QUANTIFIER ALTERNATION FOR INFINITE WORDS

THÉO PIERRON, THOMAS PLACE AND MARC ZEITOUN

Abstract. We investigate the expressive power of quantifier alternation hierarchy of first-order logic over words. This hierarchy includes the classes Σ_i (sentences having at most i blocks of quantifiers starting with an $\exists$) and $\mathcal{B}\Sigma_i$ (Boolean combinations of Σ_i sentences). So far, this expressive power has been effectively characterized for the lower levels only. Recently, a breakthrough was made over finite words, and decidable characterizations were obtained for $\mathcal{B}\Sigma_2$ and Σ_3, by relying on a decision problem called separation, and solving it for Σ_2.

The contribution of this paper is a generalization of these results to the setting of infinite words: we solve separation for Σ_2 and Σ_3, and obtain decidable characterizations of $\mathcal{B}\Sigma_2$ and Σ_3 as consequences.

1. Introduction

Regular word languages form a robust class, as they can be defined either by operational, algebraic, or logical means: they are exactly those that can be defined equivalently by finite state machines (operational view), morphisms into finite algebras (algebraic view) and monadic second order (MSO) sentences [4, 25, 8, 5] (logical view). To understand in depth the structure of this class, it is natural to classify its languages according to their descriptive complexity. The problem is to determine how complicated has to be a sentence to describe a given input language. This is a decision problem parametrized by a fragment of MSO: given an input language, can it be expressed in the fragment? This problem is called *membership* (is the language a *member* of the class defined by the fragment?).

The seminal result in this field is the membership algorithm for first-order logic (FO) over finite words, which is arguably the most prominent fragment of MSO. This algorithm was obtained in two steps. McNaughton and Papert [10] observed that the languages definable in FO are exactly the *star-free languages*: those that may be expressed by a regular expression in which complement is allowed while the Kleene star is disallowed. Furthermore, an earlier result of Schützenberger [21] shows that star-free languages are exactly the ones whose syntactic monoid is aperiodic. The syntactic monoid is a finite algebra that can be computed from any input regular language, and aperiodicity can be formulated as an equation that has to be satisfied by all elements of this algebra. Therefore, Schützenberger's result makes it possible to decide whether a regular language is star-free (and therefore definable in FO by McNaughton-Papert's result).

Following this first result, the attention turned to a deeper question: given an FO-definable language, find the "simplest" FO-sentences that define it. The standard complexity measure for FO sentences is their quantifier alternation, which counts the number of switches between blocks of $\exists$ and $\forall$ quantifiers. This measure is justified not only because it is intuitively difficult to understand a sentence with many alternations, but also because the nonelementary complexity of standard problems for FO [23] (*e.g*, satisfiability) is tied to quantifier alternation. In summary, we classify FO definable languages by counting the number of quantifier alternations needed to define them and we want to be able to decide the level of a given language (which amounts to solving membership for each level).

This leads to define the following fragments of FO: an FO sentence is Σ_i if its prenex normal form has at most i blocks of $\exists$ or $\forall$ quantifiers and starts with a block of existential ones. Note that Σ_i is not closed under complement (the negation of a Σ_i sentence is called a Π_i sentence). A sentence is $\mathcal{B}\Sigma_i$ if it is a Boolean combination of Σ_i sentences. Clearly, we have $\Sigma_i \subseteq \mathcal{B}\Sigma_i \subseteq \Sigma_{i+1}$, and these inclusions are known to be strict [3, 24]: $\Sigma_i \subsetneq \mathcal{B}\Sigma_i \subsetneq \Sigma_{i+1}$. See the figure.

$$\begin{tikzcd}
\Pi_1 \arrow[dr, dash, "\subsetneq" description] & & \Pi_2 \arrow[dr, dash, "\subsetneq" description] & & \Pi_3 \arrow[dr, dash, "\subsetneq" description] & & \Pi_4 \cdots\cdots \\
& \mathcal{B}\Sigma_1 \arrow[ur, dash, "\subsetneq" description] \arrow[dr, dash, "\subsetneq" description] & & \mathcal{B}\Sigma_2 \arrow[ur, dash, "\subsetneq" description] \arrow[dr, dash, "\subsetneq" description] & & \mathcal{B}\Sigma_3 \arrow[ur, dash, "\subsetneq" description] \arrow[dr, dash, "\subsetneq" description] & \\
\Sigma_1 \arrow[ur, dash, "\subsetneq" description] & & \Sigma_2 \arrow[ur, dash, "\subsetneq" description] & & \Sigma_3 \arrow[ur, dash, "\subsetneq" description] & & \Sigma_4 \cdots\cdots
\end{tikzcd}$$

Solving membership for levels of this hierarchy is a longstanding open problem. Following Schützenberger's approach, it was first investigated for languages of finite words. However, the question also makes sense for more complex structures, in particular for the most natural extension, infinite words. Schützenberger's result was first generalized to infinite words by Perrin [11], and a suitable algebraic framework for languages of infinite words was set up by Wilke [26]. Since a regular language of infinite words is determined by regular languages of finite words, finding a membership algorithm for languages of infinite words does not usually require to start over. Instead these algorithms are obtained by building on top of the algorithms for finite words, adding new arguments, specific to infinite words.

Regarding the hierarchy, membership is easily seen to be decidable for Σ_1. For $\mathcal{B}\Sigma_1$, the classical result of Simon [22] was generalized from finite to infinite words by Perrin and Pin [12]. For finite words, membership to Σ_2 is known to be decidable [1, 13], a result lifted to infinite words in [7, 2]. Following these results, the understanding of the hierarchy remained stuck for years until the framework was extended to new and more general problems than membership.

Rather than asking whether a language is definable in a fragment $\mathcal{F}$, these problems ask what is the best $\mathcal{F}$-definable "approximation" of this language (with respect to specific criteria). The simplest example is *$\mathcal{F}$-separation*, which takes *two* regular languages as input and asks whether there exists a third language *definable in* $\mathcal{F}$ that contains the first language and is disjoint from the second. Separation is more general than membership: asking whether a regular language is definable in $\mathcal{F}$ is the same as asking whether it can be $\mathcal{F}$-separated from its (also regular) complement. A consequence is that deciding these more general problems is usually more challenging than deciding membership. However, their investigation in the setting of finite words has also been very rewarding. A good illustration is the transfer result of [16], which states that for all i, decidability of separation for Σ_i entails decidability of membership for Σ_{i+1}. Combined with an algorithm for Σ_2-separation [16], this proved that Σ_3 has decidable membership. This result was strengthened in [14], which shows that Σ_3-separation is decidable as well, thus obtaining decidability of membership for Σ_4. Finally, in [16], it was shown that $\mathcal{B}\Sigma_2$ has decidable membership by using a generalization of separation for Σ_2 and analyzing an algorithm solving this generalization.

It remained open to know whether it was possible to generalize with the same success this new approach to the setting of infinite words. This is the investigation that we carry out in the paper. More precisely, we rely on the crucial notion of Σ_i-chains, designed in [16] for presenting and proving membership and separation algorithms for finite words. We generalize this concept to infinite words and successfully use it to prove that the following problems are decidable: Σ_2-separation, Σ_3-separation, and $\mathcal{B}\Sigma_2$ membership. This demonstrates that Σ_i-chains remain a suitable framework for presenting arguments in the setting of infinite words. On the other hand, new issues specific to infinite words arise, for example, we were not able to generalize the transfer result from Σ_i-separation to Σ_{i+1}-membership (as a consequence, membership for Σ_4 remains open). Note that a different proof for deciding $\mathcal{B}\Sigma_2$-membership has been obtained independently in [9]. It gives a decidable characterization based on topology and algebra.

Note that, for each problem, we pre-compute some information by using the corresponding algorithm designed in [16, 14] for finite words. This means that the involved algorithms from [16, 14] are used as subroutines of our algorithms.

We now present the problems in depth in Section 2, and we solve them in the rest of the paper. A detailed outline is provided at the end of Section 2.

2. Presentation of the Problem

In this section, we first define the quantifier alternation hierarchy of first-order logic. Then, we present the membership problem and the separation problem.

2.1. The Quantifier Alternation Hierarchy of First-Order Logic. Words and ω-Words.

For the whole paper, we assume that a finite alphabet A is fixed. We denote by A^+ the set of all finite nonempty words, and by A^∞ the set of all infinite words over A. In the paper, we use the following terminology: the term "word" means a finite word, and the term "ω-word" means an infinite word.

If u is a word and v is a word (resp. an ω-word), we denote by uv the word (resp. ω-word) obtained by concatenating u to the left of v. If $u \in A^+$ is a word, we denote by u^∞ the ω-word $uuuu\cdots$ obtained as the infinite concatenation of u with itself. If $u \in A^+ \cup A^\infty$ is a word or an ω-word, we denote by $\mathsf{alph}(u)$ the *alphabet* of u, *i.e.*, the set of letters of u. We call *language* (resp. *ω-language*) a subset of A^+ (resp. of A^∞), *i.e.*, a language of finite words (resp. of ω-words).

In the paper we are interested in *regular* languages and ω-languages. Regular ω-languages are those that can be equivalently defined by monadic second-order logic, finite Büchi automata or finite ω-semigroups. We work with the definition of regular ω-languages in terms of ω-semigroups, recalled in Section 3.

First-Order Logic. Any word or ω-word can be viewed as a logical structure made of a linearly ordered sequence of positions labeled over the alphabet A (finite for words and infinite for ω-words). In first-order logic (FO), one can quantify over these positions and use the following predicates.

– for each $a \in A$, a unary predicate P_a selecting all positions labeled with an a.

– a binary predicate '$<$' interpreted as the (strict) linear order over the positions.

Since any FO sentence may be interpreted both on words and ω-words, each sentence φ defines two objects: a language $L_+ = \{w \in A^+ \mid w \models \varphi\}$ and an ω-language $L_\infty = \{w \in A^\infty \mid w \models \varphi\}$. For example, the sentence $\exists x \exists y\ (x < y \wedge P_a(y))$ defines the language $A^+a \cup A^+aA^+$ and the ω-language A^+aA^∞.

Thus, we may associate two classes of objects with FO: a class of languages (we speak of FO over words) and a class of ω-languages (we speak of FO over ω-words).

Quantifier Alternation. It is usual to classify FO sentences by counting the quantifier alternations inside their prenex normal form. Set $i \in \mathbb{N}$, a sentence is said to be Σ_i (resp. Π_i) if its prenex normal form has either

– *exactly* $i-1$ quantifier alternations (*i.e.*, exactly i blocks of quantifiers) starting with an $\exists$ (resp. $\forall$), or

– *strictly less* than $i-1$ quantifier alternations (*i.e.*, strictly less than i blocks).

For example, the sentence $\exists x_1 \forall x_2 \forall x_3 \exists x_4\ \varphi(x_1, x_2, x_3, x_4)$, with φ quantifier-free, is Σ_3. Note that in general, the negation of a Σ_i sentence is not a Σ_i sentence (it is called a Π_i sentence). Hence, it is also usual to define $\mathcal{B}\Sigma_i$ sentences as those that are Boolean combinations of Σ_i and Π_i sentences.

As for full first-order logic, each level Σ_i, Π_i or $\mathcal{B}\Sigma_i$ defines two classes of objects: a class of languages and a class of ω-languages. Therefore, we obtain two hierarchies: a hierarchy of classes of languages and a hierarchy of classes of ω-languages. Both hierarchies are strict (refer to the figure in the introduction).

2.2. **Decision Problems.** Our objective is to investigate the quantifier alternation hierarchy of first-order logic over ω-words. We rely on two decision problems in order to carry out this investigation: the membership problem and the separation problem. Both problems are parametrized by a level in the hierarchy and come in two versions: a 'language' one and an 'ω-language' one. Given a level $\mathcal{F}$ in the hierarchy, the *membership problem* for $\mathcal{F}$ is as follows:

Language Membership Problem	
IN	A regular language L
OUT	Is L $\mathcal{F}$-definable ?

ω-Language Membership Problem	
IN	A regular ω-language L
OUT	Is L $\mathcal{F}$-definable ?

The separation problem is more general. Given three languages or three ω-languages K, L_1, L_2, we say that K *separates* L_1 from L_2 if $L_1 \subseteq K$ and $L_2 \cap K = \emptyset$. For $\mathcal{F}$ a level in the hierarchy, we say that L_1 is *$\mathcal{F}$-separable* from L_2 if there exists a language or ω-language that is definable in $\mathcal{F}$ and separates L_1 from L_2. Note that when $\mathcal{F}$ is not closed under complement (*e.g.*, when $\mathcal{F} = \Sigma_i$ or $\mathcal{F} = \Pi_i$), the definition is not symmetrical: L_1 may be $\mathcal{F}$-separable from L_2 while L_2 is not $\mathcal{F}$-separable from L_1. The separation problem for $\mathcal{F}$ is as follows:

Language Separation Problem	
IN	Two regular languages L_1, L_2
OUT	Is L_1 $\mathcal{F}$-separable from L_2 ?

ω-Language Separation Problem	
IN	Two regular ω-languages L_1, L_2
OUT	Is L_1 $\mathcal{F}$-separable from L_2 ?

An important remark is that membership reduces to separation: a regular language or ω-language is definable in $\mathcal{F}$ iff it is $\mathcal{F}$-separable from its (also regular) complement. This makes separation a more general problem than membership.

Both problems have been extensively studied in the literature. Indeed, it has been observed that obtaining an algorithm for the membership or separation problem associated to a particular level $\mathcal{F}$ usually yields a deep insight on $\mathcal{F}$. This is well illustrated by the most famous result of this kind, Schützenberger's Theorem [21, 10], which yields a (language) membership algorithm for FO. The result was later generalized to ω-languages by Perrin [11]. These results and the techniques used to obtain them provide not only a way to decide whether a regular (ω-)language is FO-definable, but also a generic method for constructing an FO sentence when the (ω-)language is definable. Since these first results, many efforts have been devoted for obtaining membership and separation algorithms for each level in the hierarchy. An overview of the results is presented in the following table (omitted levels are open in all cases).

Membership Problem

	Language	ω-Language
FO	Solved [21, 10]	Solved [11]
Σ_1	Solved (Folklore)	Solved (Folklore)
$\mathcal{B}\Sigma_1$	Solved [22]	Solved [12]
Σ_2	Solved [1, 13]	Solved [7]
$\mathcal{B}\Sigma_2$	Solved [16]	**Open**
Σ_3	Solved [16]	**Open**
Σ_4	Solved [14]	**Open**

Separation Problem

	Language	ω-Language
FO	Solved [17]	Solved [17]
Σ_1	Solved (Folklore)	Solved (Folklore)
$\mathcal{B}\Sigma_1$	Solved [6, 15]	Solved [18]
Σ_2	Solved [16]	**Open**
$\mathcal{B}\Sigma_2$	**Open**	**Open**
Σ_3	Solved [14]	**Open**
Σ_4	**Open**	**Open**

Our main objective is to bridge the gap between what is known for languages and what is known for ω-languages. More precisely, we want to extend the recent results of [16] and [14] to the setting of ω-words, *i.e.*, to obtain membership algorithms for $\mathcal{B}\Sigma_2$, Σ_3 and Σ_4 as well as separation algorithms for Σ_2 and Σ_3. We were able to obtain these algorithms for Σ_2, Σ_3 and $\mathcal{B}\Sigma_2$ as we state in the following theorem (we leave the case of Σ_4-membership for ω-languages open, we will come back to this point in the conclusion).

Theorem 2.1. *The following properties hold:*

a) the ω-language separation problem is decidable for Σ_2.
b) the ω-language membership problem is decidable for $\mathcal{B}\Sigma_2$.
c) the ω-language separation problem is decidable for Σ_3.

Our proof of Theorem 2.1 consists in three algorithms, one for each item in the theorem. An important remark is that each of these three algorithms depends upon an algorithm of [16] or [14] solving the corresponding problem for languages:

– We present all algorithms in a specific framework which is adapted from the one used in [16]. In particular, we reuse the key notion of "Σ_i-chain" (generalized to ω-words in a straightforward way).
– We actually reuse the language algorithms of [16] and [14] as subprocedures in our algorithms for ω-languages.

The remainder of the paper is devoted to proving Theorem 2.1. In Section 3, we recall classical notions required for our definitions and proofs: the ω-semigroup definition of regular ω-languages and logical preorders. In Section 4, we present the general framework used in the paper. In particular, we introduce a notion that will be at the core of all our algorithms: "Σ_i-chains" (which are adapted and reused from [16]). We then devote a section to each algorithm: Section 5 to Σ_2-separation, Section 6 to $\mathcal{B}\Sigma_2$-membership and Section 7 to Σ_3-separation.

3. Preliminaries

In this section, we recall some classical notions that we will need. First, we present the definition of regular ω-languages in terms of ω-semigroups. Then, we define the logical preorders that one may associate to each level Σ_i in the hierarchy.

3.1. Semigroups and ω-Semigroups.

We briefly recall the definition of regular languages and ω-languages in terms of semigroups and ω-semigroups. For details, see [12].

Semigroups. A semigroup is a set S equipped with an associative operation $s \cdot t$ (often written st). In particular, A^+ equipped with concatenation is a semigroup. Given a *finite* semigroup S, it is easy to see that there is an integer $\omega(S)$ (denoted by ω when S is understood) such that for all s of S, s^ω is idempotent: $s^\omega = s^\omega s^\omega$.

Given a language L and a morphism $\alpha : A^+ \to S$, we say that L is *recognized* by α if and only if there exists $F \subseteq S$ such that $L = \alpha^{-1}(F)$. It is well-known that a language is regular if and only if it may be recognized by a *finite* semigroup.

ω-Semigroups. An *ω-semigroup* is a pair (S_+, S_∞), where S_+ is a semigroup and S_∞ is a set. Moreover, (S_+, S_∞) is equipped with two additional products: a *mixed product* $S_+ \times S_\infty \to S_\infty$ mapping $s, t \in S_+, S_\infty$ to an element st of S_∞, and an *infinite product* $(S_+)^\infty \to S_\infty$ mapping an infinite sequence $s_1, s_2, \cdots \in (S_+)^\infty$ to an element $s_1 s_2 \cdots$ of S_∞. We require these products to satisfy all possible forms of associativity. For $s \in S_+$, we let s^∞ be the infinite product $sss\cdots \in S_\infty$. Note that (A^+, A^∞) is an ω-semigroup. See [12] for further details.

We say that (S_+, S_∞) is *finite* if both S_+ and S_∞ are. Note that even if an ω-semigroup is finite, it is not clear how to represent the infinite product, since the set of infinite sequences of S_+ is uncountable. However, it has been shown by Wilke [26] that the infinite product is fully determined by the mapping $s \mapsto s^\infty$. This makes it possible to finitely represent any finite ω-semigroup.

Morphisms of ω-semigroups are defined in the natural way. In particular, observe that any ω-semigroup morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ defines two maps: a semigroup morphism $\alpha_+ : A^+ \to S_+$ and a map $\alpha_\infty : A^\infty \to S_\infty$ (when there is no ambiguity, we shall write $\alpha(w)$ to mean $\alpha_+(w)$ if $w \in A^+$ or $\alpha_\infty(w)$ if $w \in A^\infty$). Therefore, a morphism recognizes both languages (the languages $\alpha_+^{-1}(F_+)$ for $F_+ \subseteq S_+$) and ω-languages (the ω-languages $\alpha_\infty^{-1}(F_\infty)$ for $F_\infty \subseteq S_\infty$). An ω-language is regular iff it may be recognized by a morphism into a *finite* ω-semigroup.

Syntactic Morphisms. It is known that given any regular language (resp. ω-language) L there exists a canonical morphism $\alpha_L : A^+ \to S$ (resp. $\alpha_L : (A^+, A^\infty) \to (S_+, S_\infty)$) recognizing L. This object is called the syntactic morphism of L. We refer the reader to [12] for the detailed definition of this object. In the paper we only use two properties of the syntactic morphism. The first is simply that given any regular ω-language L, one may compute its syntactic morphism from any representation of L. We state the second one below.

Fact 3.1. *Let $i \geqslant 1$ and let L be a regular ω-language. Then L is definable in $\mathcal{B}\Sigma_i$ iff so are all languages and ω-languages recognized by its syntactic morphism.*

The proof of Fact 3.1 may be found in [12] (in fact, this holds for any class of ω-languages which is a "variety" of ω-languages, not just for $\mathcal{B}\Sigma_i$). In view of this, the syntactic morphism is central for membership questions: deciding if a language is definable in $\mathcal{B}\Sigma_i$ amounts to deciding a property of its syntactic morphism. This is the approach used in our $\mathcal{B}\Sigma_2$-membership algorithm (see Section 6).

Morphisms and Separation. When working on separation, we are given two input languages or ω-languages. It is convenient to consider a single recognizing object for both inputs rather than two separate objects. This is not restrictive: given two languages (resp. two ω-languages) and two associated recognizing morphisms, one can define and compute a single morphism that recognizes them both. For example, if $L_0 \subseteq A^\infty$ is recognized by $\alpha_0 : (A^+, A^\infty) \to (S_+, S_\infty)$ and $L_1 \subseteq A^\infty$ by $\alpha_1 : (A^+, A^\infty) \to (T_+, T_\infty)$, then L_0 and L_1 are both recognized by $\alpha : (A^+, A^\infty) \to (S_+ \times T_+, S_\infty \times T_\infty)$ with $\alpha(w) = (\alpha_0(w), \alpha_1(w))$.

Alphabet Compatible Morphisms. It will be convenient to work with morphisms that satisfy an additional property. A morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ is said to be *alphabet compatible* if for all $u, v \in A^+ \cup A^\infty$, $\alpha(u) = \alpha(v)$ implies $\mathsf{alph}(u) = \mathsf{alph}(v)$. Note that when α is alphabet compatible, for all $s \in S_+ \cup S_\infty$, $\mathsf{alph}(s)$ is well defined as the unique $B \subseteq A$ such that for all $u \in \alpha^{-1}(s)$, we have $\mathsf{alph}(u) = B$ (if s has no preimage then we simply set $\mathsf{alph}(s) = \emptyset$).

To any morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$, we associate a morphism β, called the *alphabet completion* of α. The morphism β recognizes all ω-languages recognized by α and is alphabet compatible. If α is already alphabet compatible, then $\beta = \alpha$. Otherwise, observe that 2^A is a semigroup with union as the multiplication and $(2^A, 2^A)$ is therefore an ω-semigroup. Hence, we can define β as the morphism: $\beta : (A^+, A^\infty) \to (S_+ \times 2^A, S_\infty \times 2^A)$ with $\beta(w) = (\alpha(w), \mathsf{alph}(w))$.

3.2. **Logical Preorders.** To each level Σ_i in the hierarchy, one may associate preorders on the sets of words and ω-words. The definition is based on the notion of quantifier rank. The *quantifier rank* of a first-order formula is the length of the longest sequence of nested quantifiers inside the formula. For example, the following formula,

$$\exists x \ P_b(x) \wedge \neg(\exists y \ (y < x \wedge P_c(y)) \wedge (\forall y \exists z \ x < y < z \wedge P_b(y)))$$

has quantifier rank 3. It is well-known (and easy to show) that for a fixed k, there is a finite number of non-equivalent first-order formulas of rank less than k.

We may now define the preorders. Note that while we define two preorders for each level Σ_i (one on A^+ and one on A^∞), we actually use the same notation for both. Set $i \geqslant 1$ as a level in the hierarchy and $k \geqslant 1$ as a quantifier rank. Given two words $w, w' \in A^+$ (resp two ω-words $w, w' \in A^\infty$), we write $w \lesssim_i^k w'$ if and only if *any* Σ_i formula of rank at most k that is satisfied by

w is satisfied by w' as well. One may verify that $\lesssim_i^k$ is preorder. Moreover, it is immediate from the definition that the preorders get refined when k increases: $w \lesssim_i^{k+1} w' \Rightarrow w \lesssim_i^k w'$.

Denote by $\cong_i^k$ the equivalence generated by $\lesssim_i^k$: $w \cong_i^k w'$ when $w \lesssim_i^k w'$ and $w' \lesssim_i^k w$. That is, $w \cong_i^k w'$ if and only if w, w' satisfy the same Σ_i sentences (or equivalently the same $\mathcal{B}\Sigma_i$ sentences, which are just Boolean combinations of Σ_i sentences). The following fact may be verified from the definition.

Fact 3.2. *Let $k, i \geqslant 1$ and let u, v be two words or two ω-words, then*

$$(1)\ u \lesssim_i^{k+1} v \Rightarrow u \lesssim_i^k v, \qquad (2)\ u \cong_i^{k+1} v \Rightarrow u \cong_i^k v \qquad (3)\ u \lesssim_{i+1}^k v \Rightarrow u \cong_i^k v.$$

We finish the section with a few properties about the preorders $\lesssim_i^k$. The proofs are easy and omitted (they are obtained with standard Ehrenfeucht-Fraïssé arguments). We start with decomposition and composition lemmas.

Lemma 3.3 (Decomposition Lemma)**.** *Let $i, k \geqslant 1$ and let u, v be two words or two ω-words such that $u \lesssim_i^k v$. Then for any decomposition $u = u_1u_2$ of u, there exist v_1, v_2 such that $v = v_1v_2$, $u_1 \lesssim_i^{k-1} v_1$ and $u_2 \lesssim_i^{k-1} v_2$.*

Lemma 3.4 (Composition Lemma)**.** *Let $i, k \geqslant 1$, let u_1, v_1 be two words such that $u_1 \lesssim_i^k v_1$, and u_2, v_2 be either two words or two ω-words such that $u_2 \lesssim_i^k v_2$. Then $u_1u_2 \lesssim_i^k v_1v_2$ and $u_1^\infty \lesssim_i^k v_1^\infty$.*

The last composition that we state is specific to ω-words.

Lemma 3.5. *Let $i, k \geqslant 1$, $u \in A^+$ be a word and $v \in A^\infty$ be an ω-word such that $v \lesssim_i^k u^\infty$. Then for any $\ell \geqslant 2^k$, we have $u^\infty \lesssim_{i+1}^k u^\ell v$.*

In particular we will use the special case of Lemma 3.5 in which $i = 1$. In this case, one can verify that given $u \in A^+$ and $v \in A^\infty$, when $\mathsf{alph}(u) = \mathsf{alph}(v)$, we have $v \lesssim_1^k u^\infty$ for any $k \geqslant 1$. Hence we have the following corollary of Lemma 3.5.

Corollary 3.6. *Let $k \geqslant 1$, $u \in A^+$ be a word and let $v \in A^\infty$ be an ω-word such that $\mathsf{alph}(u) = \mathsf{alph}(v)$. Then for any $\ell \geqslant 2^k$, we have $u^\infty \lesssim_2^k u^\ell v$.*

4. Σ_i-Chains for ω-Languages

As explained, all algorithms for ω-languages of this paper are strongly related to the algorithms for languages of [16] and [14]. In particular, we adapt and reuse the key notion of "Σ_i-chain" which was introduced in [16]. The section is devoted to the presentation of this notion. First, we define Σ_i-chains. We then detail the link between Σ_i-chains and our decision problems, first for Σ_i, then for $\mathcal{B}\Sigma_i$.

4.1. Σ_i-Chains.

Σ_i-Chains were initially introduced in [16] as a tool designed to investigate the (language) separation problem for the logics Σ_i and $\mathcal{B}\Sigma_i$. A "set of Σ_i-chains" can be associated to any morphism $\alpha : A^+ \to S$ into a finite semigroup S. Intuitively, this set captures information about what Σ_i and $\mathcal{B}\Sigma_i$ can express about the languages recognized by α (including which ones are separable with Σ_i and $\mathcal{B}\Sigma_i$). The definition is based on the following lemma.

Lemma 4.1. *Let $i, k \geqslant 1$ and let L_1, L_2 be two languages or two ω-languages. Then L_1 is* **not** *Σ_i-separable (resp.* **not** *$\mathcal{B}\Sigma_i$-separable) from L_2 iff for all $k \geqslant 1$, there exist $w_1 \in L_1$ and $w_2 \in L_2$ such that $w_1 \lesssim_i^k w_2$ (resp. such that $w_1 \cong_i^k w_2$).*

Proof. We prove the first item, the second one is obtained similarly. Assume first that L_1 is **not** Σ_i-separable from L_2. Set $k \geqslant 1$ and consider the language or ω-language $K = \{w \mid \exists v \in L_1 \text{ s.t. } v \lesssim_i^k w\}$. By definition, $L_1 \subseteq K$ and K may be defined by a Σ_i sentence of rank k (this is because there are finitely many nonequivalent sentences of rank k). Therefore, there exists $w_2 \in L_2 \cap K$ (otherwise

K would separate L_1 from L_2, which is impossible by hypothesis). By definition of K, we get some $w_1 \in L_1$ such that $w_1 \lesssim_i^k w_2$ which terminates the proof of this direction.

For the other direction, assume that for all $k \geqslant 1$, there exist $w_1 \in L_1$ and $w_2 \in L_2$ such that $w_1 \lesssim_i^k w_2$ and let $K \supseteq L_1$ be a language or ω-language that is defined by a Σ_i sentence φ. We prove that $K \cap L_2 \neq \emptyset$, *i.e.*, that K cannot be a separator. Let k be the rank of φ, we obtain $w_1 \in L_1$ and $w_2 \in L_2$ such that $w_1 \lesssim_i^k w_2$ from our hypothesis. Since $L_1 \subseteq K$, we have $w_1 \in K$, and by definition of K, $w_1 \models \varphi$. Since φ is of rank k, by definition of $\lesssim_i^k$ we must have $w_2 \models \varphi$, *i.e.*, $w_2 \in L_2 \cap K$ which terminates the proof. $\square$

Lemma 4.1 states simple criteria equivalent to Σ_i- and $\mathcal{B}\Sigma_i$-separability. However, both criteria involve a quantification over all natural numbers. Therefore, it is not immediate that they can be decided. Indeed, since both A^+ and A^∞ are infinite sets, $\lesssim_i^k$ and $\cong_i^k$ are endlessly refined as k gets larger.

Σ_i-Chains are designed to deal with this issue. The separation problem takes two *regular* languages or ω-languages as input. Therefore, we have a single morphism that recognizes them both. For example, in the ω-language case, we have $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ (with (S_+, S_∞) a finite ω-semigroup) that recognizes both inputs. Intuitively, S_+ and S_∞ are finite abstractions of A^+ and A^∞. Therefore, we may abstract the preorders $\lesssim_i^k$ on these two finite sets: this is what Σ_i-chains are. For example, we say that $(s,t) \in (S_\infty)^2$ is a Σ_i-chain (of length 2) for α iff for all k, there exist $u, v \in A^\infty$ such that $\alpha(u) = s$, $\alpha(v) = t$ and $u \lesssim_i^k v$. For ω-languages recognized by α, it is then easy to adapt the two criteria of Lemma 4.1 to work directly with the Σ_i-chains associated to α. In other words, we reduce separation to the (still difficult) problem of computing the set of Σ_i-chains associated to a given input morphism.

Chains. Let us now define chains. Given a finite set S, a *chain over* S is simply a finite word over S (*i.e.*, an element of S^+). We shall only consider chains over S_+ and over S_∞, where S_+ and S_∞ are the two components of some ω-semigroup (S_+, S_∞). A remark about notation is in order: a word is usually denoted as the concatenation of its letters. However, since S_+ is a semigroup, this would be ambiguous: when $st \in (S_+)^+$, st could either mean a word with 2 letters s and t, or the product of s and t in S_+. To avoid confusion, we will write $(s_1, \dots, s_n)$ a chain of length n. We denote chains by $\overline{s}, \overline{t}, \dots$ and sets of chains by $\mathcal{S}, \mathcal{T}, \dots$.

If (S_+, S_∞) is an ω-semigroup, then for all $n \in \mathbb{N}$, $(S_+)^n$ is a semigroup when equipped with the componentwise multiplication $(s_1, \dots, s_n)(t_1, \dots, t_n) = (s_1t_1, \dots, s_nt_n)$. Moreover, the pair $((S_+)^n, (S_\infty)^n)$ is an ω-semigroup.

Σ_i-Chains. Fix $i \geqslant 1$ and $x \in \{+, \infty\}$. We associate a set of Σ_i-chains to any map $\beta : A^x \to S$ where S is a finite set. The set $\mathcal{C}_i[\beta] \subseteq S^+$ of Σ_i-chains for β is defined as follows. Let $\overline{s} = (s_1, \dots, s_n) \in S^+$ be a chain. We have $\overline{s} \in \mathcal{C}_i[\beta]$ if and only if for all $k \in \mathbb{N}$, there exist $w_1, \dots, w_n \in A^x$ such that:

$$w_1 \lesssim_i^k w_2 \lesssim_i^k \cdots \lesssim_i^k w_n \text{ and for all } j,\ \beta(w_j) = s_j.$$

Moreover, we denote by $\mathcal{C}_{i,n}[\beta]$ the restriction of this set to chains of length n only (*i.e.*, $\mathcal{C}_{i,n}[\beta] = \mathcal{C}_i[\beta] \cap S^n$).

Σ_i-Chains for an ω-Semigroup Morphism. It follows from the definition of Σ_i-chains that one may associate a set $\mathcal{C}_i[\alpha]$ to any semigroup morphism $\alpha : A^+ \to S$. This set is exactly the set of Σ_i-chains associated to α as defined in [16].

Moreover, given a morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ into a finite ω-semigroup (S_+, S_∞), one may associate two sets of Σ_i-chains to α: one to the morphism $\alpha_+ : A^+ \to S_+$ ($\mathcal{C}_i[\alpha_+] \subseteq (S_+)^+$) and one to the map $\alpha_\infty : A^\infty \to S_\infty$ ($\mathcal{C}_i[\alpha_\infty] \subseteq (S_\infty)^+$). We may now link Σ_i-chains to the separation problem.

4.2. **Σ_i-Chains and Separation for Σ_i.** We now connect Σ_i-chains to the separation problem. We begin with the simplest connection, which is between Σ_i-chains of length 2 and separation for Σ_i.

Theorem 4.2. *Let $i \geqslant 1$, $x \in \{+, \infty\}$ and $\beta : A^x \to S$ a map into a finite set S. Given $F_1, F_2 \subseteq S$, $L_1 = \beta^{-1}(F_1)$ and $L_2 = \beta^{-1}(F_2)$, the following are equivalent*
(1) L_1 is **not** *Σ_i-separable from L_2.*
(2) there exist $s_1 \in F_1$ and $s_2 \in F_2$ such that $(s_1, s_2) \in \mathcal{C}_{i,2}[\beta]$.

Theorem 4.2 is a straightforward consequence Lemma 4.1 (statement for Σ_i). In view of the theorem, our approach for the Σ_i-separation problem is as follows:
– for languages, we look for an algorithm computing $\mathcal{C}_{i,2}[\alpha]$ from an input morphism $\alpha : A^+ \to S$ into a finite semigroup S.
– for ω-languages, we look for an algorithm computing $\mathcal{C}_{i,2}[\alpha_\infty]$ from an input morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ into a finite ω-semigroup (S_+, S_∞). In particular, this algorithm typically involves computing $\mathcal{C}_{i,2}[\alpha_+]$ first, which can be achieved by reusing first item, *i.e.*, the algorithm for word languages.

This approach is exactly the one from [16, 14] to solve separation for Σ_2 and Σ_3 over finite words: the following theorems are proven in these papers.

Theorem 4.3 ([16])**.** *Given as input a morphism $\alpha : A^+ \to S$ into a finite semigroup S, one can compute the set $\mathcal{C}_{2,2}[\alpha]$ of Σ_2-chains of length 2 for α.*

Theorem 4.4 ([14])**.** *Given as input a morphism $\alpha : A^+ \to S$ into a finite semigroup S, one can compute the set $\mathcal{C}_{3,2}[\alpha]$ of Σ_3-chains of length 2 for α.*

We generalize these two theorems in Section 5 (for Σ_2) and Section 7 (for Σ_3) for ω-words by presenting two new algorithms. These algorithms both take a morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ as input and compute the sets $\mathcal{C}_{2,2}[\alpha_\infty]$ and $\mathcal{C}_{3,2}[\alpha_\infty]$ respectively. Note that the algorithms of Theorem 4.3 and Theorem 4.4 are reused as sub-procedures in these new algorithms for ω-languages: computing $\mathcal{C}_{2,2}[\alpha_\infty]$ and $\mathcal{C}_{3,2}[\alpha_\infty]$ requires to first compute $\mathcal{C}_{2,2}[\alpha_+]$ and $\mathcal{C}_{3,2}[\alpha_+]$ respectively.

Remark 4.5. *The algorithms of Theorems 4.3 and 4.4 both work with objects that are actually more general than Σ_i-chains: the Σ_2 algorithm works with "Σ_2-junctures" and the Σ_3 algorithm with an even more general notion: "$\Sigma_{2,3}$-trees". We do not present these more general notions because we do not need them outside of the algorithms of Theorems 4.3 and 4.4, which we use as black boxes.*

4.3. **Σ_i-Chains and Separation for $\mathcal{B}\Sigma_i$.** We finish by presenting the connection between the separation problem for $\mathcal{B}\Sigma_i$ and Σ_i-chains. This time, the connection depends on the whole set of Σ_i-chains. More precisely, it depends on yet another notion called *alternation*.

Set $x \in \{+, \infty\}$ and $\beta : A^x \to S$ as a map into a finite set S. We say that a pair $(s, t) \in S^2$ is Σ_i-alternating for β iff for all $n \geqslant 1$, $(s, t)^n \in \mathcal{C}_i[\beta]$ (where by $(s, t)^n$, we mean the chain $(s, t, s, t, \dots, s, t)$ of length $2n$).

Theorem 4.6. *Let $i \geqslant 1$, $x \in \{+, \infty\}$ and $\beta : A^x \to S$ a map into a finite set S. Given $F_1, F_2 \subseteq S$, $L_1 = \beta^{-1}(F_1)$ and $L_2 = \beta^{-1}(F_2)$, the following are equivalent,*
(1) L_1 is **not** *$\mathcal{B}\Sigma_i$-separable from L_2.*
(2) there exist $s_1 \in F_1$ and $s_2 \in F_2$ such that (s_1, s_2) is Σ_i-alternating.

Proof. There are two directions to prove. Assume first that L_1 is not $\mathcal{B}\Sigma_i$-separable from L_2. By Lemma 4.1, we know that for all $k \geqslant 1$ we have $w_1 \in L_1$ and $w_2 \in L_2$ such that $w_1 \cong_i^k w_2$. Hence, it follows that for all $k \geqslant 1$, we have $w_1 \in L_1$ and $w_2 \in L_2$ such that,

$$w_1 \lesssim_i^k w_2 \lesssim_i^k w_1 \lesssim_i^k w_2 \lesssim_i^k \cdots$$

Note that w_1, w_2 depend on k, but since S is finite, one can assume their images to be constant for infinitely many values of k. Since $w_1 \lesssim_i^{k+1} w_2 \Rightarrow w_1 \lesssim_i^k w_2$ (see Fact 3.2), we may assume that they are constant for all k, *i.e.*, that there exist $s_1 \in \beta(L_1)$ and $s_2 \in \beta(L_2)$ such that for all $k \geqslant 1$ the corresponding w_1 and w_2 are mapped to s_1 and s_2 respectively. This exactly means thats for all $n \geqslant 1$, the chain $(s_1, s_2)^n$ is a Σ_i-chain for β. Therefore, (s_1, s_2) is Σ_i-alternating, which terminates the proof of this direction.

It remains to prove the other direction. Assume that there exist $s_1 \in \beta(L_1)$ and $s_2 \in \beta(L_2)$ such that (s_1, s_2) is Σ_i-alternating. We have to prove that L_1 is **not** $\mathcal{B}\Sigma_i$-separable from L_2. We know from Lemma 4.1 that it suffices to prove that for all $k \geqslant 1$, there exist $w_1 \in L_1$ and $w_2 \in L_2$ such that $w_1 \cong_i^k w_2$.

Set $k \geqslant 1$. Since there are only finitely many nonequivalent $\mathcal{B}\Sigma_i$ formulas of rank k, the relation $\cong_i^k$ has finite index. Let ℓ be the number of equivalence classes of $\cong_i^k$. Since (s_1, s_2) is Σ_i-alternating, we know that the chain $(s_1, s_2)^\ell$ of length 2ℓ is a Σ_i-chain for β. Hence, we have 2ℓ words $u_1, \dots, u_{2\ell}$ such that for all $j \geqslant 1$, $u_{2j-1} \in L_1$ and $u_{2j} \in L_2$, and $u_1 \lesssim_i^k u_2 \lesssim_i^k \cdots \lesssim_i^k u_{2\ell}$. By choice of ℓ, the pigeonhole principle gives $j < h$ such that $u_j \cong_i^k u_h$. Since $u_j \lesssim_i^k u_{j+1} \lesssim_i^k u_h \lesssim_i^k u_j$, it follows that $u_j \cong_i^k u_{j+1}$ which terminates the proof since either $u_j \in L_1$ and $u_{j+1} \in L_2$, or $u_j \in L_2$ and $u_{j+1} \in L_1$. ☐

In view of Theorem 4.6, the separation problem for $\mathcal{B}\Sigma_i$ reduces to the computation of the Σ_i-alternating pairs. Unfortunately, whether there exists such an algorithm is open for $i \geqslant 2$, even in the case of finite words.

However, Theorem 4.6 yields an immediate corollary that applies to membership only. Given $x \in \{+, \infty\}$ and $\beta : A^x \to S$ a map into a finite set S, we say that β has *bounded Σ_i-alternation* iff there exists no Σ_i-alternating pair $(s, t) \in S^2$ for β such that $s \neq t$.

Corollary 4.7. *Let $i \geqslant 1$, $x \in \{+, \infty\}$ and $\beta : A^x \to S$ be a map into a finite set S. Then for any $F \subseteq S$, $\beta^{-1}(F)$ is $\mathcal{B}\Sigma_i$-definable if and only if β has bounded Σ_i-alternation.*

Combining Corollary 4.7 with Fact 3.1 yields a criterion for $\mathcal{B}\Sigma_i$-membership: a regular language (resp. ω-language) is definable in $\mathcal{B}\Sigma_i$ iff its syntactic morphism has bounded Σ_i-alternation. This is the approach used in [16] to obtain a (language) membership algorithm for $\mathcal{B}\Sigma_2$. More precisely, the following result is proved.

Theorem 4.8 ([16])**.** *Given as input a morphism $\alpha : A^+ \to S$ into a finite semigroup S, one can decide whether α has bounded Σ_2-alternation or not.*

In Section 6 we obtain our (ω-language) algorithm for $\mathcal{B}\Sigma_2$-membership by proving that given a morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ as input, one can decide whether α_∞ has bounded Σ_2-alternation or not. More precisely, we prove that α_∞ having bounded Σ_2-alternation is equivalent to two decidable properties of α. The first one is that α_+ has bounded Σ_2-alternation (which we can decide by Theorem 4.8) and the second is a simple equation that needs to be satisfied by (S_+, S_∞).

5. A Separation Algorithm for Σ_2

In this section, we present an algorithm for the ω-language separation problem associated to Σ_2. As expected, this algorithm is based on the computation of Σ_2-chains of length 2 (see Theorem 4.2): we prove that given a morphism α into a finite ω-semigroup, one can compute $\mathcal{C}_{2,2}[\alpha_\infty]$.

Given any *alphabet compatible* morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ into a finite ω-semigroup, we denote by $\mathrm{Calc}_{\Sigma_2}(\alpha)$ the set of all pairs

$$(r_1(s_1)^\infty,\ r_2(s_2)^\omega t_2) \in S_\infty \times S_\infty$$

with $(r_1, r_2) \in \mathcal{C}_{2,2}[\alpha_+]$, $(s_1, s_2) \in \mathcal{C}_{2,2}[\alpha_+]$, $t_2 \in \alpha(A^\infty)$ and $\mathsf{alph}(s_1) = \mathsf{alph}(t_2)$ (note that the last condition is well defined since α is alphabet compatible).

Proposition 5.1. *Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞). Then, $\mathcal{C}_{2,2}[\alpha_\infty] = \mathrm{Calc}_{\Sigma_2}(\alpha)$.*

A simple consequence of Proposition 5.1 is that the ω-language separation problem is decidable for Σ_2. Indeed, recall that for any two regular ω-languages, one may compute a single alphabet compatible ω-semigroup morphism that recognizes them both. Therefore, it follows from Theorem 4.2 that deciding Σ_2-separation amounts to having an algorithm that computes $\mathcal{C}_{2,2}[\alpha_\infty]$ from α.

We obtain this algorithm from Proposition 5.1 since $\mathrm{Calc}_{\Sigma_2}(\alpha)$ may be computed, given α as input. Indeed, by Theorem 4.3, we already know that the set $\mathcal{C}_{2,2}[\alpha_+]$ can be computed from α. Once one has this set in hand, computing $\mathrm{Calc}_{\Sigma_2}(\alpha)$ is a simple matter. Hence, we obtain the desired corollary.

Corollary 5.2. *The ω-language separation problem is decidable for Σ_2.*

An important remark is that we use Theorem 4.3 as a black box: we do not reprove that $\mathcal{C}_{2,2}[\alpha_+]$ may be computed from α_+. This is not an immediate result. In fact the proof of [16] requires to use a framework that is more general than Σ_2-chains (that of "Σ_2-junctures") as well as arguments that are independent from those that we are going to use to prove Proposition 5.1.

It remains to prove each inclusion of Proposition 5.1. We first prove the easiest one: $\mathcal{C}_{2,2}[\alpha_\infty] \supseteq \mathrm{Calc}_{\Sigma_2}(\alpha)$ (this proves correctness: all computed chains are indeed Σ_2-chains).

5.1. **Correctness Proof in Proposition 5.1.**

In this subsection, we show $\mathcal{C}_{2,2}[\alpha_\infty] \supseteq \mathrm{Calc}_{\Sigma_2}(\alpha)$. Set $(r_1, r_2) \in \mathcal{C}_{2,2}[\alpha_+]$, $(s_1, s_2) \in \mathcal{C}_{2,2}[\alpha_+]$ and $t_2 \in \alpha(A^\infty)$ such that $\mathsf{alph}(s_1) = \mathsf{alph}(t_2)$. Our objective is to prove that $(r_1(s_1)^\infty, r_2(s_2)^\omega t_2) \in \mathcal{C}_{2,2}[\alpha_\infty]$. Set $k \geqslant 1$. By definition, we need to find two ω-words $w_1 \lesssim_2^k w_2$ such that $\alpha(w_1) = r_1(s_1)^\infty$ and $\alpha(w_2) = r_2(s_2)^\omega t_2$.

By hypothesis, we have four words $x_1, x_2, y_1, y_2 \in A^+$ such that $x_1 \lesssim_2^k x_2$, $y_1 \lesssim_2^k y_2$, $\alpha(x_1) = r_1$, $\alpha(x_2) = r_2$, $\alpha(y_1) = s_1$ and $\alpha(y_2) = s_2$. Moreover, we have an ω-word $v \in A^\infty$ such $\alpha(v) = t_2$ and $\mathsf{alph}(y_1) = \mathsf{alph}(v)$. Set $w_1 = x_1(y_1)^\infty$ and $w_2 = x_2(y_2)^{2^k\omega}v$. Observe that by definition, we have $\alpha(w_1) = r_1(s_1)^\infty$ and $\alpha(w_2) = r_2(s_2)^\omega t_2$. Therefore, it remains to prove that $w_1 \lesssim_2^k w_2$.

By Corollary 3.6, we obtain that $(y_1)^\infty \lesssim_2^k (y_1)^{2^k\omega}v$. Moreover, using $y_1 \lesssim_2^k y_2$ and $v \lesssim_2^k v$ together with Lemma 3.4, we obtain $(y_1)^{2^k\omega}v \lesssim_2^k (y_2)^{2^k\omega}v$. Therefore, by transitivity $(y_1)^\infty \lesssim_2^k (y_2)^{2^k\omega}v$. Finally, we use the fact that $x_1 \lesssim_2^k x_2$ and Lemma 3.4 to conclude that $x_1(y_1)^\infty \lesssim_2^k x_2(y_2)^{2^k\omega}v$, *i.e.*, that $w_1 \lesssim_2^k w_2$. ☐

5.2. **Completeness Proof in Proposition 5.1.**

The converse inclusion $\mathcal{C}_{2,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_2}(\alpha)$ that we show now is more difficult. We devote the rest of this section to its proof. Before we start the proof, we require two additional results that we will use.

Preliminary Results. The first result that we need is a standard decomposition lemma, which may be applied to ω-words. We state it in the lemma below.

Lemma 5.3. *Let $\gamma : A^+ \to S$ be a morphism into a finite semigroup S. Then for every ω-word $w \in A^\infty$, there exists an idempotent $e \in S$ and a decomposition $w = u_0u_1u_2u_3 \cdots$ of w into infinitely many factors $u_0, u_1, u_2, \cdots \in A^+$ satisfying $\gamma(u_j) = e$ for all $j \geqslant 1$ (there is no constraint on u_0).*

The proof of Lemma 5.3 is standard and is a consequence of Ramsey Theorem over infinite graphs (see [26] for example).

In order to state the second result that we will need, we require some additional terminology about Σ_i-chains. Given $i, k, n \geqslant 1$, $x \in \{+, \infty\}$ and $\beta : A^x \to S$ a map into a finite set S, we denote by $\mathcal{C}_{i,n}^k[\beta]$ the set of all chains $(s_1, \dots, s_n) \in S^n$ for which there exist $w_1, \dots, w_n \in A^x$ satisfying

– for all j, $\beta(w_j) = s_j$.

– $w_1 \lesssim_i^k w_2 \lesssim_i^k \cdots \lesssim_i^k w_n$.

Note that by definition, $\mathcal{C}_{i,n}[\beta] = \bigcap_{k \geqslant 1} \mathcal{C}^k_{i,n}[\beta]$. We will use the following fact, which may be verified from Fact 3.2 and finiteness of S.

Fact 5.4. *Let $i, n \geqslant 1$, $x \in \{+, \infty\}$ and let $\beta : A^x \to S$ be a map into a finite set S. Then, for all $k \geqslant 1$,*

$$\mathcal{C}_{i,n}[\beta] \subseteq \mathcal{C}^{k+1}_{i,n}[\beta] \subseteq \mathcal{C}^k_{i,n}[\beta]$$

In particular, there exists ℓ (depending on i, n and β) such $\mathcal{C}_{i,n}[\beta] = \mathcal{C}^{\ell}_{i,n}[\beta]$.

Finally, let us mention the following closure properties of $\mathcal{C}_i[\beta]$, which will be used in the proof of the membership problem for $\mathcal{B}\Sigma_2$: the set $\mathcal{C}_i[\beta]$ is closed under subwords and duplication of letters. This is immediate that the definition and the fact that for all k, $\lesssim^k_i$ is transitive and reflexive.

Fact 5.5. *Set $x \in \{+, \infty\}$, $\beta : A^x \to S$ a map into a finite set S and let $(s_1, \dots, s_n) \in \mathcal{C}_i[\beta]$. Then for all $j \leqslant n$,*

$$(s_1, \dots, s_{j-1}, s_{j+1}, \dots, s_n) \in \mathcal{C}_i[\beta] \text{ and } (s_1, \dots, s_{j-1}, s_j, s_j, s_{j+1}, \dots, s_n) \in \mathcal{C}_i[\beta]$$

Remark 5.6. *A simple consequence of Fact 5.5 is that, by Higman's lemma, $\mathcal{C}_i[\beta]$ is a regular language over the alphabet S (and is therefore finitely representable). However, this fact is useless in the paper: whether an automata for this regular language can be computed is open in the cases that we consider.*

Proof of Proposition 5.1. We may now prove the inclusion $\mathcal{C}_{2,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_2}(\alpha)$ in Proposition 5.1. We set $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ as an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞). We exhibit a number $\ell \geqslant 1$ such that $\mathcal{C}^{\ell}_{2,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_2}(\alpha)$. By the inclusions in Fact 5.4, this will prove that $\mathcal{C}_{2,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_2}(\alpha)$.

We begin with the choice of the number $\ell \geqslant 1$. We know from Fact 5.4 that there exists a number ℓ_+ such that $\mathcal{C}_{2,2}[\alpha_+] = \mathcal{C}^{\ell_+}_{2,2}[\alpha_+]$. We assume without loss of generality that $\ell_+ \geqslant 2$ (by the inclusions in Fact 5.4 we may choose ℓ_+ as large as we want). Furthermore, we set $p = |S_+| + 1$. We define $\ell = \ell_+ + p$.

It now remains to prove that $\mathcal{C}^{\ell}_{2,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_2}(\alpha)$. Set $(q, q') \in \mathcal{C}^{\ell}_{2,2}[\alpha_\infty]$, we have to prove that $(q, q') \in \mathrm{Calc}_{\Sigma_2}(\alpha)$. By definition of $\mathrm{Calc}_{\Sigma_2}(\alpha)$, this means that we have to find $r_1, r_2, s_1, s_2 \in S_+$ and $t_2 \in S_\infty$ such that

$$(1) \qquad \begin{array}{l} (r_1, r_2) \in \mathcal{C}_{2,2}[\alpha_+] \\ (s_1, s_2) \in \mathcal{C}_{2,2}[\alpha_+] \\ t_2 \in \alpha(A^\infty) \text{ with } \mathsf{alph}(s_1) = \mathsf{alph}(t_2) \end{array} \qquad \text{and} \qquad \begin{array}{rcl} q & = & r_1(s_1)^\infty \\ q' & = & r_2(s_2)^\omega t_2 \end{array}$$

We proceed as follows. First, we use the definition of $\mathcal{C}^{\ell}_{2,2}[\alpha_\infty]$ to obtain two ω-words w and w' of images q and q' such that $w \lesssim^{\ell}_2 w'$. We then use the hypothesis $w \lesssim^{\ell}_2 w'$ together with our decomposition lemma, Lemma 3.3, to split w and w' into factors. Finally, we use this decomposition to find the appropriate r_1, r_2, s_1, s_2 and t such that (1) holds.

Decomposition of w and w'. Using Lemma 5.3 (with α_+ as the morphism γ) we may decompose w as an infinite product $w = u_0 u_1 u_2 \cdots$ $(u_0, u_1, u_2, \dots \in A^+)$ such that $\alpha(u_1) = \alpha(u_2) = \alpha(u_3) = \cdots$ is an idempotent e of S_+. Furthermore, note that since α is alphabet compatible, $u_1, u_2, \dots$ all share the same alphabet. Let B be this alphabet.

We now apply Lemma 3.3 p times to the ω-words $w \lesssim^{\ell}_2 w'$. This yields a decomposition $w' = u'_0 u'_1 \cdots u'_{p-1} v$ $(u'_0, u'_1, \dots, u'_{p-1} \in A^+$ and $v \in A^\infty)$ which satisfies the following fact (recall that $\ell = \ell_+ + p$),

Fact 5.7. *For all $j \leqslant p - 1$, $u_j \lesssim^{\ell_+}_2 u'_j$ and $u_p u_{p+1} \cdots \lesssim^{\ell_+}_2 v$.*

Construction of r_1, r_2, s_1, s_2 and t_2. We may now use the decomposition of w and w' to construct the appropriate r_1, r_2, s_1, s_2 and t_2 such that (1) holds.

Since $p = |S_+| + 1$, by the pigeonhole principle, we obtain $i < j \leqslant p - 1$ such that $\alpha_+(u'_0 \cdots u'_i) = \alpha_+(u'_0 \cdots u'_j) = \alpha_+(u'_0 \cdots u'_i)\alpha_+(u'_{i+1} \cdots u'_j)$. Hence, $\alpha_+(u'_0 \cdots u'_i)$ is stable by right multiplication by $\alpha_+(u'_{i+1} \cdots u'_j)$. Iterating this equality, we get

$$\alpha_+(u'_0 \cdots u'_i) = \alpha_+(u'_0 \cdots u'_i)(\alpha_+(u'_{i+1} \cdots u'_j))^\omega.$$

Set $x_1 = u_0 \cdots u_i \in A^+$, $x_2 = u'_0 \cdots u'_i \in A^+$, $y_1 = u_{i+1} \cdots u_j \in A^+$ and $y_2 = u'_{i+1} \cdots u'_j \in A^+$. Moreover, we set $r_1 = \alpha_+(x_1)$, $r_2 = \alpha_+(x_2)$, $s_1 = \alpha_+(y_1)$ and $s_2 = \alpha_+(y_2)$. Note that by the equality above, we have

$$r_2 = r_2(s_2)^\omega.$$

Finally, we set $z = u'_{i+1} \cdots u'_p v$ and $t_2 = \alpha_\infty(z)$.

It remains to prove that (1) holds. By definition, $s_1 = \alpha_+(u_{i+1} \cdots u_j)$ is the idempotent e, therefore

$$q = \alpha_\infty(w) = r_1(s_1)^\infty.$$

Moreover, we have $w' = x_2 z$, therefore,

$$q' = r_2 t_2 = r_2(s_2)^\omega t_2.$$

To conclude that (1) holds, it remains to prove that $(r_1, r_2), (s_1, s_2) \in \mathcal{C}_{2,2}[\alpha_+]$ and $\mathsf{alph}(s_1) = \mathsf{alph}(t_2)$. This is what we do now.

We know from Lemma 3.4 and Fact 5.7 that $x_1 \lesssim_2^{\ell_+} x_2$ and $y_1 \lesssim_2^{\ell_+} y_2$. This exactly says that $(r_1, r_2), (s_1, s_2) \in \mathcal{C}_{2,2}^{\ell_+}[\alpha_+]$. Therefore, by choice of ℓ_+, we have $(r_1, r_2), (s_1, s_2) \in \mathcal{C}_{2,2}[\alpha_+]$. Finally, it is simple to find a Σ_2 sentence of rank 2 that tests the alphabet of an ω-word. Since $\ell_+ \geqslant 2$ and $u_{i+1}u_{i+2} \cdots \lesssim_2^{\ell_+} z$ (see Lemma 3.4 and Fact 5.7), we therefore have

$$\mathsf{alph}(t_2) = \mathsf{alph}(z) = \mathsf{alph}(u_{i+1}u_{i+2} \cdots) = B = \mathsf{alph}(y_1) = \mathsf{alph}(s_1)$$

This terminates the proof of Proposition 5.1. □

6. A Membership Algorithm for $\mathcal{B}\Sigma_2$

In this section, we present our (ω-language) membership algorithm for $\mathcal{B}\Sigma_2$. The algorithm is stated as a decidable characterization of $\mathcal{B}\Sigma_2$ over ω-words.

Theorem 6.1. *Let $L \subseteq A^\infty$ be a regular ω-language and $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be the alphabet completion of its syntactic morphism. The following are equivalent:*

(1) L is definable in $\mathcal{B}\Sigma_2$.

(2) α_∞ has bounded Σ_2-alternation.

(3) α_+ has bounded Σ_2-alternation and α satisfies the following equation:

$$(s_2(t_2)^\omega)^\infty = (s_2(t_2)^\omega)^\omega s_1(t_1)^\infty \quad \text{for all } (s_1, s_2), (t_1, t_2) \in \mathcal{C}_{2,2}[\alpha_+] \text{ with } \mathsf{alph}(s_1) = \mathsf{alph}(t_1) \tag{2}$$

Theorems 4.8 and 4.3 entail that Item 3 in Theorem 6.1 is decidable (note however that these two theorems state difficult results of [16] whose proofs are independent from that of Theorem 6.1). Indeed, Theorem 4.8 exactly states that whether α_+ has bounded Σ_2-alternation is decidable. Moreover, once one has the set $\mathcal{C}_{2,2}[\alpha_+]$ in hand (which is computable by Theorem 4.3), deciding Equation (2) may be achieved by checking all possible combinations. Therefore, we obtain as a corollary of Theorem 6.1 that the ω-language membership problem for $\mathcal{B}\Sigma_2$ is decidable.

Corollary 6.2. *The ω-language membership problem is decidable for $\mathcal{B}\Sigma_2$.*

It now remains to prove Theorem 6.1. The most difficult (and interesting) direction is 3) $\Rightarrow$ 2). As we did in the previous section, we first prove the easier 2) $\Rightarrow$ 1) and 1) $\Rightarrow$ 3) directions.

Proof of 2) $\Rightarrow$ 1). If α_∞ has bounded Σ_2-alternation, we know from Corollary 4.7 that any ω-language recognized by α is definable in $\mathcal{B}\Sigma_2$. But one of these ω-languages is L itself, since α is the alphabet completion of its syntactic morphism. Hence, L is definable in $\mathcal{B}\Sigma_2$, which completes the proof of this implication.

Proof of 1) $\Rightarrow$ 3). Assume that L is definable in $\mathcal{B}\Sigma_2$. In particular, this means that every language or ω-language recognized by α is definable in $\mathcal{B}\Sigma_2$ (we know from Fact 3.1 that it is true for the syntactic morphism of L, so this is true as well for its alphabet completion α, as one can test the alphabet of a word in $\mathcal{B}\Sigma_2$).

Since every language recognized by α is definable in $\mathcal{B}\Sigma_2$, we know from Corollary 4.7 that α_+ has bounded Σ_2-alternation. It remains to prove that Equation (2) is satisfied. Set $(s_1,s_2),(t_1,t_2) \in \mathcal{C}_{2,2}[\alpha_+]$ with $\mathsf{alph}(s_1) = \mathsf{alph}(t_1)$. We have to prove that $(s_2(t_2)^\omega)^\infty = (s_2(t_2)^\omega)^\omega s_1(t_1)^\infty$.

As every ω-language recognized by α is definable in $\mathcal{B}\Sigma_2$, we may pick a $\mathcal{B}\Sigma_2$ sentence φ defining $\alpha^{-1}((s_2(t_2)^\omega)^\infty)$ and let k be its quantifier rank. By definition of s_1,s_2,t_1,t_2, we have words $x_1,x_2,y_1,y_2 \in A^+$ such that $x_1 \lesssim_2^k x_2$, $y_1 \lesssim_2^k y_2$, $\alpha(x_1) = s_1$, $\alpha(x_2) = s_2$, $\alpha(y_1) = t_1$, $\alpha(y_2) = t_2$ and $\mathsf{alph}(x_1) = \mathsf{alph}(y_1)$. Set

$$w_1 = (x_2(y_2)^{2^k\omega})^{2^k\omega}x_1(y_1)^\infty \quad \text{and} \quad w_2 = (x_2(y_2)^{2^k\omega})^\infty$$

Observe that $\alpha(w_1) = (s_2(t_2)^\omega)^\omega s_1(t_1)^\infty$ and $\alpha(w_2) = (s_2(t_2)^\omega)^\infty$. In particular, this means that $w_2 \models \varphi$. We prove that $w_1 \cong_2^k w_2$. By choice of k, this will prove that $w_1 \models \varphi$ and by definition of φ, that $\alpha(w_1) = (s_2(t_2)^\omega)^\infty$. This will exactly mean that $(s_2(t_2)^\omega)^\infty = (s_2(t_2)^\omega)^\omega s_1(t_1)^\infty$ and conclude the proof.

We prove $w_1 \lesssim_2^k w_2$ and $w_2 \lesssim_2^k w_1$. Note that since $\mathsf{alph}(x_1) = \mathsf{alph}(y_1)$, $x_1 \lesssim_2^k x_2$ and $y_1 \lesssim_2^k y_2$, we have $\mathsf{alph}(x_1) = \mathsf{alph}(x_2) = \mathsf{alph}(y_1) = \mathsf{alph}(y_2)$ (since the alphabet of a word may be tested in Σ_2). Therefore, we have $\mathsf{alph}(x_1(y_1)^\infty) = \mathsf{alph}(x_2(y_2)^{2^k\omega})$ and we may use Corollary 3.6 to conclude that $w_2 \lesssim_2^k w_1$. Conversely, we know that $\mathsf{alph}((x_2(y_2)^{2^k\omega})^\infty) = \mathsf{alph}(y_1)$. Therefore, we may use Corollary 3.6 again to obtain that $(y_1)^\infty \lesssim_2^k (y_1)^{2^k\omega}(x_2(y_2)^{2^k\omega})^\infty$. That $w_1 \lesssim_2^k w_2$ is then immediate from this inequality by Lemma 3.4 and transitivity of $\lesssim_2^k$.

Proof of 3) $\Rightarrow$ 2). For this last and more difficult implication, we need to introduce a preliminary result that we will use in the proof. This result is a generalized version of Proposition 5.1, which gives an effective description of the set of Σ_2-chains of length n for all $n \geqslant 2$ (whereas Proposition 5.1 is the specific case $n = 2$).

6.1. **Generalization of Proposition 5.1.**

Essentially, Proposition 5.1 gives a description of the set $\mathcal{C}_{2,2}[\alpha_\infty]$ of Σ_2-chains of length 2 for α_∞ as a function of the sets $\mathcal{C}_{2,2}[\alpha_+]$ and $\alpha(A^\infty)$. Here, we generalize this to the set $\mathcal{C}_{2,n}[\alpha_\infty]$ for an arbitrary fixed length n of Σ_2-chains. In order to understand this generalization, an important observation is that $\alpha(A^\infty)$ is exactly the set $\mathcal{C}_{2,1}[\alpha_\infty]$ of Σ_2-chains of length 1. In other words, Proposition 5.1 describes the Σ_2-chains of length 2 for α_∞ as a function of the Σ_2-chains of length 2 for α_+ and of the Σ_2-chains of length 1 for α_∞. Our generalization states that for all $n \geqslant 2$, the result still holds when replacing length 2 by length n and length 1 by length $n-1$.

Given any *alphabet compatible* morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ into a finite ω-semigroup and any $n \geqslant 2$, we denote by $\mathrm{Calc}_{\Sigma_2,n}(\alpha)$ the set of all pairs

$$(r_1(s_1)^\infty, r_2(s_2)^\omega t_2, \dots, r_n(s_n)^\omega t_n) \in (S_\infty)^n$$

with $(r_1,\dots,r_n) \in \mathcal{C}_{2,n}[\alpha_+]$, $(s_1,\dots,s_n) \in \mathcal{C}_{2,n}[\alpha_+]$, $(t_2,\dots,t_n) \in \mathcal{C}_{2,n-1}[\alpha_\infty]$ and $\mathsf{alph}(s_1) = \mathsf{alph}(t_2)$.

Proposition 6.3. *Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞) and $n \geqslant 2$. Then, $\mathcal{C}_{2,n}[\alpha_\infty] = \mathrm{Calc}_{\Sigma_2,n}(\alpha)$.*

The proof of Proposition 6.3 is a straightforward generalization of that of Proposition 5.1 and is left to the reader.

6.2. **Proof of** 3) $\Rightarrow$ 2) **in Theorem 6.1.** We now have all the material we need to prove the implication 3) $\Rightarrow$ 2) in Theorem 6.1. The proof is based on a new object that may be associated to any alphabet compatible ω-semigroup morphism. We first present this object and then explain why it may be used to prove that 3) $\Rightarrow$ 2).

Graph Associated to an ω-Semigroup Morphism. Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞). We explain how to associate a graph $G[\alpha]$ to α.

We denote by S_+^1 the monoid constructed from S_+ as follows. If S_+ is already a monoid (*i.e.*, if it has a neutral element 1_{S_+}) then $S_+^1 = S_+$. Otherwise $S_+^1 = S_+ \cup \{1_{S_+}\}$ where 1_{S_+} is defined as an artificial neutral element. We associate to α a graph $G[\alpha]$ whose set of nodes is $S_+^1 \times S_\infty$ and whose edges are labeled by subsets of the alphabet (*i.e.*, elements of 2^A). Given two nodes (s_+, s_∞) and (t_+, t_∞) and $B \in 2^A$, there is an edge,

$$(s_+, s_\infty) \xrightarrow{B} (t_+, t_\infty)$$

if and only if there exist $(p_1, p_2), (q_1, q_2) \in \mathcal{C}_2[\alpha_+]$ and $q \in \alpha(A^+)$ such that $\mathsf{alph}(p_1) = \mathsf{alph}(q_1) = \mathsf{alph}(q) = B$ and,

$$s_+ p_1 (q_1)^\infty = s_\infty \qquad \text{and} \qquad s_+ p_2 (q_2)^\omega q = t_+$$

Observe that the definition of the edge relation does not depend on t_∞ in the destination node. Given any alphabet $B \subseteq A$, we call *B-cycle* of $G[\alpha]$ a cycle of $G[\alpha]$ in which all edges are labeled by B. Finally, we say that $G[\alpha]$ is *recursive* if and only if there exist $B \subseteq A$ and a B-cycle such that this cycle contains two nodes (s_+, s_∞) and (t_+, t_∞) with $s_\infty \neq t_\infty$.

Proof of the Theorem. Now that we have defined the graph $G[\alpha]$, the implication 3) $\Rightarrow$ 2) in Theorem 6.1 is an immediate consequence of the two following lemmas.

Lemma 6.4. *Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ as an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞) and assume that α satisfies Equation* (2). *Then $G[\alpha]$ is not recursive.*

Lemma 6.5. *Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ as an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞) and assume that,*
(1) α_+ has bounded Σ_2-alternation.
(2) α_∞ has unbounded Σ_2-alternation.
Then $G[\alpha]$ is recursive.

Before we prove these two lemmas, we use them to conclude the proof of Theorem 6.1. Let $L \subseteq A^\infty$ be a regular ω-language and let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be the alphabet completion of its syntactic morphism. Assume that condition 3) holds in Theorem 6.1, *i.e.*, α_+ has bounded Σ_2-alternation and α satisfies (2). We have to prove that α_∞ has bounded Σ_2-alternation. We proceed by contradiction. Assume that α_∞ has unbounded Σ_2-alternation. We now have three hypotheses:

(a) α satisfies (2).
(b) α_+ has bounded Σ_2-alternation.
(c) α_∞ has unbounded Σ_2-alternation.

Therefore, it follows from Lemma 6.4 that $G[\alpha]$ is not recursive and from Lemma 6.5 that $G[\alpha]$ is recursive, which is a contradiction. We conclude that α_∞ has bounded Σ_2-alternation which terminates the proof of Theorem 6.1.

It now remains to prove Lemma 6.4 and Lemma 6.5. We devote a subsection to each proof.

6.3. **Proof of Lemma 6.4.** Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞) that satisfies (2). Consider a B-cycle in $G[\alpha]$ and let (s_+, s_∞) and (t_+, t_∞) be two nodes in this B-cycle. We have to prove that $s_\infty = t_\infty$. We first prove that we may actually assume that (s_+, s_∞) and (t_+, t_∞) are the only nodes in the cycle. This follows from the next fact.

Fact 6.6. *For all $B \subseteq A$, $\xrightarrow{B}$ is transitive.*

Proof. Let (r_+, r_∞), (s_+, s_∞) and (t_+, t_∞) be three nodes such that,

$$(r_+, r_\infty) \xrightarrow{B} (s_+, s_\infty) \xrightarrow{B} (t_+, t_\infty)$$

By definition of the left edge, we have $(p_1, p_2), (q_1, q_2) \in \mathcal{C}_2[\alpha_+]$ and $q \in \alpha(A^+)$ such that $\mathsf{alph}(p_1) = \mathsf{alph}(q_1) = \mathsf{alph}(q) = B$, $r_+ p_1 (q_1)^\infty = r_\infty$ and $r_+ p_2 (q_2)^\omega q = s_+$.

Moreover, it follows from the definition of the right edge that we have q' such that $\mathsf{alph}(q') = B$ and $s_+ q' = t_+$. Set $q'' = qq'$, we now have $\mathsf{alph}(p_1) = \mathsf{alph}(q_1) = \mathsf{alph}(q'') = B$, $r_+ p_1 (q_1)^\infty = r_\infty$ and $r_+ p_2 (q_2)^\omega q'' = t_+$. We conclude that $(r_+, r_\infty) \xrightarrow{B} (t_+, t_\infty)$. ☐

It is immediate from Fact 6.6 that,

$$(s_+, s_\infty) \xrightarrow{B} (t_+, t_\infty) \quad \text{and} \quad (t_+, t_\infty) \xrightarrow{B} (s_+, s_\infty)$$

By definition of B-labeled edges we have $(p_1, p_2), (q_1, q_2), (p'_1, p'_2), (q'_1, q'_2) \in \mathcal{C}_2[\alpha_+]$ and $q, q' \in S_+$ such that,

a) $\mathsf{alph}(p_1) = \mathsf{alph}(q_1) = \mathsf{alph}(q) = \mathsf{alph}(p'_1) = \mathsf{alph}(q'_1) = \mathsf{alph}(q') = B$.
b) $s_+ p_1 (q_1)^\infty = s_\infty$.
c) $s_+ p_2 (q_2)^\omega q = t_+$.
d) $t_+ p'_1 (q'_1)^\infty = t_\infty$.
e) $t_+ p'_2 (q'_2)^\omega q' = s_+$.

We now use these equalities to prove that s_∞ and t_∞ are equal. Using *c)* and *e)*, we obtain $s_+ = t_+ p'_2 (q'_2)^\omega q' = s_+ p_2 (q_2)^\omega q p'_2 (q'_2)^\omega q'$. Therefore, s_+ is stable by right multiplication by $p_2 (q_2)^\omega q p'_2 (q'_2)^\omega q'$, hence

$$\begin{array}{rcl} s_+ & = & s_+ (p_2 (q_2)^\omega q p'_2 (q'_2)^\omega q') \\ & = & s_+ (p_2 (q_2)^\omega q p'_2 (q'_2)^\omega q')^{\omega+1} \end{array} \tag{3}$$

whence by *a)*

$$\begin{array}{rcl} s_\infty & = & s_+ (p_2 (q_2)^\omega q p'_2 (q'_2)^\omega q')^{\omega+1} p_1 (q_1)^\infty \\ & = & s_+ p_2 (q_2)^\omega \cdot (q p'_2 (q'_2)^\omega q' p_2 (q_2)^\omega)^\omega \cdot q p'_2 (q'_2)^\omega q' p_1 (q_1)^\infty \end{array}$$

Now, by closure of $\mathcal{C}_2[\alpha_+]$ under product and since $(p_1, p_2) \in \mathcal{C}_2[\alpha_+]$, we obtain that

$$(q p'_2 (q'_2)^\omega q' p_1, q p'_2 (q'_2)^\omega q' p_2) \in \mathcal{C}_2[\alpha_+].$$

One can also verify that $\mathsf{alph}(q p'_2 (q'_2)^\omega q' p_1) = B = \mathsf{alph}(q_1)$. Therefore, we may apply (2) to obtain

$$\begin{array}{rcl} s_\infty & = & s_+ p_2 (q_2)^\omega (q p'_2 (q'_2)^\omega q' p_2 (q_2)^\omega)^\infty \\ & = & s_+ (p_2 (q_2)^\omega q p'_2 (q'_2)^\omega q')^\infty \end{array}$$

We now prove that $t_\infty = s_+ (p_2 (q_2)^\omega q p'_2 (q'_2)^\omega q')^\infty$ as well. By (3) and Items *d)* and *c)*, we get

$$\begin{array}{rcl} t_\infty & = & s_+ (p_2 (q_2)^\omega q p'_2 (q'_2)^\omega q')^{\omega+1} p_2 (q_2)^\omega q p'_1 (q'_1)^\infty \\ & = & s_+ p_2 (q_2)^\omega q p'_2 (q'_2)^\omega \cdot (q' p_2 (q_2)^\omega q p'_2 (q'_2)^\omega)^\omega \cdot q' p_2 (q_2)^\omega q p'_1 (q'_1)^\infty \end{array}$$

Since by hypothesis, we have $p'_1, q'_1 \in \mathcal{C}_2[\alpha_+]$, we get by closure under product that

$$(q'p_2(q_2)^\omega qp'_1, q'p_2(q_2)^\omega qp'_2) \in \mathcal{C}_2[\alpha_+].$$

One can also verify that $\mathsf{alph}(q'p_2(q_2)^\omega qp'_1) = B = \mathsf{alph}(q'_1)$. Therefore, we may apply (2) to obtain,

$$\begin{array}{rcl} t_\infty & = & s_+p_2(q_2)^\omega qp'_2(q'_2)^\omega(q'p_2(q_2)^\omega qp'_2(q'_2)^\omega)^\infty \\ & = & s_+(p_2(q_2)^\omega qp'_2(q'_2)^\omega q')^\infty \end{array}$$

We conclude that $s_\infty = t_\infty$ which terminates the proof. □

6.4. **Proof of Lemma 6.5.** Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞) such that,

(1) α_+ has bounded alternation.
(2) α_∞ has unbounded alternation.

We have to prove that $G[\alpha]$ is recursive. Let $B \subseteq A$. We say that a node (s_+, s_∞) of $G[\alpha]$ is a B-*generator* if there exists a Σ_2-alternating pair $(s_1, s_2) \in (S_\infty)^2$ for α_∞ such that $s_\infty = s_+s_1 \neq s_+s_2$ and $\mathsf{alph}(s_1) = B$.

Lemma 6.7. *$G[\alpha]$ contains at least one B-generator for some $B \subseteq A$.*

Proof. This is because α_∞ has unbounded Σ_2-alternation. By definition this means that there exists at least one Σ_i-alternating pair $(s_1, s_2) \in S_\infty$ for α_∞ such that $s_1 \neq s_2$. It is then immediate that $(1_{S_+}, s_1)$ is an $\mathsf{alph}(s_1)$-generator. □

Set B as a minimal alphabet (with respect to inclusion) such that there exists a B-generator. That $G[\alpha]$ is recursive is a consequence of the next lemma.

Lemma 6.8. *Let (s_+, s_∞) be any B-generator of $G[\alpha]$. Then, there exists a node (t_+, t_∞) such that*

(1) $s_\infty \neq t_\infty$.
(2) (t_+, t_∞) is a B-generator.
(3) there is a B-labeled edge from (s_+, s_∞) to (t_+, t_∞).

Since $G[\alpha]$ is a finite graph it is immediate from Lemma 6.8 and our choice of B that it must contain a B-cycle in which there are two nodes (s_+, s_∞) and (t_+, t_∞) such that $s_\infty \neq t_\infty$. We conclude that $G[\alpha]$ is recursive. It now remains to prove Lemma 6.8. In particular, this is where we use Proposition 6.3. We devote the remainder of this section to this proof.

Let (s_+, s_∞) be a B-generator of $G[\alpha]$ and let $(s_1, s_2) \in (S_\infty)^2$ be the Σ_2-alternating pair such that $s_\infty = s_+s_1 \neq s_+s_2$. We have to construct (t_+, t_∞) satisfying the conditions of Lemma 6.8. We proceed as follows. First we choose an integer $\ell \geqslant 1$ and use the fact that (s_1, s_2) is Σ_2-alternating to conclude that $(s_1, s_2)^\ell \in \mathcal{C}_{2,2\ell}[\alpha_\infty]$. We then use this result together with Proposition 6.3 to construct the desired node (t_+, t_∞).

Let us begin with the choice of ℓ. This choice is based on the two following facts.

Fact 6.9. *There exists $n_+ \geqslant 1$ such that for any $t_1, t_2 \in S_+$ and any $n \geqslant n_+$, if $(t_1, t_2)^n \in \mathcal{C}_2[\alpha_+]$, then (t_1, t_2) is Σ_2-alternating.*

Fact 6.10. *There exists $n_\infty \geqslant 1$ such that for any $t_1, t_2 \in S_\infty$ and any $n \geqslant n_\infty$, if $(t_1, t_2)^n \in \mathcal{C}_2[\alpha_\infty]$, then (t_1, t_2) is Σ_2-alternating.*

The two facts share identical proofs. We show the first one (it suffices to replace α_+ by α_∞ and $\mathcal{C}_2[\alpha_+]$ by $\mathcal{C}_2[\alpha_\infty]$ to obtain the second one). If for all $t_1, t_2 \in S_+$ and $n \geqslant 1$, we have $(t_1, t_2)^n \in \mathcal{C}_2[\alpha_+]$, then choose $n_+ = 1$. Otherwise, since $\mathcal{C}_2[\alpha_+]$ is closed under subwords (Fact 5.5), if $(t_1, t_2)^k \notin \mathcal{C}_2[\alpha_+]$, then for all $j \geqslant k$, we have $(t_1, t_2)^j \notin \mathcal{C}_2[\alpha_+]$ as well. Therefore, one can define n_+ as the largest integer k such that there exist $t_1, t_2 \in S_+$ with $(t_1, t_2)^{k-1} \in \mathcal{C}_2[\alpha_+]$ but $(t_1, t_2)^k \notin \mathcal{C}_2[\alpha_+]$ (with the convention that $(t_1, t_2)^0 \in \mathcal{C}_2[\alpha_+]$).

We now set $h = \max(n_+, n_\infty)$ and we choose $\ell = |S_+|^4 \times |S_\infty|^2 \times h$. Since (s_1, s_2) is Σ_2-alternating, we have in particular $(s_1, s_2)^\ell \in \mathcal{C}_{2,2\ell}[\alpha_\infty]$. We now use this fact together with Proposition 6.3 to construct (t_+, t_∞).

Since $(s_1, s_2)^\ell \in \mathcal{C}_{2,2\ell}[\alpha_\infty]$, we may use Proposition 6.3 to obtain $(p_1, \dots, p_{2\ell}) \in \mathcal{C}_{2,2\ell}[\alpha_+]$, $(q_1, \dots, q_{2\ell}) \in \mathcal{C}_{2,2\ell}[\alpha_+]$ and $(t_2, \dots, t_{2\ell}) \in \mathcal{C}_{2,2\ell-1}[\alpha_\infty]$ such that:

– $\mathsf{alph}(t_2) = \mathsf{alph}(q_1)$
– $p_1(q_1)^\infty = s_1$.
– for all $i \geqslant 1$, $p_{2i}(q_{2i})^\omega t_{2i} = s_2$.
– for all $i \geqslant 1$, $p_{2i+1}(q_{2i+1})^\omega t_{2i+1} = s_1$.

We set $C = \mathsf{alph}(p_1)$ and $D = \mathsf{alph}(q_1) = \mathsf{alph}(t_2)$. Observe that since $p_1(q_1)^\infty = s_1$, we have $C \cup D = \mathsf{alph}(s_1) = B$.

Using the pigeonhole principle and our choice of ℓ, we obtain a set of h indices $I = \{i_1, \dots, i_h\}$ such that for all $i, j \in I$,

$$\begin{array}{lllllllll} p_{2i} & = & p_{2j} & q_{2i} & = & q_{2j} & t_{2i} & = & t_{2j} \\ p_{2i+1} & = & p_{2j+1} & q_{2i+1} & = & q_{2j+1} & t_{2i+1} & = & t_{2j+1} \end{array}$$

We define, $p'_2 = p_{2i}$, $p'_3 = p_{2i+1}$, $q'_2 = q_{2i}$, $q'_3 = q_{2i+1}$, $t'_2 = t_{2i}$ and $t'_3 = t_{2i+1}$ (for $i \in I$). Since Σ_2-chains are closed under subwords (see Fact 5.5) and I is of size h, we know that

$$\begin{array}{lll} (p_1, p'_2, p'_3, \dots, p'_2, p'_3) & \in & \mathcal{C}_{2,2h+1}[\alpha_+] \\ (q_1, q'_2, q'_3, \dots, q'_2, q'_3) & \in & \mathcal{C}_{2,2h+1}[\alpha_+] \\ (t'_2, t'_3, \dots, t'_2, t'_3) & \in & \mathcal{C}_{2,2h}[\alpha_\infty] \end{array}$$

In particular, this means that $(p'_2, p'_3)^h \in \mathcal{C}_{2,2h}[\alpha_+]$, $(q'_2, q'_3)^h \in \mathcal{C}_{2,2h}[\alpha_+]$ and $(t'_2, t'_3)^h \in \mathcal{C}_{2,2h}[\alpha_\infty]$. By choice of h, it follows that (p'_2, p'_3), (q'_2, q'_3) and (t'_2, t'_3) are Σ_2-alternating for α_+ and α_∞. Furthermore, since α_+ has bounded Σ_2-alternation, it follows that $p'_2 = p'_3$ and $q'_2 = q'_3$. Let us summarize what we have so far. We have $(p_1, p'_2), (q_1, q'_2) \in \mathcal{C}_2[\alpha_+]$ and $(t'_2, t'_3) \in \mathcal{C}_2[\alpha_\infty]$ which is Σ_2-alternating for α_∞ such that

(1) $s_1 = p_1(q_1)^\infty$.
(2) $s_2 = p'_2(q'_2)^\omega t'_2$.
(3) $s_1 = p'_2(q'_2)^\omega t'_3$.

We may now define the node (t_+, t_∞). Set $t_+ = s_+ p'_2(q'_2)^\omega$ and $t_\infty = s_+ p'_2(q'_2)^\omega t'_2$. We prove that this node satisfies the conditions of Lemma 6.8. We have $t_\infty = s_+ s_2$, which is different from $s_+ s_1 = s_\infty$ by definition of (s_1, s_2). Hence, we have $s_\infty \neq t_\infty$. Moreover, we have $(t'_2, t'_3) \in \mathcal{C}_2[\alpha_\infty]$ which is Σ_2-alternating and,

$$\begin{array}{lllllll} t_+ t'_2 & = & s_+ p'_2(q'_2)^\omega t'_2 & = & s_+ s_2 & = & t_\infty \\ t_+ t'_3 & = & s_+ p'_2(q'_2)^\omega t'_3 & = & s_+ s_1 & = & s_\infty \end{array}$$

and we already know that $s_\infty \neq t_\infty$. Therefore, (t_+, t_∞) is an $\mathsf{alph}(t'_2)$-generator. Furthermore, is it simple to verify that as an element of the Σ_2-chain, $(t_2, \dots, t_\ell)$, t'_2 has the same alphabet as t_2, *i.e.*, $\mathsf{alph}(t'_2) = D \subseteq B$. It follows that (t_+, t_∞) is a D-generator and by minimality of B that $D = B$.

Finally we prove that there is a B-labeled edge from (s_+, s_∞) to (t_+, t_∞). Observe that

– $s_+(p_1(q_1)^\omega)(q_1)^\infty = s_\infty$.
– $s_+(p'_2(q'_2)^\omega)(q'_2)^\omega(q'_2)^\omega = t_+$.

Therefore, since we already know that $(p_1(q_1)^\omega, p'_2(q'_2)^\omega), (q_1, q'_2) \in \mathcal{C}_2[\alpha_+]$, it suffices to prove that $\mathsf{alph}(p_1(q_1)^\omega) = \mathsf{alph}(q_1) = \mathsf{alph}((q'_2)^\omega) = B$ to conclude that there is a B-labeled edge from (s_+, s_∞) to (t_+, t_∞). By definition, we have $\mathsf{alph}(p_1(q_1)^\omega) = C \cup D = B$. Similarly, we have $\mathsf{alph}(q_1) = D$ and we have already established that $D = B$. Finally, since $(q_1, q'_2) \in \mathcal{C}_2[\alpha_+]$, we have $\mathsf{alph}(q'_2) = \mathsf{alph}(q_1) = B$ and $\mathsf{alph}((q'_2)^\omega) = B$, which terminates the proof. ☐

7. A Separation Algorithm for Σ_3

We now present our algorithm for the ω-language separation problem associated to Σ_3. As for Σ_2, this algorithm is based on Theorem 4.2: we present a procedure for computing $\mathcal{C}_{3,2}[\alpha_\infty]$ from an input morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$.

However, in this case, this computation requires a new ingredient. This new ingredient is a generalization of Σ_i-chains that we call *mixed chains*.

Mixed Chains. Set $x \in \{+, \infty\}$ and $\beta : A^x \to S$ as a map into some finite set S. We define a set $\mathcal{M}[\beta] \subseteq S^3$. Let $\overline{s} = (s_1, s_2, s_3) \in S^3$ be a chain over S. We have $\overline{s} \in \mathcal{M}[\beta]$ if and only if for all $k \in \mathbb{N}$, there exist $w_1, w_2, w_3 \in A^x$ such that,

$$\beta(w_1) = s_1,\ \beta(w_2) = s_2,\ \beta(w_3) = s_3 \quad \text{and} \quad w_1 \lesssim_2^k w_2 \lesssim_3^k w_3$$

Mixed chains were first introduced in [14] under a different name: "$\Sigma_{2,3}$-trees" (here, 2 and 3 denote Σ_2 and Σ_3). In fact "$\Sigma_{2,3}$-trees" are a more general notion: what we call mixed chains are a particular case of "$\Sigma_{2,3}$-trees". Essentially $\Sigma_{2,3}$-trees are trees of depth 3 whose nodes are labeled by elements of a finite set S and mixed chains are the special case when there is only a single branch in the tree.

An important remark is that we will not present any algorithm for computing mixed chains. On the other hand, our algorithm for computing $\mathcal{C}_{3,2}[\alpha_\infty]$ from an ω-semigroup morphism α is parametrized by the set of mixed chains $\mathcal{M}[\alpha_+]$. That $\mathcal{M}[\alpha_+]$ may be computed from α_+ is a very difficult result of [14], stated below.

Theorem 7.1 ([14]). *Given as input a morphism $\alpha : A^+ \to S$ into a finite semigroup S, one can compute the set $\mathcal{M}[\alpha]$ of mixed chains for α.*

We may now present our separation algorithm for Σ_3 over ω-words. Set $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ as an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞). We define $\mathrm{Calc}_{\Sigma_3}(\alpha) \subseteq (S_\infty)^2$ as the set of all pairs

$$\big(r_2(s_2(t_2)^\omega)^\infty,\ r_3(s_3(t_3)^\omega)^\omega s_1(t_1)^\infty\big)$$

with $(r_2, r_3) \in \mathcal{C}_{3,2}[\alpha_+]$, $(s_1, s_2, s_3) \in \mathcal{M}[\alpha_+]$, $(t_1, t_2, t_3) \in \mathcal{M}[\alpha_+]$ and $\mathsf{alph}(s_1) = \mathsf{alph}(t_1)$. Since we know from Theorem 7.1 that one may compute $\mathcal{M}[\alpha_+]$ from α, it is immediate from the definition that one may compute $\mathrm{Calc}_{\Sigma_3}(\alpha)$ from α.

Proposition 7.2. *Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞). Then, $\mathcal{C}_{3,2}[\alpha_\infty] = \mathrm{Calc}_{\Sigma_3}(\alpha)$.*

As for Σ_2, it is immediate that Proposition 7.2 yields an algorithm for Σ_3-separation over ω-words. Indeed, it provides an algorithm computing $\mathcal{C}_{3,2}[\alpha_\infty]$ from any alphabet compatible morphism α, which suffices to decide Σ_3-separation.

Corollary 7.3. *The ω-language separation problem is decidable for Σ_3.*

It remains to prove Proposition 7.2. We proceed as for Σ_2. We first prove the easiest inclusion $\mathcal{C}_{3,2}[\alpha_\infty] \supseteq \mathrm{Calc}_{\Sigma_3}(\alpha)$.

Proof of the inclusion $\boldsymbol{\mathcal{C}_{3,2}[\alpha_\infty] \supseteq \mathrm{Calc}_{\Sigma_3}(\alpha)}$. Let $(r_2, r_3) \in \mathcal{C}_{3,2}[\alpha_+]$, $(s_1, s_2, s_3) \in \mathcal{M}[\alpha_+]$ and $(t_1, t_2, t_3) \in \mathcal{M}[\alpha_+]$ be chains such that $\mathsf{alph}(s_1) = \mathsf{alph}(t_1)$. We have to prove that the pair $(r_2(s_2(t_2)^\omega)^\infty, r_3(s_3(t_3)^\omega)^\omega s_1(t_1)^\infty)$ belongs to $\mathcal{C}_{3,2}[\alpha_\infty]$. Set $k \geqslant 1$, we need to find two ω-words $w_2 \lesssim_3^k w_3$ such that $\alpha(w_2) = r_2(s_2(t_2)^\omega)^\infty$ and $\alpha(w_3) = r_3(s_3(t_3)^\omega)^\omega s_1(t_1)^\infty$. The definition gives words $x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3$ such that

– $\alpha(x_j) = r_j$, $\alpha(y_j) = s_j$, $\alpha(z_j) = t_j$

– $x_2 \lesssim_3^k x_3$, $y_1 \lesssim_2^k y_2 \lesssim_3^k y_3$ and $z_1 \lesssim_2^k z_2 \lesssim_3^k z_3$.

Moreover, as $\mathsf{alph}(s_1) = \mathsf{alph}(t_1)$, we have $\mathsf{alph}(y_1) = \mathsf{alph}(z_1)$. We define $w_2 = x_2(y_2(z_2)^{2^k\omega})^\infty$ and $w_3 = x_3(y_3(z_3)^{2^k\omega})^{2^k\omega}y_1z_1^\infty$. It is immediate from this definition that $\alpha(w_2) = r_2(s_2(t_2)^\omega)^\infty$ and that $\alpha(w_3) = r_3(s_3(t_3)^\omega)^\omega s_1(t_1)^\infty$. It remains to prove that $w_2 \lesssim_3^k w_3$.

We first prove $y_1z_1^\infty \lesssim_2^k (y_2(z_2)^{2^k\omega})^\infty$. Since $\mathsf{alph}(y_1) = \mathsf{alph}(z_1)$, we may use Corollary 3.6 to obtain $z_1^\infty \lesssim_2^k (z_1)^{2^k\omega}(y_1(z_1)^{2^k\omega})^\infty$. By Lemma 3.4 and transitivity,

$$y_1z_1^\infty \lesssim_2^k (y_1(z_1)^{2^k\omega})^\infty \lesssim_2^k (y_2(z_2)^{2^k\omega})^\infty \tag{4}$$

We may now use (4) together with Lemma 3.5 to obtain that $(y_2(z_2)^{2^k\omega})^\infty \lesssim_3^k (y_2(z_2)^{2^k\omega})^{2^k\omega}y_1z_1^\infty$. Using Lemma 3.4 and transitivity again, we obtain that

$$x_2(y_2(z_2)^{2^k\omega})^\infty \lesssim_3^k x_3(y_3(z_3)^{2^k\omega})^{2^k\omega}y_1z_1^\infty$$

This exactly says that $w_2 \lesssim_3^k w_3$ which concludes the proof. □

It remains to prove the second inclusion of Proposition 7.2, that is, $\mathcal{C}_{3,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_3}(\alpha)$. Note that the proof relies on Lemma 5.3 and Fact 5.4 that we presented for the proof of the Σ_2 algorithm. We will also need an additional result about mixed chains (it is essentially the 'mixed chains' version of Fact 5.4).

7.1. **Preliminary Result.** Given $x \in \{+,\infty\}$ and $\beta : A^x \to S$ a map into a finite set S, for all $k \geqslant 1$, we denote by $\mathcal{M}^k[\beta]$ the set of all chains $(s_1, s_2, s_3) \in S^3$ such that there exist $w_1, w_2, w_3 \in A^x$ satisfying,
– for all j, $\beta(w_j) = s_j$.
– $w_1 \lesssim_2^k w_2 \lesssim_3^k w_3$.
Note that by definition, $\mathcal{M}[\beta] = \bigcap_{k\geqslant 1} \mathcal{M}^k[\beta]$. Moreover, the following fact may be verified from the definition (this uses Fact 3.2 and the fact that S is finite).

Fact 7.4. *Let $x \in \{+,\infty\}$ and let $\beta : A^x \to S$ be a map into a finite set S. Then, for all $k \geqslant 1$,*

$$\mathcal{M}[\beta] \subseteq \mathcal{M}^{k+1}[\beta] \subseteq \mathcal{M}^k[\beta].$$

Moreover, there exists ℓ (depending on β) such that $\mathcal{M}[\beta] = \mathcal{M}^\ell[\beta]$.

7.2. **Proof of Proposition 7.2.** We now prove that $\mathcal{C}_{3,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_3}(\alpha)$ in Proposition 7.2. We follow a proof template which is similar to the one we used to prove Proposition 5.1. Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite ω-semigroup (S_+, S_∞). We exhibit a number $\ell \geqslant 1$ such that $\mathcal{C}^\ell_{3,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_3}(\alpha)$ (by Fact 5.4, this suffices since $\mathcal{C}_{3,2}[\alpha_\infty] \subseteq \mathcal{C}^\ell_{3,2}[\alpha_\infty]$ for any ℓ).

We begin by choosing the number ℓ. We know from Fact 5.4 and Fact 7.4 that there exists ℓ_+ such that $\mathcal{M}[\alpha_+] = \mathcal{M}^{\ell_+}[\alpha_+]$ and $\mathcal{C}_{3,2}[\alpha_+] = \mathcal{C}^{\ell_+}_{3,2}[\alpha_+]$. Moreover, we may assume without loss of generality that $\ell_+ \geqslant 2$ (we may choose ℓ_+ as large as we want). Set $p = |S_+| + 1$. We define $\ell' = \ell_+ + p^2$ and $\ell = \ell' + p$.

We have to prove that $\mathcal{C}^\ell_{3,2}[\alpha_\infty] \subseteq \mathrm{Calc}_{\Sigma_3}(\alpha)$. Set $(q, q') \in \mathcal{C}^\ell_{3,2}[\alpha_\infty]$, we prove that $(q, q') \in \mathrm{Calc}_{\Sigma_3}(\alpha)$. By definition, of $\mathrm{Calc}_{\Sigma_3}(\alpha)$, we have to find $r_2, r_3, s_1, s_2, s_3, t_1, t_2$ and t_3 in S_+ such that,

$$\begin{array}{l} (r_2, r_3) \in \mathcal{C}_{3,2}[\alpha_+] \\ (s_1, s_2, s_3) \in \mathcal{M}[\alpha_+] \\ (t_1, t_2, t_3) \in \mathcal{M}[\alpha_+] \\ \mathsf{alph}(s_1) = \mathsf{alph}(t_1) \end{array} \quad \text{and} \quad \begin{array}{lcl} q & = & r_2(s_2(t_2)^\omega)^\infty \\ q' & = & r_3(s_3(t_3)^\omega)^\omega s_1(t_1)^\infty \end{array} \tag{5}$$

The existence of $r_2, r_3, s_1, s_2, s_3, t_1, t_2$ and t_3 in S_+ is a consequence of the following lemma.

Lemma 7.5. *There exist words $u_2, u_3, x_2, x_3, y_1, y_2, y_3, z_1, z_2$ and z_3 in A^+ such that*

(1) $x_2 \lesssim_3^{\ell_+} x_3$, $y_1 \lesssim_2^{\ell_+} y_2 \lesssim_3^{\ell_+} y_3$, $z_1 \lesssim_2^{\ell_+} z_2 \lesssim_3^{\ell_+} z_3$ and $u_2 \lesssim_3^{\ell_+} u_3$.

(2) $\mathit{alph}(u_3y_1) = \mathit{alph}(z_1)$*.*
(3) $q = \alpha(x_2(y_2(z_2)^\omega u_2)^\infty)$ *and* $q' = \alpha(x_3(y_3(z_3)^\omega u_3)^\omega y_1(z_1)^\infty)$*.*

Before proving Lemma 7.5, let us explain how we use it to conclude the proof of Proposition 7.2. We set

$$\begin{array}{rcl} r_2 & = & \alpha(x_2(y_2(z_2)^\omega u_2)^{\omega-1}y_2(z_2)^\omega) \\ r_3 & = & \alpha(x_3(y_3(z_3)^\omega u_3)^{\omega-1}y_3(z_3)^\omega) \end{array} \qquad \begin{array}{rcl} s_1 & = & \alpha(u_3y_1) \\ s_2 & = & \alpha(u_2y_2) \\ s_3 & = & \alpha(u_3y_3) \end{array} \qquad \begin{array}{rcl} t_1 & = & \alpha(z_1) \\ t_2 & = & \alpha(z_2) \\ t_3 & = & \alpha(z_3) \end{array}$$

We have to prove that these choices satisfy the conditions in (5). The facts that $(r_2, r_3) \in \mathcal{C}_{3,2}[\alpha_+]$, $(s_1, s_2, s_3) \in \mathcal{M}[\alpha_+]$ and $(t_1, t_2, t_3) \in \mathcal{M}[\alpha_+]$ is a simple consequence of the first item in Lemma 7.5 and our choice of ℓ_+. Let us detail the case of (s_1, s_2, s_3) (other cases are handled similarly). By the first item in Lemma 7.5, we know that $y_1 \lesssim_2^{\ell_+} y_2 \lesssim_3^{\ell_+} y_3$ and $u_2 \lesssim_3^{\ell_+} u_3$. Also notice the second inequality means that $u_3 \lesssim_2^{\ell_+} u_2 \lesssim_3^{\ell_+} u_3$ (this is comes from the last item in Fact 3.2). We may now apply Lemma 3.4 to obtain that $u_3y_1 \lesssim_2^{\ell_+} u_2y_2 \lesssim_3^{\ell_+} u_3y_3$. By definition of s_1, s_2, s_3, this exactly says that $(s_1, s_2, s_3) \in \mathcal{C}_{3,2}^{\ell_+}[\alpha_+]$ and we chose ℓ_+ so that $\mathcal{C}_{3,2}^{\ell_+}[\alpha_+] = \mathcal{C}_{3,2}[\alpha_+]$. We conclude that $(s_1, s_2, s_3) \in \mathcal{M}[\alpha_+]$.

Moreover, that $\mathsf{alph}(s_1) = \mathsf{alph}(t_1)$ is immediate from the definitions of s_1 and t_1 since we have $\mathsf{alph}(u_3y_1) = \mathsf{alph}(z_1)$ in Lemma 7.5.
Finally, we prove that $q = r_2(s_2(t_2)^\omega)^\infty$ and $q' = r_3(s_3(t_3)^\omega)^\omega s_1(t_1)^\infty$. We have:

$$\begin{array}{rcl} r_2(s_2(t_2)^\omega)^\infty & = & \alpha(x_2(y_2(z_2)^\omega u_2)^{\omega-1}y_2(z_2)^\omega(u_2y_2(z_2)^\omega)^\infty) \\ & = & \alpha(x_2(y_2(z_2)^\omega u_2)^{\omega-1}(y_2(z_2)^\omega u_2)^\infty) \\ & = & \alpha(x_2(y_2(z_2)^\omega u_2)^\infty) \\ & = & q \quad \text{by the third item in Lemma 7.5} \end{array}$$

and

$$\begin{array}{rcl} r_3(s_3(t_3)^\omega)^\omega s_1(t_1)^\infty & = & \alpha(x_3(y_3(z_3)^\omega u_3)^{\omega-1}y_3(z_3)^\omega(u_3y_3(z_3)^\omega)^\omega u_3y_1(z_1)^\infty) \\ & = & \alpha(x_3(y_3(z_3)^\omega u_3)^{\omega-1}(y_3(z_3)^\omega u_3)^{\omega+1}y_1(z_1)^\infty) \\ & = & \alpha(x_3(y_3(z_3)^\omega u_3)^{2\omega}y_1(z_1)^\infty) \\ & = & \alpha(x_3(y_3(z_3)^\omega u_3)^\omega y_1(z_1)^\infty) \\ & = & q' \quad \text{by the third item in Lemma 7.5} \end{array}$$

Therefore, our choices of $r_2, r_3, s_1, s_2, s_3, t_1, t_2$ and t_3 satisfy the conditions of (5), which exactly means that $(q, q') \in \mathrm{Calc}_{\Sigma_3}(\alpha)$ and we are finished with the proof of Proposition 7.2.

It now remains to prove Lemma 7.5. We proceed in two steps. We first prove the existence of a set of words and ω-words that satisfy weaker properties than the desired words u_2, u_3, x_2, x_3, y_1, y_2, y_3, z_1, z_2 and z_3 in Lemma 7.5. We then use this first set of words and ω-words to construct the desired words u_2, u_3, x_2, x_3, y_1, y_2, y_3, z_1, z_2 and z_3. We devote a subsection to each step.

7.3. **First Step in the Proof of Lemma 7.5.**

As explained, this first step consists in the construction of a set of words and ω-words that we will then use in the second step to construct the words $u_2, u_3, x_2, x_3, y_1, y_2, y_3, z_1, z_2$ and z_3 of Lemma 7.5. We state the construction in the lemma below. Recall that we have set $p = |S_+| + 1$, $\ell' = p^2 + \ell_+$ and $\ell = p + \ell$, and that $(q, q') \in \mathcal{C}_{3,2}^{\ell}[\alpha_\infty]$.

Lemma 7.6. *There exist words* $w_0, w_1, w_0', w_1' \in A^+$ *and an* ω*-word* $w_\infty' \in A^\infty$ *such that:*
(1) $w_0 \lesssim_3^{\ell'} w_0'$ *and* $w_1 \lesssim_3^{\ell'} w_1'$*.*
(2) $w_\infty' \lesssim_2^{\ell'} (w_1)^\infty$
(3) $q = \alpha(w_0(w_1)^\infty)$ *and* $q' = \alpha(w_0'(w_1')^\omega w_\infty')$*.*

We devote the subsection to the proof of Lemma 7.6. By definition, $(q, q') \in \mathcal{C}_{3,2}^{\ell}[\alpha_\infty]$, which means that there exist two ω-words $w, w' \in A^\infty$ of images q, q' under α such that $w \lesssim_3^{\ell} w'$. We construct $w_0, w_1, w'_0, w'_1 \in A^+$ and $w'_\infty \in A^\infty$ by decomposing w and w'. Our first move is to decompose w as an infinite product using Lemma 5.3. However, if we use α_+ only as the morphism for making the decomposition, we will not have a strong enough link between the factors to prove the desired result. For this reason, we apply the lemma for a morphism carrying more information than α_+ does. Let us define this morphism.

We know since Lemma 3.4 that over A^+, the equivalence $\cong_3^{\ell'}$ is a congruence for concatenation, hence the quotient $A^+/\cong_3^{\ell'}$ is a semigroup. Moreover, since there are only finitely many non equivalent $\mathcal{B}\Sigma_3$ sentences of quantifier rank ℓ', this semigroup is finite. We write

$$\begin{array}{rccl} \gamma: & A^+ & \to & S_+ \times (A^+/\cong_3^{\ell'}) \\ & u & \mapsto & (\alpha(u), [u]_{\cong_3^{\ell'}}) \end{array}$$

where $[u]_{\cong_3^{\ell'}}$ denotes the $\cong_3^{\ell'}$-equivalence class of u. We now apply Lemma 5.3 to $w \in A^\infty$ with γ as the morphism. We obtain a decomposition $w = u_0u_1u_2\cdots$ $(u_0, u_1, u_2, \cdots \in A^+)$ such that $\gamma(u_1) = \gamma(u_2) = \gamma(u_3) = \cdots$ is an idempotent of $S_+ \times (A^+/\cong_3^{\ell'})$. In other words, the decompositions satisfies the two following properties:

– there exists an idempotent $e \in S_+$ such that all factors of the form $u_iu_{i+1}\cdots u_j$ with $1 \leqslant i \leqslant j$ have image e under α.
– all factors of the form $u_iu_{i+1}\cdots u_j$ with $1 \leqslant i \leqslant j$ are $\cong_3^{\ell'}$-equivalent.

We now use this decomposition of w and the hypothesis $w \lesssim_3^\ell w'$ to decompose w' as well. Since $w \lesssim_3^\ell w'$, we may apply Lemma 3.3 p times to the w and w' to obtain a decomposition $w' = u'_0\cdots u'_{p-1}u'_\infty$ of w' $(u'_0, u'_1, \ldots, u'_{p-1} \in A^+$ and $u'_\infty \in A^\infty)$ which satisfies the following fact (recall that $\ell = \ell' + p$).

Fact 7.7. *For all $j \leqslant p-1$, $u_j \lesssim_3^{\ell'} u'_j$ and $u_pu_{p+1}\cdots \lesssim_3^{\ell'} u'_\infty$.*

Since we chose $p = |S_+| + 1$, we may now use the pigeonhole principle to obtain $i < j \leqslant p-1$ such that $\alpha(u'_0\cdots u'_i) = \alpha(u'_0\cdots u'_i \cdot u'_{i+1}\cdots u'_j)$, that is, $\alpha(u'_0\cdots u'_i)$ is stable by right multiplication by $\alpha(u'_{i+1}\cdots u'_j)$. This gives us the following fact.

Fact 7.8. *We have $\alpha(u'_1\cdots u'_i) = \alpha(u'_1\cdots u'_i(u'_{i+1}\cdots u'_j)^\omega)$.*

We may now define the words $w_0, w_1, w'_0, w'_1 \in A^+$ and the ω-word $w'_\infty \in A^\infty$ in the lemma. We set,

$$\begin{array}{rclcrclcrcl} w_0 & = & u_0\cdots u_i & \quad & w_1 & = & u_{i+1}\cdots u_j & \quad & & & \\ w'_0 & = & u'_0\cdots u'_i & & w'_1 & = & u'_{i+1}\cdots u'_j & & w'_\infty & = & u'_{i+1}\cdots u'_{p-1}u'_\infty \end{array}$$

It now remains to verify that these choices satisfies the conditions of the lemma. That $w_0 \lesssim_3^{\ell'} w'_0$ and $w_1 \lesssim_3^{\ell'} w'_1$ is immediate from Fact 7.7 and Lemma 3.4.

We now prove that $w'_\infty \lesssim_2^{\ell'} (w_1)^\infty$. We know from Fact 7.7 and Lemma 3.4 that $u_{i+1}u_{i+2}\ldots \lesssim_3^{\ell'} w'_\infty$. Moreover, we know by construction and the choice of the morphism γ that $w_1 = u_{i+1}\cdots u_j \cong_3^{\ell'} u_h$ for all $h \geqslant 1$. In particular, this means that $w_1 \lesssim_3^{\ell'} u_h$ for all $h \geqslant 1$. Using Lemma 3.4 again, we obtain that $(w_1)^\infty \lesssim_3^{\ell'} u_{i+1}u_{i+2}\cdots$ and by transitivity that $(w_1)^\infty \lesssim_3^{\ell'} w'_\infty$. Finally, we obtain that $w'_\infty \lesssim_2^{\ell'} (w_1)^\infty$ using the last item in Fact 3.2.

It remains to prove that $q = \alpha(w_0(w_1)^\infty)$ and $q' = \alpha(w'_0(w'_1)^\omega w'_\infty)$. By definition, $\alpha(w_0(w_1)^\infty) = \alpha(w_0)e^\infty = \alpha(u_0u_1u_2\cdots) = \alpha(w) = q$. Moreover, $w'_0w'_\infty = w'$ by definition. Therefore, $\alpha(w'_0w'_\infty) = \alpha(w') = q'$. Finally, since $\alpha(w'_0) = \alpha(w'_0(w'_1)^\omega)$ (this is Fact 7.8), we obtain that $\alpha(w'_0(w'_1)^\omega w'_\infty) = q'$ which terminates the proof of Lemma 7.6.

This finishes the first step in the proof of Lemma 7.5. We present the second and last step in the next subsection.

7.4. **Second Step in the Proof of Lemma 7.5.** We finish the proof of Lemma 7.5 by using the words and ω-words obtained from Lemma 7.6 to construct the desired words u_2, u_3, x_2, x_3, y_1, y_2, y_3, z_1, z_2 and z_3.

Let $w_0, w_1, w_0', w_1' \in A^+$ and $w'_\infty \in A^\infty$ that satisfy the conditions of Lemma 7.6. We begin by using the hypothesis $w'_\infty \lesssim_2^{\ell'} (w_1)^\infty$ to decompose w'_∞ and $(w_1)^\infty$. We apply Lemma 5.3 to w'_∞ (with α_+ as the morphism) to construct a decomposition $w'_\infty = v_0'v_1'v_2' \cdots$ such that $\alpha(v_1') = \alpha(v_2') = \cdots$ is an idempotent f of S_+.

Fact 7.9. *For any $h \geqslant 0$, $\alpha(v_0' \cdots v_h')f^\infty = \alpha(w'_\infty)$.*

An other property that we will use is that since α is alphabet compatible, for all $h \geqslant 1$ the v_h' have the same alphabet. We call B this alphabet.

We now use this decomposition of w'_∞ together with the fact that $w'_\infty \lesssim_2^{\ell'} (w_1)^\infty$ to decompose $(w_1)^\infty$ as well. We apply Lemma 3.3 p^2 times to the ω-words w'_∞ and $(w_1)^\infty$. This yields a decomposition $(w_1)^\infty = v_0 \cdots v_{p^2-1}v_\infty$ of $(w_1)^\infty$ ($v_0, v_1, \dots, v_{p^2-1} \in A^+$ and $v_\infty \in A^\infty$) which satisfies the following fact (recall that $\ell' = \ell_+ + p^2$).

Fact 7.10. *For all $h \leqslant p^2 - 1$, $v_h' \lesssim_2^{\ell_+} v_h$ and $v'_{p^2}v'_{p^2+1} \cdots \lesssim_2^{\ell_+} v_\infty$.*

Observe that since $v_0 \cdots v_{p^2-1}$ is a finite prefix of $(w_1)^\infty$, there exists a number $n \geqslant 1$ such that it is a prefix of $(w_1)^n$. Therefore, there exists $v_{p^2} \in A^+$ such that $v_0 \cdots v_{p^2-1}v_{p^2} = (w_1)^n$. Furthermore, we know from the second item in Lemma 7.6 and Lemma 3.4 that,

$$v_0 \cdots v_{p^2-1}v_{p^2} = (w_1)^n \lesssim_3^{\ell'} (w_1')^n$$

Therefore, we may apply Lemma 3.3 p^2 times to obtain a decomposition $(w_1')^n = v_0'' \cdots v''_{p^2-1}v''_{p^2}$ of $(w_1')^n$ that satisfies the following fact (recall that $\ell' = \ell_+ + p^2$).

Fact 7.11. *For all $h \leqslant p^2$, $v_h \lesssim_3^{\ell_+} v_h''$.*

Finally, since $p = |S_+| + 1$, we have $p^2 \geqslant |S_+|^2 + 1$. Therefore, we may apply the pigeonhole principle to obtain i, j such that $0 \leqslant i < j \leqslant p^2 - 1$, $\alpha(v_0 \cdots v_i) = \alpha(v_0 \cdots v_j)$ and $\alpha(v_0'' \cdots v_i'') = \alpha(v_0'' \cdots v_j'')$. In particular, we obtain the following fact.

Fact 7.12. *We have*

$$\begin{array}{rcl} \alpha(v_0 \cdots v_i) & = & \alpha(v_0 \cdots v_i(v_{i+1} \cdots v_j)^\omega) \\ \alpha(v_0'' \cdots v_i'') & = & \alpha(v_0'' \cdots v_i''(v_{i+1}'' \cdots v_j'')^\omega) \end{array}$$

We are now ready to construct the words $u_1, u_2, x_2, x_3, y_1, y_2, y_3, z_1, z_2$ and z_3 in A^+ from Lemma 7.5. We set,

$$\begin{array}{rcl} x_2 & = & w_0 \\ x_3 & = & w_0' \end{array} \qquad \begin{array}{rcl} y_1 & = & v_0' \cdots v_i' \\ y_2 & = & v_0 \cdots v_i \\ y_3 & = & v_0'' \cdots v_i'' \end{array} \qquad \begin{array}{rcl} z_1 & = & v_{i+1}' \cdots v_j' \\ z_2 & = & v_{i+1} \cdots v_j \\ z_3 & = & v_{i+1}'' \cdots v_j'' \end{array} \qquad \begin{array}{rcl} u_2 & = & v_{i+1} \cdots v_{p^2} \\ u_3 & = & v_{i+1}'' \cdots v_{p^2}'' \end{array}$$

It remains to verify that these words satisfy the conditions of the lemma. We begin with the inequalities. That $x_2 \lesssim_3^{\ell_+} x_3$ is immediate by choice of w_0, w_0' in Lemma 7.6. The inequalities $y_1 \lesssim_2^{\ell_+} y_2 \lesssim_3^{\ell_+} y_3$, $z_1 \lesssim_2^{\ell_+} z_2 \lesssim_3^{\ell_+} z_3$ and $u_2 \lesssim_3^{\ell_+} u_3$ are immediate from Fact 7.10, Fact 7.11 and Lemma 3.4.

Let us now prove that $\mathsf{alph}(u_3y_1) = \mathsf{alph}(z_1) = B$ (recall that B is the shared alphabet of all v_h' for $h \geqslant 1$). Since y_1 is a product of v_h' for $h \geqslant 1$, it is immediate that $\mathsf{alph}(y_1) = B$. Furthermore, using the same argument, it is also immediate that $\mathsf{alph}(z_1) = B$. Therefore, it suffices to prove that $\mathsf{alph}(u_3) \subseteq B$ to conclude that $\mathsf{alph}(u_3y_1) = B = \mathsf{alph}(z_1)$. We know that $u_2 \lesssim_3^{\ell_+} u_3$, therefore $\mathsf{alph}(u_3) = \mathsf{alph}(u_2)$ ($\ell_+ \geqslant 2$ and the alphabet may be tested with a Σ_3 sentence of rank

2). Moreover, by definition, u_2 is a suffix of $(w_1)^p$. Therefore, $\mathsf{alph}(u_2) \subseteq \mathsf{alph}((w_1)^p) = \mathsf{alph}(w_1)$. Finally, we know from Fact 7.10 that $v_{p^2}v_{p^2+1}\cdots \lesssim_2^{\ell+} v_\infty$. Since v_∞ is a suffix of $(w_1)^\infty$, we have $\mathsf{alph}(v_\infty) = \mathsf{alph}(w_1) = B$ and $\mathsf{alph}(u_3) = \mathsf{alph}(u_2) \subseteq B$.

We finish with the proof of the last item in Lemma 7.5: $q = \alpha(x_2(y_2(z_2)^\omega u_2)^\infty)$ and $q' = \alpha(x_3(y_3(z_3)^\omega u_2)^\omega y_1(z_1)^\infty)$. First, by definition, we have $x_2 = w_0$ and $x_3 = w'_0$. Furthermore, we know from Fact 7.12 that

$$\alpha(y_2(z_2)^\omega u_2) = \alpha(y_2u_2) = \alpha((w_1)^n) \quad \text{and} \quad \alpha(y_3(z_3)^\omega u_3) = \alpha(y_3u_3) = \alpha((w'_1)^n)$$

Finally, we obtain from Fact 7.9 that $\alpha(y_1(z_1)^\infty) = \alpha(w'_\infty)$. By combining all of this, we obtain,

$$\begin{array}{lclcl}\alpha(x_2(y_2(z_2)^\omega u_2)^\infty) & = & \alpha(w_0((w_1)^n)^\infty) & = & \alpha(w_0(w_1)^\infty) \\ \alpha(x_3(y_3(z_3)^\omega u_2)^\omega y_1(z_1)^\infty) & = & \alpha(w'_0((w'_1)^n)^\omega w'_\infty) & = & \alpha(w'_0(w'_1)^\omega w'_\infty)\end{array}$$

We conclude that $q = \alpha(x_2(y_2(z_2)^\omega u_2)^\infty)$ and $q' = \alpha(x_3(y_3(z_3)^\omega u_2)^\omega y_1(z_1)^\infty)$ from the third item in Lemma 7.6. This terminates the proof of Lemma 7.5. ☐

8. Conclusion

We proved that for ω-languages, the separation problem is decidable for Σ_2 and Σ_3 and the membership problem is decidable for $\mathcal{B}\Sigma_2$. Note that using a theorem of [19], these results may be lifted to the variants of the same logics whose signature has been enriched with a predicate "+1" that is interpreted as the successor relation. This means that over ω-words, separation is decidable for $\Sigma_2(<,+1)$ and $\Sigma_3(<,+1)$ and membership is decidable for $\mathcal{B}\Sigma_2(<,+1)$.

A gap remains between languages and ω-languages: we leave open the case of Σ_4-membership for ω-languages while it is known to be decidable for languages [14]. The language algorithm was based on two results: 1) the decidability of Σ_3-separation [14] and 2) an effective reduction of Σ_{i+1}-membership to Σ_i-separation [16] (which is generic for all $i \geqslant 1$). In the ω-language setting, we are missing the second result and it is not clear if a similar reduction exists.

References

[1] M. Arfi. Polynomial operations on rational languages. In *Proceedings of the 4th Annual Symposium on Theoretical Aspects of Computer Science, STACS'87*, Lect. Notes Comp. Sci., pages 198–206, Berlin, Heidelberg, 1987. Springer-Verlag.

[2] M. Bojańczyk. The common fragment of ACTL and LTL. In R. Amadio, editor, *Foundations of Software Science and Computational Structures, FoSSaCS'08*, volume 4962 of *Lect. Notes Comp. Sci.*, pages 172–185. Springer-Verlag, 2008.

[3] J. A. Brzozowski and R. Knast. The dot-depth hierarchy of star-free languages is infinite. *Journal of Computer and System Sciences*, 16(1):37–55, 1978.

[4] J. R. Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

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

[6] W. Czerwiński, W. Martens, and T. Masopust. Efficient separability of regular languages by subsequences and suffixes. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming, ICALP'13*, Lect. Notes Comp. Sci., pages 150–161, Berlin, Heidelberg, 2013. Springer-Verlag.

[7] V. Diekert and M. Kufleitner. Fragments of first-order logic over infinite words. *Theory of Computing Systems*, 48(3):486–516, 2011.

[8] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961.

[9] M. Kufleitner and T. Walter. Level two of the quantifier alternation hierarchy over infinite words. *CoRR*, abs/1509.06207, 2015.

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

[11] D. Perrin. Recent results on automata and infinite words. In *Proceedings of the 9th International Symposium on Mathematical Foundations of Computer Science, MFCS'84*, Lect. Notes Comp. Sci., pages 134–148, Berlin, Heidelberg, 1984. Springer-Verlag.

[12] D. Perrin and J.-É. Pin. *Infinite Words*. Elsevier, 2004.

[13] J.-É. Pin and P. Weil. Polynomial closure and unambiguous product. *Theory of Computing Systems*, 30(4):383–422, 1997.

[14] 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.

[15] T. Place, L. van Rooijen, and M. Zeitoun. Separating regular languages by piecewise testable and unambiguous languages. In *Proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science, MFCS'13*, Lect. Notes Comp. Sci., pages 729–740, Berlin, Heidelberg, 2013. Springer-Verlag.

[16] T. Place and M. Zeitoun. Going higher in the first-order quantifier alternation hierarchy on words. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming, ICALP'14*, Lect. Notes Comp. Sci., pages 342–353, Berlin, Heidelberg, 2014. Springer-Verlag.

[17] T. Place and M. Zeitoun. Separating regular languages with first-order logic. In *Proceedings of the Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL'14) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'14)*, pages 75:1–75:10, New York, NY, USA, 2014. ACM.

[18] T. Place and M. Zeitoun. Separating ω-languages without quantifier alternation. Unpublished, 2015.

[19] T. Place and M. Zeitoun. Separation and the successor relation. In preparation, long version of [20], 2015.

[20] T. Place and M. Zeitoun. Separation and the successor relation. In E. W. Mayr and N. Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 662–675, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

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

[22] I. Simon. Piecewise testable events. In *Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages*, pages 214–222, Berlin, Heidelberg, 1975. Springer-Verlag.

[23] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, STOC '73*, pages 1–9, New York, NY, USA, 1973. ACM.

[24] W. Thomas. A concatenation game and the dot-depth hierarchy. In *Computation Theory and Logic*, pages 415–426. Springer-Verlag, Berlin, Heidelberg, 1987.

[25] B. A. Trakhtenbrot. Finite automata and logic of monadic predicates. *Doklady Akademii Nauk SSSR*, 149:326–329, 1961. In Russian.

[26] T. Wilke. An Eilenberg theorem for ∞-languages. In *Proceedings of the 18th International Colloquium on Automata, Languages and Programming, ICALP'91*, volume 510 of *Lect. Notes Comp. Sci.*, pages 588–599, Berlin, Heidelberg, 1991. Springer-Verlag.

Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France