Quantifier Alternation for Infinite Words

Théo Pierron, Thomas Place(✉), and Marc Zeitoun

LaBRI, UMR 5800, University of Bordeaux, 33400 Talence, France
tplace@labri.fr

Abstract. We investigate the expressive power of the 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.

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 [4,5,8,27] (logical view). To understand the structure of this class in depth, it is natural to classify its languages according to their descriptive complexity. The problem is to determine how complicated a sentence has to be 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 [23] 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,

B. Jacobs and C. Löding (Eds.): FOSSACS 2016, LNCS 9634, pp. 234–251, 2016.
DOI: 10.1007/978-3-662-49630-5_14

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 [25] (*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 (cf. figure). Clearly, we have $\Sigma_i \subseteq \mathcal{B}\Sigma_i \subseteq \Sigma_{i+1}$, and these inclusions are known to be strict [3,26]: $\Sigma_i \subsetneq \mathcal{B}\Sigma_i \subsetneq \Sigma_{i+1}$.

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

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 [28]. 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 [24] was generalized from finite to infinite words by Perrin and Pin [12]. For finite words, membership to Σ_2 is known to be decidable [1,15], a result lifted to infinite words in [2,7]. 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 [18], 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 [18], this proved that Σ_3 has decidable membership. This result was strengthened in [16], which shows that Σ_3-separation is decidable as well, thus obtaining decidability of membership for Σ_4. Finally, in [18], 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 [18] 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 also that, for each problem, we pre-compute some information by using the corresponding algorithm designed in [16,18] for finite words. This means that the involved algorithms from [16,18] are used as subroutines of our algorithms.

It is worth noting that the decidability of the membership problem for $\mathcal{B}\Sigma_2$ over infinite words has been obtained independently in [9]. While the algorithm is essentially the same as our own, its proof is completely different.

We now present the problems in depth in Sect. 1, and we solve them in the rest of the paper. A detailed outline is provided at the end of Sect. 1. Due to lack of space, some proofs are postponed to the full version of this paper, see [13].

1 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.

1.1 The Quantifier Alternation Hierarchy of First-Order Logic

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

First-Order Logic. Any word or infinite word can be viewed as a logical structure made of a linearly ordered sequence of positions (finite for words and infinite for infinite words) labeled over alphabet A. 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 infinite words, each sentence φ defines two objects: a language $L_+ = \{w \in A^+ \mid w \models \varphi\}$ and a language of infinite words $L_\infty = \{w \in A^\infty \mid w \models \varphi\}$. For example, the sentence $\exists x \exists y\ (x < y \wedge P_a(y))$ defines the language $A^+a \cup A^+aA^+$ and the language of infinite words 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 languages of infinite words (we speak of FO over infinite words).

Quantifier Alternation. It is usual to classify FO sentences by counting the quantifier alternations inside their prenex normal form. Let $i \in \mathbb{N}$, 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 quantifier blocks) 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 $\exists x_1 \forall x_2 \forall x_3 \exists x_4\ \varphi$, 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 languages of infinite words. Therefore, we obtain two hierarchies: a hierarchy of classes of languages and a hierarchy of classes of languages of infinite words, both of which are known to be strict [3,26].

1.2 Decision Problems

Our objective is to investigate the quantifier alternation hierarchy of first-order logic over infinite words. We rely on two decision problems in order to carry out this investigation: the membership problem and the separation problem. The input of these problems are *regular* languages of finite and infinite words. They are those languages that can be equivalently defined by monadic second-order logic, finite Büchi automata or finite Wilke algebras. We will use Wilke algebras, whose definition is recalled in Sect. 2. Both problems are parametrized by a level in the hierarchy and come therefore in two versions: a 'language' one and a 'language of infinite words' one. Let $\mathcal{F}$ be a level in the hierarchy.

Membership. The *membership problem* for level $\mathcal{F}$ is as follows:

For Finite Words

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

For Infinite words

IN	A regular language of infinite words L
OUT	Is L $\mathcal{F}$-definable ?

Separation. The separation problem is more general. Given three languages or three languages of infinite words 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, L_1 is said $\mathcal{F}$*-separable* from L_2 if there exists an $\mathcal{F}$-definable language or language of infinite words that separates L_1 from L_2. Note that when $\mathcal{F}$ is not closed under complement (*e.g.*, for $\mathcal{F} = \Sigma_i$), 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:

For Finite Words

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

For Infinite words

IN	Two regular languages of infinite words 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 of words or infinite words is definable in $\mathcal{F}$ iff it is $\mathcal{F}$-separable from its (also regular) complement: separation is 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 [10,23], which yields a membership algorithm for FO over words. The result was later generalized to FO over infinite words by Perrin [11]. These results and the techniques used to obtain them provide not only a way to decide whether a regular language of finite or infinite words is FO-definable, but also a generic method for constructing a defining FO sentence, when possible. 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

	Words	Infinite words
FO	Solved [23,10]	Solved [11]
Σ_1	Solved [1]	Solved [14,12]
$\mathcal{B}\Sigma_1$	Solved [24]	Solved [12]
Σ_2	Solved [1,15]	Solved [7,2]
$\mathcal{B}\Sigma_2$	Solved [18]	**This paper**
Σ_3	Solved [18]	**This paper**
Σ_4	Solved [16]	**Open**

Separation Problem

	Words	Infinite words
FO	Solved [19]	Solved [19]
Σ_1	Solved (Folklore)	Solved (Folklore)
$\mathcal{B}\Sigma_1$	Solved [6,17]	Solved [20]
Σ_2	Solved [18]	**This paper**
$\mathcal{B}\Sigma_2$	**Open**	**Open**
Σ_3	Solved [16]	**This paper**
Σ_4	**Open**	**Open**

Our objective is to bridge the gap between the knownledge for languages and that for languages of infinite words. More precisely, we want to extend the results of [16,18] to the setting of infinite 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 stated in the next theorem. Note that the Σ_3-membership algorithm follows from its separation algorithm. We leave open the case of Σ_4-membership for languages of infinite words.

Theorem 1. *The following properties hold:*

(a) the separation problem is decidable for Σ_2 over infinite words.
(b) the membership problem is decidable for $\mathcal{B}\Sigma_2$ over infinite words.
(c) the separation problem is decidable for Σ_3 over infinite words.

Our proof of Theorem 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 [18] or [16] solving the corresponding problem for finite words:

- We present all algorithms in a specific framework which is adapted from the one used in [18]. In particular, we reuse the key notion of "Σ_i-chain" (generalized to infinite words in a straightforward way).
- We actually reuse the algorithms for finite words of [16,18] as subprocedures in our algorithms for languages of infinite words.

The remainder of the paper is devoted to proving Theorem 1. In Sect. 2, we recall classical notions required for our definitions and proofs: the algebraic definition of regular languages of infinite words and logical preorders. In Sect. 3, we present the general framework that we use. 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 [18]). We then devote a section to each algorithm: Sect. 4 to Σ_2-separation, Sect. 5 to $\mathcal{B}\Sigma_2$-membership and Sect. 6 to Σ_3-separation.

2 Preliminaries

We recall some classical notions that we will need. First, we present the definition of regular languages of infinite words in terms of Wilke algebras. Then, we define the logical preorders that one may associate to each level Σ_i in the hierarchy.

2.1 Semigroups and Wilke Algebras

We briefly recall the definition of regular languages and languages of infinite words in terms of semigroups and Wilke algebras. 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 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.

Wilke Algebras. A *Wilke algebra* 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 $s_1, s_2, \dots \in (S_+)^\infty$ 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 a Wilke algebra. See [12] for further details (we use a distinct notation from [12], where what we write s^ω, s^∞ is noted s^π, s^ω, respectively).

We say that (S_+, S_∞) is *finite* if both S_+ and S_∞ are. Note that even if a Wilke algebra 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 [28] that the infinite product is fully determined by the mapping $s \mapsto s^\infty$. This makes it possible to finitely represent any finite Wilke algebra.

Morphisms of Wilke algebras are defined in the natural way. In particular, observe that any morphism of Wilke algebra $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ 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 languages of infinite words (the languages of infinite words $\alpha_\infty^{-1}(F_\infty)$ for $F_\infty \subseteq S_\infty$). A language of infinite words is regular iff it may be recognized by a morphism into a *finite* Wilke algebra.

Syntactic Morphisms. It is known that given any regular language (resp. language of infinite words) 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 that given any regular language of infinite words L, one can compute its syntactic morphism from any representation of L. We state the second one below.

Fact 2. *Let $i \geqslant 1$ and let L be a regular language of infinite words. Then L is definable in $\mathcal{B}\Sigma_i$ iff so are all languages of words and infinite words recognized by its syntactic morphism.*

The proof of Fact 2 may be found in [12] (in fact, this holds for any class of languages of infinite words which forms a "variety" of languages of infinite words, 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 membership algorithm for $\mathcal{B}\Sigma_2$ (see Sect. 5).

Morphisms and Separation. When working on separation, we are given two input languages or languages of infinite words. 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 languages of infinite words) 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 languages of infinite words 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 a Wilke algebra. Hence, we let β be the morphism: $\beta : (A^+, A^\infty) \to (S_+ \times 2^A, S_\infty \times 2^A)$ with $\beta(w) = (\alpha(w), \mathsf{alph}(w))$.

2.2 Logical Preorders

To each level Σ_i in the hierarchy, one may associate preorders on the sets of words and infinite 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 sentence,

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

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 sentences of rank less than k.

We now define the preorders. Note that while we define two preorders for each level Σ_i (one on A^+, one on A^∞), we actually use the same notation for both. Let $i \geqslant 1$ be a level in the hierarchy and $k \geqslant 1$ as a quantifier rank. Given two words $w, w' \in A^+$ (resp two infinite words $w, w' \in A^\infty$), we write $w \lesssim_i^k w'$ if and only if *any* Σ_i sentence of rank at most k satisfied by w is satisfied by w' as well. By contrapositive, since the negation of a Σ_i sentence is in Π_i, we have $w \lesssim_i^k w'$ iff any Π_i sentence of rank at most k satisfied by w' is also satisfied by w.

One may verify that $\lesssim_i^k$ is preorder. Moreover, it is immediate that the preorders get refined when k or i increase: $w \lesssim_i^{k+1} w'$ or $w \lesssim_{i+1}^k w'$ imply $w \lesssim_i^k w'$. Since a Π_{i+1} sentence is in Σ_i, $w \lesssim_{i+1}^k w'$ also implies $w' \lesssim_i^k w$.

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 nothing but Boolean combinations of Σ_i sentences). The following fact sums up what we just observed.

Fact 3. *Let $k, i \geqslant 1$ and let u, v be two words or two infinite 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 4 (Decomposition Lemma). *Let $i, k \geqslant 1$ and let u, v be two words or two infinite words such that $u \lesssim_i^k v$. Then for any decomposition $u = u_1 u_2$ of u, there exist v_1, v_2 such that $v = v_1 v_2$, $u_1 \lesssim_i^{k-1} v_1$ and $u_2 \lesssim_i^{k-1} v_2$.*

Lemma 5 (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 infinite words such that $u_2 \lesssim_i^k v_2$. Then $u_1 u_2 \lesssim_i^k v_1 v_2$ and $u_1^\infty \lesssim_i^k v_1^\infty$.*

The last composition that we state is specific to infinite words.

Lemma 6. *Let $i, k \geqslant 1$, $u \in A^+$ be a word and $v \in A^\infty$ be an infinite 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 6 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 6.

Corollary 7. *Let $k \geqslant 1$, $u \in A^+$ be a word and let $v \in A^\infty$ be an infinite word such that $\mathsf{alph}(u) = \mathsf{alph}(v)$. Then for any $\ell \geqslant 2^k$, we have $u^\infty \lesssim_2^k u^\ell v$.*

3 Σ_i-Chains for Language of Infinite Words

All algorithms for infinite words of this paper are strongly related to the finite words algorithms of [16,18]. In particular, we adapt and reuse the key notion of "Σ_i-chain" which was introduced in [18]. 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 $\mathcal{B}\Sigma_i$.

Σ_i-Chains were initially introduced in [18] as a tool designed to investigate the separation problem over finite words 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 classical lemma.

Lemma 8. *Let $i, k \geqslant 1$ and L_1, L_2 be two languages or two languages of infinite words. 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. $w_1 \cong_i^k w_2$).*

Lemma 8 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 languages of infinite words as input. Therefore, we have a single morphism that recognizes them both. For example, in the case of infinite words, we have $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$, with (S_+, S_∞) a finite Wilke algebra, that recognizes both inputs. Intuitively, S_+ and S_∞ are finite abstractions of A^+ and A^∞. Consequently, 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 if 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 languages of infinite words recognized by α, it is then easy to adapt the two criteria of Lemma 8 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 Wilke algebra (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 $(s_1, \dots, s_n)$ for a chain of length n. We denote chains by $\bar{s}, \bar{t}, \dots$ and sets of chains by $\mathcal{S}, \mathcal{T}, \dots$

If (S_+, S_∞) is a Wilke algebra, 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 a Wilke algebra (in which the mixed and infinite products are defined componentwise as well).

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 $\bar{s} = (s_1, \dots, s_n) \in S^+$ be a chain. We have $\bar{s} \in \mathcal{C}_i[\beta]$ if and only if for all $k \in \mathbb{N}$, there exist $w_1, \dots, w_n \in A^x$ such that:

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

We let $\mathcal{C}_{i,n}[\beta]$ be the restriction of this set to chains of length n: $\mathcal{C}_{i,n}[\beta] = \mathcal{C}_i[\beta] \cap S^n$.

Σ_i-Chains Associated to a 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 [18].

Moreover, given a morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ into a finite Wilke algebra (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.

3.1 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 9. *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 9 is a straightforward consequence of the statement for Σ_i in Lemma 8. 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 languages of infinite words, 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 Wilke algebra (S_+, S_∞). Typically, this algorithm involves computing $\mathcal{C}_{i,2}[\alpha_+]$ first, which can be achieved by reusing the first item, *i.e.*, the algorithm for word languages.

This approach is exactly the one used in [16,18] to solve separation for Σ_2 and Σ_3 over finite words: the following theorems are proven in these papers.

Theorem 10 (see [18]). *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 11 (see [16]). *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 Sect. 4 (for Σ_2) and Sect. 6 (for Σ_3) for infinite 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. The algorithms of Theorems 10 and 11 are reused as subprocedures in these new algorithms for languages of infinite words: computing $\mathcal{C}_{2,2}[\alpha_\infty]$ and $\mathcal{C}_{3,2}[\alpha_\infty]$ requires to first compute $\mathcal{C}_{2,2}[\alpha_+]$ and $\mathcal{C}_{3,2}[\alpha_+]$.

Remark 12. The algorithms of Theorems 10 and 11 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 10 and 11, which we use as black boxes.

3.2 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*.

Let $x \in \{+, \infty\}$ and $\beta : A^x \to S$ be 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 $n \geqslant 1$, we have $(s,t)^n \in \mathcal{C}_i[\beta]$ (where by $(s,t)^n$, we mean the chain $(s,t,s,t,\dots,s,t)$ of length $2n$).

Theorem 13. *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.*

The proof of Theorem 13 is based on the second part of Lemma 8. In view of the theorem, the separation problem for $\mathcal{B}\Sigma_i$ reduces to the computation of the Σ_i-alternating pairs, which is unfortunately open for $i \geqslant 2$, even on finite words.

Regarding membership however, Theorem 13 yields an immediate corollary. For $x \in \{+, \infty\}$ and $\beta : A^x \to S$ a map into a finite set S, we say that β has *bounded Σ_i -alternation* iff every Σ_i-alternating pair $(s,t) \in S^2$ for β satisfies $s = t$.

Corollary 14. *Let $i \geqslant 1$, $x \in \{+, \infty\}$ and $\beta : A^x \to S$ be a map into a finite set S. Then all sets $\beta^{-1}(F)$ for $F \subseteq S$ are $\mathcal{B}\Sigma_i$-definable if and only if β has bounded Σ_i-alternation.*

Combining Corollary 14 with Fact 2 yields a criterion for $\mathcal{B}\Sigma_i$-membership: a regular language of finite or infinite words is definable in $\mathcal{B}\Sigma_i$ iff its syntactic morphism has bounded Σ_i-alternation. This is used in [18] to obtain a (language) membership algorithm for $\mathcal{B}\Sigma_2$. More precisely, the following result is proved.

Theorem 15 (see [18]). *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 Sect. 5 we obtain our algorithm for $\mathcal{B}\Sigma_2$-membership over infinite words 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 is that α_+ has bounded Σ_2-alternation (which we can decide by Theorem 15). The second is a simple equation that (S_+, S_∞) needs to satisfy.

4 A Separation Algorithm for Σ_2

In this section, we present an algorithm for the separation problem associated to Σ_2 over infinite words. As expected, this algorithm is based on the computation of Σ_2-chains of length 2 (see Theorem 9): we prove that given a morphism α into a finite Wilke algebra, one can compute $\mathcal{C}_{2,2}[\alpha_\infty]$.

For an *alphabet compatible* morphism $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ into a finite Wilke algebra, 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 this last condition is well defined since α is alphabet compatible. Recall that s_1^∞ is the infinite product $s_1 s_1 \ldots$, and s_2^ω the idempotent power of s_2 in S_+.

Proposition 16. *Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite Wilke algebra (S_+, S_∞). Then, $\mathcal{C}_{2,2}[\alpha_\infty] = \mathrm{Calc}_{\Sigma_2}(\alpha)$.*

A consequence of Proposition 16 is that the separation problem is decidable for Σ_2 over infinite words. Indeed, recall that for any two regular languages of infinite words, one may compute a single alphabet compatible Wilke algebra morphism that recognizes them both. Therefore, it follows from Theorem 9 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 16 since $\mathrm{Calc}_{\Sigma_2}(\alpha)$ may be computed, given α as input. Indeed, by Theorem 10, we already know that the set $\mathcal{C}_{2,2}[\alpha_+]$ can be computed from α. Hence, we obtain the desired corollary.

Corollary 17. *Over infinite words, the separation problem is decidable for Σ_2.*

An important remark is that we use Theorem 10 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 [18] 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 16.

It remains to prove Proposition 16. We illustrate the algorithm by proving the easier inclusion: $\mathcal{C}_{2,2}[\alpha_\infty] \supseteq \mathrm{Calc}_{\Sigma_2}(\alpha)$ (this proves correctness: all computed chains are indeed Σ_2-chains). The converse inclusion (corresponding to completeness: all Σ_2-chains are computed) is available in the long version of the paper.

Correctness Proof: $\boldsymbol{\mathcal{C}_{2,2}[\alpha_\infty] \supseteq \mathrm{Calc}_{\Sigma_2}(\alpha)}$. Let $(r_1, r_2) \in \mathcal{C}_{2,2}[\alpha_+]$, $(s_1, s_2) \in \mathcal{C}_{2,2}[\alpha_+]$ and $t_2 \in \alpha(A^\infty)$ such that $\mathsf{alph}(s_1) = \mathsf{alph}(t_2)$. Our objective is to prove that $(r_1(s_1)^\infty, r_2(s_2)^\omega t_2) \in \mathcal{C}_{2,2}[\alpha_\infty]$. Let $k \geqslant 1$. By definition, we need to find two infinite 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 infinite word $z \in A^\infty$ such $\alpha(z) = t_2$ and $\mathsf{alph}(y_1) = \mathsf{alph}(z)$. Let $w_1 = x_1(y_1)^\infty$ and $w_2 = x_2(y_2)^{2^k\omega} z$. 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 7, we obtain that $(y_1)^\infty \lesssim_2^k (y_1)^{2^k\omega} z$. Moreover, using $y_1 \lesssim_2^k y_2$ and $z \lesssim_2^k z$ together with Lemma 5, we obtain $(y_1)^{2^k\omega} z \lesssim_2^k (y_2)^{2^k\omega} z$. Therefore, by transitivity $(y_1)^\infty \lesssim_2^k (y_2)^{2^k\omega} z$. Finally, we use the fact that $x_1 \lesssim_2^k x_2$ and Lemma 5 to conclude that $x_1(y_1)^\infty \lesssim_2^k x_2(y_2)^{2^k\omega} z$, *i.e.*, that $w_1 \lesssim_2^k w_2$. $\square$

5 A Membership Algorithm for $\mathcal{B}\Sigma_2$

We now present our membership algorithm for $\mathcal{B}\Sigma_2$ over infinite words. The algorithm is stated as a decidable characterization of $\mathcal{B}\Sigma_2$ over infinite words.

Theorem 18. *Let $L \subseteq A^\infty$ be regular and let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ 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:

$$(st^\omega)^\infty = (st^\omega)^\omega st^\infty \text{ for all } s,t \in \alpha(A^+) \text{ such that } \mathsf{alph}(s) = \mathsf{alph}(t) \quad (1)$$

We know that Item 3 in Theorem 18 is decidable. Indeed, Theorem 15 states that whether α_+ has bounded Σ_2-alternation is decidable (note however that this is a difficult result of [18] whose proof is independent from that of Theorem 18). Moreover, verifying that (1) is satisfied may be achieved by checking all possible combinations. Therefore, we obtain the following corollary of Theorem 18.

Corollary 19. *The membership problem over infinite words is decidable for $\mathcal{B}\Sigma_2$.*

It now remains to prove Theorem 18. That 2) $\Rightarrow$ 1) is immediate from Corollary 14. The most difficult (and interesting) direction is 3) $\Rightarrow$ 2). Due to lack of space, it is proved in the long version of this paper. As we did in the previous section, we illustrate the theorem by proving the easier 1) $\Rightarrow$ 3) direction.

Proof of 1) $\Rightarrow$ 3). Let L be $\mathcal{B}\Sigma_2$-definable. In particular, this means that every language of finite or infinite words recognized by α is definable in $\mathcal{B}\Sigma_2$ (we know from Fact 2 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 $\mathcal{B}\Sigma_2$, Corollary 14 entails that α_+ has bounded Σ_2-alternation. It remains to establish Eq. (1). For $s,t \in \alpha(A^+)$ such that $\mathsf{alph}(s) = \mathsf{alph}(t)$, let us show that $(st^\omega)^\infty = (st^\omega)^\omega st^\infty$.

Let k such that for any $r \in S_\infty$, $\alpha^{-1}(r)$ may be defined by a $\mathcal{B}\Sigma_2$ sentence of quantifier rank less than k (k exists since all these languages of infinite words are definable in $\mathcal{B}\Sigma_2$). By choice of k, for any two infinite words $u,v \in A^\infty$, we have $u \cong_2^k v \Rightarrow \alpha(u) = \alpha(v)$. Therefore, in order to conclude, it suffices to find two infinite words u,v of images $(st^\omega)^\infty$ and $(st^\omega)^\omega st^\infty$ and such that $u \cong_2^k v$.

By definition of s,t, we have words $x,y \in A^+$ such that $\alpha(x) = s$, $\alpha(y) = t$ and $\mathsf{alph}(x) = \mathsf{alph}(y)$. Let $u = (xy^{2^k\omega})^\infty$ and $v = (xy^{2^k\omega})^{2^k\omega}xy^\infty$. It is imediate that u and v have images $(st^\omega)^\infty$ and $(st^\omega)^\omega st^\infty$. It remains to prove that $u \cong_2^k v$.

We prove that $u \lesssim_2^k v$ and $v \lesssim_2^k u$. Observe that $\mathsf{alph}(xy^{2^k\omega}) = \mathsf{alph}(xy^\infty)$. Hence, we get $u \lesssim_2^k v$ from Corollary 7. Conversely, we know that $\mathsf{alph}((xy^{2^k\omega})^\infty) = \mathsf{alph}(y)$. Therefore, we may use Corollary 7 again to obtain $y^\infty \lesssim_2^k y^{2^k\omega}(xy^{2^k\omega})^\infty$. That $v \lesssim_2^k u$ is then immediate from this inequality by Lemma 5.

6 A Separation Algorithm for Σ_3

We present our algorithm for the separation problem associated to Σ_3 over infinite words. As for Σ_2, this algorithm is based on Theorem 9: we give a procedure 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. Let $x \in \{+, \infty\}$ 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 $\bar{s} = (s_1, s_2, s_3) \in S^3$ be a chain over S. We have $\bar{s} \in \mathcal{M}[\beta]$ if and only if for all $k \in \mathbb{N}$, there exist $w_1, w_2, w_3 \in A^x$ such that,

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

Note the definition involves both the preorder "$\lesssim_2^k$" associated to Σ_2 and the preorder "$\lesssim_3^k$" associated to Σ_3 (hence the name "mixed chains"). 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 a 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 [16], stated below.

Theorem 20 (see [16]). *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.*

Remark 21. The presentation of Theorem 20 is different in [16]. It is proved that one can compute the set of "$\Sigma_{2,3}$-trees" associated to α. 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.

We may now present our separation algorithm for Σ_3 over infinite words. Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite Wilke algebra (S_+, S_∞). We define $\text{Calc}_{\Sigma_3}(\alpha) \subseteq (S_\infty)^2$ 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 20 that one may compute $\mathcal{M}[\alpha_+]$ from α, it is immediate from the definition that one may compute $\text{Calc}_{\Sigma_3}(\alpha)$ from α.

Proposition 22. *Let $\alpha : (A^+, A^\infty) \to (S_+, S_\infty)$ be an alphabet compatible morphism into a finite Wilke algebra (S_+, S_∞). Then, $\mathcal{C}_{3,2}[\alpha_\infty] = \text{Calc}_{\Sigma_3}(\alpha)$.*

As for Σ_2, Proposition 22 immediately yields an algorithm for Σ_3-separation over infinite 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 23. *The separation problem over infinite words is decidable for Σ_3.*

It remains to prove Proposition 22. We proceed as for Σ_2. Again, we only prove the easier inclusion and postpone the other to the long version of this paper.

Proof of $\mathcal{C}_{3,2}[\alpha_\infty] \supseteq \mathrm{Calc}_{\Sigma_3}(\alpha)$. 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 $(r_2(s_2(t_2)^\omega)^\infty, r_3(s_3(t_3)^\omega)^\omega s_1(t_1)^\infty) \in \mathcal{C}_{3,2}[\alpha_\infty]$. Let $k \geqslant 1$, we need to find two infinite 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 $x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3$ with:

- $\alpha(x_j) = r_j$, $\alpha(y_j) = s_j$, $\alpha(z_j) = t_j$
- $x_2 \lesssim_3^k x_3$, $y_1 \lesssim_2^k y_2 \lesssim_3^k y_3$ 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 $w_3 = x_3(y_3(z_3)^{2^k\omega})^{2^k\omega} y_1 z_1^\infty$. 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 $y_1 z_1^\infty \lesssim_2^k (y_2(z_2)^{2^k\omega})^\infty$. Since $\mathsf{alph}(y_1) = \mathsf{alph}(z_1)$, we may use Corollary 7 to obtain $z_1^\infty \lesssim_2^k (z_1)^{2^k\omega}(y_1(z_1)^{2^k\omega})^\infty$. By Lemma 5 and transitivity,

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

We may now use (2) together with Lemma 6 to obtain that $(y_2(z_2)^{2^k\omega})^\infty \lesssim_3^k (y_2(z_2)^{2^k\omega})^{2^k\omega} y_1 z_1^\infty$. Using Lemma 5 and transitivity again, we obtain that

$$x_2(y_2(z_2)^{2^k\omega})^\infty \lesssim_3^k x_3(y_3(z_3)^{2^k\omega})^{2^k\omega} y_1 z_1^\infty$$

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

7 Conclusion

We proved that for languages of infinite words, the separation problem is decidable for Σ_2 and Σ_3 and that the membership problem is decidable for $\mathcal{B}\Sigma_2$. Note that using a theorem of [21], these results may be lifted to the variants of these logics whose signature has been enriched with a predicate "+1", that is interpreted as the successor relation. This means that over infinite 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 languages of infinite words: we leave open the case of Σ_4-membership for languages of infinite words while it is known to be decidable for languages [16]. The language algorithm was based on two ingredients: (1) the decidability of Σ_3-separation [16] and (2) an effective reduction of Σ_{i+1}-membership to Σ_i-separation [18] (which is generic for all $i \geqslant 1$). In the setting of languages of infinite words, we are missing the second result and it is not clear whether a similar reduction exists.

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

References

1. Arfi, M.: Polynomial operations on rational languages. In: Brandenburg, F.J., Vidal-Naquet, G., Wirsing, M. (eds.) STACS'87. LNCS, vol. 247, pp. 198–206. Springer, Heidelberg (1987)
2. Bojańczyk, M.: The common fragment of ACTL and LTL. In: Amadio, R.M. (ed.) FOSSACS 2008. LNCS, vol. 4962, pp. 172–185. Springer, Heidelberg (2008)
3. Brzozowski, J.A., Knast, R.: The dot-depth hierarchy of star-free languages is infinite. J. Comput. Syst. Sci. **16**(1), 37–55 (1978)
4. Büchi, J.R.: Weak second-order arithmetic and finite automata. Math. Logic Q. **6**(1–6), 66–92 (1960)
5. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Logic, Methodology and Philosophy of Science (Proc. 1960 Internat. Congr.), pp. 1–11. Stanford Univ. Press, Stanford (1962)
6. Czerwiński, W., Martens, W., Masopust, T.: Efficient separability of regular languages by subsequences and suffixes. In: Fomin, F.V., Freivalds, R., Kwiatkowska, M., Peleg, D. (eds.) ICALP 2013, Part II. LNCS, vol. 7966, pp. 150–161. Springer, Heidelberg (2013)
7. Diekert, V., Kufleitner, M.: Fragments of first-order logic over infinite words. Theory Comput. Syst. **48**(3), 486–516 (2011)
8. Elgot, C.C.: Decision problems of finite automata design and related arithmetics. Trans. Am. Math. Soc. **98**(1), 21–51 (1961)
9. Kufleitner, M., Walter, T.: Level two of the quantifier alternation hierarchy over infinite words. CoRR, abs/1509.06207 (2015)
10. McNaughton, R., Papert, S.A.: Counter-Free Automata. MIT Press, Cambridge (1971)
11. Perrin, D.: Recent results on automata and infinite words. In: Chytil, M.P., Koubek, V. (eds.) MFCS 1984. LNCS, vol. 176, pp. 134–148. Springer, Heidelberg (1984)
12. Perrin, D., Pin, J.É.: Infinite Words. Elsevier, Amsterdam (2004)
13. Pierron, T., Place, T., Zeitoun, M.: Quantifier alternation for infinite words. CoRR, abs/1511.09011 (2015)
14. Pin, J.É.: Positive varieties and infinite words. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998. LNCS, vol. 1380, pp. 76–87. Springer, Heidelberg (1998)
15. Pin, J.É., Weil, P.: Polynomial closure and unambiguous product. Theory Comput.Syst. **30**(4), 383–422 (1997)
16. Place, T.: Separating regular languages with two quantifier alternations. In: Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2015), pp. 202–213. IEEE (2015)
17. Place, T., van Rooijen, L., Zeitoun, M.: Separating regular languages by piecewise testable and unambiguous languages. In: Chatterjee, K., Sgall, J. (eds.) MFCS 2013. LNCS, vol. 8087, pp. 729–740. Springer, Heidelberg (2013)
18. Place, T., Zeitoun, M.: Going higher in the first-order quantifier alternation hierarchy on words. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014, Part II. LNCS, vol. 8573, pp. 342–353. Springer, Heidelberg (2014)

19. Place, T., Zeitoun, M.: Separating regular languages with first-order logic. In: 2014 Proceedings of the Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL 2014), 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2014), pp. 75:1–75:10. ACM, New York (2011)
20. Place, T., Zeitoun, M.: Separating ω-languages without quantifier alternation (2015) (Unpublished)
21. Place, T., Zeitoun, M.: Separation and the successor relation. In preparation, long version of [22] (2015)
22. Place, T., Zeitoun, M.: Separation and the successor relation. In: Mayr, E.W., Ollinger, N. (eds.) 32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015). Leibniz International Proceedings in Informatics (LIPIcs), vol. 30, pp. 662–675. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl (2015)
23. Schützenberger, M.P.: On finite monoids having only trivial subgroups. Inf. Control **8**(2), 190–194 (1965)
24. Simon, I.: Piecewise testable events. In: Brakhage, H. (ed.) Automata Theory and Formal Languages. LNCS, vol. 33, pp. 214–222. Springer, Heidelberg (1975)
25. Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time (preliminary report). In: Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, STOC 1973, pp. 1–9. ACM, New York (1973)
26. Thomas, W.: A concatenation game and the dot-depth hierarchy. In: Börger, E. (ed.) Computation Theory and Logic. LNCS, vol. 270, pp. 415–426. Springer, Heidelberg (1987)
27. Trakhtenbrot, B.A.: Finite automata and logic of monadic predicates. Dokl. Akad. Nauk SSSR **149**, 326–329 (1961). In Russian
28. Wilke, T.: An Eilenberg theorem for ∞-languages. In: Leach Albert, J., Monien, B., Rodríguez Artalejo, M. (eds.) ICALP 1991. LNCS, vol. 510, pp. 588–599. Springer, Heidelberg (1991)