Logical Methods in Computer Science
Vol. 12(1:5)2016, pp. 1–30
www.lmcs-online.org

Submitted Jun. 4, 2014
Published Mar. 9, 2016

# SEPARATING REGULAR LANGUAGES WITH FIRST-ORDER LOGIC

THOMAS PLACE AND MARC ZEITOUN

LaBRI, Bordeaux University, France
*e-mail address*: {thomas.place, marc.zeitoun}@labri.fr

Abstract. Given two languages, a separator is a third language that contains the first one and is disjoint from the second one. We investigate the following decision problem, called *separation*: given two regular languages of finite words, decide whether there exists a first-order definable separator. A more general problem was solved in an algebraic framework by Henckell in 1988, although the connection with separation was pointed out only in 1996, by Almeida. The result was then generalized by Henckell, Steinberg and Rhodes in 2010. In this paper, we present a new, self-contained and elementary proof of it, which actually covers the original result of Henckell.

We prove that in order to answer this question, sufficient information can be extracted from semigroups recognizing the input languages, using a fixpoint computation, similar to that originally proposed by Henckell. Given as input a morphism recognizing both languages to be separated, this yields an Exptime algorithm for checking first-order separability. Moreover, the correctness proof of this algorithm yields a stronger result, namely a description of a possible separator. More precisely, one can compute a bound on the quantifier rank of potential separators, as well as a first-order formula that describes a separator, if there exists one. Finally, we prove that this technique can be generalized to answer the same question for regular languages of infinite words.

## 1. Introduction

In this paper, we investigate a decision problem on word languages: the *separation problem*. The problem is parametrized by a class $\mathsf{Sep}$ of *separator languages* and is as follows: given as input two regular word languages, decide whether there exists a third language in $\mathsf{Sep}$ containing the first language while being disjoint from the second one.

More than the decision procedure itself, the primary motivation for investigating this type of problem is the insight it gives on the class $\mathsf{Sep}$. Indeed, the separation problem is a generalization of the *membership problem*, which is often considered as the right approach to understand the expressive power of a class of languages. In this restricted problem, one

*2012 ACM CCS:* [**Theory of computation**]: Formal languages and automata theory—Regular languages.

*Key words and phrases:* Words, Infinite Words, Regular Languages, Semigroups, First-Order Logic, Expressive Power, Ehrenfeucht-Fraïssé games, Separation.

Supported by ANR 2010 BLAN 0202 01 FREC. This study has also been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux – CPU (ANR-10-IDEX-03-02).


DOI:10.2168/LMCS-12(1:5)2016

only needs to decide whether a single input regular language already belongs to the class Sep under investigation. Intuitively, in order to get such a decision procedure, one has to consider *all* regular languages simultaneously, which requires a strong understanding of the *expressive power* of Sep. Since regular languages are closed under complement, testing membership can be achieved by testing whether the input is separable from its complement. Therefore, membership can be reduced to separation, which makes separation more general.

It turns out that separation is actually strictly more general than membership and solving it requires a deeper understanding of the class Sep. More than the expressive power, it requires an understanding of the *discriminating power* of Sep. This means that while intrinsically more difficult, solving the separation problem is also more rewarding than solving the membership problem. In both cases, the problem amounts to finding a language in Sep. However, in the membership case, there is only one candidate, which is already known: the input. Therefore, we start with a fixed recognizing device for this unique candidate and powerful tools are available, *viz.* the syntactic semigroup of the language, which is now accepted as the natural tool for solving the membership problem for word languages. In the separation case, there can be infinitely many candidates as separators, which means that there is no fixed recognition device that we can use. An even harder question then is to actually construct a separator language in Sep.

Investigating the deeper separation problem can also be relevant when a pure membership approach fails. Many natural classes of languages are built on top of weaker classes. For example, in logic, more powerful classes can be built on top of weaker ones by adding predicates to the signature. When investigating the membership problem, a natural approach would be to first obtain a solution for the weaker class and then to transfer it to the extended one. However, this approach fails in general. Actually, many extensions of classes of languages are known not to preserve decidability of membership [1, 25, 4]. The reason is that such a transfer result requires more information on the original class than what a solution to membership provides. This makes the deeper separation problem a more promising setting as already noted in [3, 29]. A recent example is the quantifier alternation hierarchy of first-order logic: in [23], it was proved that solving the separation problem for level $i$ in this hierarchy yields the solution for the membership problem at level $(i+1)$.

**First-order logic.** In this paper, we choose Sep as the class of *languages definable by first-order sentences* (*i.e.*, sets of words that satisfy some first-order sentence). In this context, the separation problem can be rephrased as follows: given two regular languages as input, decide whether there exists a first-order sentence that is satisfied by all words of the first language, and by no word of the second one. Thus, such a formula witnesses that the input languages are disjoint.

Within monadic second order logic, which defines on finite words all regular languages, first-order logic is often considered as the yardstick. It is a robust class having several characterizations [10]. It corresponds to star-free languages, and has the same expressive power as linear temporal logic [13]. In particular, it was the first natural class for which the membership problem was proved to be decidable. This result, known as Schützenberger's theorem [27, 15], served as a template and a starting point of a line of research that successfully solved the membership problem for a wide assortment of classes of regular languages. This makes first-order logic the natural candidate to serve as the example for devising a general approach to the separation problem.

Schützenberger's theorem states that first-order definable languages are exactly those whose syntactic semigroup is aperiodic, *i.e.*, has only trivial subgroups. Since the syntactic

semigroup of a language is computable and aperiodicity is a decidable property, this yields a decision procedure for membership. Schützenberger's original proof has been refined over the years. Our own proof for separation by first-order logic actually generalizes a more recent proof by Wilke [36]. Similar results [33, 16] make it possible to decide first-order definability for languages of infinite words, or finite or infinite Mazurkiewicz traces. See [10] for a survey.

**Contributions and main ideas.** We obtain our separation algorithm for first-order logic by relying on a specific framework. A key idea is that, in order to separate two regular languages with first-order logic, one needs to consider more languages than just these two. One has to consider a single morphism from $A^+$ into a finite semigroup $S$ that recognizes them both and solve the separation problem simultaneously for all pairs of languages that are recognized by it. Indeed, the set of all languages recognized by a semigroup morphism has structure: considering them all as a whole allows us to exploit this structure.

More precisely, our framework is designed to reduce separation to the following more general problem. Given a morphism $\alpha$ from $A^+$ into a finite semigroup $S$, we want to construct an FO-partition of $A^+$ (a **finite** partition of $A^+$ into first-order definable languages) that is an "optimal approximation" of the languages recognized by $\alpha$. The main point is that a necessary condition for an FO-partition to be optimal (for $\alpha$) is that any two recognized languages are FO-separable *if and only if* they can be separated by a language built as a union of languages in the FO-partition (however this condition is not sufficient, which is why this problem is more general).

Our solution is presented as follows. First, we obtain a fixpoint algorithm that, given a morphism $\alpha$ as input, computes an object that we call the *optimal imprint with respect to FO on* $\alpha$. Intuitively, this "optimal imprint" contains information about the FO-partitions that are optimal for $\alpha$. In fact, this information includes which pairs of languages recognized by $\alpha$ are FO-separable. In other words, this yields a decision procedure for the separation problem associated to FO. This fixpoint algorithm is complemented by a generic technique for constructing optimal FO-partitions by induction (this is actually a byproduct of the correctness proof of the fixpoint algorithm). This is of particular interest as this yields an inductive way to build first-order separators when they exist.

An important observation is that the "*optimal imprint with respect to FO on* $\alpha$" that our fixpoint algorithm computes is actually an alternate definition of a previously known notion: the so-called *aperiodic pointlike sets* (whose original definition is algebraic and very different from the one we use in this paper). While we never use this fact in the paper, it connects our results to those of Henckell [11] (see also [12, 26] which answers the problem for even more general classes). Indeed, in [11], Henckell does not consider the separation problem: his main objective is to find an algorithm that computes these aperiodic pointlike sets. In fact, the connection between the separation problem and the pointlike sets was only observed later by Almeida [2][1]. Hence, our fixpoint algorithm and its proof can be viewed as a new proof of Henckell's result: one can compute the aperiodic pointlike sets of a semigroup.

Note however that our approach is vastly different from that of Henckell. In particular, it is more rewarding with respect to the separation problem. Indeed, the motivations and the proofs of [11, 12] are purely algebraic and provide no intuition on the underlying logic. Our contributions differ from those of [11, 12] in several ways.

[1]This connection is the analogue of the equivalence (1) $\iff$ (2) in our Theorem 4.6.

- First, we give a new and self-contained proof that the separation problem by first-order languages is decidable. It is independent from those of [11, 12], and relies on elementary ideas and notions from language theory only, making it accessible to computer scientists. We do not use any involved construction from semigroup theory: we work directly with the logic itself. As mentioned above, the proof refines Wilke's membership algorithm [36].
- Second, when the input languages are separable, our approach makes it possible to inductively compute a first-order formula that defines a separator: we have a generic way to construct optimal FO-partitions. In addition, we provide a bound on the *expected quantifier rank* of a potential separator.
- Third, as a consequence of our algorithm, we obtain an EXPTIME upper bound (while complexity is not investigated in [11], a rough analysis yields an EXPSPACE upper bound).
- Finally, the techniques of [11, 12] are tailored to work with finite words only. We also solve the separation problem for languages of *infinite words* by first-order definable languages, by a smooth extension of our techniques.

Since we do not follow the proofs of [11, 12], it is not surprising that we obtain a different algorithm. However, we are able to derive two variations of it, which allows us to give an alternate and elementary correctness proof of Henckell's original algorithms.

**Related work.** First-order logic has a number of important fragments. The separation question makes sense when choosing such natural subclasses as classes of separators. It has already been solved for the case of local fragments, such as locally testable (LT) and locally threshold testable languages (LTT), although the problem is already NP-hard starting from two DFAs as input, while membership is known to be polynomial [5]. The algebraic varieties associated to the classes LT and LTT in Eilenberg's correspondence are well-known, namely the class $\mathsf{LSl}$ of all finite local semilattices, and the semidirect product $\mathsf{Acom} * \mathsf{D}$ of commutative and aperiodic semigroups with right zero semigroups. Using these correspondences and the algebraic interpretation of separation given in [2], algebraic proofs were given, both for LT [8, 7] and for LTT, via [5, 31, 28, 30]. Although indirect, these proofs actually provide more information than what is needed for separation alone. A direct and elementary approach for both classes was also presented in [20, 22].

The separation problem is also decidable for the fragment of first-order logic made of Boolean combinations of $\Sigma_1(<)$ sentences (that is, first order sentences without any quantifier alternation), as a consequence of [2, 3], and then obtained directly and independently in [9, 21]. It has then been shown to be decidable for the first fragments of first-order logic in the quantifier alternation hierarchy, namely the ones consisting of $\Sigma_2(<)$ [23], respectively of $\Sigma_3(<)$ sentences [19] (*i.e*, first order sentences of the form $\exists^*\forall^*\varphi$, respectively of the form $\exists^*\forall^*\exists^*\varphi$, with $\varphi$ quantifier-free). In view of the aforementioned transfer result, this yields decidability of membership for the next level, $\Sigma_4(<)$. Within this hierarchy, membership remains open for level 5 and above (hence, separation is open for level 4 and above).

Finally, the problem has also been investigated for the fragment $\mathrm{FO}^2(<)$ of first-order logic using 2 variables only, and again has been proven to be decidable [21].

**Paper outline.** We first give the necessary definitions and terminology: languages and semigroups for finite words are defined in Section 2 and first-order logic is defined in Section 3. Section 4 is devoted to the presentation of our algorithm solving first-order separation through the computation of sets that cannot be distinguished by first-order logic. Sections 5 and 6 are devoted to proving the soundness and completeness of this algorithm, respectively. In Section 7, we present alternate versions of our algorithm. In Section 8, we recall the

preliminary definitions for tackling the separation problem in the setting of infinite words. In Section 9, we state a generalization to infinite words of our algorithm, for which we prove soundness in Section 10.1 and completeness in Section 10.2.

This paper is the journal version of [24]. From the conference version, the missing proofs have been added, separation is now presented in a generic, language-theoretic setting, and the proof of the algorithm has been entirely rewritten so that it now constructs an actual separator by induction when it exists.

## 2. Preliminaries

In this section, we provide terminology for words, semigroups and languages. All the definitions are for finite words. We delay the definitions for infinite words to Section 8.

**Semigroups.** A semigroup is a set $S$ equipped with an associative operation $s \cdot t$ (often written $st$). A monoid is a semigroup $S$ having an identity element $1_S$, *i.e.*, such that $s \cdot 1_S = 1_S \cdot s = s$ for all $s \in S$. Finally, a group is a monoid such that every element $s$ has an inverse $s^{-1}$, *i.e.*, such that $s \cdot s^{-1} = s^{-1} \cdot s = 1_S$.

Given a *finite* semigroup $S$, it is folklore and easy to see that there is an integer $\omega(S)$ (denoted by $\omega$ when $S$ is understood) such that for all $s$ in $S$, $s^\omega$ is idempotent: $s^\omega = s^\omega s^\omega$.

**Words, Languages, Morphisms.** We fix a finite alphabet $A$. We denote by $A^+$ the set of all nonempty finite words and by $A^*$ the set of all finite words over $A$. If $u, v$ are words, we denote by $u \cdot v$ or by $uv$ the word obtained by the concatenation of $u$ and $v$. Observe that $A^+$ (resp. $A^*$) equipped with the concatenation operation is a semigroup (resp. a monoid).

For convenience, we only consider languages that do not contain the empty word. That is, a language is a subset of $A^+$ (this does not affect the generality of the argument). We work with regular languages, *i.e.*, languages definable by *nondeterministic finite automata* (NFA).

We shall exclusively work with the algebraic representation of regular languages in terms of semigroups. We say that a language $L$ *is recognized by a semigroup* $S$ if there exists a semigroup morphism $\alpha : A^+ \to S$ and a subset $F \subseteq S$ such that $L = \alpha^{-1}(F)$. It is well known that a language is regular if and only if it can be recognized by a *finite* semigroup. Moreover, from any NFA recognizing some language $L$, one can compute a canonical semigroup recognizing $L$, called the *syntactic semigroup* of $L$.

When working on separation, we consider as input two regular languages $L_0, L_1$. It will be convenient to have a single semigroup recognizing both of them, rather than having to deal with two objects. Let $S_0, S_1$ be semigroups recognizing $L_0, L_1$ together with the associated morphisms $\alpha_0, \alpha_1$, respectively. Then, $S_0 \times S_1$ equipped with the componentwise multiplication $(s_0, s_1) \cdot (t_0, t_1) = (s_0 t_0, s_1 t_1)$ is a semigroup that recognizes both $L_0$ and $L_1$ with the morphism $\alpha : w \mapsto (\alpha_0(w), \alpha_1(w))$. From now on, we work with such a single semigroup recognizing both languages, and we call $\alpha$ the associated morphism.

**Semigroup of Subsets.** As explained in the introduction, our separation algorithm works by computing special subsets of a semigroup recognizing both input languages. Intuitively, these subsets are those that cannot be distinguished by first-order logic. More precisely, by *special subset*, we mean that any first-order definable language has an image under $\alpha$ that either contains *all* elements of the subset, or *none* of them. For this reason, we work with the semigroup of subsets. Let $S$ be a semigroup. Observe that the set $2^S$ of subsets of $S$

equipped with the operation

$$T \cdot T' = \{s \cdot s' \mid s \in T, \quad s' \in T'\}$$

is a semigroup, that we call the *semigroup of subsets of* $S$. Note that $S$ can be viewed as a subsemigroup of $2^S$, since $S$ is isomorphic to the semigroup $\big\{\{s\} \mid s \in S\big\} \subseteq 2^S$. We denote by $\mathcal{R}, \mathcal{S}, \mathcal{T}, \ldots$ subsemigroups of a semigroup of subsets.

**Downset $\downarrow \mathcal{S}$.** Let $\mathcal{S} \subseteq 2^S$ be any subset of $2^S$. We define the *downset* $\downarrow \mathcal{S}$ of $\mathcal{S}$ as

$$\downarrow \mathcal{S} = \{T \in 2^S \mid \exists T' \in \mathcal{S}, \quad T \subseteq T'\}.$$

Clearly, we have $\mathcal{S} \subseteq \downarrow \mathcal{S}$. Moreover, if $\mathcal{S}$ is a subsemigroup of $2^S$, it is easy to check that $\downarrow \mathcal{S}$ is a subsemigroup as well.

**Union $\llbracket \mathcal{S} \rrbracket$.** For $\mathcal{S} \subseteq 2^S$ any subset of $2^S$, we define $\llbracket \mathcal{S} \rrbracket \subseteq S$, the *union* of $\mathcal{S}$, as the set

$$\llbracket \mathcal{S} \rrbracket = \bigcup_{T \in \mathcal{S}} T \subseteq S$$

We call *index* of $\mathcal{S}$ the size of its union, *i.e.*, $|\llbracket \mathcal{S} \rrbracket|$. By definition, we have the following fact.

**Fact 2.1.** *Set $\mathcal{S} \subseteq 2^S$ and $\mathcal{T} \subseteq 2^S$, then $\llbracket \mathcal{S} \cdot \mathcal{T} \rrbracket = \llbracket \mathcal{S} \rrbracket \cdot \llbracket \mathcal{T} \rrbracket$.*

## 3. First-Order Logic and Separation

This section is devoted to the definition of first-order logic on words. See [34, 10, 32] for details on this classical notion.

**First-Order Logic.** We view words as logical structures composed of a sequence of positions labeled over $A$. We denote by $<$ the linear order over the positions. We work with first-order logic FO($<$) using a unary predicate $a(x)$ for each $a \in A$, which selects positions $x$ labeled with an $a$, as well as a binary predicate for the linear order $<$. A language $L$ is said to be *first-order definable* if there exists an FO($<$) formula $\varphi$ such that $L = \{w \in A^+ \mid w \models \varphi\}$. We write FO for the class of all first-order definable languages.

There are many known characterizations of the class of first-order definable languages. Kamp's Theorem [13] states that it is exactly the class of languages definable in linear temporal logic LTL. In [15], it was proven that this is also the class of languages that can be recognized with *counter-free automata* as well as the class of star-free languages (*i.e.*, languages definable by a regular expression that may use complement, but does not use the Kleene star). This result bridged the gap with Schützenberger's Theorem [27], which characterizes star-free languages as those whose syntactic semigroup is *aperiodic* (*i.e.*, all its elements $s$ satisfy the equality $s^\omega = s^{\omega+1}$). The separation algorithm that we present in this paper can be viewed as a generalization of Schützenberger's Theorem. In particular, we reprove this theorem as a simple corollary of our algorithm. Note that conversely, using Schützenberger's result as a black box doesn't seem to help much to obtain a simpler proof with our approach.

**Separation**. Given languages $L, L_0, L_1$, we say that $L$ *separates* $L_0$ from $L_1$ if

$$L_0 \subseteq L \text{ and } L_1 \cap L = \varnothing.$$

Given a class of languages $\mathcal{C}$, the pair $(L_0, L_1)$ is said to be $\mathcal{C}$-*separable* if some language $L \in \mathcal{C}$ separates $L_0$ from $L_1$. Note that when $\mathcal{C}$ is closed under complementation (for example

when $\mathcal{C} = \mathrm{FO}(<)$), $(L_0, L_1)$ is $\mathcal{C}$-separable if and only if $(L_1, L_0)$ is. Therefore, we simply say that $L_0$ and $L_1$ are $\mathcal{C}$-separable in this case.

In this paper, we present an algorithm that decides whether two regular languages are FO-separable. Let us give an example of two languages that are not FO-separable.

**Example 3.1.** Let $K_0 = (aa)^*$, $K_1 = (aa)^*a$ and

$$L_0 = (bK_0bK_1)^+,$$
$$L_1 = (bK_0bK_1)^*bK_0.$$

It is well known that $a^{2^k}$ and $a^{2^k-1}$ cannot be distinguished by any FO-sentence of quantifier rank $k$, see *e.g.* [32] (recall here that the *quantifier rank* of a first-order formula $\varphi$ is the length of the largest sequence of nested quantifiers in $\varphi$ — the rank is a usual way to classify first-order formulas). Therefore, $K_0$ and $K_1$ are not FO-separable. Reusing this argument then shows that $L_0$ and $L_1$ are not FO-separable either. We shall explain below how this is detected by our algorithm.

The main result of the paper is the following theorem.

**Theorem 3.2.** *Let $L_0, L_1$ be two regular languages recognized by a morphism $\alpha : A^+ \to S$ into a finite semigroup. The two following items hold.*

1. *One can decide in* Exptime *with respect to $|S|$ whether $L_0$ and $L_1$ are FO-separable.*
2. *When $L_0$ and $L_1$ are FO-separable, one can construct an actual FO-separator defined by a formula of quantifier rank at most $|A|2^{|S|^2}$.*

The proof of Theorem 3.2 is postponed to Sections 4, 5 and 6. In Section 4 we present our decision procedure and Section 5 and 6 are devoted to proving soundness and completeness of this procedure. Note that most of our efforts are aimed to obtaining an algorithm for Item 1 of the theorem. We actually obtain the second item as a byproduct of the completeness proof of Section 6: this proof is constructive and can be used to build an actual separator by induction (which turns out to have rank at most $|A|2^{|S|^2}$), when it exists.

## 4. Separation Algorithm

In this section, we define a general framework which is tailored to the investigation of the separation problem. We then use it to obtain a separation algorithm in the special case when the class of separators is given by first-order logic, *i.e.*, to prove Theorem 3.2. An important remark is that the problem that we actually consider and solve is slightly more general than separation. In particular, this problem takes an arbitrary number of languages as input rather than just two. Let us first explain our motivation for considering such a generalization.

In the separation problem, we are given a single semigroup morphism $\alpha$ that recognizes the two input languages $L_0, L_1$ that need to be separated. However, in general, a single morphism recognizes several different languages, not just these two. Moreover, while these other languages are not the ones we aim at separating, $L_0$ and $L_1$ are built-up from them. This makes all these languages relevant when working with $L_0$ and $L_1$. Therefore, our approach is to consider them all simultaneously in a problem that generalizes separation: *computing an FO-partition that is optimal* for the morphism $\alpha$.

We organize the section in three parts. First, we present our framework in a general context (*i.e.*, for an arbitrary class of separators $\mathcal{C}$) and connect it to the separation problem. In the second part, we apply this framework to first-order logic and use it to obtain a separation algorithm and to prove Theorem 3.2. Finally, in the third part, we illustrate this algorithm on Example 3.1.

4.1. **Definition.** For the definitions, we assume that an arbitrary class of languages $\mathcal{C}$ over our fixed alphabet $A^+$ is fixed. Moreover, we need $\mathcal{C}$ to satisfy the three following properties:

(1) $\mathcal{C}$ is nonempty and closed under boolean operations.
(2) $\mathcal{C}$ is closed under right and left quotients: for any $w \in A^+$ and $L \in \mathcal{C}$, we have

$$w^{-1}L \stackrel{\text{def}}{=} \{u \in A^+ \mid wu \in L\} \in \mathcal{C} \quad \text{and} \quad Lw^{-1} \stackrel{\text{def}}{=} \{u \in A^+ \mid uw \in L\} \in \mathcal{C}.$$

(3) $\mathcal{C}$ only consists of regular languages.

It is straightforward to verify that FO satisfies these three properties. Note that the objects that we define below make sense even when $\mathcal{C}$ does not satisfy these three properties. However, we will need these properties to make the connection with the separation problem.

**$\mathcal{C}$-Partitions and Imprints.** Assume that an alphabet $A$ is fixed. A *$\mathcal{C}$-partition* (of $A^+$) is a **finite** partition $\mathbf{K} = \{K_1, \ldots, K_m\}$ of $A^+$ such that all languages $K_i$ in $\mathbf{K}$ belong to $\mathcal{C}$. Note that since $\mathcal{C}$ is non-empty and closed under boolean operations, $A^+$ belongs to $\mathcal{C}$. Therefore, there exists at least one $\mathcal{C}$-partition, namely $\{A^+\}$.

When we have a morphism $\alpha : A^+ \to S$ and a $\mathcal{C}$-partition $\mathbf{K}$ in hand, our main interest will be to know how good $\mathbf{K}$ is at separating languages recognized by $\alpha$: what are the languages recognized by $\alpha$ that can be separated by a union of languages in $\mathbf{K}$? This information is captured by a new object that we associate to each $\mathcal{C}$-partition and each morphism, the *imprint of the partition on the morphism*.

Given a morphism $\alpha : A^+ \to S$ into a finite semigroup $S$ and a $\mathcal{C}$-partition $\mathbf{K}$. The *imprint of* $\mathbf{K}$ *on* $\alpha$ is defined as the set

$$\mathcal{I}[\alpha](\mathbf{K}) = \{T \in 2^S \mid \text{there exists } K \in \mathbf{K} \text{ such that } T \subseteq \alpha(K)\}.$$

In other words, $T \in \mathcal{I}[\alpha](\mathbf{K})$ if and only if there exists a language in $\mathbf{K}$ that intersects $\alpha^{-1}(t)$ for all $t \in T$. Observe that by definition, an imprint on $\alpha$ is a subset of $2^S$. Hence, since $S$ is finite, there are finitely many possible imprints on $\alpha$. We present three simple properties of imprints. The first one states that an imprint always contains some trivial elements.

**Fact 4.1.** Let $\alpha : A^+ \to S$ be a morphism into a finite semigroup $S$ and $\mathbf{K}$ be a $\mathcal{C}$-partition. Then, $\{\{\alpha(w)\} \mid w \in A^+\} \subseteq \mathcal{I}[\alpha](\mathbf{K})$.

*Proof.* For any $w \in A^+$, there exists $K \in \mathbf{K}$ such that $w \in K$ ($\mathbf{K}$ is a partition of $A^+$). Hence, $\{\alpha(w)\} \subseteq \alpha(K)$ and $\{\alpha(w)\} \in \mathcal{I}[\alpha](\mathbf{K})$. □

The second property states that any imprint is closed under downset.

**Fact 4.2.** Let $\alpha : A^+ \to S$ be a morphism into a finite semigroup $S$ and $\mathbf{K}$ be a $\mathcal{C}$-partition. Then $\mathcal{I}[\alpha](\mathbf{K}) = {\downarrow}\mathcal{I}[\alpha](\mathbf{K})$.

*Proof.* By definition, $\mathcal{I}[\alpha](\mathbf{K}) \subseteq {\downarrow}\mathcal{I}[\alpha](\mathbf{K})$. Let us prove the converse inclusion. Set $T \in {\downarrow}\mathcal{I}[\alpha](\mathbf{K})$. By definition, there exists $T' \in \mathcal{I}[\alpha](\mathbf{K})$ such that $T \subseteq T'$. By definition of imprints, we obtain $K \in \mathbf{K}$ such that $T' \subseteq \alpha(K)$. Therefore, $T \subseteq T' \subseteq \alpha(K)$ and $T \in \mathcal{I}[\alpha](\mathbf{K})$. Note that we have shown that $\mathcal{I}[\alpha](\mathbf{K}) = {\downarrow}\alpha(\mathbf{K})$. □

The third property connects imprints to the separation problem: the imprint of $\mathbf{K}$ on $\alpha$ records which languages recognized by $\alpha$ can be separated with $\mathbf{K}$.

**Lemma 4.3.** *Let $\alpha : A^+ \to S$ be a morphism into a finite semigroup $S$ and $\mathbf{K}$ be a $\mathcal{C}$-partition. Let $L_1, L_2$ be two languages recognized by $\alpha$ and let $T_1, T_2 \subseteq S$ be the corresponding accepting sets. The two following conditions are equivalent:*
(1) *for all $t_1 \in T_1$ and all $t_2 \in T_2$, we have $\{t_1, t_2\} \notin \mathcal{I}[\alpha](\mathbf{K})$.*
(2) *$L_1$ and $L_2$ can be separated by a union of languages in $\mathbf{K}$.*

*Proof.* Assume first that Item (1) holds and set $K = \bigcup_{\{K' \in \mathbf{K} | K' \cap L_1 \neq \varnothing\}} K'$. Since $\mathbf{K}$ is a partition of $A^+$, we have $L_1 \subseteq K$ by definition. Moreover, we know from Item (1) that no language $K' \in \mathbf{K}$ intersects both $L_1$ and $L_2$. It follows that $K \cap L_2 = \varnothing$: $K$ separates $L_1$ from $L_2$ and Item (2) holds.

Assume now that Item (2) holds. Since $\mathbf{K}$ is a partition, this means that no language $K \in \mathbf{K}$ intersects both $L_1$ and $L_2$. It follows from the definition of imprints that for all $t_1 \in T_1$ and all $t_2 \in T_2$, we have $\{t_1, t_2\} \notin \mathcal{I}[\alpha](\mathbf{K})$. □

An important remark is that, in general, the imprint of $\mathbf{K}$ on $\alpha$ contains more than just separation related information. For example, assume that $S = \{s_1, s_2, s_3\}$ and consider two $\mathcal{C}$-partitions $\mathbf{K}$ and $\mathbf{K}'$ having the following imprints on $\alpha$:

$$\begin{aligned} \mathcal{I}[\alpha](\mathbf{K}) &= \{\varnothing, \{s_1\}, \{s_2\}, \{s_3\}, \{s_1, s_2\}, \{s_1, s_3\}, \{s_2, s_3\}, \{s_1, s_2, s_3\}\}, \\ \mathcal{I}[\alpha](\mathbf{K}') &= \{\varnothing, \{s_1\}, \{s_2\}, \{s_3\}, \{s_1, s_2\}, \{s_1, s_3\}, \{s_2, s_3\}\}. \end{aligned}$$

From the separation point of view, we know from Lemma 4.3 that $\mathbf{K}$ and $\mathbf{K}'$ are equally useless (they cannot be used to separate any pair of nonempty languages recognized by $\alpha$). However, we also know from the imprints that $\mathbf{K}'$ is "better" as it contains no language that intersects $\alpha^{-1}(s_1), \alpha^{-1}(s_2)$ and $\alpha^{-1}(s_3)$ at the same time.

In view of Lemma 4.3, the smaller the imprint on $\alpha$ of a $\mathcal{C}$-partition is, the better this $\mathcal{C}$-partition is at separating languages recognized by $\alpha$. We use this remark to define the notion of *optimal* $\mathcal{C}$-partition.

**Optimal $\mathcal{C}$-Partitions.** Given a morphism $\alpha : A^+ \to S$ into a finite semigroup $S$ and a $\mathcal{C}$-partition $\mathbf{K}$, we say that $\mathbf{K}$ is *optimal for $\alpha$* if for any $\mathcal{C}$-partition $\mathbf{K}'$,

$$\mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathcal{I}[\alpha](\mathbf{K}')$$

We can use the fact that $\mathcal{C}$ is closed under intersection to prove that for any morphism $\alpha$, there always exists a $\mathcal{C}$-partition that is optimal for $\alpha$.

**Lemma 4.4.** *Let $\alpha : A^+ \to S$ be a morphism into a finite semigroup $S$. Then there exists a $\mathcal{C}$-partition that is optimal for $\alpha$.*

*Proof.* We prove that for any two $\mathcal{C}$-partitions $\mathbf{K}'$ and $\mathbf{K}''$, there exists a third $\mathcal{C}$-partition $\mathbf{K}$ such that, $\mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathcal{I}[\alpha](\mathbf{K}')$ and $\mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathcal{I}[\alpha](\mathbf{K}'')$. Since there are only finitely possible imprints on $\alpha$, the lemma will follow.

We set $\mathbf{K} = \{K' \cap K'' \mid K' \in \mathbf{K}' \text{ and } K'' \in \mathbf{K}''\}$. Since $\mathbf{K}'$ and $\mathbf{K}''$ were $\mathcal{C}$-partitions and $\mathcal{C}$ is closed under intersection, $\mathbf{K}$ remains a $\mathcal{C}$-partition. Finally, it is immediate from the definitions that $\mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathcal{I}[\alpha](\mathbf{K}')$ and $\mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathcal{I}[\alpha](\mathbf{K}'')$. □

Observe that the proof of Lemma 4.4 is non-constructive. Given a morphism $\alpha$, computing an actual optimal $\mathcal{C}$-partition for $\alpha$ is a difficult problem in general. In fact, as seen in Theorem 4.6 below, this is more general than solving $\mathcal{C}$-separability for any pair of languages recognized by $\alpha$. Before we present this theorem, let us make an important observation about optimal $\mathcal{C}$-partitions.

By definition, given a morphism $\alpha$, all $\mathcal{C}$-partitions that are optimal for $\alpha$ have the same imprint on $\alpha$. Hence, this unique imprint is a canonical object for $\mathcal{C}$ and $\alpha$. We call it the *optimal imprint with respect to* $\mathcal{C}$ *on* $\alpha$ and we denote it by $\mathcal{I}_{\mathcal{C}}[\alpha]$:

$$\mathcal{I}_{\mathcal{C}}[\alpha] = \mathcal{I}[\alpha](\mathbf{K}) \quad \text{for any optimal } \mathcal{C}\text{-partition } \mathbf{K} \text{ of } \alpha.$$

Note that, as an imprint, $\mathcal{I}_{\mathcal{C}}[\alpha]$ satisfies Fact 4.1 and Fact 4.2: $\{\{\alpha(w)\} \mid w \in A^+\} \subseteq \mathcal{I}_{\mathcal{C}}[\alpha]$ and $\mathcal{I}_{\mathcal{C}}[\alpha] = {\downarrow}\mathcal{I}_{\mathcal{C}}[\alpha]$. Moreover, using our three hypotheses on $\mathcal{C}$ (note that this is where we need the second and third ones), one can prove another convenient property: $\mathcal{I}_{\mathcal{C}}[\alpha]$ is a subsemigroup of $2^S$ (*i.e.*, it is closed under multiplication).

**Lemma 4.5.** *Let $\alpha : A^+ \to S$ be a morphism into a finite semigroup $S$. Then $\mathcal{I}_{\mathcal{C}}[\alpha]$ is a subsemigroup of $2^S$: for all $R, T \in \mathcal{I}_{\mathcal{C}}[\alpha]$, $RT \in \mathcal{I}_{\mathcal{C}}[\alpha]$.*

*Proof.* Let $R, T \in \mathcal{I}_{\mathcal{C}}[\alpha]$ with $R = \{r_1, \ldots, r_m\}$ and $T = \{t_1, \ldots, t_n\}$. We prove that $RT \in \mathcal{I}_{\mathcal{C}}[\alpha]$. Let $\mathbf{K}$ be a $\mathcal{C}$-partition of $A^+$. We have to prove $RT \in \mathcal{I}[\alpha](\mathbf{K})$, *i.e.*, that there exists $K \in \mathbf{K}$ such that $RT \subseteq \alpha(K)$. This is a consequence of the following claim.

**Claim.** *There exist $u_1, \ldots, u_m \in A^+$ and $v_1, \ldots, v_n \in A^+$ such that $\alpha(u_i) = r_i$ for $i \leqslant m$ and $\alpha(v_j) = t_j$ for $j \leqslant n$ and,*

- *for any $w \in A^+$, there exists $K \in \mathbf{K}$ such that $u_1 w, \ldots, u_m w \in K$.*
- *for any $w \in A^+$, there exists $K \in \mathbf{K}$ such that $wv_1, \ldots, wv_n \in K$.*

Before we prove the claim, let us finish the proof of the lemma. Using the first item of the claim for $w = v_1$, we obtain a language $K \in \mathbf{K}$ such that $u_1v_1, u_2v_1, \ldots, u_mv_1 \in K$. Similarly, for all $i \leqslant m$, we can use the second item of the claim for $w = u_i$ and we obtain a language $K_i \in \mathbf{K}$ such that $u_iv_1, u_iv_2, \ldots, u_iv_n \in K_i$. Note that each language $K_i$ contains the word $u_iv_1$, which also belongs to $K$. Hence, since $\mathbf{K}$ is a partition of $A^+$, we have $K = K_1 = \cdots = K_m$ and $K$ contains the word $u_iv_j$ for all $i \leqslant m$ and $j \leqslant n$. Since $\alpha(u_iv_j) = r_it_j$, this exactly says that $RT \subseteq \alpha(K)$, which terminates the proof of the lemma.

It now remains to prove the claim. We prove the existence of the words $v_1, \ldots, v_n$. The proof for that of $u_1, \ldots, u_m$ is symmetric. Observe that for any $w \in A^+$, the set $\mathbf{L}_w = \{w^{-1}K \mid K \in \mathbf{K}\}$ is a $\mathcal{C}$-partition of $A^+$ (recall that $\mathcal{C}$ is assumed to be closed under quotients). Moreover, since $\mathcal{C}$ contains only regular languages, all $K \in \mathbf{K}$ are regular languages and by Myhill-Nerode Theorem, they have finitely many left quotients. It follows that the set $\{\mathbf{L}_w \mid w \in A^+\}$ is finite. Hence using the fact that $\mathcal{C}$ is closed under boolean operations, we can construct a new $\mathcal{C}$-partition $\mathbf{L}$ that refines all partitions $\mathbf{L}_w$ for $w \in A^+$.

Since $\mathbf{L}$ is a $\mathcal{C}$-partition and $T \in \mathcal{I}_{\mathcal{C}}[\alpha]$, we know that there exists $L \in \mathbf{L}$ such that $T \subseteq \alpha(L)$. This means that $L$ contains $n$ words $v_1, \ldots, v_n \in A^+$ such that $\alpha(v_j) = t_j$ for $j \leqslant n$. We now prove that $v_1, \ldots, v_n \in A^+$ satisfy the conditions of the claim. Set $w \in A^+$, we know that there exists $L' \in \mathbf{L}_w$ such that $L \subseteq L'$ ($\mathbf{L}$ refines $\mathbf{L}_w$). This means that $v_1, \ldots, v_n \in L'$. Finally, by definition, $L' = w^{-1}K$ for some $K \in \mathbf{K}$, hence $wv_1, \ldots, wv_n \in K$. ∎

**From $\mathcal{C}$-Partitions to Separation.** We can now connect $\mathcal{C}$-partitions and optimal imprints to the separation problem for $\mathcal{C}$.

**Theorem 4.6.** *Let $\alpha : A^+ \to S$ be a morphism into a finite semigroup $S$. Let $L_1, L_2$ be two languages recognized by $\alpha$ and let $T_1, T_2 \subseteq S$ be the corresponding accepting sets. The following properties are equivalent:*

(1) *$L_1$ and $L_2$ are $\mathcal{C}$-separable.*
(2) *for all $t_1 \in T_1$ and all $t_2 \in T_2$, we have $\{t_1, t_2\} \notin \mathcal{I}_{\mathcal{C}}[\alpha]$.*
(3) *for any $\mathcal{C}$-partition $\mathbf{K}$ that is optimal for $\alpha$, $L_1$ and $L_2$ are separable by a union of languages in $\mathbf{K}$.*

*Proof.* We prove that (3) $\Rightarrow$ (1) $\Rightarrow$ (2) $\Rightarrow$ (3). Let us first assume that (3) holds, *i.e.*, that for any $\mathcal{C}$-partition $\mathbf{K}$ that is optimal for $\alpha$, $L_1$ and $L_2$ are separable by a union of languages in $\mathbf{K}$. Since there exists at least one $\mathcal{C}$-partition that is optimal for $\alpha$ (see Lemma 4.4), $L_1$ can be separated from $L_2$ with a union of languages in $\mathcal{C}$. Since $\mathcal{C}$ is closed under union, this separator is in $\mathcal{C}$ and (1) holds.

We now prove that (1) $\Rightarrow$ (2). Assume that (1) holds, *i.e.*, that $L_1$ is $\mathcal{C}$-separable from $L_2$. This means that there exists a language $K \in \mathcal{C}$ such that $L_1 \subseteq K$ and $K \cap L_2 = \varnothing$ (*i.e.*, $L_2 \subseteq A^+ \setminus K$). Since $\mathcal{C}$ is closed under complementation, $A^+ \setminus K \in \mathcal{C}$ and $\mathbf{K} = \{K, A^+ \setminus K\}$ is a $\mathcal{C}$-partition. By construction, for all $t_1 \in T_1$ and all $t_2 \in T_2$, $\{t_1, t_2\} \notin \mathcal{I}[\alpha](\mathbf{K})$. Hence $\{t_1, t_2\} \notin \mathcal{I}_{\mathcal{C}}[\alpha]$ since $\mathcal{I}_{\mathcal{C}}[\alpha] \subseteq \mathcal{I}[\alpha](\mathbf{K})$ by definition.

It remains to prove that (2) $\Rightarrow$ (3). Assume that for all $t_1 \in T_1$ and all $t_2 \in T_2$, $\{t_1, t_2\} \notin \mathcal{I}_{\mathcal{C}}[\alpha]$ and let $\mathbf{K}$ be an optimal $\mathcal{C}$-partition for $\alpha$. Since $\mathbf{K}$ is optimal, we know from our hypothesis that for all $t_1 \in T_1$ and all $t_2 \in T_2$, $\{t_1, t_2\} \notin \mathcal{I}[\alpha](\mathbf{K})$. Hence it follows from Lemma 4.3 that $L_1$ can be separated from $L_2$ by a union of languages in $\mathbf{K}$. □

In view of Theorem 4.6, given a class of languages $\mathcal{C}$ that satisfies the appropriate properties, a general approach to the separation problem can be devised as follows.

(1) Present an algorithm which takes a morphism $\alpha : A^+ \to S$ as input and computes $\mathcal{I}_{\mathcal{C}}[\alpha]$. Thanks to Theorem 4.6, this allows us to decide whether any two languages recognized by the input morphism $\alpha$ are $\mathcal{C}$-separable.

    Typically, this algorithm should be a lowest fixpoint: $\mathcal{I}_{\mathcal{C}}[\alpha]$ is computed as the smallest set $\mathrm{Sat}_{\mathcal{C}}(\alpha)$ which contains the singletons $\{\alpha(w)\}$ for $w \in \alpha(A^+)$ (see Fact 4.1) and is closed under a set of rules that is specific to $\mathcal{C}$. Note that, in view of Fact 4.2, one of these rules should always be closure under downset $\downarrow$, and in view of Lemma 4.5, one of these rules should always be closure under multiplication.
(2) To prove that the algorithm is sound, *i.e.*, that $\mathrm{Sat}_{\mathcal{C}}(\alpha) \subseteq \mathcal{I}_{\mathcal{C}}[\alpha]$, one needs to prove that for any computed set $T$, the imprint of any $\mathcal{C}$-partition $\mathbf{K}$ must contain $T$. This is usually simple and involves Ehrenfeucht-Fraïssé arguments.
(3) To prove that the algorithm is complete, *i.e.*, that $\mathcal{I}_{\mathcal{C}}[\alpha] \subseteq \mathrm{Sat}_{\mathcal{C}}(\alpha)$, one needs to construct a $\mathcal{C}$-partition $\mathbf{K}$ whose imprint on $\alpha$ is included in $\mathrm{Sat}_{\mathcal{C}}(\alpha)$. By definition, this proves that $\mathcal{I}_{\mathcal{C}}[\alpha] \subseteq \mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathrm{Sat}_{\mathcal{C}}(\alpha)$, hence, this proves completeness. We actually get more from this construction: combining it with the knowledge that the algorithm is also correct, we obtain $\mathcal{I}_{\mathcal{C}}[\alpha] = \mathcal{I}[\alpha](\mathbf{K})$: the $\mathcal{C}$-partition $\mathbf{K}$ that we construct is actually optimal for $\alpha$. By Item (3) in Theorem 4.6, we get a way to construct an actual separator in $\mathcal{C}$ of two $\mathcal{C}$-separable languages recognized by $\alpha$.

This terminates the presentation of the general approach. We now apply it to the special case when the class $\mathcal{C}$ is FO.

4.2. **A Separation Algorithm for First-Order Logic.** Fix a morphism $\alpha : A^+ \to S$ into a finite semigroup $S$. We describe a lowest fixpoint algorithm for computing the optimal imprint with respect to FO on $\alpha$: $\mathcal{I}[\alpha]$. Note that from now on we work with FO only. Therefore, we simply write $\mathcal{I}[\alpha]$ for $\mathcal{I}_{\mathrm{FO}}[\alpha]$.

Set $\mathcal{S}$ as a subsemigroup of $2^S$. We define $\mathrm{Sat}(\mathcal{S})$, the *saturation of* $\mathcal{S}$, as the smallest subset of $2^S$ that contains $\mathcal{S}$ and is closed under the three following operations:

(1) Downset: $\mathrm{Sat}(\mathcal{S}) = {\downarrow}\mathrm{Sat}(\mathcal{S})$.
(2) Multiplication: for any $T, T' \in \mathrm{Sat}(\mathcal{S})$, we have $TT' \in \mathrm{Sat}(\mathcal{S})$.
(3) FO-Closure: for any $T \in \mathrm{Sat}(\mathcal{S})$, $T^\omega \cup T^{\omega+1} \in \mathrm{Sat}(\mathcal{S})$.

Note that it is immediate that one can compute $\mathrm{Sat}(\mathcal{S})$ from $\mathcal{S}$ using a lowest fixpoint algorithm. Finally, we define $\mathrm{Sat}(\alpha)$ as $\mathrm{Sat}(\mathcal{S})$ for $\mathcal{S} = \{\{\alpha(w)\} \mid w \in A^+\}$.

An interesting observation is that only Operation (3) is specific to first-order logic in the definition of Sat. Indeed, we already know from the generic presentation that $\mathcal{I}[\alpha]$ contains $\{\{\alpha(w)\} \mid w \in A^+\}$ and is closed under downset and multiplication. This terminates the presentation of the algorithm, we state its correctness in the following proposition.

**Proposition 4.7.** *Set $\alpha : A^+ \to S$ as a morphism into a finite semigroup $S$. Then,*

$$\mathcal{I}[\alpha] = \mathrm{Sat}(\alpha).$$

Since $\mathrm{Sat}(\alpha)$ is computable, Proposition 4.7 immediately implies that so is $\mathcal{I}[\alpha]$. Using Theorem 4.6, this yields the decidability of the separation problem for first-order logic. A simple analysis of the lowest fixpoint procedure shows an Exptime complexity upper bound. This proves the first item of Theorem 3.2, as we now show.

*Proof of the first item of Theorem 3.2.* By Theorem 4.6, it suffices to prove that one can compute $\mathcal{I}[\alpha]$ in Exptime in the size of $S$. Indeed, it then suffices to test whether there exists $T \in \mathcal{I}[\alpha]$ such that $\alpha(L_1) \cap T \neq \varnothing$ and $\alpha(L_2) \cap T \neq \varnothing$. This can also be achieved in Exptime by testing all possible candidates $T$. By Proposition 4.7, we know that computing $\mathcal{I}[\alpha]$ can be done by computing $\mathrm{Sat}(\alpha)$.

By definition, $\mathrm{Sat}(\alpha) \subseteq 2^S$. This means that the number of steps the algorithm needs to reach the fixpoint is at most exponential in $S$. Therefore, it suffices to prove that each step can be done in Exptime to conclude that the whole computation can also be done in Exptime. Each step requires computing $T^\omega \cup T^{\omega+1}$ for at most $|2^S|$ subsets $T$. Each computation can be done in Exptime, since $T^\omega$ is equal to some $T^m$ for $m \leqslant |2^S|$ such that $T^m = T^{2m}$. □

We postpone the proof of Item (2) of Theorem 3.2 (the bound on the quantifier rank of the separator) to Section 6 where we prove the difficult direction of Proposition 4.7: $\mathcal{I}[\alpha] \subseteq \mathrm{Sat}(\alpha)$. As explained, the proof amounts to constructing an optimal FO-partition for $\alpha$.

Another interesting observation about our saturation algorithm is that it can be viewed as a generalization of Schützenberger's Theorem [27, 15]. Indeed, a language is first-order *definable* if and only if its syntactic semigroup is aperiodic. One definition of aperiodicity is that a semigroup is aperiodic if and only if it satisfies the identity $s^\omega = s^{\omega+1}$. The counterpart to this definition can be found in the main operation of our saturation procedure, Operation (3) (which is the only non-generic operation). This observation raises another question: could Operation (3) be replaced to reflect alternate definitions of aperiodicity while retaining Proposition 4.7? We shall see in Section 7 that this is indeed possible. Another

consequence of this observation is that we can reprove Schützenberger's Theorem as a simple corollary of Proposition 4.7.

**Corollary 4.8** (Schützenberger's Theorem)**.** *Let $L$ be a regular language. Then $L$ can be defined in FO if and only if its syntactic semigroup is aperiodic.*

*Proof.* It is known that a language is definable in FO if and only if all languages recognized by its syntactic semigroup are definable in FO as well (this is actually not specific to FO and true for all classes of languages that are "Varieties", see [18] for example). Hence, if $S$ is the syntactic semigroup of $L$ and $\alpha : A^+ \to S$ the associated (surjective) morphism, $L$ is definable in FO if and only if $\{\alpha^{-1}(s) \mid s \in S\}$ is an FO-partition. The imprint on $\alpha$ of this FO-partition is $\{\{\alpha(w)\} \mid w \in A^+\} \cup \{\varnothing\}$, which is equal to $\{\{s\} \mid s \in S\} \cup \{\varnothing\}$, since $\alpha$ is surjective. Therefore, $L$ is definable in FO if and only if $\mathrm{Sat}(\alpha) = \mathcal{I}[\alpha] = \{\{s\} \mid s \in S\} \cup \{\varnothing\}$ (see Proposition 4.7). By definition of $\mathrm{Sat}(\alpha)$, this is equivalent to $s^\omega = s^{\omega+1}$ for all $s \in S$. □

It now remains to prove Proposition 4.7. In Section 5, we prove that $\mathrm{Sat}(\alpha) \subseteq \mathcal{I}[\alpha]$. This corresponds to soundness of the algorithm: all computed sets indeed belong to $\mathcal{I}[\alpha]$. Finally, in Section 6, we focus on the proof of the most difficult direction, which is the second one: $\mathcal{I}[\alpha] \subseteq \mathrm{Sat}(\alpha)$. It implies completeness of the algorithm, that is, that every set belonging to $\mathcal{I}[\alpha]$ is actually computed by the algorithm.

We finish this section by running the algorithm, to show that it detects that the languages of Example 3.1 are not FO-separable.

4.3. **Example 3.1, continued.** To start our algorithm, we first need a semigroup morphism recognizing both $L_0$ and $L_1$. Observe that both languages are recognized by the automaton below, with 4 as final state for $L_0$, and 2 as final state for $L_1$. Therefore, its transition semigroup $S$ recognizes both languages[2]. The recognizing morphism $\alpha : A^+ \to S$ maps a

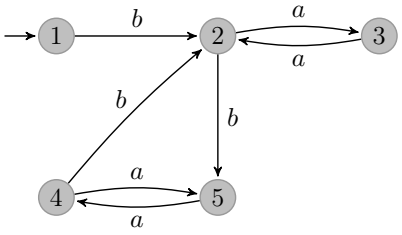


Figure 1: Automaton recognizing both $L_0$ and $L_1$

word to the partial function it defines from states to states. We still denote the images of $a, b \in A$ by $a, b \in S$, respectively. It is easy to see that $L_0 = \alpha^{-1}(b^2a)$ and $L_1 = \alpha^{-1}(\{b, b^2ab\})$.

We use Theorem 4.6 to show that $L_0$ and $L_1$ are not FO-separable: we have to find $s_0 \in \alpha(L_0)$ and $s_1 \in \alpha(L_1)$ such that $\{s_0, s_1\} \in \mathcal{I}[\alpha]$. We claim that $s_0 = b^2a$ and $s_1 = b^2ab$ satisfy this property. We actually show that $\{s_0, s_1\}$ is computed as an element of $\mathrm{Sat}(\alpha)$, which, by Proposition 4.7, implies that it belongs to $\mathcal{I}[\alpha]$.

[2]Recall that the transition semigroup consists of all partial mappings induced by words from the state set to itself. It is easy to see that it recognizes the language accepted by the automaton, see [18, Sec. 3.1].

By definition, $\{a\}, \{b\} \in \mathrm{Sat}(\alpha)$. Then, note that $\{a\}^\omega = \{a^2\}$ and $\{a\}^{\omega+1} = \{a\}$. Therefore, by definition of Operation (3), we have $\{a, a^2\} \in \mathrm{Sat}(\alpha)$. Using Operation (2), we then obtain that $X = \{a, aa\} \cdot \{b\} = \{ab, aab\} \in \mathrm{Sat}(\alpha)$. Now, Operation (3) yields $Y = X^\omega \cup X^{\omega+1} \in \mathrm{Sat}(\alpha)$. Computing $Y$ shows that $\{bab, bab^2\} \subseteq Y$. Finally, using Operation (2), we obtain that $T = \{b\} \cdot Y \cdot \{a, a^2\} \in \mathrm{Sat}(\alpha)$. One can then verify that $\{b^2a, b^2ab\} \subseteq T$. Therefore, we get from Operation (1) that $\{b^2a, b^2ab\} \in \mathrm{Sat}(\alpha)$, as claimed.

## 5. Soundness of the Algorithm

In this section we prove soundness of our algorithm, that is the inclusion $\mathrm{Sat}(\alpha) \subseteq \mathcal{I}[\alpha]$ in Proposition 4.7. Recall that we work with a morphism $\alpha : A^+ \to S$ into a finite semigroup $S$. By definition of $\mathrm{Sat}(\alpha)$, we need to prove that $\mathcal{I}[\alpha]$ contains the set $\{\{\alpha(w)\} \mid w \in A^+\}$ and is closed under Downset, Multiplication and FO-Closure.

We already know from Fact 4.1 that $\mathcal{I}[\alpha]$ contains the set $\{\{\alpha(w)\} \mid w \in A^+\}$. Furthermore, closure under Downset and Multiplication follows from Fact 4.2 and Lemma 4.5. Therefore, we only need to prove that $\mathcal{I}[\alpha]$ is closed under FO-Closure. This is what we do now. To present the argument, we need an alternate definition of $\mathcal{I}[\alpha]$.

Given two words $w, w' \in A^+$ and $k \in \mathbb{N}$, we write $w \equiv_k w'$ to denote the fact that $w$ and $w'$ satisfy the same FO-formulas of quantifier rank $k$. One can verify that for all $k$, $\equiv_k$ is an equivalence relation of finite index and that each class can be defined in FO.

**Lemma 5.1.** *Let $T \in 2^S$. Then $T \in \mathcal{I}[\alpha]$ if and only if for all $k \in \mathbb{N}$, there exists an equivalence class $W$ of $\equiv_k$ such that $T \subseteq \alpha(W)$.*

*Proof.* Assume first that for all $k \in \mathbb{N}$, there exists an equivalence class $W$ of $\equiv_k$ such that $T \subseteq \alpha(W)$. To prove that $T \in \mathcal{I}[\alpha]$, we show that $T \in \mathcal{I}[\alpha](\mathbf{K})$ for any FO-partition $\mathbf{K}$. By definition, there exists $k \in \mathbb{N}$ such that all languages in $\mathbf{K}$ can be defined by a formula of quantifier rank $k$ (recall that FO-partitions are finite partitions). It follows that elements of $\mathbf{K}$ are unions of classes of $\equiv_k$, hence $W$ is included in some language $K$ of $\mathbf{K}$. By hypothesis, $T \subseteq \alpha(W)$. It follows that $T \subseteq \alpha(W) \subseteq \alpha(K)$, which terminates the proof of this direction.

Conversely, assume that $T \in \mathcal{I}[\alpha]$ and set $k \in \mathbb{N}$. Set $\mathbf{K}$ as the partition of $A^+$ into equivalence classes of $\equiv_k$. By definition, $\mathbf{K}$ is an FO-partition. Hence, there exists an equivalence class $W \in \mathbf{K}$ of $\equiv_k$ such that $T \subseteq \alpha(W)$. □

We can now prove that $\mathcal{I}[\alpha]$ is closed under FO-closure to conclude the soundness proof: set $T = \{t_1, \ldots, t_n\} \in \mathcal{I}[\alpha]$ and set $R = T^\omega \cup T^{\omega+1}$, we need to prove that $R \in \mathcal{I}[\alpha]$. We use Lemma 5.1: for all $k \in \mathbb{N}$, we shall find an equivalence class $V$ of $\equiv_k$ such that $R \subseteq \alpha(V)$.

Set $k \in \mathbb{N}$. Using the other direction of Lemma 5.1 for $T$, we obtain an equivalence class $W$ of $\equiv_k$ such that $T \subseteq \alpha(W)$. Set $V = W^{2^k\omega} \cup W^{2^k\omega+1}$. By definition, $R \subseteq \alpha(V)$. Therefore, it suffices to prove that $V$ is included in an equivalence class of $\equiv_k$ (*i.e.*, that all words in $V$ are pairwise equivalent). This is a consequence of the following lemma, which is easy and folklore, and whose simple proof is omitted here. It relies on Ehrenfeucht-Fraïssé games, details can be found in [32].

**Lemma 5.2.** *Set $k \in \mathbb{N}$, then.*

(1) *For all $u_1, u_2, v_1, v_2 \in A^+$, if $u_1 \equiv_k v_1$ and $u_2 \equiv_k v_2$, then $u_1 \cdot u_2 \equiv_k v_1 \cdot v_2$.*
(2) *For all $u \in A^+$ and all $k > 0$, we have $u^{2^k} \equiv_k u^{2^k+1}$.*

Let us now conclude the proof. Pick some arbitrarily chosen word $w \in W$. By Lemma 5.2 (1), it is immediate that any word of $V$ is $\equiv_k$-equivalent to either $u_0 = w^{2^k\omega} \in W^{2^k\omega}$ or to $u_1 = w^{2^k\omega+1} \in W^{2^k\omega+1}$. To conclude that all words of $V$ are $\equiv_k$-equivalent, it remains to prove that $u_0 \equiv_k u_1$, which follows directly from Lemma 5.2 (2).

## 6. Completeness of the Algorithm

In this section, we prove the most interesting inclusion from Proposition 4.7: $\mathcal{I}[\alpha] \subseteq \mathrm{Sat}(\alpha)$. We use induction to construct an FO-partition $\mathbf{K}$ whose imprint on $\alpha$ belongs to $\mathrm{Sat}(\alpha)$. This proves that $\mathcal{I}[\alpha] \subseteq \mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathrm{Sat}(\alpha)$. For the rest of this section, we assume fixed a morphism $\alpha : A^+ \to S$ into a finite semigroup. We state the induction in the following proposition. Note that, in order to set up the induction, we have to start from a morphism from some free monoid $B^+$, where $B$ is an *arbitrary* alphabet, into an *arbitrary* subsemigroup $\mathcal{S}$ of $2^S$. This is because, following a proof from Wilke [36], we argue by induction on the size of (the index of) the semigroup, and of the alphabet.

**Proposition 6.1.** *Let $\mathcal{S}$ be a subsemigroup of $2^S$ and $\beta : B^+ \to \mathcal{S}$ be a surjective morphism. Then there exists an FO-partition $\mathbf{K}$ of $B^+$ such that for all $K \in \mathbf{K}$,*

(1) $\llbracket \beta(K) \rrbracket \in \mathrm{Sat}(\mathcal{S})$.
(2) *$K$ can be defined by a first-order formula of rank at most $|B| \cdot 2^{|\llbracket \mathcal{S} \rrbracket|^2}$.*

Before proving Proposition 6.1, we apply it to conclude the proof of Proposition 4.7. Set $\mathcal{S} = \{\{\alpha(w)\} \mid w \in A^+\}$ and $\beta : A^+ \to \mathcal{S}$ defined by $\beta(w) = \{\alpha(w)\}$ (note that $\beta$ is surjective). Recall that by definition, $\mathrm{Sat}(\alpha) = \mathrm{Sat}(\mathcal{S})$. From Proposition 6.1, we obtain an FO-partition $\mathbf{K}$ of $A^+$ such that, for all $K \in \mathbf{K}$,

(1) $\alpha(K) = \llbracket \beta(K) \rrbracket \in \mathrm{Sat}(\alpha)$.
(2) $K$ can be defined by a first-order formula of rank at most $|A| \cdot 2^{|S|^2}$.

It is now immediate from Item (1) and the fact that $\mathrm{Sat}(\alpha)$ is closed under downset that $\mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathrm{Sat}(\alpha)$. We conclude that $\mathcal{I}[\alpha] \subseteq \mathcal{I}[\alpha](\mathbf{K}) \subseteq \mathrm{Sat}(\alpha)$ which terminates the proof of Proposition 4.7. Moreover, we already know from Section 5 that $\mathcal{I}[\alpha] \supseteq \mathrm{Sat}(\alpha)$, so we actually obtain that $\mathcal{I}[\alpha] = \mathcal{I}[\alpha](\mathbf{K})$: $\mathbf{K}$ is optimal for $\alpha$.

This is of particular interest. Indeed, we know from Theorem 4.6 that for any two languages recognized by $\alpha$ that are FO-separable, one can construct a separator as a union of languages in $\mathbf{K}$. Therefore, since our proof of Proposition 6.1 is constructive ($\mathbf{K}$ is built by induction), we obtain a method for constructing an FO-separator for any pair of FO-separable languages recognized by $\alpha$. Finally, we know from Item (2) that this separator has quantifier rank at most $|A| \cdot 2^{|S|^2}$ which yields the second item in Theorem 3.2.

**Corollary 6.2** (Second item in Theorem 3.2)**.** *Given two languages $L_0$ and $L_1$ that are recognized by $\alpha$, if they are FO-separable, then one can effectively construct an actual separator with a formula of quantifier rank at most $|A|2^{|S|^2}$.*

Note that a rough analysis of the procedure that constructs separators yields a 2-Exptime upper bound on the complexity in the size of $S$. This is because while the rank of the formula is "only" exponentially large in $S$, its size is one exponential larger in general.

Another interesting remark is that while this paper is written from a logical perspective (we prove that $\mathbf{K}$ is an FO-partition by constructing a first-order formula for each language in $\mathbf{K}$), the construction is not specific to first-order logic. In other words, the construction could

be easily adapted to obtain *star-free expressions*, LTL *formulas* or *counter-free automata* defining the languages in $\mathbf{K}$.

It now remains to prove Proposition 6.1. The rest of this section is devoted to this proof. We set $\mathcal{S}$ as a subsemigroup of $2^S$ and $\beta : B^+ \to \mathcal{S}$ as a morphism as in the statement of the proposition. The proof is a generalization of Wilke's argument [36] for deciding first-order definability. As explained above, the proof is constructive. We construct the partition $\mathbf{K}$ as well as the first-order formulas that define its languages. We proceed by induction on the following two parameters listed by order of importance:

($a$) the index $|[\![\mathcal{S}]\!]|$ of $\mathcal{S}$,
($b$) the size of $B$.

The proof is divided into three cases:

- first, we consider the case when $|B| = 1$.
- otherwise, we distinguish two subcases, depending on a property of $\beta$ called *tameness*.

6.1. **Special Case: $|B| = 1$.** In that case, $B$ is a singleton $\{b\}$. Hence all words are of the form $b^n$ for some $n \geqslant 1$. It follows from a standard semigroup theory argument that there exists $m \leqslant |\mathcal{S}| \leqslant 2^{|[\![\mathcal{S}]\!]|}$ such that $\beta(b^m) = \beta(b^\omega)$. We partition $B^+$ into $m$ languages.

For all $1 \leqslant i < m$, we set $K_i$ as the singleton $\{b^i\}$. Finally, we set $K_m = \{b^j \mid j \geqslant m\}$. It is immediate by definition that $\mathbf{K} = \{K_1, \ldots, K_m\}$ is a partition of $B^+$. Moreover, one can easily construct first-order formulas of rank at most $m \leqslant 2^{|[\![\mathcal{S}]\!]|} \leqslant 2^{|[\![\mathcal{S}]\!]|^2}$ for $K_1, \ldots, K_m$: $\mathbf{K}$ is an FO-partition and we obtain Item (2). It remains to prove Item (1).

For $i < m$, we have $[\![\beta(K_i)]\!] = \beta(b^i) \in \mathcal{S} \subseteq \mathrm{Sat}(\mathcal{S})$. Hence Item (1) is satisfied. Assume now that $i = m$. By definition of $K_m$ and $\omega$,

$$[\![\beta(K_m)]\!] = \bigcup_{j \geqslant 0} \beta(b^{j+\omega}) = (\beta(b)^\omega \cup \beta(b)^{\omega+1}) \cdots (\beta(b)^\omega \cup \beta(b)^{2\omega-1})$$

Since $\mathrm{Sat}(\mathcal{S})$ is closed under multiplication, it suffices to prove that for all $j$, we have $\beta(b)^\omega \cup \beta(b)^{\omega+j} \in \mathrm{Sat}(\mathcal{S})$. By definition, for any $j \geqslant 0$, $\beta(b)^{\omega+j} \in \mathcal{S} \subseteq \mathrm{Sat}(\mathcal{S})$. Moreover, observe that

$$\beta(b)^\omega \cup \beta(b)^{\omega+j} = (\beta(b)^{\omega+j})^\omega \cup (\beta(b)^{\omega+j})^{\omega+1}.$$

Therefore, $\beta(b)^\omega \cup \beta(b)^{\omega+j} \in \mathrm{Sat}(\mathcal{S})$ is immediate by Operation (3).

This terminates the case $|B| = 1$. For the remainder of the proof, we now assume that $|B| \geqslant 2$. As explained above, we distinguish two cases depending on a property of $\beta$.

**Tameness.** We say that $\beta$ is *tame* if

$$\forall b \in B,\ [\![\mathcal{S}]\!] = \beta(b) \cdot [\![\mathcal{S}]\!] \text{ and } [\![\mathcal{S}]\!] = [\![\mathcal{S}]\!] \cdot \beta(b).$$

6.2. **Case 1: $\beta$ is tame.** This is the base case: we don't use induction. We use tameness to prove that $\llbracket \mathcal{S} \rrbracket \in \mathrm{Sat}(\mathcal{S})$. Therefore, it suffices to choose, $\mathbf{K} = \{B^+\}$ since $\llbracket \beta(B^+) \rrbracket = \llbracket \mathcal{S} \rrbracket$ (recall that $\beta$ is assumed to be surjective). This is a consequence of the following lemma:

**Lemma 6.3.** *There exists a group $\mathcal{G} \subseteq \mathcal{S}$ such that $\llbracket \mathcal{G} \rrbracket = \llbracket \mathcal{S} \rrbracket$.*

We first use Lemma 6.3 to finish the proof of this case. Let $\mathcal{G} = \{T_1, \ldots, T_n\}$ be a group as given by the lemma. We prove that $\llbracket \mathcal{G} \rrbracket \in \mathrm{Sat}(\mathcal{S})$. Since $\mathcal{G}$ is a group, we get $T_i^\omega = 1_\mathcal{G}$, so $T_i = T_1^\omega \cdots T_{i-1}^\omega T_i^{\omega+1} T_{i+1}^\omega \cdots T_n^\omega$ for all $i$. Combining these equalities gives us the equality

$$\llbracket \mathcal{G} \rrbracket = (T_1^\omega \cup T_1^{\omega+1}) \cdots (T_n^\omega \cup T_n^{\omega+1}).$$

By definition, for all $i$, $T_i \in \mathcal{S}$, hence $T_i \in \mathrm{Sat}(\mathcal{S})$ since $\beta$ is surjective. It then follows from Multiplication Closure, FO-Closure and the equality above that $\llbracket \mathcal{G} \rrbracket \in \mathrm{Sat}(\mathcal{S})$. Since $\llbracket \mathcal{G} \rrbracket = \llbracket \mathcal{S} \rrbracket$, this terminates the proof in Case 1.

It remains to prove Lemma 6.3. We first prove that while $\mathcal{S}$ might not be a group itself, it is what we call a *pseudo-group*.

**Pseudo-groups.** Let $\mathcal{T}$ be a subsemigroup of $2^S$. We say that $\mathcal{T}$ is a *pseudo-group* if for all $T \in \mathcal{T}$, $\llbracket \mathcal{T} \rrbracket = T \cdot \llbracket \mathcal{T} \rrbracket$ and $\llbracket \mathcal{T} \rrbracket = \llbracket \mathcal{T} \rrbracket \cdot T$.

**Lemma 6.4.** *$\mathcal{S}$ is a pseudo-group.*

*Proof.* Set $T \in \mathcal{S}$. We prove that $\llbracket \mathcal{S} \rrbracket = \llbracket \mathcal{S} \rrbracket \cdot T$. The equality $\llbracket \mathcal{S} \rrbracket = T \cdot \llbracket \mathcal{S} \rrbracket$ is symmetrical. Since $\beta$ is surjective, there exists $w \in B^+$ such that $T = \beta(w)$. We proceed by induction on the length of $w$. If $w$ is of length 1, this is by tameness of $\beta$.

Assume now that the result holds for words of length $k$ and that $w$ is of length $k+1$. This means that $w = ub$ with $u$ a word of length $k$. By induction hypothesis, we get that $\llbracket \mathcal{S} \rrbracket = \llbracket \mathcal{S} \rrbracket \cdot \beta(u)$. Moreover, using tameness, we get that $\llbracket \mathcal{S} \rrbracket = \llbracket \mathcal{S} \rrbracket \cdot \beta(b)$. It follows that $\llbracket \mathcal{S} \rrbracket = \llbracket \mathcal{S} \rrbracket \cdot \beta(u) \cdot \beta(b) = \llbracket \mathcal{S} \rrbracket \cdot \beta(w)$, which concludes the proof. □

We now finish the proof of Lemma 6.3. We prove that any pseudo-group $\mathcal{T} \subseteq \mathcal{S}$ that is not already a group strictly contains a subsemigroup $\mathcal{R}$ that remains a pseudo-group, and such that $\llbracket \mathcal{R} \rrbracket = \llbracket \mathcal{T} \rrbracket$. Applying this result iteratively to $\mathcal{S}$ yields the desired group $\mathcal{G}$.

Let $\mathcal{T} \subseteq \mathcal{S}$ be a pseudo-group that is not already a group. An easy and standard argument implies that there must exist $R \in \mathcal{T}$ such that $R \cdot \mathcal{T} \subsetneq \mathcal{T}$ or $\mathcal{T} \cdot R \subsetneq \mathcal{T}$. By symmetry assume that it is the former and set $\mathcal{R} = R \cdot \mathcal{T}$. By definition, $\mathcal{R}$ is closed under product and is therefore a semigroup. It remains to prove that $\llbracket \mathcal{R} \rrbracket = \llbracket \mathcal{T} \rrbracket$ and that $\mathcal{R}$ is a pseudo-group.

By definition, we have $\llbracket \mathcal{R} \rrbracket = R \cdot \llbracket \mathcal{T} \rrbracket$ and $R \cdot \llbracket \mathcal{T} \rrbracket = \llbracket \mathcal{T} \rrbracket$ since $\mathcal{T}$ is a pseudo-group. We conclude that $\llbracket \mathcal{R} \rrbracket = \llbracket \mathcal{T} \rrbracket$. Finally, set $RT \in \mathcal{R}$, we need to prove that $\llbracket \mathcal{R} \rrbracket = RT \cdot \llbracket \mathcal{R} \rrbracket$ and $\llbracket \mathcal{R} \rrbracket = \llbracket \mathcal{R} \rrbracket \cdot RT$. Both equalities are immediate since $\llbracket \mathcal{R} \rrbracket = \llbracket \mathcal{T} \rrbracket$ and $\mathcal{T}$ is a pseudo-group.

6.3. **Case 2: $\beta$ is not tame.** This is the case where we use induction. By hypothesis on $\beta$, there exists $b \in B$ such that $\llbracket \mathcal{S} \rrbracket \neq \beta(b) \cdot \llbracket \mathcal{S} \rrbracket$ or $\llbracket \mathcal{S} \rrbracket \neq \llbracket \mathcal{S} \rrbracket \cdot \beta(b)$. By symmetry, we assume the former, *i.e.*,

$$\llbracket \mathcal{S} \rrbracket \neq \beta(b) \cdot \llbracket \mathcal{S} \rrbracket.$$

We set $b$ as this letter for the remainder of the proof.

Recall that we have to construct an FO-partition $\mathbf{K}$ of $B^+$ satisfying Items (1) and (2) in Proposition 6.1. We begin by giving a brief overview of the construction. Set $C = B \setminus \{b\}$.

Observe that any word $w \in B^+$ can be uniquely decomposed in three (possibly empty) parts: a prefix in $C^+$, an infix in $(b^+C^+)^+$ and a suffix in $b^+$. Our construction works by using induction to construct finite partitions of the sets of possible prefixes, infixes and suffixes, which yields a partition of the whole set $B^+$. For the sets of prefixes and suffixes (*i.e.*, $C^+$ and $b^+$), the partitions are simply obtained by induction on the size of alphabet. For the set of infixes (*i.e.* $(b^+C^+)^+$) the argument is more involved and is obtained by induction on the index of $\mathcal{S}$. We now make the construction more precise. We begin by defining the partitions of the sets of possible prefixes, infixes and suffixes, in the three lemmas below.

**Lemma 6.5** (Partition of the prefixes)**.** *There exists a finite partition* $\mathbf{L}$ *of* $C^+$ *such that for any language* $L \in \mathbf{L}$*:*

(1) $[\![\beta(L)]\!] \in \mathrm{Sat}(\mathcal{S})$.
(2) *there exists a first-order formula of rank at most* $(|B| - 1) \cdot 2^{|[\![\mathcal{S}]\!]|^2}$ *that defines* $L$*.*

**Lemma 6.6** (Partition of the suffixes)**.** *There exists a finite partition* $\mathbf{H}$ *of* $b^+$ *such that for any language* $H \in \mathbf{H}$*:*

(1) $[\![\beta(H)]\!] \in \mathrm{Sat}(\mathcal{S})$.
(2) *there exists a first-order formula of rank at most* $2^{|[\![\mathcal{S}]\!]|^2}$ *that defines* $H$*.*

**Lemma 6.7** (Partition of the infixes)**.** *There exists a finite partition* $\mathbf{K}'$ *of* $(b^+C^+)^+$ *such that for any language* $K \in \mathbf{K}'$*:*

(1) $[\![\beta(K)]\!] \in \mathrm{Sat}(\mathcal{S})$.
(2) *there exists a first-order formula of rank at most* $|B| \cdot 2^{|[\![\mathcal{S}]\!]|^2} - 2$ *that defines* $K$*.*

Lemmas 6.5, 6.6 and 6.7 are proved using both our induction hypotheses. Before we present these proofs, we use the three lemmas to construct the desired FO-partition $\mathbf{K}$ of $B^+$ and conclude the proof of Proposition 6.1. We define $\mathbf{K}$ as follows:

$$\begin{aligned} & \{LK'H \mid L \in \mathbf{L}, K' \in \mathbf{K}' \text{ and } H \in \mathbf{H}\} \\ \cup\ & \{K'H \mid K' \in \mathbf{K}' \text{ and } H \in \mathbf{H}\} \\ \mathbf{K} = \cup\ & \{LH \mid L \in \mathbf{L} \text{ and } H \in \mathbf{H}\} \\ \cup\ & \{LK' \mid L \in \mathbf{L} \text{ and } K' \in \mathbf{K}'\} \\ \cup\ & \mathbf{L} \cup \mathbf{K}' \cup \mathbf{H} \end{aligned}$$

That $\mathbf{K}$ is partition of $B^+$ is immediate since $\mathbf{L}, \mathbf{K}'$ and $\mathbf{H}$ are partitions and any word in $B^+$ can be *uniquely* decomposed as the concatenation of a prefix in $b^+$, an infix in $(b^+C^+)^+$ and a suffix in $C^+$ (each one possibly empty, but not the three of them together). That $\mathbf{K}$ is actually an FO-partition is a consequence of the following fact which describes a standard construction for first-order logic over words.

**Fact 6.8.** Set $k \geqslant 0$ and two languages $L_1, L_2$, each defined by a first-order formula of rank at most $k$. Then $L_1L_2$ can be defined by a first-order formula of rank at most $k + 1$.

*Proof.* A word is in $L_1L_2$ if and only it can be cut into a prefix in $L_1$ and a suffix in $L_2$. Therefore, in first-order logic, it suffices to quantify existentially the position $x$ at which the cut is made and then use the formulas that define $L_1$ and $L_2$ (modified so that quantifications are restricted to the left or to the right of $x$) to test whether the prefix and suffix belong to $L_1$ and $L_2$. By construction, the formula we obtain has rank at most $k + 1$. □

Since any language in $\mathbf{K}$ is the concatenation of at most three languages that are defined by first-order formulas of rank at most $|B|\cdot 2^{|||\mathcal{S}|||^2}-2$ (see the first items in the three lemmas), we obtain that $\mathbf{K}$ is an FO-partition as well as Item (2) in Proposition 6.1.

Finally, Item (1) in Proposition 6.1 (*i.e.*, for all $K \in \mathbf{K}$, $[\![\beta(K)]\!] \in \mathrm{Sat}(\mathcal{S})$) is an immediate consequence of Item (1) in the three lemmas and the fact that $\mathrm{Sat}(\mathcal{S})$ is closed under multiplication. This terminates the proof of Proposition 6.1.

It now remains to prove Lemmas 6.5, 6.6 and 6.7. We first take care of Lemma 6.5 and Lemma 6.6 which are immediate by induction. Indeed, Lemma 6.5 is obtained by applying the induction hypothesis on the second parameter (the size of the alphabet) to the restriction of $\beta$ to $C^+$ (recall that $C = B \setminus \{b\}$). Furthermore, Lemma 6.6 is exactly the special case when the alphabet is of size one which is already proved.

The proof of Lemma 6.7 is more involved and is where we use induction on the index of $\mathcal{S}$ as well as our choice of the letter $b$ (*i.e.*, the fact that $\beta$ is not tame). We devote the remainder of the section to this proof.

Recall that our goal is to find a partition of $(b^+C^+)^+$ that meets the conditions of the lemma. We proceed in three steps. First, we use Lemma 6.5 and Lemma 6.6 (*i.e.*, our FO-partitions of $b^+$ and $C^+$) to abstract the set $b^+C^+$ as a finite alphabet $\mathfrak{B}$ and in turn the set $(b^+C^+)^+$ as the set of all words in $\mathfrak{B}^+$. This allows us to abstract the restriction of $\beta$ to $(b^+C^+)^+$ as a semigroup morphism $\gamma : \mathfrak{B}^+ \to \mathcal{T}$ into a new semigroup $\mathcal{T} \subseteq \mathcal{S}$. Then in a second step, we use the fact that $\beta$ is not tame (through our choice of $b$) to prove that $\mathcal{T}$ has smaller index than $\mathcal{S}$. This enables us to apply induction to $\gamma$ and obtain an FO-partition of $\mathfrak{B}^+$. Finally, in the third step, we construct the desired partition of $(b^+C^+)^+$ from that of $\mathfrak{B}^+$.

**Proof of Lemma 6.7, Step 1: Abstraction of $(b^+C^+)^+$.** We begin with the definition of the new alphabet $\mathfrak{B}$. Intuitively, we want to simply set $\mathfrak{B} = \mathbf{H} \times \mathbf{L}$. Indeed, we know by construction of $\mathbf{H}$ and $\mathbf{L}$ that $\{HL \mid H \in \mathbf{H} \text{ and } L \in \mathbf{L}\}$ is a partition of $b^+C^+$. Therefore, such an alphabet $\mathfrak{B}$ would be a satisfying abstraction of $b^+C^+$. However, there is an issue with this definition: in the proof, we do not keep track of the size of the partitions $\mathbf{H}$ and $\mathbf{L}$. Therefore, such a definition does not allow us to control the size of $\mathfrak{B}$. This is a problem for proving Item (2) in Lemma 6.7 as the bound on the quantifier rank of the formulas obtained by induction depends on the size of the alphabet. For this reason we use the following slightly different definition:

$$\mathfrak{B} = \left\{ [\![\beta(HL)]\!] \mid H \in \mathbf{H} \text{ and } L \in \mathbf{L} \right\} \subseteq 2^S.$$

Observe that to any word $w \in b^+C^+$, one can associate a unique letter $(w)_{\mathfrak{B}} \in \mathfrak{B}$: since $\mathbf{H}$ and $\mathbf{L}$ are partitions, there exist unique $H \in \mathbf{H}$ and $L \in \mathbf{L}$ such that $w \in HL$, we simply set $(w)_{\mathfrak{B}} = [\![\beta(HL)]\!]$. This means that $\mathfrak{B}$ defines a finite partition of $b^+C^+$ (it is even an FO-partition by Fact 6.8): two words are in the same class of the partition if they yield the same letter over $\mathfrak{B}$. We extend the definition to words $w \in (b^+C^+)^+$: any such $w$ can be uniquely decomposed as $w = w_1 \cdots w_n$ with $w_1, \ldots, w_n \in b^+C^+$, we set $(w)_{\mathfrak{B}} = (w_1)_{\mathfrak{B}} \cdots (w_n)_{\mathfrak{B}} \in \mathfrak{B}^+$. In particular $\mathfrak{B}^+$ defines an infinite partition of $(b^+C^+)^+$.

We finish with the definition of the morphism $\gamma$. We set $\mathcal{T}$ as the subsemigroup of $2^S$ generated by $\mathfrak{B}$. Finally, set $\gamma : \mathfrak{B}^+ \to \mathcal{T}$ defined by simply evaluating in $\mathcal{T}$ the product of the letters of a word in $\mathfrak{B}^+$. The following fact is immediate from the definitions. It links $\gamma$ to $\beta$.

**Fact 6.9.** For any $w \in (b^+C^+)^+$, $\beta(w) \subseteq \gamma((w)_{\mathfrak{B}})$.

**Proof of Lemma 6.7, Step 2: Constructing a partition of $\mathfrak{B}^+$.** We use induction to partition $\mathfrak{B}^+$. That we may apply induction to $\gamma$ is a consequence of the following fact, which is where we use our choice of $b$ (*i.e.*, the fact that $\beta$ is not tame).

**Fact 6.10.** The index of $\mathcal{T}$ is strictly smaller than the index of $\mathcal{S}$.

*Proof.* By definition, for any language $H \in \mathbf{H}$, we have $[\![\beta(H)]\!] \subseteq [\![\beta(b^+)]\!]$. Hence,

$$[\![\mathcal{T}]\!] \subseteq [\![\beta(b^+)]\!] \cdot [\![\mathcal{S}]\!] = [\![\beta(b^+) \cdot \mathcal{S}]\!] \subseteq \beta(b) \cdot [\![\mathcal{S}]\!].$$

By definition of $b$, we know that $\beta(b) \cdot [\![\mathcal{S}]\!] \subsetneq [\![\mathcal{S}]\!]$. We conclude that $[\![\mathcal{T}]\!] \subsetneq [\![\mathcal{S}]\!]$ which terminates the proof. □

It follows from Fact 6.10 that we may apply induction on our first induction parameter (the index of $\mathcal{S}$) to $\gamma$ and obtain an FO-partition $\mathbf{F}$ of $\mathfrak{B}^+$ such that for all $F \in \mathbf{F}$:

(1) $[\![\gamma(F)]\!] \in \mathrm{Sat}(\mathcal{T})$.
(2) $F$ can be defined with a first-order formula of rank at most $|\mathfrak{B}| \cdot 2^{|[\![\mathcal{T}]\!]|^2}$.

**Proof of Lemma 6.7, Step 3: Constructing the partition $\mathbf{K'}$ of $(b^+C^+)^+$.** For any $F \in \mathbf{F}$, we define $K_F = \{w \in (b^+C^+)^+ \mid (w)_{\mathfrak{B}} \in F\}$. Finally, we set $\mathbf{K'} = \{K_F \mid F \in \mathbf{F}\}$. Since $\mathbf{F}$ is a partition of $\mathfrak{B}^+$, it is immediate that $\mathbf{K'}$ is a partition of $(b^+C^+)^+$. It now remains to prove that Items (1) and (2) in Lemma 6.7 hold.

Let us first prove that Item (1) is satisfied. Set $K \in \mathbf{K'}$. By definition, $K = K_F$ for some $F \in \mathbf{F}$. By definition of $K_F$ and Fact 6.9, we have that,

$$[\![\beta(K)]\!] \subseteq [\![\gamma(F)]\!]$$

Moreover, by construction of $\mathbf{F}$, we know that $[\![\gamma(F)]\!] \in \mathrm{Sat}(\mathcal{T})$. Finally, since $\mathcal{T} \subseteq \mathcal{S}$, we have $\mathrm{Sat}(\mathcal{T}) \subseteq \mathrm{Sat}(\mathcal{S})$. Using closure under downset, we obtain that $[\![\beta(K)]\!] \in \mathrm{Sat}(\mathcal{S})$ which terminates the proof of Item (1).

It now remains to prove that Item (2) holds. Set $K \in \mathbf{K'}$. We need to construct an FO formula of rank at most $|B| \cdot 2^{|[\![\mathcal{S}]\!]|^2} - 2$ that defines $K$. Note that it is immediate that $(b^+C^+)^+$ can be defined in FO (with a formula of rank 2): this amounts to testing that the first letter in the word is a $b$ and the last is a letter of $C$. Therefore it suffices to construct a formula $\varphi_K$ such that for all $w \in (b^+C^+)^+$, $w \models \varphi_K$ if and only if $w \in K$.

By construction, there exists $F \in \mathbf{F}$ such that $K = K_F$ as well as an FO formula $\Psi_F$ (over $\mathfrak{B}$) of rank less than $|\mathfrak{B}| \cdot 2^{|[\![\mathcal{T}]\!]|^2}$ that defines $F$. By definition of $K = K_F$, it suffices to construct $\varphi_K$ so that for any $w \in (b^+C^+)^+$,

$$w \models \varphi_K \quad \text{if and only if} \quad (w)_{\mathfrak{B}} \models \Psi_F$$

The construction is standard, we build $\varphi_K$ by modifying $\Psi_F$. Consider a word $w \in (b^+C^+)^+$. We say that a position $x$ in $w$ is *distinguished* if and only if $x$ is labeled by a "$b$" and position $(x+1)$ has label in $C$. In other words $x$ is the rightmost $b$-labeled position of an infix in $b^+C^+$ of $w$. Recall that by definition, every letter of $(w)_{\mathfrak{B}}$ abstracts an infix in $b^+C^+$ of $w$. Therefore, one can associate a position $\hat{x}$ of $(w)_{\mathfrak{B}}$ to every distinguished position $x$ of $w$.

**Fact 6.11.** For every $\mathfrak{b} \in \mathfrak{B}$, there exists a first-order formula $\overline{\mathfrak{b}}(x)$ of rank at most $(|B| - 1) \cdot 2^{|[\![\mathcal{S}]\!]|^2}$ such that for any $w \in (b^+C^+)^+$ and any distinguished position $x$ of $w$:

$$w, x \models \overline{\mathfrak{b}}(x) \text{ if and only if } w_{\mathfrak{B}}, \hat{x} \models \mathfrak{b}(\hat{x}). \tag{6.1}$$

*Proof.* This amounts to testing whether the maximal infix in $b^+$ ending at position $x$ in $w$ and the maximal infix in $C^+$ starting at position $x+1$ in $w$ are in the appropriate languages of $\mathbf{H}$ and $\mathbf{L}$ that yield letter $\mathfrak{b}$. This can easily be done with rank at most $(|B|-1)\cdot 2^{|\!|\!|\mathcal{S}|\!|\!|^2}$ since any language in $\mathbf{H}$ or $\mathbf{L}$ can be defined by a formula of rank at most $(|B|-1)\cdot 2^{|\!|\!|\mathcal{S}|\!|\!|^2}$ (see Lemma 6.5 and Lemma 6.6). □

The desired formula $\varphi_K$ is obtained from $\Psi_F$ by restricting quantifications to distinguished positions and replacing each atomic subformula of the form $\mathfrak{b}(x)$ by the formula $\overline{\mathfrak{b}}(x)$. This can clearly be done in first-order logic. Observe that this formula has rank at most $r = |\mathfrak{B}|\cdot 2^{|\!|\!|\mathcal{T}|\!|\!|^2} + (|B|-1)\cdot 2^{|\!|\!|\mathcal{S}|\!|\!|^2}$. Since $|\mathfrak{B}| \leqslant 2^{|\!|\!|\mathcal{T}|\!|\!|}$ and $1 \leqslant ||\!|\mathcal{T}|\!|| \leqslant |\!|\!|\mathcal{S}|\!|\!| - 1$, we obtain:

$$r \leqslant |B|\cdot 2^{|\!|\!|\mathcal{S}|\!|\!|^2} - (2^{|\!|\!|\mathcal{S}|\!|\!|^2} - 2^{|\!|\!|\mathcal{S}|\!|\!|^2 - |\!|\!|\mathcal{S}|\!|\!|}) \leqslant |B|\cdot 2^{|\!|\!|\mathcal{S}|\!|\!|^2} - 2$$

Note that the last inequality is justified by the fact that $|\!|\!|\mathcal{S}|\!|\!| \geqslant 2$, which holds in this case since otherwise $\mathcal{S}$ would be the trivial group. This terminates the proof of Lemma 6.7.

## 7. Alternate Algorithms

In this section, we connect our algorithm with the ones of Henckell [11] and Henckell, Rhodes and Steinberg [12, 26]. These algorithms (there are two of them) differ in the specific FO-operation. Although the change is minor and the correspondence between these algorithms is easy to prove, this is what brings a complexity improvement, from Expspace to Exptime. We include this easy section to bridge the gap between all three algorithms. Note that the fact that the two given by Henckell are equivalent was already shown in [11].

The well-known decidable characterization of first-order logic by Schützenberger [27, 15] states that a language is first-order *definable* if and only if its syntactic semigroup is *aperiodic*. In the literature, there are many equivalent definitions of aperiodicity. In this paper, we consider three of them: one is equational, the second considers subgroups and the third considers the $\mathscr{H}$-classes. The relation '$\mathscr{H}$' is one of Green's relations which are well known in semigroup theory. Two elements $s, s'$ of a semigroup $S$ are $\mathscr{H}$-equivalent if $s = s'$ or there exist $t_\ell, t'_\ell, t_r, t'_r \in S$ such that $st_r = s'$, $s't'_r = s$, $t_\ell s = s'$ and $t'_\ell s' = s$. We state the three equivalent definitions.

**Lemma 7.1** (Folklore, see [18])**.** *A finite semigroup $S$ is aperiodic if and only if it satisfies one of the following equivalent statements:*

(1) *for all $s \in S$, $s^\omega = s^{\omega+1}$.*
(2) *all subgroups in $S$ are trivial.*
(3) *all $\mathscr{H}$-classes in $S$ are trivial.*

Our saturation procedure Sat can be viewed as a generalization of the first definition of aperiodicity. Indeed, Operation (3) reflects the equation $s^\omega = s^{\omega+1}$. In this section, we present two alternate and equivalent saturation procedures that reflect the two other definitions. Let $\alpha : A^+ \to S$ be a morphism into a finite semigroup.

Let $\mathcal{S}$ be a subsemigroup of $2^S$. We set $\mathrm{Sat}_G(\mathcal{S})$ as the smallest subset of $2^S$ that contains $\mathcal{S}$ and is closed under downset, multiplication and the following operation:

$$\text{for all } \mathcal{G} \subseteq \mathrm{Sat}_G(\mathcal{S}) \text{ that is a group, } |\!|\!|\mathcal{G}|\!|\!| \in \mathrm{Sat}_G(\mathcal{S}). \tag{7.1}$$

Similarly, $\mathrm{Sat}_H(\mathcal{S})$ is the smallest subset of $2^S$ that contains $\mathcal{S}$ and is closed under downset, multiplication and the following operation:

$$\text{for all } \mathcal{H} \subseteq \mathrm{Sat}_H(\mathcal{S}) \text{ that is an } \mathscr{H}\text{-class, } [\![\mathcal{H}]\!] \in \mathrm{Sat}_H(\mathcal{S}). \tag{7.2}$$

$\mathrm{Sat}_G$ reflects the second definition of aperiodicity and $\mathrm{Sat}_H$ the third. In the following proposition, we state that the three saturation procedures are equivalent and can therefore all be used to compute $\mathcal{I}[\alpha]$ by Proposition 4.7.

**Proposition 7.2.** *Let $\mathcal{S}$ be a subsemigroup of $2^S$. Then,*

$$\mathrm{Sat}(\mathcal{S}) = \mathrm{Sat}_G(\mathcal{S}) = \mathrm{Sat}_H(\mathcal{S}).$$

Note that the saturation procedure $\mathrm{Sat}_H$ is essentially Henckell's original algorithm [11], where $\mathrm{Sat}_G$ was also shown to be a correct saturation operation. We finish the section by proving Proposition 7.2.

*Proof.* We prove that $\mathrm{Sat}(\mathcal{S}) \subseteq \mathrm{Sat}_H(\mathcal{S}) \subseteq \mathrm{Sat}_G(\mathcal{S}) \subseteq \mathrm{Sat}(\mathcal{S})$. Let us first prove that $\mathrm{Sat}(\mathcal{S}) \subseteq \mathrm{Sat}_H(\mathcal{S})$.

**$\mathrm{Sat}(\mathcal{S}) \subseteq \mathrm{Sat}_H(\mathcal{S})$.** By definition of $\mathrm{Sat}(\mathcal{S})$ and $\mathrm{Sat}_H(\mathcal{S})$, this amounts to proving that $\mathrm{Sat}_H(\mathcal{S})$ is closed under FO-closure: for any $T \in \mathrm{Sat}_H(\mathcal{S})$, $T^\omega \cup T^{\omega+1} \in \mathrm{Sat}_H(\mathcal{S})$.

Set $T \in \mathrm{Sat}_H(\mathcal{S})$. Observe that $T^{\omega+1}$ and $T^\omega$ are $\mathscr{H}$-equivalent elements in the semigroup $\mathrm{Sat}_H(\mathcal{S})$, and are therefore both contained in some $\mathscr{H}$-class $\mathcal{H} \subseteq \mathrm{Sat}_H(\mathcal{S})$. By definition of $\mathrm{Sat}_H(\mathcal{S})$, we then have $[\![\mathcal{H}]\!] \in \mathrm{Sat}_H(\mathcal{S})$. Hence, $T^\omega \cup T^{\omega+1} \subseteq [\![\mathcal{H}]\!] \in \mathrm{Sat}_H(\mathcal{S})$, which ends the proof since $\mathrm{Sat}_H(\mathcal{S})$ is closed under downset.

**$\mathrm{Sat}_H(\mathcal{S}) \subseteq \mathrm{Sat}_G(\mathcal{S})$.** This inclusion, which is easy to prove, follows from [11]. We give here a proof for the sake of completeness.

By definition of $\mathrm{Sat}_H(\mathcal{S})$ and $\mathrm{Sat}_G(\mathcal{S})$, this amounts to proving that $\mathrm{Sat}_G(\mathcal{S})$ is closed under 7.2: for all $\mathcal{H} \subseteq \mathrm{Sat}_G(\mathcal{S})$ that is an $\mathscr{H}$-class, $[\![\mathcal{H}]\!] \in \mathrm{Sat}_G(\mathcal{S})$.

Let $\mathcal{H} \subseteq \mathrm{Sat}_G(\mathcal{S})$ be an $\mathscr{H}$-class. We claim that either $\mathcal{H}$ is a singleton, or there exists a group $\mathcal{G}$ in $\mathrm{Sat}_G(\mathcal{S})$ and $R \in \mathrm{Sat}_G(\mathcal{S})$ such that $\mathcal{H} = R \cdot \mathcal{G}$. If $\mathcal{H}$ is a singleton, then $[\![\mathcal{H}]\!]$ is the unique element of $\mathcal{H}$ which belongs to $\mathrm{Sat}_G(\mathcal{S})$. Otherwise, using closure under multiplication, it follows that $[\![\mathcal{H}]\!] = R[\![\mathcal{G}]\!] \in \mathrm{Sat}_G(\mathcal{S})$ since $[\![\mathcal{G}]\!] \in \mathrm{Sat}_G(\mathcal{S})$ by Operation 7.1 in the definition of $\mathrm{Sat}_G$.

It remains to prove the claim (which actually is not specific to subsemigroups of a semigroup of subset): every $\mathscr{H}$-class $\mathcal{H}$ of a semigroup $\mathcal{T}$ is either a singleton, or of the form $R \cdot \mathcal{G}$, for $R \in \mathcal{T}$ and $\mathcal{G}$ a group in $\mathcal{T}$. Let $\mathrm{Stab} = \{T \in \mathcal{T} \mid \mathcal{H} \cdot T = \mathcal{H}\}$. If $\mathcal{H}$ is not a singleton, then Green's Lemma implies that Stab is nonempty, and therefore it is a subsemigroup of $\mathcal{T}$. Let $\mathcal{G}$ be an $\mathscr{H}$-class of its minimal ideal. By standard results in semigroup theory [18, Chapter V], $\mathcal{G}$ is a group. Let us check that $\mathcal{H} = H \cdot \mathcal{G}$, for any $H \in \mathcal{H}$. Indeed, let $H \in \mathcal{H}$ and let $E$ be the identity of $\mathcal{G}$. Since $E \in \mathrm{Stab}$, we have $H = H'E$ for some $H' \in \mathcal{H}$, and so $HE = H$. Let now $H_1 \in \mathcal{H}$. By definition, we have $H_1 = H \cdot X$ for some $X \in \mathcal{T}$. Note that since $\mathcal{G}$ is in the minimal ideal, we have $EXE \in \mathcal{G}$. Hence $H_1 = H_1E = HXE = H(EXE) \in H\mathcal{G}$. This proves the claim and establishes the inclusion.

**$\mathrm{Sat}_G(\mathcal{S}) \subseteq \mathrm{Sat}(\mathcal{S})$.** By definition of $\mathrm{Sat}_G(\mathcal{S})$ and $\mathrm{Sat}(\mathcal{S})$, this amounts to proving that $\mathrm{Sat}(\mathcal{S})$ is closed under (7.1): for all $\mathcal{G} \subseteq \mathrm{Sat}(\mathcal{S})$ that is a group, $[\![\mathcal{G}]\!] \in \mathrm{Sat}(\mathcal{S})$.

Set $\mathcal{G} \subseteq \mathrm{Sat}(\mathcal{S})$ that is a group and set $\mathcal{G} = \{T_1, \ldots, T_n\}$ with $T_i \in \mathrm{Sat}(\mathcal{S})$ and let $1_\mathcal{G}$ be the identity element of $\mathcal{G}$. Since $\mathcal{G}$ is a group, for all $i$, $T_i^\omega = 1_\mathcal{G}$. In particular this means

that for all $i$, $T_i = T_1^\omega \cdots T_{i-1}^\omega T_i^{\omega+1} T_{i+1}^\omega \cdots T_n^\omega$. By combining these equalities, we get

$$[\![\mathcal{G}]\!] = T_1 \cup \cdots \cup T_n \subseteq (T_1^\omega \cup T_1^{\omega+1}) \cdots (T_n^\omega \cup T_n^{\omega+1}).$$

It follows from FO-closure and closure under multiplication that $[\![\mathcal{G}]\!] \in \mathrm{Sat}(\mathcal{S})$. □

## 8. Infinite Words

An advantage of our technique for proving Theorem 3.2 is that it generalizes smoothly to the setting of infinite words, *i.e.*, it can be adapted to prove that FO-separability is decidable for infinite words. Both the algorithm itself and its proof are very similar to those of the finite words setting. In particular, we retain all results that we already have for finite words:

- we get an Exptime upper bound on the complexity of the problem.
- we get an exponential upper bound on the quantifier rank of a potential separator.
- the proof is constructive: if a separator exists, one can be constructed by induction.

The remainder of the paper is devoted to the presentation of this generalization. In this section, we introduce $\omega$-words and generalize our definitions to this setting: we define $\omega$-languages, $\omega$-semigroups and first-order logic over $\omega$-words. We postpone the presentation of the separation algorithm itself (which requires generalizing our framework to $\omega$-words) to the next section, Section 9. Finally, Section 10 is devoted to the proof of this algorithm.

### 8.1. Regular Languages of $\omega$-words.

**$\omega$-words and $\omega$-languages.** Recall that $A$ is a finite alphabet. We denote by $A^\infty$ the set of infinite words, called *$\omega$-words* over $A$. Note that we still use the term "word" to mean an element of $A^+$. If $u$ is a word and $v$ an $\omega$-word, we denote by $u \cdot v$ or $uv$ the $\omega$-word obtained by concatenating $u$ to the left of $v$, and by $u^\infty$ the $\omega$-word obtained by infinite concatenation of $u$ with itself[3]. An *$\omega$-language* is a subset of $A^\infty$. *Regular* $\omega$-languages are those that are accepted by *nondeterministic Büchi automata* (NBA). Again, we will only work with the algebraic representation of $\omega$-languages that we recall below.

**$\omega$-semigroups.** We briefly recall the definition of $\omega$-semigroups, which play the role of semigroups in the setting of $\omega$-words. For more details, we refer the reader to [17].

An *$\omega$-semigroup* is a pair $(S_+, S_\infty)$ where $S_+$ is a semigroup and $S_\infty$ is a set. Moreover, $(S_+, S_\infty)$ is equipped with two additional products: a *mixed product* $S_+ \times S_\infty \to S_\infty$ that maps $s, t \in S_+, S_\infty$ to an element denoted $st$, and an *infinite product* $(S_+)^\infty \to S_\infty$ that maps an infinite sequence $s_1, s_2, \cdots \in (S_+)^\infty$ to an element of $S_\infty$ denoted by $s_1 s_2 \cdots$. We require these products as well as the semigroup product of $S_+$ to satisfy all possible forms of associativity (see [17] for details). Finally, we denote by $s^\infty$ the element $sss\cdots$. Observe that $(A^+, A^\infty)$ is an $\omega$-semigroup.

The notions of subsemigroups and morphisms can be adapted to $\omega$-semigroups. In particular, if $T_+$ is a subsemigroup of $S_+$ and $T_\infty$ is the set obtained by applying the infinite product to all sequences of $T_+$, then $(T_+, T_\infty)$ is a sub-$\omega$-semigroup of $(S_+, S_\infty)$ called the *sub-$\omega$-semigroup generated by $T_+$*.

An $\omega$-semigroup is said to be *finite* if both $S_+$ and $S_\infty$ are finite. Note that even if an $\omega$-semigroup is finite, it is not obvious that a finite representation of the infinite product

[3]In the literature, the $\omega$-word $u^\infty$ is usually denoted by $u^\omega$. Here, we use this non standard notation in order to avoid confusion with the idempotent power $\omega$ in semigroups.

exists. However, it was proven by Wilke [35] that the infinite product is fully determined by the mapping $s \mapsto s^\infty$, yielding a finite representation for finite $\omega$-semigroups. An $\omega$-language $L$ is said to be *recognized* by an $\omega$-semigroup $(S_+, S_\infty)$ if there exists $F \subseteq S_\infty$ as well as a morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ such that $L = \alpha^{-1}(F)$. It is well known that an $\omega$-language is regular if and only if it is recognized by a *finite* $\omega$-semigroup. Moreover [35], from any NBA recognizing $L$, one can compute a canonical smallest $\omega$-semigroup recognizing $L$, called the *syntactic $\omega$-semigroup of $L$*.

As for finite words, when working on separation, it is convenient to consider a single recognizing object for both input languages rather than two separate objects. Again, this is not restrictive: given two $\omega$-languages and two associated recognizing $\omega$-semigroups, one can define (and compute) a single $\omega$-semigroup that recognizes both languages by taking the Cartesian product of the two original $\omega$-semigroups.

**Semigroup of Subsets.** For an $\omega$-semigroup $(S_+, S_\infty)$, note that $(2^{S_+}, 2^{S_\infty})$ is an $\omega$-semigroup with the products defined in the natural way.

### 8.2. First-Order Logic over $\omega$-words.

As for words, an $\omega$-word can be viewed as a sequence of positions that are labeled over $A$ (the difference being that in the case of $\omega$-words, the sequences are infinite: there is a leftmost position but no rightmost one). Therefore, first-order formulas as we defined them can also be interpreted on $\omega$-words and we can simply say that an $\omega$-language $L \subseteq A^\infty$ is first-order definable if and only if there exists an FO formula $\varphi$ such that $L = \{w \in A^\infty \mid w \models \varphi\}$.

First-order logic over $\omega$-words shares similar properties with first-order logic over words. First, the equivalence with star-free languages still holds for $\omega$-languages: an $\omega$-language is first-order definable if and only if it is star-free [14, 33]. Furthermore, Schützenberger's Theorem was generalized to $\omega$-languages by Perrin [16]: a regular $\omega$-language is star-free (and hence FO) if and only if the finite semigroup $S_+$ of its syntactic $\omega$-semigroup $(S_+, S_\infty)$ is aperiodic. Note that we obtain an alternate proof of this theorem as a simple consequence of our separation algorithm.

Our main theorem for $\omega$-languages is similar to Theorem 3.2 for languages and is as follows.

**Theorem 8.1.** *Let $L_0, L_1$ be regular $\omega$-languages recognized by a morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ into a finite $\omega$-semigroup. The two following items hold.*

1. *One can decide in* Exptime *with respect to $|S_+|$ whether $L_0$ and $L_1$ are FO-separable.*
2. *When $L_0$ and $L_1$ are FO-separable, one can construct an actual separator with a formula of quantifier rank at most $|A|2^{|S_+|^2} + 1$.*

The proof of Theorem 8.1 is very similar to the one of Theorem 3.2 and relies on the same objects: FO-partitions and optimal imprints (generalized to $\omega$-words). In particular, this means that our proof remains constructive: it yields an inductive way to construct an actual separator, *i.e.*, an FO-partition of $A^\infty$ that is optimal for the input morphism $\alpha$, when it exists (a rough analysis yields a 2-Exptime complexity in $|S_+|$).

## 9. Separation Algorithm for Infinite Words

In this section, we present our separation algorithm for first-order logic over $\omega$-words. As we explained, this algorithm is based on a generalization of our finite words framework to $\omega$-words. Therefore, we divide this section in two parts. In the first part, we generalize FO-partitions, imprints and Theorem 4.6 to $\omega$-words. Then, in the second part, we present our separation algorithm.

9.1. **Definition.** Recall that a finite alphabet $A$ is fixed. For the definitions, we let $\mathcal{C}$ as an arbitrary class consisting of languages and $\omega$-languages (*i.e.*, $\mathcal{C} \subseteq 2^{A^+} \cup 2^{A^\infty}$). Moreover, we assume that

- when restricted to languages, $\mathcal{C}$ is nonempty, closed under Boolean operations and quotients, and contains only regular languages,
- when restricted to $\omega$-languages, $\mathcal{C}$ is nonempty and closed under Boolean operations.

One can verify that FO satisfies these conditions. Note that since $\mathcal{C}$ is assumed to contain both languages and $\omega$-languages, one can consider two kinds of $\mathcal{C}$-partitions: $\mathcal{C}$-partitions of $A^+$ and $\mathcal{C}$-partitions of $A^\infty$.

Set $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ as an arbitrary morphism into a finite $\omega$-semigroup $(S_+, S_\infty)$. Observe that any such morphism $\alpha$ can be decomposed into two maps: a morphism $\alpha_+ : A^+ \to S_+$ into a finite semigroup $S_+$ and a map $\alpha_\infty : A^\infty \to S_\infty$ into a finite set $S_\infty$.

Since $\alpha_+$ is a morphism, we may directly apply our definition of imprints for finite words to it: if $\mathbf{K}$ is a $\mathcal{C}$-partition of $A^+$, then $\mathcal{I}[\alpha_+](\mathbf{K}) \subseteq 2^{S_+}$ is well-defined. Similarly, by hypothesis on $\mathcal{C}$, the optimal $\mathcal{C}$-partitions of $A^+$ for $\alpha_+$ are well-defined as those having the smallest possible imprint on $\alpha_+$: $\mathcal{I}_\mathcal{C}[\alpha_+]$. In particular, we know from Lemma 4.5 and our hypothesis on $\mathcal{C}$ that $\mathcal{I}_\mathcal{C}[\alpha_+]$ is a subsemigroup of $2^{S_+}$.

It turns out that aside from Lemma 4.5, these definitions do not require $\alpha_+$ to be a semigroup morphism. Hence, they can also be applied to $\alpha_\infty$. If $\mathbf{K}$ is a $\mathcal{C}$-partition of $A^\infty$, then the imprint of $\mathbf{K}$ on $\alpha_\infty$ is defined by,

$$\mathcal{I}[\alpha_\infty](\mathbf{K}) = \{T \in 2^{S_\infty} \mid \text{there exists } K \in \mathbf{K} \text{ such that } T \subseteq \alpha_\infty(K)\} \subseteq 2^{S_\infty}.$$

Note that one can verify that imprints on $\alpha_\infty$ still verify Fact 4.1 (*i.e.*, for all $w \in A^\infty$, $\{\alpha_\infty(w)\} \in \mathcal{I}[\alpha_\infty](\mathbf{K})$) and Fact 4.2 (*i.e.*, $\mathcal{I}[\alpha_\infty](\mathbf{K})$ is closed under downset). Finally, the optimal $\mathcal{C}$-partitions of $A^\infty$ for $\alpha_\infty$ are defined as those having the smallest possible imprint on $\alpha_\infty$: $\mathcal{I}_\mathcal{C}[\alpha_\infty]$ (as before, we need the fact that $\mathcal{C}$ is closed under intersection to prove that there exists at least one optimal $\mathcal{C}$-partition, see Lemma 4.4).

**Remark 9.1.** *Note that in this case, since $S_\infty$ is not a semigroup, it is not true that $\mathcal{I}_\mathcal{C}[\alpha_\infty]$ is a semigroup. However, with additional hypotheses on $\mathcal{C}$ (which correspond to the usual generalization of closure under quotients to classes of $\omega$-languages), one could prove that the pair $(\mathcal{I}_\mathcal{C}[\alpha_+], \mathcal{I}_\mathcal{C}[\alpha_\infty])$ is a sub-$\omega$-semigroup of $(2^{S_+}, 2^{S_\infty})$. We will prove this property in the special case where $\mathcal{C} = FO$.*

We can now generalize Theorem 4.6 to $\omega$-words.

**Theorem 9.2.** *Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be a morphism into a finite $\omega$-semigroup $(S_+, S_\infty)$. Let $L_1, L_2 \subseteq A^\infty$ be two $\omega$-languages recognized by $\alpha$ and let $T_1, T_2 \subseteq S_\infty$ be the corresponding accepting sets. The following properties are equivalent:*

(1) *$L_1$ and $L_2$ are $\mathcal{C}$-separable.*

(2) *for all* $t_1 \in T_1$ *and all* $t_2 \in T_2$, $\{t_1, t_2\} \notin \mathcal{I}_{\mathcal{C}}[\alpha_\infty]$.
(3) *for any* $\mathcal{C}$*-partition* $\mathbf{K}$ *of* $A^\infty$ *that is optimal for* $\alpha_\infty$, $L_1$ *and* $L_2$ *are separable by a union of languages in* $\mathbf{K}$.

The proof of Theorem 9.2 is identical to that of Theorem 4.6. In view of the theorem, generalizing our approach to $\omega$-languages amounts to finding an algorithm that computes $\mathcal{I}_{\mathcal{C}}[\alpha_\infty]$ from a morphism $\alpha$ into a finite $\omega$-semigroup. We now present such an algorithm.

9.2. **Separation Algorithm.** We can now generalize our separation algorithm to the setting of $\omega$-words. Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be a morphism into a finite $\omega$-semigroup $(S_+, S_\infty)$. From now on, we only work with the class FO, therefore, we simply write $(\mathcal{I}[\alpha_+], \mathcal{I}[\alpha_\infty])$ to denote the pair $(\mathcal{I}_{\mathrm{FO}}[\alpha_+], \mathcal{I}_{\mathrm{FO}}[\alpha_\infty])$. We present an algorithm for computing this pair.

We already know how to compute $\mathcal{I}[\alpha_+]$ from $\alpha_+$: $\mathcal{I}[\alpha_+] = \mathrm{Sat}(\mathcal{S}_+)$ with "Sat" defined in Section 4 (see Proposition 4.7). It turns out that $\mathcal{I}[\alpha_\infty]$ can easily be computed from $\mathcal{I}[\alpha_+]$. For $\mathcal{S} \subseteq 2^{S_+}$, let $\mathrm{Sat}_\infty(\mathcal{S})$ be the smallest subset of $2^{S_\infty}$ closed under the following operations:

(1) For any $T \in \mathcal{S}$, we have $T^\infty \in \mathrm{Sat}_\infty(\mathcal{S})$.
(2) For any $T \in \mathcal{S}$ and $T' \in \mathrm{Sat}_\infty(\mathcal{S})$, we have $TT' \in \mathrm{Sat}_\infty(\mathcal{S})$.
(3) $\mathrm{Sat}^\infty(\mathcal{S})$ is closed under downset: $\mathrm{Sat}_\infty(\mathcal{S}) = {\downarrow}\,\mathrm{Sat}_\infty(\mathcal{S})$.

In other words, $\mathrm{Sat}_\infty(\mathcal{S})$ is the smallest subset of $2^{S_\infty}$ that is closed under downset and such that $(\mathcal{S}, \mathrm{Sat}_\infty(\mathcal{S}))$ is a sub-$\omega$-semigroup of $(2^{S_+}, 2^{S_\infty})$. This smallest subset of $2^{S_\infty}$ clearly exists. Finally, we set $\mathrm{Sat}_\infty(\alpha)$ as $\mathrm{Sat}_\infty(\mathcal{I}[\alpha_+])$.

**Proposition 9.3.** *Set* $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ *as a morphism into a finite* $\omega$*-semigroup* $(S_+, S_\infty)$. *Then,*

$$\mathcal{I}[\alpha_\infty] = \mathrm{Sat}_\infty(\alpha).$$

Since we already know how to compute $\mathcal{I}[\alpha_+]$ in Exptime with respect to $|S_+|$ (see Proposition 4.7), it follows from Proposition 9.3 that one can compute $\mathcal{I}[\alpha_\infty]$ in Exptime with respect to $|S_+|$ as well. It then follows from Theorem 9.2 that this generalizes our upper bound on the complexity of the separation problem to $\omega$-languages: one can decide in Exptime whether two $\omega$-languages are FO-separable. Therefore, we obtain the first item in Theorem 8.1 as a corollary. We will obtain the second item as a byproduct of the proof of Proposition 9.3.

Another important remark is that it follows from Proposition 9.3 that $(\mathcal{I}[\alpha_+], \mathcal{I}[\alpha_\infty])$ is a sub-$\omega$-semigroup of $(2^{S_+}, 2^{S_\infty})$. As explained in Remark 9.1, this property is not specific to FO. On the other hand, what is specific to FO is that $\mathcal{I}[\alpha_\infty]$ is the **smallest** subset of $2^{S^\infty}$ that is closed under downset and such that $(\mathcal{I}[\alpha_+], \mathcal{I}[\alpha_\infty])$ is a sub-$\omega$-semigroup of $(2^{S_+}, 2^{S_\infty})$.

Finally, a consequence of Proposition 9.3 is that we obtain Perrin's theorem [16] as a corollary, just as we obtained Schützenberger's one [27] as a corollary of Proposition 4.7.

**Corollary 9.4.** *Let* $L$ *be a regular* $\omega$*-language. Then* $L$ *can be defined in FO if and only if its syntactic* $\omega$*-semigroup* $(S_+, S_\infty)$ *is such that* $S_+$ *is aperiodic.*

*Proof.* The proof is similar to that of Corollary 4.8. It is known that an $\omega$-language is definable in FO if and only if all languages and $\omega$-languages recognized by its syntactic $\omega$-semigroup are definable in FO as well (as before, this is actually not specific to FO and

true for all classes of $\omega$-languages that are "Varieties", see [17] for example). It follows that, if $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ is the syntactic $\omega$-semigroup of $L$, then $L$ is definable in FO if and only if $\mathcal{I}[\alpha_+]$ and $\mathcal{I}[\alpha_\infty]$ contain only singletons and the empty set. One can then verify from Proposition 4.7 that this is is equivalent to $S_+$ satisfying, $s^\omega = s^{\omega+1}$ for all $s \in S_+$. □

It now remains to prove Proposition 9.3. We present this proof in the next section, Section 10.

## 10. Correctness of the Infinite Words Algorithm

This section is devoted to the proof Proposition 9.3. We fix a morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ into a finite $\omega$-semigroup $(S_+, S_\infty)$ for the whole section. We have to prove that $\mathcal{I}[\alpha_\infty] = \mathrm{Sat}_\infty(\alpha)$. We separate the proof in two parts, each one corresponding to an inclusion.

10.1. **Soundness of the Algorithm.** We begin with the easiest inclusion: $\mathrm{Sat}_\infty(\alpha) \subseteq \mathcal{I}[\alpha_\infty]$. This corresponds to soundness of the algorithm: it only computes sets belonging to $\mathcal{I}[\alpha_\infty]$. By definition of $\mathrm{Sat}_\infty(\alpha)$, we need to prove that:

- For any $T \in \mathcal{I}[\alpha_+]$, we have $T^\infty \in \mathcal{I}[\alpha_\infty]$.
- For any $T \in \mathcal{I}[\alpha_+]$ and $T' \in \mathcal{I}[\alpha_\infty]$, we have $TT' \in \mathcal{I}[\alpha_\infty]$.
- $\mathcal{I}[\alpha_\infty]$ is closed under downset.

That $\mathcal{I}[\alpha_\infty]$ is closed under downset is immediate from the definition ($\mathcal{I}[\alpha_\infty]$ is an imprint). We prove the two other items. The proof relies on the generalization of the equivalence $\equiv_k$ to $\omega$-words: given two $\omega$-words $w, w' \in A^\infty$ and $k \in \mathbb{N}$, we write $w \equiv_k w'$ to denote the fact that $w$ and $w'$ satisfy the same formulas of quantifier rank $k$. One can verify that Lemma 5.1 still holds for $\omega$-words.

**Lemma 10.1.** *Let $T \in 2^{S_\infty}$. Then $T \in \mathcal{I}[\alpha_\infty]$ if and only if for all $k \in \mathbb{N}$, there exists an equivalence class $W \subseteq A^\infty$ of $\equiv_k$ such that $T \subseteq \alpha(W)$.*

We can now finish the proof of soundness. Set $T \in \mathcal{I}[\alpha_+]$ and $T' \in \mathcal{I}[\alpha_\infty]$. We use Lemmas 10.1 to prove that $T^\infty \in \mathcal{I}[\alpha_\infty]$ and $TT' \in \mathcal{I}[\alpha_\infty]$. Set $k \in \mathbb{N}$.

By Lemmas 5.1 and 10.1, we obtain an equivalence class $W \subseteq A^+$ of $\equiv_k$ (over finite words) and an equivalence class $W' \subseteq A^\infty$ of $\equiv_k$ (over $\omega$-words) such that $T \subseteq \alpha(W)$ and $T' \subseteq \alpha(W')$. We know from Lemma 10.1 that it suffices to prove that $W^\infty$ and $WW'$ are included in equivalence classes of $\equiv_k$ in order to conclude that $T^\infty \in \mathcal{I}[\alpha_\infty]$ and $TT' \in \mathcal{I}[\alpha_\infty]$. This can be easily verified using a generalization of the first item of Lemma 5.2 to $\omega$-words: for any $w \in W$ and $w' \in W'$, one can verify that any $\omega$-word in $W^\infty$ is $\equiv_k$-equivalent to $w^\infty$ and that any $\omega$-word in $WW'$ is $\equiv_k$-equivalent to $ww'$.

10.2. **Completeness of the Algorithm.** We now turn to the most interesting inclusion in Proposition 9.3: $\mathcal{I}[\alpha_\infty] \subseteq \mathrm{Sat}_\infty(\alpha)$. The proof is a generalization of that of Proposition 6.1 to the setting of $\omega$-words. In particular, the proof remains constructive: we use induction to construct an FO-partition $\mathbf{K}$ of $A^\infty$ whose imprint on $\alpha_\infty$ is included in $\mathrm{Sat}_\infty(\alpha)$. This proves that $\mathcal{I}[\alpha_\infty] \subseteq \mathcal{I}[\alpha_\infty](\mathbf{K}) \subseteq \mathrm{Sat}_\infty(\alpha)$. The induction is stated in the following proposition.

**Proposition 10.2.** *Let $(\mathcal{S}_+, \mathcal{S}_\infty)$ be a sub-$\omega$-semigroup of $(2^{S_+}, 2^{S_\infty})$ and let $\beta : (B^+, B^\infty) \to (\mathcal{S}_+, \mathcal{S}_\infty)$ be a surjective morphism. Then there exists an FO-partition $\mathbf{K}$ of $B^\infty$ such that for all $K \in \mathbf{K}$:*

(1) $\llbracket \beta(K) \rrbracket \in \mathrm{Sat}_\infty(\mathrm{Sat}(\mathcal{S}_+))$.
(2) *$K$ can be defined by a first-order formula of rank at most* $|B| \cdot 2^{|\llbracket \mathcal{S} \rrbracket|^2} + 1$.

Let us first use Proposition 10.2 to conclude the proof of Proposition 9.3. Set $\mathcal{S}_+ = \{\{\alpha(w)\} \mid w \in A^+\}$, $\mathcal{S}_\infty = \{\{\alpha(w)\} \mid w \in A^\infty\}$ and $\beta : (B^+, B^\infty) \to (\mathcal{S}_+, \mathcal{S}_\infty)$ defined by $\beta(w) = \{\alpha(w)\}$ for $w \in A^+ \cup A^\infty$ (note that $\beta$ is surjective). Recall that we already know from Proposition 4.7 that $\mathcal{I}[\alpha_+] = \mathrm{Sat}(\mathcal{S}_+)$. Therefore, by definition, $\mathrm{Sat}_\infty(\alpha) = \mathrm{Sat}_\infty(\mathrm{Sat}(\mathcal{S}_+))$. From Proposition 10.2, we obtain an FO-partition $\mathbf{K}$ of $A^\infty$ such that for all $K \in \mathbf{K}$,

(1) $\alpha(K) = \llbracket \beta(K) \rrbracket \in \mathrm{Sat}_\infty(\alpha)$.
(2) any $K \in \mathbf{K}$ can be defined by a first-order formula of rank at most $|A| \cdot 2^{|S_+|^2} + 1$.

It is now immediate from Item 1 and the fact that $\mathrm{Sat}_\infty(\alpha)$ is closed under downset that $\mathcal{I}[\alpha_\infty](\mathbf{K}) \subseteq \mathrm{Sat}_\infty(\alpha)$. We conclude that $\mathcal{I}[\alpha_\infty] \subseteq \mathcal{I}[\alpha_\infty](\mathbf{K}) \subseteq \mathrm{Sat}_\infty(\alpha)$ which terminates the proof of Proposition 9.3. Moreover, since we already know that $\mathrm{Sat}_\infty(\alpha) \subseteq \mathcal{I}[\alpha_\infty]$, we actually have $\mathcal{I}[\alpha_\infty] = \mathcal{I}[\alpha_\infty](\mathbf{K})$: $\mathbf{K}$ is optimal for $\alpha_\infty$. Therefore, we obtain the second item in Theorem 8.1 from Item (2) of Proposition 10.2.

**Corollary 10.3** (Second item in Theorem 8.1)**.** *Given two $\omega$-languages $L_0$ and $L_1$ that are recognized by $\alpha$, if they are FO-separable, then one can construct an actual separator with a formula of quantifier rank at most $|A|2^{|S_+|^2} + 1$.*

It remains to prove Proposition 10.2. We generalize the techniques we used to prove Proposition 6.1. Note that in several cases, the construction will require building an FO-partition of $B^+$ (or of a subset of $B^+$). In this cases, we will simply use Proposition 6.1. As for Proposition 6.1, we construct $\mathbf{K}$ by induction on the following two parameters listed by order of importance:

($a$) the index $|\llbracket \mathcal{S}_+ \rrbracket|$ of $\mathcal{S}_+$,
($b$) the size of $B$.

Observe that the case $|B| = 1$ is trivial in this setting: in that case $B^\infty$ is a singleton. We now assume that $|B| > 1$ and distinguish two subcases, depending on whether the restriction of $\beta$ to $B^+$ is *tame*. Recall that we say that $\beta$ is *tame* if for all $b \in B$, $\llbracket \mathcal{S}_+ \rrbracket = \beta(b) \cdot \llbracket \mathcal{S}_+ \rrbracket$ and $\llbracket \mathcal{S}_+ \rrbracket = \llbracket \mathcal{S}_+ \rrbracket \cdot \beta(b)$.

**Case 1: $\beta$ is tame.** As we have seen in the proof of Proposition 6.1, in that case we have $\llbracket \mathcal{S}_+ \rrbracket \in \mathrm{Sat}(\mathcal{S}_+)$. By surjectivity of $\beta$ it is immediate that $(\llbracket \mathcal{S}_+ \rrbracket)^\infty = \llbracket \mathcal{S}_\infty \rrbracket$. Therefore, by Item (1) in the definition of $\mathrm{Sat}_\infty$, $\llbracket \mathcal{S}_\infty \rrbracket \in \mathrm{Sat}_\infty(\mathrm{Sat}(\mathcal{S}_+))$. It is therefore sufficient to set $\mathbf{K} = \{B^\infty\}$ to satisfy Item (1) and Item (2) in the proposition.

**Case 2: $\beta$ is not tame.** By hypothesis on $\beta$, there exists $b \in B$ such that $\llbracket \mathcal{S}_+ \rrbracket \neq \beta(b) \cdot \llbracket \mathcal{S}_+ \rrbracket$ or $\llbracket \mathcal{S}_+ \rrbracket \neq \llbracket \mathcal{S}_+ \rrbracket \cdot \beta(b)$. By symmetry, we assume the former, *i.e.*, $\llbracket \mathcal{S}_+ \rrbracket \neq \beta(b) \cdot \llbracket \mathcal{S}_+ \rrbracket$. We set $b$ as this letter for the remainder of the proof.

Recall that we have to construct an FO-partition $\mathbf{K}$ of $B^\infty$ satisfying Items (1) and (2) in Proposition 10.2. Set $C = B \setminus \{b\}$ and observe that $B^\infty$ is the (disjoint) union of the following five sets:

$$B^\infty = b^\infty \cup B^* C b^\infty \cup C^\infty \cup B^* b C^\infty \cup C^*(b^+C^+)^\infty. \tag{10.1}$$

Therefore, it suffices to find FO-partitions satisfying Items (1) and (2) for all five sets to obtain the desired partition of $B^\infty$. These partitions are defined from FO-partitions of $C^+$ and $B^+$ (obtained from Proposition 6.1), of $b^\infty$ and $C^\infty$ (obtained by induction on $|B|$ in Proposition 10.2) and of $C^*(b^+C^+)^\infty$ (obtained by induction on the index of $\mathcal{S}_+$ in

Proposition 10.2). Since the construction is similar for all five sets, we only detail the case of $C^*(b^+C^+)^\infty$ (other cases are handled similarly). The construction is based on the following two lemmas.

**Lemma 10.4** (Partition of $C^+$)**.** *There exists a finite partition* $\mathbf{L}$ *of* $B^+$ *such that for any language* $L \in \mathbf{L}$*:*
(1) $[\![\beta(L)]\!] \in \mathrm{Sat}(\mathcal{S}_+)$.
(2) *there exists a first-order formula of rank at most* $|C| \cdot 2^{|[\![\mathcal{S}_+]\!]|^2}$ *that defines* $L$.

**Lemma 10.5** (Partition of $(b^+C^+)^\infty$)**.** *There exists a finite partition* $\mathbf{K'}$ *of* $(b^+C^+)^\infty$ *such that for any language* $K \in \mathbf{K'}$*:*
(1) $[\![\beta(K)]\!] \in \mathrm{Sat}_\infty(\mathrm{Sat}(\mathcal{S}_+))$.
(2) *there exists a first-order formula of rank at most* $|B| \cdot 2^{|[\![\mathcal{S}_+]\!]|^2}$ *that defines* $K$.

Lemma 10.4 is obtained by applying Proposition 6.1 to the restriction of $\beta$ to $C^+$. The proof of Lemma 10.5 is a straightforward generalization to $\omega$-words of the proof of Lemma 6.7 and is left to the reader (note that this is where our choice of $b$ and induction on the index of $\mathcal{S}_+$ are used).

Let us now explain how to construct the desired FO-partition of $C^*(b^+C^+)^\infty$. Consider the following partition $\mathbf{K''}$ of $C^*(b^+C^+)^\infty$,

$$\mathbf{K''} = \{K' \mid K' \in \mathbf{K'}\} \cup \{LK' \mid L \in \mathbf{L} \text{ and } K' \in \mathbf{K'}\}.$$

It is immediate from the fact that $\mathbf{L}$ and $\mathbf{K'}$ are partitions that $\mathbf{K''}$ is a partition of $C^*(b^+C^+)^\infty$. Moreover, it follows from Item (1) of Lemmas 10.4 and 10.5 and the second item in the definition of $\mathrm{Sat}_\infty$ that $\mathbf{K''}$ satisfies the first item in Proposition 10.2: for all $K'' \in \mathbf{K''}$, $[\![\beta(K'')]\!] \in \mathrm{Sat}_\infty(\mathrm{Sat}(\mathcal{S}_+))$. Finally, that the second item in Proposition 10.2 holds (*i.e.*, that any language in $\mathbf{K''}$ can be defined by a FO formula of rank at most $|B| \cdot 2^{|[\![\mathcal{S}_+]\!]|^2} + 1$) comes from the following fact (which generalizes Fact 6.8 to $\omega$-words).

**Fact 10.6.** Set $k \geqslant 0$. Let $L_1$ be a language and $L_2$ be an $\omega$-language, each defined by a first-order formula of rank at most $k$. Then $L_1L_2$ can be defined by a first-order formula of rank at most $k + 1$.

## 11. Conclusion

We have given simple and self-contained proofs that one can decide in Exptime whether two regular languages of finite or infinite words are separable by first-order logic. Further, we have obtained an upper bound on the quantifier rank of an expected separator. We have also described a procedure to compute, given as input a morphism $\alpha$ into a finite semigroup, a finite set of FO-formulas whose associated languages form a partition of $A^+$, and such that any two FO-separable languages recognized by $\alpha$ can be separated by a disjunction of some of these formulas. These formulas are computed inductively along the correctness proof of our algorithm.

There are some open questions left in this line of research. First, we do not know if the bounds are tight. We conjecture that the problem is Exptime-complete starting from semigroups. A related question is the complexity, starting from NFAs. Our results imply a 2-Exptime upper bound (for DFAs, checking first-order definability is Pspace-complete [6]). Moreover, we do not know whether the bounds on the quantifier depth and the size of the

expected separator are tight. Finally, it is likely that these techniques can be extended to other settings without much difficulty, as for finite or infinite Mazurkiewicz traces. A much more interesting and challenging problem is to look at separation for tree languages, where, for first-order logic, even getting a decidable characterization is open despite many recent attempts.

## References


[1] D. Albert, R. Baldinger, and J. Rhodes. Undecidability of the identity problem for finite semigroups. *The Journal of Symbolic Logic*, 57(1):179–192, 1992.

[2] J. Almeida. Some algorithmic problems for pseudovarieties. *Publ. Math. Debrecen*, 54:531–552, 1999. Proc. of Automata and Formal Languages, VIII.

[3] J. Almeida and M. Zeitoun. The pseudovariety J is hyperdecidable. *RAIRO Inform. Théor. Appl.*, 31(5):457–482, 1997.

[4] K. Auinger. On the decidability of membership in the global of a monoid pseudovariety. *IJAC*, 20(2):181–188, 2010.

[5] D. Beauquier and J. E. Pin. Languages and scanners. *Theoret. Comput. Sci.*, 84(1):3–21, 1991.

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

[7] J. C. Costa. Free profinite locally idempotent and locally commutative semigroups. *J. Pure Appl. Algebra*, 163(1):19–47, 2001.

[8] J. C. Costa and C. Nogueira. Complete reducibility of the pseudovariety LSl. *Internat. J. Algebra Comput.*, 19(02):247–282, 2009.

[9] W. Czerwiński, W. Martens, and T. Masopust. Efficient separability of regular languages by subsequences and suffixes. In *ICALP'13*, volume 7966 of *Lect. Notes Comp. Sci.*, pages 150–161. Springer, 2013.

[10] V. Diekert and P. Gastin. First-order definable languages. In *Logic and Automata: History and Perspectives*, volume 2, pages 261–306. Amsterdam Univ. Press, 2008.

[11] K. Henckell. Pointlike sets: the finest aperiodic cover of a finite semigroup. *J. Pure Appl. Algebra*, 55(1-2):85–126, 1988.

[12] K. Henckell, J. Rhodes, and B. Steinberg. Aperiodic pointlikes and beyond. *Internat. J. Algebra Comput.*, 20(2):287–305, 2010.

[13] H. W. Kamp. *Tense Logic and the Theory of Linear Order*. Phd thesis, CS Department, University of California at Los Angeles, USA, 1968.

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

[15] R. McNaughton and S. Papert. *Counter-Free Automata*. MIT Press, 1971.

[16] D. Perrin. Recent results on automata and infinite words. In *MFCS'84*, volume 176 of *Lect. Notes Comp. Sci.*, pages 134–148. Springer, 1984.

[17] D. Perrin and J. E. Pin. *Infinite Words*. Elsevier, 2004.

[18] J. E. Pin. Mathematical foundations of automata theory, 2016. `http://www.liafa.jussieu.fr/~jep/PDF/MPRI/MPRI.pdf`.

[19] T. Place. Separating regular languages with two quantifier alternations. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'15)*, pages 202–213. IEEE, 2015.

[20] T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by locally testable and locally threshold testable languages. In *FSTTCS'13*, volume 24 of *LIPIcs*, pages 363–375, 2013.

[21] T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *MFCS'13*, volume 8087 of *Lect. Notes Comp. Sci.*, pages 729–740. Springer, 2013.

[22] T. Place, L. van Rooijen, and M. Zeitoun. On separation by locally testable and locally threshold testable languages. *Logical Methods in Computer Science*, 10(3:24):1–28, 2014.

[23] T. Place and M. Zeitoun. Going higher in the first-order quantifier alternation hierarchy on words. In *ICALP'14*, 2014.

[24] T. Place and M. Zeitoun. Separating regular languages with first-order logic. In *CSL-LICS'14*, 2014.

[25] J. Rhodes. Undecidability, automata, and pseudovarities of finite semigroups. *IJAC*, 9(3-4):455–474, 1999.
[26] J. Rhodes and B. Steinberg. *The q-theory of Finite Semigroups*. Springer, 2008.
[27] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Inform. Control*, 8:190–194, 1965.
[28] B. Steinberg. On pointlike sets and joins of pseudovarieties. *Internat. J. Algebra Comput.*, 8(2):203–231, 1998.
[29] B. Steinberg. A delay theorem for pointlikes. *Semigroup Forum*, 63(3):281–304, 2001.
[30] B. Steinberg. A delay theorem for pointlikes. *Sem. Forum*, 63(3):281–304, 2001.
[31] H. Straubing. Finite semigroup varieties of the form $\mathbf{V} * \mathbf{D}$. *J. Pure Appl. Algebra*, 36(C):53–94, 1985.
[32] H. Straubing. *Finite Automata, Formal Logic and Circuit Complexity*. Birkhauser, 1994.
[33] W. Thomas. Star-free regular sets of omega-sequences. *Inform. and Control*, 42(2):148–156, 1979.
[34] W. Thomas. Languages, automata, and logic. In *Handbook of formal languages*. Springer, 1997.
[35] T. Wilke. An Eilenberg theorem for $\infty$-languages. In *ICALP'91*, volume 510 of *Lect. Notes Comp. Sci.*, pages 588–599. Springer, 1991.
[36] T. Wilke. Classifying discrete temporal properties. In *STACS'99*, volume 1563 of *Lect. Notes Comp. Sci.*, pages 32–46. Springer, 1999.