n on z_n for $0 < n < N$ that lead from q to q. Therefore, we can construct an infinite run ρ of $\mathcal{A}$ on u being the concatenation of the runs ρ'_n on the words z_n for $n \in \mathbb{N}$. We show that if r_c is a reset of a counter c in ρ that appears after the word z_N then $\mathrm{val}(c, \rho, r_c) \leq 2 \cdot l$. Since there are only finitely many resets of counters before the word z_N, this bound suffices to show that the run ρ is accepting.

 Observe that the increments in ρ correspond to the increments in the runs ρ_n. Also, ρ performs all the resets that appear in runs ρ_n except the resets at the end of the words. There can be at most one such skipped reset in a row because every counter is reset in every run ρ'_n. Therefore, $\mathrm{val}(c, \rho, r_c) \leq 2 \cdot l$.

Case T = S: Let q, τ be parameters such that $z \in \mathrm{L}(\mathcal{A}_{q,\tau})$. Therefore, for every $l \in \mathbb{N}$ we have $z \in \overline{\mathrm{L}(\mathcal{A}_{q,\tau} > l)}$. As Z is convergent to z and languages $\overline{\mathrm{L}(\mathcal{A}_{q,\tau} > l)}$ are open, it means that

$$\forall_l \; \exists_N \; \forall_{n \geq N} \; \mathrm{val}(\mathcal{A}_{q,\tau}, z_n) > l. \tag{5.1}$$

 As above we construct a run ρ over u that first leads on z_0 from some state of I to q and later consists of a concatenation of runs over words z_n. Let ρ'_0 be any run of $\mathcal{A}$ that leads from I to q on z_0. For $n > 0$ we pick a run ρ_n in such a way that it is accepting and[2]

$$\mathrm{val}(\rho_n) = \mathrm{val}(\mathcal{A}_{q,\tau}, z_n).$$

Observe that by (5.1), we obtain

$$\lim_{n\to\infty} \mathrm{val}(\mathcal{A}_{q,\tau}, z_n) = \lim_{n\to\infty} \mathrm{val}(\rho_n) = \infty. \tag{5.2}$$

 For $n > 0$ by ρ'_n be denote the run of $\mathcal{A}$ on z_n induced by ρ_n. Similarly as in the previous case, runs ρ'_n for $n \in \mathbb{N}$ can be combined into a run ρ of $\mathcal{A}$ on u. By the construction of $\mathcal{A}_{q,\tau}$, ρ resets every counter infinitely often.

 Let r_c be a position in u where a counter $c \in \Gamma$ is reset during ρ. Assume that r_c is contained in a word z_n and $n > 1$ — we do not care about first two words.

 Consider two cases:

$(\tau(c) = \rightarrow)$: In that case ρ performs the same increments and resets of c as the runs ρ_n. Therefore, $\mathrm{val}(c, \rho, r_c) \geq \mathrm{val}(\rho_n)$.

---

[2]Since there are only finitely many runs of an automaton on a finite word, there always exists a run realising the value $\mathrm{val}(\mathcal{A}_{q,\tau}, z_n)$, no matter whether the value is finite or not.

($\tau(c) = \leftarrow$): If r_c is not the first reset of c in ρ'_n then the value of c before r_c in ρ is the same as in ρ_n. Assume that r_c is the first reset of c in ρ'_n. Note that ρ_{n-1} performs an additional reset of c at the end of z_{n-1}. This reset does not appear in ρ so $\mathrm{val}(c, \rho, r_c) \geq \mathrm{val}(\rho_{n-1})$.

In all the cases
$$\mathrm{val}(c, \rho, r_c) \geq \min\left(\mathrm{val}(\rho_{n-1}), \mathrm{val}(\rho_n)\right),$$
so the values of c before successive resets tend to infinity by (5.2). It means that ρ is an accepting run and $u \in \mathrm{L}(\mathcal{A})$.

6. Separation for ω-languages

In this section we show the main result of the paper. The technique is to lift the separation results for T-regular languages of profinite words into the ω-word case.

Theorem 6.1. Let $\mathrm{T} \in \{\mathrm{B}, \mathrm{S}\}$. If L_1, L_2 are disjoint languages of ω-words both recognised by ωT-automata then there exists an ω-regular language L_{sep} such that
$$L_1 \subseteq L_{\mathrm{sep}} \quad \text{and} \quad L_2 \subseteq L_{\mathrm{sep}}^c.$$

Additionally, the construction is effective.

The rest of the section is devoted to showing this theorem. As observed by Thomas Colcombet, in the case of $\mathrm{T} = \mathrm{B}$ the thesis can be proved directly, without referring to profinite words. This simpler proof is presented in Section 7 with his kind permission. However, the $\mathrm{T} = \mathrm{B}$ case is also treated here for two reasons: first it reveals the symmetry and generality of the "profinite approach", second it can be used as a guideline for the more complex case of $\mathrm{T} = \mathrm{S}$.

Let $i \in \{1, 2\}$ and M^i_{trans} denote the trace-monoid for an ωT-automaton $\mathcal{A}_i$ recognising L_i. Let $h_i = h^i_{\mathcal{A}_i}$ be the canonical homomorphisms from $\widehat{A^*}$ to M^i_{trans}. Define Tp^i as the set of types $t_i = (s_i, e_i)$ in the trace-monoid M^i_{trans}.

For every type $t_i = (s_i, e_i) \in \mathrm{Tp}^i$ define $M^i_{t_i} \subseteq \widehat{A^*}$ as the T-regular language of profinite words given by Theorem 5.1 for $\mathcal{A} = \mathcal{A}_i$ and $t = t_i$. By the statement of the theorem we know that $M^i_{t_i} \subseteq \overline{h_i^{-1}(e_i)}$.

Definition 6.2. For a pair of types $t_1 = (s_1, e_1) \in \mathrm{Tp}^1$, $t_2 = (s_2, e_2) \in \mathrm{Tp}^2$, we say that t_1, t_2 are *coherent* if there exist finite words $w_s, w_e \in A^*$ such that: $h_i(w_s) = s_i$ and $h_i(w_e) = e_i$ for $i = 1, 2$.

An important application of Theorem 5.1 is the following lemma.

Lemma 6.3. If a pair of types $t_1 \in \mathrm{Tp}^1$, $t_2 \in \mathrm{Tp}^2$ is coherent then the languages $M^1_{t_1}, M^2_{t_2}$ are disjoint.

Proof. Take coherent types $t_1 = (s_1, e_1)$ and $t_2 = (s_2, e_2)$.

Assume that there exists a profinite word $w \in M^1_{t_1} \cap M^2_{t_2}$. Since $w \in \overline{h_i^{-1}(e_i)}$ for $i = 1, 2$, there exists a sequence $W = w_1, w_2, \ldots$ of finite words converging to w such that $h_1(w_n) = e_1$ and $h_2(w_n) = e_2$ for all $n > 0$. Moreover, by coherency of t_1, t_2 there exists a finite word w_0 such that $h_1(w_0) = s_1$ and $h_2(w_0) = s_2$. Let $u = w_0 w_1 w_2 \ldots$ We show that $u \in L_1 \cap L_2$ — a contradiction.

Take $i \in \{1, 2\}$. Observe that $u = w_0 w_1 \ldots$ is a decomposition of u of h_i-type t_i. Additionally observe that the sequence W converges to w and w belongs to $M_{t_i}^i$. So, by Theorem 5.1 we have $u \in L_i$. $\quad\square$

Take a pair of coherent types t_1, t_2. Since the languages $M_{t_1}^1, M_{t_2}^2$ are disjoint, we can use Theorem 4.1 to find a separating profinite-regular language $\overline{R_{t_1,t_2}} \subseteq \widehat{A^*}$ such that

$$M_{t_1}^1 \subseteq \overline{R_{t_1,t_2}} \quad \text{and} \quad M_{t_2}^2 \subseteq \overline{R_{t_1,t_2}}^c.$$

Now we can introduce the ω-regular language L_{sep} separating L_1, L_2.

Definition 6.4. Consider a coherent pair of types (t_1, t_2). Let S_{t_1,t_2} be defined as follows: S_{t_1,t_2} is the language of ω-words u such that there exists a decomposition $u = w_0 w_1 \ldots$ of types t_1, t_2 with respect to h_1, h_2, such that every grouping of $(w_n)_{n \in \mathbb{N}}$ from some point on belongs to the regular language R_{t_1,t_2}.

Note that the above definition can be expressed in MSO so S_{t_1,t_2} is an ω-regular language.

Let L_{sep} be the ω-regular language defined as

$$L_{\text{sep}} = \bigcup_{t_1,t_2} S_{t_1,t_2},$$

where the sum ranges over pairs of coherent types.

Clearly L_{sep} is an ω-regular language. What remains is to show the following lemma.

Lemma 6.5. The language L_{sep} separates L_1 and L_2.

Proof. First observe that $L_1 \subseteq L_{\text{sep}}$. Take $u \in L_1$. We want to construct a decomposition $W = w_0, w_1, \ldots$ of u such that:

- the h^i-type of W is t_i for $i = 1, 2$ and some pair of coherent types (t_1, t_2) in $\text{Tp}^1 \times \text{Tp}^2$,
- the sequence W is strongly convergent to some profinite word $w \in \widehat{A^*}$.

The sequence W is obtained in steps. First we use Theeorem 2.7 to find a decomposition of u with respect to both monoids $M_{\text{trans}}^1, M_{\text{trans}}^2$ at the same time. Such decomposition satisfies the first bullet above. Then, using Theorem 2.12, we can group our sequence into W in such a way that W is strongly convergent.

By Theorem 5.1, there exists a grouping Z of W that converges to a profinite word $z \in M_{t_1}^1 \subseteq \overline{R_{t_1,t_2}}$. But since W is strongly convergent, $z = w$. Therefore, by the strong convergence of W, every grouping of W converges to $w \in \overline{R_{t_1,t_2}}$. So every grouping of u from some point on belongs to R_{t_1,t_2} as in the definition of L_{sep}. Therefore, $u \in L_{\text{sep}}$.

Now we show that $L_2 \cap L_{\text{sep}} = \emptyset$. Assume otherwise, that there exists an ω-word $u \in L_2 \cap L_{\text{sep}}$. Since $u \in L_{\text{sep}}$, there exists a coherent pair of types t_1, t_2 such that $u \in S_{t_1,t_2}$. Therefore, u can be decomposed as $u = w_0 w_1 \ldots$ of types t_1, t_2 respectively. Let $W = w_0, w_1, \ldots$ Because $u \in L_2$ so by Theorem 5.1 there exists a grouping Z of W with a limit $z \in M_{t_2}^2$. But by the definition of S_{t_1,t_2} almost all words in Z belong to R_{t_1,t_2} so $z \in \overline{R_{t_1,t_2}}$. Since $\overline{R_{t_1,t_2}} \cap M_{t_2}^2 = \emptyset$, we have the required contradiction. $\quad\square$

Now we can deduce the corollary from the introduction.

Corollary 6.6. If a given language of ω-words L and its complement L^c are both ωB-regular (resp. ωS-regular) then L is (effectively) ω-regular.

Proof. Let L be a language of ω-words such that L and L^c are both ωT-regular. By Theorem 6.1 there exists an ω-regular language L_{sep} that separates L and L^c. But in that case $L_{\text{sep}} = L$ so L is ω-regular.   □

7. A direct proof of separation for ωB-regular languages

As observed by Thomas Colcombet, the separation property for ωB-regular languages can be shown directly, without referring to profinite words. This simpler proof is presented here with his kind permission.

Let $\mathrm{L}(\mathcal{A})$ be an ωB-regular language recognised by an ωB-automaton $\mathcal{A}$. Consider a Büchi automaton $\mathcal{A}'$ obtained from $\mathcal{A}$ by removing all the counter operations (similarly to Lemma 4.5) and requiring that every counter is reset infinitely often.

Clearly, the language recognized by $\mathcal{A}'$ is ω-regular and $\mathrm{L}(\mathcal{A}) \subseteq \mathrm{L}(\mathcal{A}')$.

Claim 7.1. If $u = wzz\ldots$ is an ultimately periodic ω-word in $\mathrm{L}(\mathcal{A}')$ then $u \in \mathrm{L}(\mathcal{A})$.

Proof. Observe that $\mathcal{A}'$ has an ultimately periodic accepting run ρ on u. By the acceptance condition of $\mathcal{A}'$, every counter c of $\mathcal{A}$ is reset infinitely often during ρ. Since ρ is ultimately periodic, the values of the counter c are bounded in ρ. Therefore, $u \in \mathrm{L}(\mathcal{A})$.   □

It means that, since an ω-regular language is entirely defined by the ultimately periodic ω-words it contains [Büc62], $\mathrm{L}(\mathcal{A}')$ is the least ω-regular language that contains $\mathrm{L}(\mathcal{A})$. It also means that it depends only on $\mathrm{L}(\mathcal{A})$ but not on the specific automaton $\mathcal{A}$ that recognizes it. Let us call this language $\text{Closure}(\mathrm{L}(\mathcal{A}))$.

Consider now two ωB-regular languages of empty intersection L_1 and L_2. Assume $\text{Closure}(L_1)$ intersects $\text{Closure}(L_2)$ then, since these languages are ω-regular, there is an ultimately periodic ω-word in this intersection. But according to Claim 7.1, this ultimately periodic ω-word belongs to both L_1 and L_2. A contradiction.

It follows that if L_1 and L_2 are disjoint then $\text{Closure}(L_1)$ (respectively $\text{Closure}(L_2)$) are separators. Also, this construction shows that in order to construct a separator of two ωB-regular languages, only one language needs to be known.

8. Acknowledgements

References

[Alm03]   Jorge Almeida. Profinite semigroups and applications. In *Structural Theory of Automata, Semi-groups, and Universal Algebra*, pages 7–18, 2003.

[AMN12]   André Arnold, Henryk Michalewski, and Damian Niwiński. On the separation question for tree languages. In *STACS*, pages 396–407, 2012.

[BC06]   Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In *LICS*, pages 285–296, 2006.

[BKT12]   Mikołaj Bojańczyk, Eryk Kopczyński, and Szymon Toruńczyk. Ramseys theorem for colors from a metric space. *Semigroup Forum*, 85:182–184, 2012.

[Büc62]   Julius Richard Büchi. On a decision method in restricted second-order arithmetic. In *Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.

[CL10]   Thomas Colcombet and Christof Löding. Regular cost functions over finite trees. In *LICS*, pages 70–79, 2010.

[Col09]   Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *ICALP (2)*, pages 139–150, 2009.

[Col13]   Thomas Colcombet. Fonctions régulières de coût. Habilitation thesis, Université Paris Diderot—Paris 7, 2013.

[HMN09]   Szczepan Hummel, Henryk Michalewski, and Damian Niwiński. On the Borel inseparability of game tree languages. In *STACS*, pages 565–575, 2009.

[Kec95]   Alexander Kechris. *Classical descriptive set theory*. Springer-Verlag, New York, 1995.

[KV99]   Orna Kupferman and Moshe Y. Vardi. The weakness of self-complementation. In *STACS*, pages 455–466, 1999.

[Pin09]   Jean-Éric Pin. Profinite methods in automata theory. In *STACS*, pages 31–50, 2009.

[PP04]   Dominique Perrin and Jean-Éric Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Elsevier, 2004.

[Rab70]   Michael O. Rabin. Weakly definable relations and special automata. In *Proceedings of the Symposium on Mathematical Logic and Foundations of Set Theory*, pages 1–23. North-Holland, 1970.

[Tor11]   Szymon Toruńczyk. *Languages of profinite words and the limitedness problem*. PhD thesis, University of Warsaw, 2011.

[Tor12]   Szymon Toruńczyk. Languages of profinite words and the limitedness problem. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *ICALP (2)*, volume 7392 of *Lecture Notes in Computer Science*, pages 377–389. Springer, 2012.