Regular Languages of Words Over Countable Linear Orderings[⋆]

Olivier Carton[1], Thomas Colcombet[2], and Gabriele Puppis[3]

[1] University Paris Diderot, LIAFA
olivier.carton@liafa.jussieu.fr
[2] CNRS/LIAFA
thomas.colcombet@liafa.jussieu.fr
[3] Department of Computer Science, Oxford University
gabriele.puppis@cs.ox.ac.uk

Abstract. We develop an algebraic model for recognizing languages of words indexed by countable linear orderings. This notion of recognizability is effectively equivalent to definability in monadic second-order (MSO) logic. The proofs also imply the first known collapse result for MSO logic over countable linear orderings.

1 Introduction

This paper continues a long line of research aiming at understanding the notions of regularity for languages of infinite objects, i.e., of infinite words and trees. This research results both in decision procedures for the *monadic second-order* (*MSO*) *logic* and in a fine comprehension of the mechanisms involved in different models of recognition. More specifically, the paper answers the following interrogations:

> *What is a good notion of regularity for languages of words indexed by countable linear orderings? Is it equivalent to definability in MSO? What are the correct tools for studying this notion?*

Several results, in particular in terms of decidability, partially answered the above questions (see related work below). Our study gives a deeper insight in the phenomena, e.g. answering (positively) the following previously open question:

> *Does there exist a collapse result for MSO logic over countable linear orders, as Büchi's result shows a collapse of MSO logic to its existential fragment for words indexed by ω?*

The central objects in this paper are *words* indexed by countable linear orderings, i.e., total orders over countable sets together with functions mapping elements to letters in some finite alphabet. Languages are just sets of countable words and MSO logic gives a formalism for describing such languages in terms of formulas involving quantification over elements and sets of elements (a formula naturally defines the language of all words that makes the formula true).

[⋆] Supported by the ANR project JADE: 'Jeux et Automates, Décidabilité et Extensions'

Related work. Büchi initiated the study of MSO using the tools of language theory. He established that every language of ω-words (i.e., the particular case of words indexed by the ordinal ω) definable in MSO is effectively recognized by a suitable form of automaton [4]. A major advance has been obtained by Rabin, who extended this result to infinite trees [8]. One consequence of Rabin's result is that MSO is decidable over the class of all countable linear orderings. Indeed, every linear ordering can be seen as a set of nodes of the infinite tree, with the order corresponding to the infix ordering on nodes. Another proof of the decidability of the MSO theory of countable orders has been given by Shelah using the composition method [12]. This is an automaton-free approach to logic based on syntactic operations on formulas and inspired from Feferman and Vaught [6]. The same paper of Shelah is also important for another result it contains: the undecidability of the MSO theory of the real line (the reals with order). However, for ω-words as for infinite trees, the theory is much richer than simply the decidability of MSO. In particular, MSO is known to be equivalent to several formalisms, such as automata and, in the ω-word case, regular expressions and some forms of algebras, which give a very deep insight in the structure of languages. The decidability proof for MSO does not provide such an understanding.

A branch of research has been pursued to raise the equivalence between logic, automata, and algebra to infinite words beyond ω-words. In [3], Büchi introduced ω_1-automata on transfinite words to prove the decidability of MSO logic for ordinals less than ω_1. Besides the usual transitions, ω_1-automata are equipped with limit transitions of the form $P \to q$, with P set of states, which are used in a Muller-like way to process words indexed over ordinals. Büchi proved that his automata have the same expressive power as MSO logic for ordinals less than ω_1. The key ingredient is the closure under complementation of ω_1-automata.

In [2], ω_1-automata have been extended to $\diamond$-automata by introducing limit transitions of the form $q \to P$ to process words over linear orderings. In [10], $\diamond$-automata are proven to be closed under complementation with respect to countable and scattered orderings. This last result implies that $\diamond$-automata have the same expressive power as MSO logic over countable and scattered orderings [1]. However, it was already noticed in [1] that $\diamond$-automata are strictly weaker than MSO logic over countable (possibly non-scattered) linear orderings: indeed, the closure under complementation fails as there is an automaton that accepts all words with non-scattered domains, whereas there is none for scattered words.

In this paper, we unify those branches of research. We provide an algebraic framework and a notion of recognizability which happens to be equivalent to the definability in MSO logic. Our approach both extends the decidability approach of Rabin and Shelah, and provides new results concerning the expressive power of MSO logic over countable linear orders. In preliminary versions of this work, we devised an equivalent automaton model. This notion is less natural and it is not presented in this short paper.

Structure of the paper. After the preliminaries in Section 2, we present $\circ$-algebras and the corresponding tools and results in Section 3. In Section 4 we translate MSO formulas to $\circ$-algebras and in Section 5 we establish the converse.

2 Preliminaries

Linear orderings. A *linear ordering* $\alpha = (X, <)$ is a non-empty set X equipped with a total order $<$. Two linear orderings have same *order type* if there is an order-preserving bijection between their domains. We denote by ω, ω^*, ζ, η the order types of $(\mathbb{N}, <)$, $(-\mathbb{N}, <)$, $(\mathbb{Z}, <)$, $(\mathbb{Q}, <)$, respectively. Unless strictly necessary, we do not distinguish between a linear ordering and its order type. A *sub-ordering* of α is a subset I of α equipped with the same ordering relation (we denote it by $\alpha|_I$). Given two subsets I, J of α, we write $I < J$ iff $x < y$ for all $x \in I$ and all $y \in J$. A subset I of α is said to be *convex* if for all $x, y \in I$ and all $z \in \alpha$, $x < z < y$ implies $z \in I$. A linear ordering α is *dense* if for all $x < y \in \alpha$, there is $z \in \alpha$ such that $x < z < y$. It is *scattered* if none of its sub-orderings is both dense and non-singleton.

The *sum* $\alpha_1 + \alpha_2$ of two linear orderings $\alpha_1 = (X_1, <_1)$ and $\alpha_2 = (X_2, <_2)$ (up to renaming, assume that X_1 and X_2 are disjoint) is the linear ordering $(X_1 \uplus X_2, <)$, where $<$ coincides with $<_1$ on X_1, with $<_2$ on X_2, and, furthermore, it satisfies $X_1 < X_2$. More generally, given a linear ordering $\alpha = (X, <)$ and, for each $i \in X$, a linear ordering $\beta_i = (Y_i, <_i)$ (assume that the sets Y_i are pairwise disjoint), we denote by $\sum_{i \in \alpha} \beta_i$ the linear ordering $(Y, <')$, where $Y = \bigcup_{i \in X} Y_i$ and, for every $i, j \in X$, every $x \in Y_i$, and every $y \in Y_j$, $x <' y$ iff either $i = j$ and $x <_i y$ hold or $i < j$ holds.

Additional material on linear orderings can be found in [11].

Condensations. A standard way to prove properties of linear orderings is to decompose them into basic components (e.g., finite sequences, ω-sequences, ω^*-sequences, and η-orderings). This can be done by exploiting the notion of condensation. Precisely, a *condensation* of a linear ordering α is an equivalence relation $\sim$ over it such that for all $x < y < z$ in α, $x \sim z$ implies $x \sim y \sim z$ (this is equivalent to enforcing the condition that every equivalence class of $\sim$ is a convex subset). The ordering relation of α induces a corresponding total order on the *quotient* $\alpha/\!\sim$, which is called the *condensed order*. Conversely, any partition C of α into convex subsets induces a condensation $\sim_C$ such that $x \sim y$ iff x and y belong to the same convex subset $I \in C$.

Words and languages. We use a generalized notion of word, which coincides with the notion of labeled linear ordering. Given a linear ordering α and a finite alphabet A, a *word over A with domain α* is a mapping of the form $w : \alpha \to A$. Hereafter, we shall always consider words up to isomorphism of the domain, unless specifically required. Moreover, we only consider words of countable domains. The set of all words (of countable domain) over an alphabet A is denoted by A°. Given a word w with domain α and a non-empty subset I of α, we denote by $w|_I$ the *sub-word* resulting from the restriction of the domain of w to I. Furthermore, if I is convex, then $w|_I$ is said to be a *factor* of w.

Certain words will play a crucial role in the sequel. In particular, a word $w : \alpha \to A$ is said to be a *perfect shuffle of* A if (i) the domain α is isomorphic to $\mathbb{Q}$ and (ii) for every symbol $a \in A$, the set $w^{-1}(a) = \{x \in \alpha \mid w(x) = a\}$

is dense in α. Recall that $\mathbb{Q}$ is the unique, up to isomorphism, countable non-singleton dense linear ordering with no end-points. Likewise, for every finite set A, there is a unique, up to isomorphism, perfect shuffle of A.

Given two words $u : \alpha \to A$ and $v : \beta \to A$, we denote by uv the word with domain $\alpha + \beta$ where each position $x \in \alpha$ (resp., $x \in \beta$) is labeled by $u(x)$ (resp., $v(x)$). The concatenation of words is easily generalized to the infinite concatenation $\prod_{i\in\alpha} w_i$, where α is a linear ordering and each w_i has domain β_i, the result being a word with domain $\sum_{i\in\alpha} \beta_i$. We also define the *shuffle* of a tuple of words $w_1, ..., w_k$ as the word $\{w_1, ..., w_k\}^\eta = \prod_{i\in\mathbb{Q}} w_{f(i)}$, where f is the unique perfect shuffle of $\{1, ..., k\}$ with domain $\mathbb{Q}$.

A *language* is a set of words over a certain alphabet. For some technical reasons, it is convenient to avoid the presence of the empty word ε in a language. Thus, unless otherwise specified, we use the term word to mean a labeled, countable, non-empty linear ordering. The operations of juxtaposition, ω-iteration, ω^*-iteration, shuffle, etc. are extended to languages in the obvious way.

3 Semigroups and algebras for countable linear orderings

This section is devoted to the presentation of algebraic objects suitable for defining a notion of recognizable $\circ$-languages. As it is already the case for ω-words, our definitions come in two flavors, $\circ$-semigroups (corresponding to ω-semigroups) and $\circ$-algebras (corresponding to Wilke-algebras). We prove the equivalence of the two notions when the underlying set is finite.

Countable products. The objective is to have a notion of products indexed by countable linear orderings, and possessing several desirable properties (in particular, generalized associativity and existence of finite presentations).

Definition 1. *A* (generalized) product *over a set S is a function π from S° to S such that, for every $a \in S$, $\pi(a) = a$ and, for every word u and every condensation $\sim$ of its domain,*

$$\pi(u) \;=\; \pi\Big(\prod\nolimits_{I\in\alpha/\sim} \pi(u|_I)\Big) \qquad \text{(generalized associativity)}$$

The pair $\langle S, \pi\rangle$ is called a $\circ$-semigroup*.*

As an example, the operation $\prod$ is a generalized product over A°. Hence, $\langle A^\circ, \prod\rangle$ is a $\circ$-semigroup (in particular, it is the *free* $\circ$-semigroup generated by A).

A *morphism* from a $\circ$-semigroup (S, π) to another $\circ$-semigroup (S', π') is a mapping $\varphi : S \to S'$ such that, for every word $w : \alpha \to S$, $\varphi(\pi(w)) = \pi'(\widetilde{\varphi}(w))$, where $\widetilde{\varphi}$ is the component-wise extension of φ to words. A $\circ$-language $L \subseteq A^\circ$ is called *recognizable by $\circ$-semigroups* if there exists a morphism φ from $\langle A^\circ, \prod\rangle$ to some finite semigroup $\langle S, \pi\rangle$ (here finite means that S is finite) such that $L = \varphi^{-1}(F)$ for some $F \subseteq S$ (equivalently, $\varphi^{-1}(\varphi(L)) = L$).

Recognizability by $\circ$-semigroup has the expressive power we aim at, however, the product π requires to be represented, a priori, by an infinite table. This is not usable as it stands for decision procedures. That is why, given a finite ∘-semigroup $\langle S, \pi\rangle$, we define the following (finitely presentable) algebraic operators:

- $\cdot : S^2 \rightarrow S$, mapping a pair of elements $a, b \in S$ to the element $\pi(ab)$,
- $\tau : S \rightarrow S$, mapping an element $a \in S$ to the element $\pi(a^\omega)$,
- $\tau^* : S \rightarrow S$, mapping an element $a \in S$ to the element $\pi(a^{\omega^*})$,
- $\kappa : \mathscr{P}(S)\backslash\{\emptyset\} \rightarrow S$, mapping a non-empty set $\{a_1, ..., a_k\}$ to $\pi\big(\{a_1, ..., a_k\}^\eta\big)$.

One says that $\cdot, \tau, \tau^*$ and κ are *induced by* π. From now on, we shall use the operator $\cdot$ with infix notation (e.g., $a \cdot b$) and the operators τ, τ^*, and κ with superscript notation (e.g., a^τ, $\{a_1, ..., a_k\}^\kappa$). The resulting algebraic structure $\langle S, \cdot, \tau, \tau^*, \kappa\rangle$ has the property of being a ∘-algebra, defined as follows:

Definition 2. *A structure $\langle S, \cdot, \tau, \tau^*, \kappa\rangle$, with $\cdot : S^2 \rightarrow S$, $\tau, \tau^* : S \rightarrow S$, and $\kappa : \mathscr{P}(S) \setminus \{\emptyset\} \rightarrow S$, is called a* ∘-algebra *if:*

(A1) $(S, \cdot)$ *is a semigroup, namely, for every $a, b, c \in S$, $a \cdot (b \cdot c) = (a \cdot b) \cdot c$,*

(A2) τ *is* compatible to the right*, namely, for every $a, b \in S$ and every $n > 0$, $(a \cdot b)^\tau = a \cdot (b \cdot a)^\tau$ and $(a^n)^\tau = a^\tau$,*

(A3) τ^* *is* compatible to the left*, namely, for every $a, b \in S$ and every $n > 0$, $(b \cdot a)^{\tau^*} = (a \cdot b)^{\tau^*} \cdot a$ and $(a^n)^{\tau^*} = a^{\tau^*}$,*

(A4) κ *is* compatible with shuffles*, namely, for every non-empty subset P of S, every element c in P, every subset Q of P, and every non-empty subset R of $\{P^\kappa,\ a \cdot P^\kappa,\ P^\kappa \cdot b,\ a \cdot P^\kappa \cdot b\ :\ a, b \in P\}$, we have $P^\kappa = P^\kappa \cdot P^\kappa = P^\kappa \cdot c \cdot P^\kappa = (P^\kappa)^\tau = (P^\kappa \cdot c)^\tau = (P^\kappa)^{\tau^*} = (c \cdot P^\kappa)^{\tau^*} = (Q \cup R)^\kappa$.*

The typical ∘-algebra is:

Lemma 1. *For every alphabet A, $\langle A^\circ, \cdot, \omega, \omega^*, \eta\rangle$ is a ∘-algebra.*

Proof. By a systematic analysis of Axioms A1-A4. □

Furthermore, as we mentioned above, every ∘-semigroup induces a ∘-algebra.

Lemma 2. *For every ∘-semigroup $\langle S, \pi\rangle$, $\langle S, \cdot, \tau, \tau^*, \kappa\rangle$ is a ∘-algebra, where the operators $\cdot, \tau, \tau^*$, and κ are those induced by π.*

Extension of ∘-algebras to countable products. Here, we aim at proving a converse to Lemma 2, namely, that every *finite* ∘-algebra $\langle S, \cdot, \tau, \tau^*, \kappa\rangle$ can be uniquely extended into a unique ∘-semigroup $\langle S, \pi\rangle$ (Theorem 1). We assume all words are over the alphabet S. The objective of the construction is to attach to each word u over S a 'value' in S. Furthermore, this value needs to be unique.

The central objects in this proof are *evaluation trees*, i.e., infinite trees describing how a word in S° can be evaluated into an element of S. We begin with condensation trees which are convenient representations for nested condensations. The nodes of a condensation tree are convex subsets of the linear ordering and the descendant relation is the inclusion. The set of children of each node defines a condensation. Furthermore, in order to provide an induction parameter, we require that the branches of a condensation tree are finite (but their length may not be uniformly bounded).

Definition 3. *A* condensation tree *over a linear ordering α is a set T of non-empty convex subsets of α such that:*

- *$\alpha \in T$,*
- *for all I, J in T, either $I \subseteq J$ or $J \subseteq I$ or $I \cap J = \emptyset$,*
- *for all $I \in T$, the union of all $J \in T$ such that $J \subsetneq I$ is either I or $\emptyset$,*
- *every subset of T totally ordered by inclusion is finite.*

Elements in T are called *nodes*. The node α is called the *root* of the tree. Nodes minimal for $\subseteq$ are called *leaves*. Given $I, J \in T$ such that $I \subsetneq J$ and there exist no $K \in T$ such that $I \subsetneq K \subsetneq J$, then I is called a *child* of J and J the *parent* of I. According to the definition, if I is an *internal node*, i.e., is not a leaf, then it has a set of children $\texttt{children}_T(I)$ which forms a partition of I. This partition consisting of convexes, it corresponds naturally to a condensation of $\alpha|_I$. When the tree T is clear from the context, we will denote by $\texttt{children}(I)$ the set of all children of I in T and, by extension, the corresponding condensation and the corresponding condensed linear ordering.

Since the branches of a condensation tree are finite, an ordinal rank, called *foundation rank*, can be associated with such a tree. This is the smallest ordinal β that enables a labeling of the nodes by ordinals less than or equal to β such that the label of each node is strictly greater than the labels of its children. This rank allows us to make proofs by induction (see also [7] for similar definitions).

We now introduce evaluation trees. Intuitively, these are condensation trees where each internal node has an associated value in S that can be 'easily computed' from the values of its children. Here we consider a word u 'easy to compute' if the following function π_0 is defined on u:

Definition 4. *Let π_0 be the partial function from S° to S such that:*

- *$\pi_0(ab) = a \cdot b$ for all $a, b \in S$,*
- *$\pi_0(s^\omega) = s^\tau$ for all $s \in S$,*
- *$\pi_0(s^{\omega^*}) = s^{\tau^*}$ for all $s \in S$,*
- *$\pi_0(P^\eta) = P^\kappa$ for all non-empty sets $P \subseteq S$,*
- *in any other case π_0 is undefined.*

An evaluation tree *over a linear ordering α is a pair $\mathcal{T} = \langle T, \gamma \rangle$ in which T is a condensation tree over α and γ is a function from T to S such that for every internal node $I \in T$, $\gamma(I) = \pi_0(\gamma(\texttt{children}(I)))$, where $\gamma(\texttt{children}(I))$ denotes the word with domain $\texttt{children}(I)$ labeling each position $J \in \texttt{children}(I)$ with $\gamma(J)$ (note that we assume that $\pi_0(\gamma(\texttt{children}(I))$ is defined). The* value *of $\langle T, \gamma \rangle$ is $\gamma(\alpha)$, i.e., the value of the root.*

An evaluation tree *$\mathcal{T} = \langle T, \gamma \rangle$ over a word u is an evaluation tree over the domain of u such that the leaves of T are singletons and $\gamma(\{x\}) = u(x)$ for all x in the domain of u.*

The next propositions are central in the study of evaluation trees.

Proposition 1. *For every word u, there exists an evaluation tree over u.*

The proof here resembles the construction used by Shelah in his proof of decidability of the monadic second-order theory of orders from [12]. In particular, it uses the theorem of Ramsey [9], as well as a lemma stating that every non-trivial word indexed by a dense linear ordering has a perfect shuffle as a factor. We remark that the above proposition does not use any of the Axioms A1-A4.

Proposition 2. *Two evaluation trees over the same word have the same value.*

The proof of this result is quite involved and it heavily relies on the use of Axioms A1-A4 (each axiom can be seen as an instance of Proposition 2 in some special cases of computation trees of height 2). The proof makes also use of Proposition 1. Note that, as opposed to Proposition 1, Proposition 2 has no counterpart in [12].

Using the above results, the proof of the following result is relatively easy:

Theorem 1. *For every finite $\circ$-algebra $\langle S, \cdot, \tau, \tau^*, \kappa \rangle$, there exists a unique product π that defines $\langle S, \cdot, \tau, \tau^*, \kappa \rangle$.*

Proof. Given a word w with domain α, one defines $\pi(w)$ to be the value of some evaluation tree over w (the evaluation tree exists by Proposition 1 and the value $\pi(w)$ is unique by Proposition 2).

We prove that π is associative. Let $\sim$ be a condensation of the domain α. For all $I \in \alpha/\sim$, let $\mathcal{T}_I$ be some evaluation tree over $w|_I$. Let also $\mathcal{T}'$ be some evaluation tree over the word $w' = \prod_{I \in \alpha/\sim} \pi(w|_I)$. One constructs an evaluation tree $\mathcal{T}$ over w by first lifting $\mathcal{T}'$ from the linear ordering $\alpha/\sim$ to α (this is done by replacing each node J in $\mathcal{T}'$ by $\bigcup J$) and then substituting each leaf of $\mathcal{T}'$ corresponding to some class $I \in \alpha/\sim$ with the subtree $\mathcal{T}_I$. The last step is possible (i.e., respects the definition of evaluation tree) since the value of each evaluation tree $\mathcal{T}_I$ is $\pi(w|_I)$, which coincides with the value $w'(I)$ at the leaf I of $\mathcal{T}'$. By Proposition 2, the resulting evaluation tree $\mathcal{T}$ has the same value as $\mathcal{T}'$ and this witnesses that $\pi(w) = \pi\left(\prod_{I \in \alpha/\sim} \pi(w|_I)\right)$.

What remains to be done is to prove that indeed the above choice of π defines $\cdot, \tau, \tau^*, \kappa$. This requires a case by case analysis. □

Let us conclude with a decidability result.

Theorem 2. *Emptiness of $\circ$-languages recognizable by $\circ$-algebras is decidable.*

Proof (principle of the algorithm). It is sufficient to describe an algorithm which given a set of elements $A \subseteq S$ computes the set $X = \{\pi(u) \ : \ u \in A^\circ\}$. For this, one just has to saturate A under the operations $\cdot, \tau, \tau^*, \kappa$ yielding the set $\langle A \rangle$. Indeed, it is easy to prove that $\langle A \rangle \subseteq X$, since this inclusion holds for A and is preserved under each operation of the saturation. The converse inclusion is established using Proposition 1. □

4 From monadic second-order logic to $\circ$-algebras

Let us recall that monadic second-order (MSO) logic is the extension of first-order logic with set quantifiers. We assume the reader to have some familiarity with this logic as well as with the Büchi approach for translating MSO formulas into automata. A good survey can be found in [13]. We refer to the *$\forall$-fragment* as the set of formulas that start with a block of universal set quantifiers, followed by a first-order formula. The *$\exists\forall$-fragment* consists of formulas starting with a block of existential set quantifiers followed by a formula of the $\forall$-fragment.

Here, we mimic Büchi's technique and show a relatively direct consequence of the above results, namely that MSO formulas can be translated to $\circ$-algebras:

Proposition 3. *The MSO definable languages are effectively $\circ$-recognizable.*

Let us remark that we could have equally well used the composition method of Shelah for establishing Proposition 3. Indeed, given an MSO definable language, a $\circ$-algebra recognizing it can be directly extracted from [12].

Our chosen proof for Proposition 3 follows Büchi's approach, namely, we establish sufficiently many closure properties of $\circ$-recognizable language. Then, each construction of the logic can be translated into an operation on languages. To disjunction corresponds union, to conjunction corresponds intersection, to negation corresponds complement, etc. We assume the reader to be familiar with this approach (in particular the coding of the valuations of free variables).

The closure under intersection, union, and complement are, as usual, easy to obtain. The languages corresponding to atomic predicates are also very easily shown to be $\circ$-recognizable. What remains to be proved is the closure under projection. Given a language of $\circ$-words L over some alphabet A, and a mapping h from A to another alphabet B, the *projection of L by h* is simply $h(L)$ (h being extended component-wise to $\circ$-words, and $\circ$-languages). It is classical that this projection operation is what is necessary for obtaining the closure under existential quantification at the logical level. Hence, we just need to prove:

Lemma 3. *The $\circ$-recognizable languages are effectively closed under projections.*

Proof (sketch). We first describe the construction for a given $\circ$-semigroup $\langle S, \pi\rangle$. The projection is obtained, as it is usual, by a powerset construction, i.e., we aim at providing a $\circ$-product over $\mathscr{P}(S)$. Given two words u and U over S and $\mathscr{P}(S)$ respectively, we write $u \in U$ when $\mathcal{D}om(u) = \mathcal{D}om(U)$ and $u(x) \in U(x)$ for all $x \in \mathcal{D}om(U)$. We define the mapping $\widetilde{\pi}$ from $(\mathscr{P}(S))^\circ$ to $\mathscr{P}(S)$ by

$$\widetilde{\pi}(U) \ =^{\text{def}} \ \{\pi(u) \ : \ u \in U\} \qquad \text{for all } U \in (\mathscr{P}(S))^\circ.$$

Let us show that $\widetilde{\pi}$ is associative. Consider a word U over $\mathscr{P}(S)$ and a condensation $\sim$ of its domain. Then,

$$\begin{aligned}\widetilde{\pi}(U) \ &= \ \Big\{\pi(u) \ : \ u \in U\Big\} \ = \ \Big\{\pi\Big(\prod\nolimits_{I\in\alpha/\sim} \pi(u|_I)\Big) \ : \ u \in U\Big\} \\ &= \ \Big\{\pi\Big(\prod\nolimits_{I\in\alpha/\sim} a_I\Big) \ : \ a_I \in \widetilde{\pi}(U|_I) \text{ for all } I \in \alpha/{\sim}\Big\} \ = \ \widetilde{\pi}\Big(\prod\nolimits_{I\in\alpha/\sim} \widetilde{\pi}(U|_I)\Big)\,,\end{aligned}$$

where the second equality is derived from the associativity of π. Hence $(\mathscr{P}(S), \widetilde{\pi})$ is a $\circ$-semigroup. It is just a matter of writing to show that $\langle \mathscr{P}(S), \widetilde{\pi} \rangle$ recognizes any projection of a language recognized by $\langle S, \pi \rangle$.

Thanks to Lemma 2 and Theorem 1, the above construction can be performed at the level of the $\circ$-algebra $\langle S, \cdot, \tau, \tau^*, \kappa \rangle$, namely, there exists a $\circ$-algebra $\langle \mathscr{P}(S), \widetilde{\cdot}, \widetilde{\tau}, \widetilde{\tau}^*, \widetilde{\kappa} \rangle$ corresponding to $\langle \mathscr{P}(S), \widetilde{\pi} \rangle$. The problem is that this may, a priori, be non-effective. However, using a more careful analysis, it is possible to show the effectiveness of the construction.

Let us give some intuition on how one can compute $P^{\widetilde{\kappa}} = \widetilde{\pi}(P^\eta)$ given a set $P = \{A_1, ..., A_k\}$, with $A_1, ..., A_k \subseteq S$ and $k \geq 1$. This is the most difficult operator. Since $P^{\widetilde{\kappa}} = \{\pi(u) \ : \ u \in U, \ u \in P^\eta\}$, this is very similar to computing $\{\pi(u) \ : \ u \in A^\circ\}$ as done in the proof of Theorem 2. One just needs to restrict the set of considered words u to be the ones such that $u \in U$ for some $U \in P^\eta$. This can be achieved by performing a product of S with a $\circ$-algebra which recognizes the single-word language $\{P^\eta\}$, and then applying the saturation process of Theorem 2 on the resulting $\circ$-algebra. □

5 From ∘-algebras to monadic second-order logic

We have seen in the previous section that every MSO formula defines a $\circ$-recognizable language. In this section, we sketch the proof of the converse.

Theorem 3. *The $\circ$-recognizable languages are effectively MSO definable. Furthermore, such languages are definable in the $\exists\forall$-fragment of MSO logic.*

We fix for the remaining of the section a morphism h from $\langle A^\circ, \prod \rangle$ to a $\circ$-semigroup $\langle S, \pi \rangle$, with S finite. Let F be some subset of S. Let also $\cdot, \tau, \tau^*, \kappa$ be defined from π. Our goal is to show that $L = h^{-1}(F)$ is MSO definable. It is sufficient for this to show that for every $s \in S$, the language

$$\pi^{-1}(s) \ = \ \{w \in S^\circ \ : \ \pi(w) = s\} \ ,$$

is defined by some MSO formula φ_s. This establishes that $L = \bigcup_{s \in F} h^{-1}(s)$ is defined by the disjunction $\bigvee_{s \in F} \varphi'_s$, where φ'_s is obtained from φ_s by replacing every occurrence of an atom $t(x)$, with $t \in S$, by $\bigvee_{a \in h^{-1}(t) \cap A} a(x)$.

A good approach for defining $\pi^{-1}(s)$ is to use a formula which, given $w \in S^\circ$, guesses some object ‘witnessing’ $\pi(w) = s$. The only objects that we have seen so far and that are able to ‘witness’ $\pi(w) = s$ are evaluation trees. Unfortunately, there is no way an MSO formula can guess an evaluation tree, since their height cannot be uniformly bounded. That is why we use another kind of object for witnessing $\pi(w) = a$: the so-called Ramsey split, introduced just below.

Ramsey splits. Ramsey splits are not directly applied to words, but to additive labellings. An *additive labeling* σ from a linear ordering α to a semigroup $\langle S, \cdot \rangle$ (in particular, this will be a $\circ$-semigroup in our case) is a function that maps any pair of elements $x < y$ from α to an element $\sigma(x, y) \in S$ in such a way that $\sigma(x, y) \cdot \sigma(y, z) = \sigma(x, z)$ for all $x < y < z$ in α.

Given two positions $x < y$ in a word w, denote by $[x, y)$ the interval $\{z : x \leq z < y\}$. Given a word w and two positions $x < y$ in it, we define $\sigma_w(x, y) \in S$ to be $\pi(w|_{[x,y)})$. We just mention σ whenever w is clear from the context. Quite naturally, σ_w is additive since for all $x < y < z$, we have $\sigma(x, y) \cdot \sigma(y, z) = \pi\big(w|_{[x,y)}\big) \cdot \pi\big(w|_{[y,z)}\big) = \pi\big(w|_{[x,y)} w|_{[y,z)}\big) = \pi\big(w|_{[x,z)}\big) = \sigma(x, z)$.

Definition 5. *A* split *of height n of a linear ordering α is a function $g : \alpha \to [1, n]$. Two elements $x, y \in \alpha$ are called (k-)*neighbors *iff $g(x) = g(y) = k$ and $g(z) \leq k$ for all $z \in [x, y] \cup [y, x]$ (note that neighborhood is an equivalence). The split g is called* Ramsey for some additive labeling *σ iff for all equivalence classes $X \subseteq \alpha$ for the neighborhood relation, there is an idempotent $e \in S$ such that $\sigma(x, y) = e$ for all $x < y$ in X.*

Theorem 4 (Colcombet [5]). *For every finite semigroup $\langle S, \cdot\rangle$, every linear ordering α, and every additive labeling σ from α to $\langle S, \cdot\rangle$, there is a split of α which is Ramsey for σ and which has height at most $2|S|$.*

From ○-recognizable to MSO definable. The principle is to construct a formula which, given a word w, guesses a split of height at most $2|S|$, and uses it for representing the function which to every convex set I associates $\pi(w|_I)$. For the explanations, we assume that some word w is fixed, that its domain is α, and that σ is the additive labeling over α derived from w. We remark, however, that all constructions are uniform and do not depend on w.

We aim at constructing a formula $\texttt{evaluate}_s$, for each $s \in S$, which holds over a word w iff $\pi(w) = s$. The starting point is to guess:

- a split g of α of height at most $2|S|$, and;
- a function e mapping each position $x \in \alpha$ to an idempotent of S.

The intention is that a choice of g, e by the formula is good when the split g is Ramsey for σ and the function e maps each x to the idempotent $e(x)$ that arises when the neighborhood class of x is considered in the definition of Ramseyness. In such a case, we say that (g, e) is *Ramsey*. Observe that neither g nor e can be represented by a single monadic variable. However, since both g and e are functions from α to sets of bounded size ($2|S|$ for g, and $|S|$ for e), one can guess them using a fixed number of monadic variables. This kind of coding is standard, and from now on we shall use explicitly the mappings g and e in MSO formulas.

Knowing a Ramsey pair (g, e) is an advance toward computing the value of a word. Indeed, Ramsey splits can be used as 'accelerating structures' in the sense that every computation of some $\pi(w|_I)$ for a convex subset I becomes significantly easier when a Ramsey split is known, namely, first-order definable. This is formalized by the following lemma.

Lemma 4. *For all $s \in S$, there is a first-order formula $\texttt{value}_s(g, e, X)$, such that for every convex subset I:*

- *if (g, e) is Ramsey, then $\texttt{value}_s(g, e, I)$ holds iff $\pi(w|_I) = s$,*
- *if both $\texttt{value}_s(g, e, I)$ and $\texttt{value}_t(g, e, I)$ hold, then $s = t$.*

One sees those formulas as defining a partial function `value` mapping g, e, I to some element $s \in S$ (the second item enforces that there is no ambiguity about the value, namely, that this is a function and not a relation). From now we simply use the notation $\texttt{value}(g, e, I)$ as if it were a function.

One needs now to enforce that $\texttt{value}(g, e, I)$ coincides with $\pi(w|_I)$, even without assuming that (g, e) is Ramsey. For this, one uses condensations. A priori, a condensation is not representable by monadic variables, since it is a binary relation. However, any set $X \subseteq \alpha$ naturally defines the relation $\approx_X$ such that $x \approx_X y$ iff either $[x, y] \subseteq X$, or $[x, y] \cap X = \emptyset$. It is easy to check that this relation is a condensation. A form of converse result also holds:

Lemma 5. *For all condensations $\sim$, there is X such that $\sim$ and $\approx_X$ coincide.*

Lemma 5 tells us that it is possible to work with condensations as if they were monadic variables. In particular, we use condensation variables in the sequel, which in fact are implemented by the set obtained from Lemma 5.

Given a convex subset I of α and some condensation $\sim$ of $\alpha|_I$, we denote by $w[I, \sim]$ the word with domain $\beta = (\alpha|_I)/\sim$ in which every $\sim$-equivalence class J is labeled by $\texttt{value}(g, e, J)$. One defines the formula $\texttt{consistency}(g, e)$ which checks that for all convex subsets I and all condensations $\sim$ of $\alpha|_I$ (thanks to Lemma 5), the following conditions hold:

(C1) if I is a singleton $\{x\}$, then $\texttt{value}(g, e, I) = w(x)$,

(C2) if $w[I, \sim] = st$ for some $s, t \in S$, then $\texttt{value}(g, e, I) = s \cdot t$,

(C3) if $w[I, \sim] = s^\omega$ for some $s \in S$, then $\texttt{value}(g, e, I) = s^\tau$,

(C4) if $w[I, \sim] = s^{\omega^*}$ for some $s \in S$, then $\texttt{value}(g, e, I) = s^{\tau^*}$,

(C5) if $w[I, \sim] = P^\eta$ for some $P \subseteq S$, then $\texttt{value}(g, e, I) = P^\kappa$.

For some fixed I and $\sim$, the above tests require access to the elements $w[I, \sim](J)$, where J is a $\sim$-equivalence class of $\alpha|_I$. Since $\sim$-equivalence of two positions $x, y \in \alpha|_I$ is first-order definable, we know that for every position $x \in \alpha|_I$, the element $\texttt{value}(g, e, [x]_\sim)$ is first-order definable from x. This shows that the above properties can be expressed by first-order formulas and hence $\texttt{consistency}(g, e)$ is in the $\forall$-fragment.

The last key argument is to propagate the 'local consistency' constraints C1–C5 to a 'global consistency' property. This is done by the following lemma.

Lemma 6. *If $\texttt{consistency}(g, e)$ holds, then $\texttt{value}(g, e, I) = \pi(w|_I)$ for all convex subsets I of α.*

This lemma implies Theorem 3. We claim indeed that, given $s \in S$, the language $\pi^{-1}(s)$ is defined by the following formula in the $\exists\forall$-fragment of MSO:

$$\texttt{evaluate}_s \ \stackrel{\text{def}}{=} \ \exists g. \exists e.\ \texttt{consistency}(g, e) \wedge \texttt{value}(g, e, \alpha) = s\ .$$

Let $\pi(w) = s$. One can find a Ramsey pair (g, e) using Theorem 4. Lemma 4 then implies $\pi(w|_I) = \texttt{value}(g, e, I)$ for all convex subsets I. Since π is a product, the constraints C1–C5 are satisfied and $\texttt{consistency}(g, e)$ holds. This proves that $\texttt{evaluate}_s$ holds. Conversely, if $\texttt{evaluate}_s$ holds, then $\texttt{consistency}(g, e)$ holds for some (g, e). Lemma 6 then implies $\pi(w) = \pi(w|_\alpha) = \texttt{value}(g, e, \alpha) = s$. □

6 Conclusion

We have introduced an algebraic notion of recognizability for languages of countable words and we have shown the correspondence with the family of languages definable in MSO logic. As a byproduct of this result, it follows that MSO logic interpreted over countable words collapses to its ∃∀-fragment (hence, since it is closed under complementation, it also collapses to its ∀∃-fragment). This collapse result is optimal, in the sense that there exist definable languages that are not definable in the ∃-fragment, nor in the ∀-fragment. An example of such a language is the set of all scattered words over $\{a\}$ and all non-scattered words over $\{b\}$: checking that a word is scattered requires a universal quantification over the sub-orderings of its domain and, conversely, checking that a word is not scattered requires an existential quantification.

Acknowledgments We are grateful to Achim Blumensath and the anonymous referees for their numerous comments on this work.

References

[1] N. Bedon, A. Bès, O. Carton, and C. Rispal. Logic and rational languages of words indexed by linear orderings. *Theoretical Computer Science*, 46(4):737–760, 2010.

[2] V. Bruyère and O. Carton. Automata on linear orderings. *Journal of Computer and System Sciences*, 73(1):1–24, 2007.

[3] J.R. Büchi. Transfinite automata recursions and weak second order theory of ordinals. In *Proc. Int. Congress Logic, Methodology, and Philosophy of Science, Jerusalem 1964*, pages 2–23. North Holland, 1964.

[4] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proceedings of the International Congress for Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

[5] T. Colcombet. Factorisation forests for infinite words and applications to countable scattered linear orderings. *Theoretical Computer Science*, 411:751–764, 2010.

[6] S. Feferman and R. Vaught. The first-order properties of products of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.

[7] A.S. Kechris. *Classical Descriptive Set Theory.* Graduate Texts in Mathematics. Springer, 1995.

[8] M.O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[9] F.P. Ramsey. On a problem of formal logic. In *Proceedings of the London Mathematical Society*, volume 30, pages 264–286, 1929.

[10] C. Rispal and O. Carton. Complementation of rational sets on countable scattered linear orderings. *International Journal of Foundations of Computer Science*, 16(4):767–786, 2005.

[11] J.G. Rosenstein. *Linear Orderings.* Academic Press, 1982.

[12] S. Shelah. The monadic theory of order. *Annals of Mathematics*, 102:379–419, 1975.

[13] W. Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.