A context-free linear ordering with an undecidable first-order theory*

Arnaud Carayol[1] and Zoltán Ésik[2]

[1] Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est, France
[2] Institute of Informatics, University of Szeged, Hungary

Abstract. The words of a context-free language, ordered by the lexicographic ordering, form a context-free linear ordering. It is well-known that the linear orderings associated with deterministic context-free languages have a decidable monadic second-order theory. In stark contrast, we give an example of a context-free language whose lexicographic ordering has an undecidable first-order theory.

1 Introduction

When the alphabet of a language L is linearly ordered, we may equip L with the lexicographic ordering. It is known that every countable linear ordering is isomorphic to the lexicographic ordering of a (prefix) language.

The lexicographic orderings of regular languages (i.e., the regular linear orderings) were studied in [1–4, 12, 15, 20, 24, 28]. These linear orderings agree with the leaf orderings of the regular trees, and are all automatic linear orderings as defined in [22]. It follows from results in [20] that all scattered regular linear orderings have finite Hausdorff rank, or finite condensation rank (FC-rank), as defined in [27]. In fact all automatic linear orderings have finite FC-rank [22]. Moreover, an ordinal is the order type of a regular well-ordering if and only if it is strictly less than ω^ω.

The study of the lexicographic orderings of context-free languages (context-free linear orderings) was initiated in [4] and further developed in [5, 6, 8, 16–18] and was extended to languages generated by *deterministic* higher order grammars in [7].

It follows from early results in [13] that the lexicographic orderings of deterministic context-free languages are (up to isomorphism) identical to the leaf orderings of the algebraic trees, cf. [5]. In [4], it was shown that every ordinal less

* Arnaud Carayol has been supported by the project AMIS (ANR 2010 JCJC 0203 01 AMIS). Both authors received partial support from the project TÁMOP-4.2.1/B-09/1/KONV-2010-0005 "Creating the Center of Excellence at the University of Szeged", supported by the European Union and co-financed by the European Regional Fund. Zoltán Ésik was also partly supported by the National Foundation of Hungary for Scientific Research, grant no. K 75249, and by a chair Labex Bézout as part of the program "Investissements d'Avenir" (ANR-10-LABX-58).

than ω^{ω^ω} is the order type of a deterministic context-free linear ordering and it was conjectured that a well-ordering is isomorphic to a context-free linear ordering if and only if its order type is less than ω^{ω^ω}. This conjecture was confirmed in [5] for deterministic context-free linear orderings, and in [18] for context-free linear orderings. Moreover, it was shown in [6] and [18] that the FC-rank of every scattered deterministic context-free linear ordering and in fact every scattered context-free linear ordering is less than ω^ω. Since the FC-rank of a well-ordering is less than ω^ω exactly when its order type is less than ω^{ω^ω}, it follows in conjunction with results proved in [4] that a well-ordering is isomorphic to the lexicographic ordering of a context-free language or deterministic context-free language if and only if its order type is less than ω^{ω^ω}. Exactly the same ordinals are the order types of the tree automatic well-orderings, see [14]. Eventually, it was proved in [8] that the FC-rank of every context-free linear ordering is less than ω^ω. However, the question whether there exists a context-free linear ordering that is not a deterministic context-free linear ordering remained open.

Since deterministic context-free linear orderings belong to the pushdown hierarchy [9–11, 25], they all have decidable monadic second-order theories. In fact, there exists an algorithm that takes two inputs, an $LR(1)$ grammar (or equivalently a deterministic pushdown automaton) and a sentence of the monadic second-order logic of linear orders and tells whether the sentence holds in the lexicographic ordering of the language generated by the grammar. Such a decision procedure does not exist for all context-free grammars and monadic second-order or even first-order sentences, since as shown in [16], it is undecidable to tell whether a context-free linear ordering (given by a context-free grammar) is dense. In contrast, it is decidable whether a context-free linear ordering is a well-ordering or a scattered ordering.

In this paper we prove that there is a context-free linear ordering whose first-order theory is undecidable. Thus there exists a context-free linear ordering which is not the lexicographic ordering of a deterministic context-free language. The context-free language defining this linear ordering is a finite disjoint union of deterministic context-free languages. Hence our undecidability result holds for the class of unambiguous context-free linear orderings.

As a corollary, we also obtain the existence of a (unambiguous) context-free language whose associated tree has an undecidable monadic second-order theory. The tree of a language is composed of the set of all prefixes of the words of the language as set of vertices, and its ancestor relation is simply the prefix relation. This result in turn proves the existence a context-free language that cannot be accepted by any deterministic collapsible pushdown automaton (an extension of the classical notion of pushdown automaton with nested stacks and links [19]), as shown previously by Paweł Parys using a pumping argument [26].

The paper is organised as follows. In Section 2, we recall basic definitions on linear orderings. Definitions concerning first-order logic and the structures associated with languages are given in Section 3. Section 1 presents our main result and its corollaries are given in Section 5. Section 6 concludes the paper.

2 Linear orderings

A piece of notation: for a nonnegative integer n, we will denote the set $\{1, \ldots, n\}$ by $[n]$.

When A is an alphabet, we let A^* denote the set of all finite words over A, including the empty word ϵ. The set A^+ is $A^* - \{\epsilon\}$. We let u^R denote the mirror image of a word $u \in A^*$.

A *linear ordering* [27] $(I, <)$ is a set I equipped with a strict linear order relation $<$. As usual, we will write $x \leq y$ for $x, y \in I$ if $x < y$ or $x = y$. A linear ordering $(I, <)$ is finite or countable if I is. A *morphism* of linear orderings is an order preserving map. Note that every morphism is necessarily injective. When $(I, <)$ and $(J, <')$ are linear orderings such that $I \subseteq J$ and the embedding $I \hookrightarrow J$ is a morphism, we call $(I, <)$ a *subordering* of $(J, <')$. In this case the relation $<$ is the restriction of the relation $<'$ onto I and we usually write just I for $(I, <)$.

An *isomorphism* is a bijective morphism. Isomorphic linear orderings are said to have the same *order type*. The order types of the positive integers $\mathbb{N}$, negative integers $\mathbb{N}_-$, all integers $\mathbb{Z}$, and the rationals $\mathbb{Q}$, ordered as usual, are denoted ω, ω^*, ζ and η, respectively. As usual, the finite order types may be identified with the nonnegative integers.

Recall that a linear ordering $(I, <)$ is *dense* if it has at least two elements and for every $x, y \in I$ with $x < y$ there is some $z \in I$ with $x < z < y$. A *quasi-dense* linear ordering is a linear ordering that has a dense subordering, and a *scattered* linear ordering is a linear ordering that is not quasi-dense. For example, $\mathbb{N}$ and $\mathbb{Z}$ are scattered, $\mathbb{Q}$ is dense, and the ordering obtained by replacing each or some point in $\mathbb{Q}$ with a 2-element linear ordering is quasi-dense but not dense. Clearly, every subordering of a scattered linear ordering is scattered. It is well-known that a linear ordering is quasi-dense if and only if it has a subordering of order type η. Moreover, up to isomorphism, there are 4 countable dense linear orderings, the ordering $\mathbb{Q}$ of the rationals possibly equipped with a least or greatest element, or both.

When $(I, <)$ is a linear ordering and for each $i \in I$, $(J_i, <_i)$ is a linear ordering, the *ordered sum*

$$\sum_{i \in I} (J_i, <_i)$$

is the disjoint union $\bigcup_{i \in I}(J_i \times \{i\})$ equipped with the order relation $(x, i) < (y, j)$ if and only if either $i < j$, or $i = j$ and $x <_i y$. When each $(J_i, <_i)$ is the linear ordering $(J, <')$, we call the ordered sum the *product* of $(I, <)$ and $(J, <')$, denoted $(I, <) \times (J, <')$. Finite ordered sums are also denoted as $(I_1, <_1) + \cdots + (I_n, <_n)$. Since the operation of ordered sum preserves isomorphism, we may also define ordered sums of order types. For example, $1 + \eta + 1$ is the order type of the rationals equipped with both a least and a greatest element. It is known that every scattered sum of scattered linear orderings is scattered. This means that if $(I, <)$ is scattered as is each $(J_i, <_i)$, then $\sum_{i \in I}(J_i, <_i)$ is also scattered. A sum over a dense linear ordering $(I, <)$ is referred to as a *dense sum*.

3 First-order logic

A *signature* is a ranked set σ of symbols. We let $|R|$ denote the arity (≥ 1) of the symbol R. A *relational structure* S over σ is given by a tuple $(D, (R^S)_{R\in\sigma})$, where D is the domain of S, and where for all $R \in \sigma$, the interpretation of R in S denoted R^S is a subset of $D^{|R|}$. When S is clear from the context, we just write R for its interpretation R^S.

Let $S = (D, (R^S)_{R\in\sigma})$ and $S' = (D', (R^{S'})_{R\in\sigma})$ be two structures over σ. An *isomorphism* h from D to D' is a bijection from D to D' such that for all $R \in \sigma$ and for all $u_1, \ldots, u_{|R|} \in D$, $(u_1, \ldots, u_{|R|}) \in R^S$ if and only if $(h(u_1), \ldots, h(u_{|R|})) \in R^{S'}$. We let $S \cong S'$ denote the existence of an isomorphism between S and S'.

A linear ordering $(I, <_I)$ is naturally represented as a structure over the signature σ_{ord} with one symbol $<$ of arity 2. Its domain is the set I and the symbol $<$ is interpreted as $<_I$.

First-order formulas use first-order variables, which are interpreted by elements of the structure and are denoted by lower case letters $x, y \ldots$. Atomic first-order formulas are of the form $R(x_1, \ldots, x_{|R|})$, where R is a relation symbol from the signature and $x_1, \ldots, x_{|R|}$ are first-order variables, or $x = y$ for first-order variables x, y with the obvious semantics. Complex formulas are built as usual from atomic ones by the use of Boolean connectives and quantifiers. Free and bound occurrences of variables in a formula are defined as usual. We write $\varphi(x_1, \ldots, x_n)$ to denote that the formula φ has free variables in $\{x_1, \ldots, x_n\}$. A *closed formula* has no free variables.

For a formula $\varphi(x_1, \ldots, x_n)$ and elements of the domain $u_1, \ldots, u_n$, we write $S \models \varphi[u_1, \ldots, u_n]$ to denote that the structure S *satisfies* the formula φ when the free variable x_i, $i \in [n]$, is interpreted as u_i. For a closed formula φ, we simply write $S \models \varphi$.

For example, the following formula over the signature σ_{ord} expresses that the structure is a linear ordering:

$$\begin{aligned} &\forall x\, \forall y \;\; x < y \to \neg(y < x) \\ \wedge\, &\forall x\, \forall y \;\; x < y \vee y < x \vee x = y \\ \wedge\, &\forall x\, \forall y\, \forall z \;\; (x < y \wedge y < z) \to x < z \end{aligned}$$

3.1 First-order interpretations

First-order interpretation is a transformation defining a structure in another structure using first-order logic.

Definition 1. *A first-order interpretation from a signature σ to a signature σ' is given by a tuple $(\delta, (\varphi_R)_{R\in\sigma'})$, where δ is a formula over σ with one free variable x_1, and for each symbol $R \in \sigma'$, φ_R is a formula over σ with free variables $x_1, \ldots, x_{|R|}$.*

Applying a first-order interpretation $\mathcal{I}$ to a structure S over the signature σ gives rise to a structure over the signature σ', denoted $\mathcal{I}(S)$. Its domain is the

set $D' = \{u \in D \mid S \models \delta[u]\}$. A symbol $R \in \sigma'$ is interpreted in $\mathcal{I}(S)$ as the set of all tuples satisfying φ_R :

$$\{(u_1, \ldots, u_{|R|}) \in (D')^{|R|} \mid S \models \varphi_R[u_1, \ldots, u_{|R|}]\}.$$

An example of first-order interpretation is given in Section 3.2.

3.2 Structures associated with words and languages

Let A be a finite alphabet. A word w over A can be represented by a structure over the signature $\sigma_A = \{P_a \mid a \in A\} \cup \sigma_{\text{ord}}$ with $|P_a| = 1$ for all $a \in A$. This structure, denoted S_w, has the set $[|w|]$ of positions in the word as its domain. The symbol $<$ is interpreted (in S_w) as the natural order and for all $a \in A$, P_a is interpreted as the predicate marking all occurrences of the letter a.

For $A = \mathbf{2} = \{0, 1\}$, the formula φ over the signature σ_A given below expresses that a word starts with the letter 0 (i.e., for all $w \in A^*$, $S_w \models \varphi$ if and only if w stars with 0) :

$$\exists x \ \ (\forall y \ \neg(y < x) \wedge P_0(x)) \ .$$

Similarly, when $L \subseteq A^*$, we define the structure S_L over the signature σ_A associated with a language L. This structure is obtained by taking the disjoint union of all the structures S_w for $w \in L$. Note that as soon as L contains two nonempty words, the relation $<$ is no longer a linear ordering.

The following formula is satisfied by the languages in which all nonempty words start with the letter 0.

$$\forall x \ \ (\forall y \ \neg(y < x) \rightarrow P_0(x))$$

Fig. 1. The structure S_L associated with the language $L = 1^*0$.

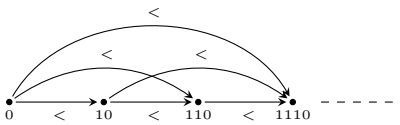
3.3 Lexicographic ordering and countable words associated with a language

We will consider countable linear orderings that arise as lexicographic orderings of languages. Suppose that A is an alphabet which is linearly ordered by the relation $<$. Then we define a strict partial order $<_s$ on A^* by $u <_s v$ if and only if $u = xay$ and $v = xbz$ for some $x, y, z \in A^*$ and $a, b \in A$ with $a < b$. We also define $u <_p v$ if and only if u is a *proper* prefix of v, and $u <_\ell v$ if and

only if $u <_s v$ or $u <_p v$. The *lexicographic order* relation $<_\ell$ turns A^* into a linear ordering. In particular, any language $L \subseteq A^*$ gives rise to a structure over the signature σ_{ord} denoted O_L and called the *lexicographic ordering of* L. The domain of O_L is the language L and the symbol $<$ is interpreted as the lexicographic ordering $<_\ell$.

We say that a language $L \subseteq A^*$ is scattered, dense, etc. if its lexicographic ordering has the corresponding property. Moreover, we say that a lexicographic ordering is a *regular* or a *context-free linear ordering* if it is isomorphic to the lexicographic ordering of a regular or context-free language. *Deterministic context-free linear orderings* are defined in the same way.

Example 1. Consider the alphabet $\mathbf{2} = \{0, 1\}$ ordered by $0 < 1$. The lexicographic ordering of the regular language 1^*0 is of order type ω and is depicted below.



Similarly the lexicographic orderings of the regular languages 0^*1, $0^+1 + 1^+0$ are of order type ω^* and ζ, respectively. The lexicographic ordering of $(00+11)^*01$ is η. The context-free linear ordering $(\bigcup_{n\geq 0} 1^n 0(1^*0)^n, <_\ell)$ is of order type $1+\omega+\omega^2+\cdots = \omega^\omega$. The context-free linear orderings $(\bigcup_{n\geq 1} 1^n 0(0(0^+1+1^+0)+10^{<n}), <_\ell)$ and $(\bigcup_{n\geq 1} 1^n 0(0(00+11)^*01+1(1^*0)^n), <_\ell)$ have respective order types $\zeta + 1 + \zeta + 2 + \cdots$ and $\eta + \omega + \eta + \omega^2 + \cdots$.

A *countable word* (called *arrangement* in [12]) over an alphabet B is a countable linear ordering whose elements are labelled by letters of B. Each language over an ordered alphabet A not containing the empty word gives rise to a *countable word* W_L over A, which is represented by a structure over the signature σ_A. Its domain is the language L. The symbol $<$ is interpreted as the lexicographic order $<_\ell$, and for all $a \in A$, P_a is interpreted as the set of words of L ending with the letter a. We say that a countable word is *context-free* if it is isomorphic to the countable word of some context-free language.

Lemma 1. *Every context-free linear ordering (resp. word) can be represented by a prefix context-free language not containing the empty word.*

Proof. We establish the result for context-free words. Let $A = \{a_1, \ldots, a_n\}$ with $a_1 < \cdots < a_n$, and let $L \subseteq A^+$ be a context-free language which does not contain the empty word.

Let $A' = \{a'_1, \ldots, a'_n\}$ be an alphabet disjoint from A and let $\pi : A^* \mapsto (A')^*$ be the morphism mapping a_i to a'_i for all $i \in [n]$.

Consider the context-free language L' over $A \cup A'$ ordered by $a_1 < \cdots < a_n < a'_1 < \cdots < a'_n$ defined by

$$L' = \{\pi(wa)a \mid wa \in L \text{ and } a \in A\}.$$

The language L' is prefix (as L' is included in $(A')^+A$ and A and A' are assumed to be disjoint). To conclude the proof, we observe that the mapping $\theta : L \mapsto L'$ mapping $wa \in L$ to $\pi(wa)a \in L'$ is an isomorphism from W_L to $W_{L'}$. □

Context-free words are clearly closed under substitution. Thus we have:

Lemma 2. *Let L be context-free language over an ordered alphabet A which does not contain the empty word, and suppose that for each $a \in A$, P_a is a context-free linear ordering. Then the ordered sum*

$$\sum_{u \in L} P_{\lambda(u)} \qquad \text{where } \lambda(u) \text{ designates the last letter of } u,$$

obtained by replacing each $u \in L$ ending with $a \in A$ by a copy of P_a, is a context-free linear ordering.

Proof. By Lemma 1, we can assume w.l.o.g. that L is prefix. For all $a \in A$, let L_a be a context-free language (which does not contain the empty word) defining the context-free linear ordering P_a. The ordered sum $\sum_{u \in L} P_{u(|u|)}$ is isomorphic to the context-free linear ordering defined by

$$\{waL_a \mid wa \in L \text{ and } a \in A\}.$$

□

This property in turn implies that context-free words can be defined in context-free linear orderings.

Lemma 3. *Let A be an ordered alphabet and let L be a context-free language not containing the empty word. There exists a context-free language L' and a first-order interpretation $\mathcal{I}$ such that W_L is isomorphic to $\mathcal{I}(O_{L'})$.*

Proof. Let L be a context-free language not containing the empty word over the alphabet $A = \{a_1, \ldots, a_n\}$. Consider the linear ordering O obtained by replacing in W_L each vertex labelled a_i by a copy of a linear ordering of order type $\zeta + i + \zeta$. As for all $i \geq 0$, $\zeta + i + \zeta$ is a context-free linear ordering, we obtain by Lemma 2 that O is a context-free linear ordering. Let L' be a context-free language such that $O_{L'}$ is isomorphic to O.

We now define a first-order interpretation transforming $O_{L'}$ into W_L. The first-order interpretation $\mathcal{I}$ only keeps vertices that have no predecessor. These vertices correspond to the first vertex in between two consecutive copies of ζ. Therefore these vertices are in a one to one correspondence with the elements of L. The order relation $<$ is inherited by $\mathcal{I}$. The predicate P_{a_1} is defined for those vertices with no successors, hence guaranteeing that the vertex (which must have no predecessor) lies in a copy of $\zeta + 1 + \zeta$. Similarly P_{a_2} is defined for vertices having a successor with no successor, etc. Formally the interpretation is defined by

$$\begin{aligned}
&\delta(x) = \forall y\ \neg\mathrm{Succ}(y, x)\\
&\varphi_<(x, y) = x < y\\
&\varphi_{P_{a_i}}(x_1) = \exists x_2 \ldots \exists x_i \ \ \textstyle\bigwedge_{j \in [i-1]} \mathrm{Succ}(x_j, x_{j+1}) \wedge \forall y\ \neg\mathrm{Succ}(x_i, y)
\end{aligned}$$

where $\mathrm{Succ}(x, y)$ is the formula expressing that y is the successor of x. □

3.4 Tree of language

The tree of a language L over A is a structure, denoted T_L, over the signature $\sigma_{\text{anc}} = \sigma_A \cup \{\prec\}$, where $\prec$ is of arity 2. The domain is the the set of prefixes of the language L. For all $a \in A$, P_a is interpreted as the set of words of L ending with the letter a, and $\prec$ is interpreted as the prefix relation $<_p$.

It is possible to define the tree of a language over a more restricted signature $\sigma_{\text{suc}} = \sigma_A \cup \{\text{Succ}\}$, where Succ is interpreted as the direct successor relation.

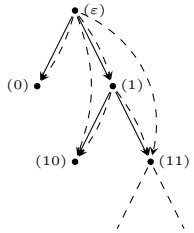


Fig. 2. The tree T_L of the language $L = 1^*0$ where full edges represent the Succ relation and dashed edges represent the relation $\prec$. The root is not labelled, all leaves are labelled by 0, and all other nodes by 1.

Lemma 4. *For any language L over an ordered alphabet A, the linear ordering O_L can be first-order interpreted in T_L over the signature σ_{anc}.*

4 Main undecidability result

This section is devoted to establishing the following theorem.

Theorem 1. *There exists a context-free word with an undecidable first-order theory. Furthermore, such a context-free word can be defined by a finite disjoint union of deterministic context-free languages.*

We now proceed with the proof of Theorem 1. The key ingredient of the proof are the languages obtained by a special form of product, denoted $\otimes$, of deterministic context-free languages.

Definition 2. *Let $L_1, \ldots, L_n$ be languages over the alphabet A. We let $L_1 \otimes \cdots \otimes L_n$ denote the language over the alphabet $A \times \mathbf{2}^n$ containing all nonempty words*

$$(a_1, \bar{b}_1) \cdots (a_m, \bar{b}_m)$$

such that for all $i \in [m]$, a_i belongs to A and $\bar{b}_i$ belongs to $\mathbf{2}^n$ and furthermore for all $\ell \in [n]$, the ℓ-th component of the "flag" $\bar{b}_i$ is equal to 1 if and only if the word $a_1 \cdots a_i$ belongs to L_ℓ.

Intuitively the ℓth bit of the ith letter of the attached flag signals if the prefix of length i projected on A belongs to the language L_ℓ.

Example 2. Let A be the alphabet $\{a, b, c, d\}$. Consider the two deterministic context-free languages $L_1 = \{w \in A^* \mid |w|_a = |w|_b\}$ and $L_2 = \{w \in A^* \mid |w|_c = |w|_d\}$. The language $L_1 \otimes L_2$ contains the word

$$\begin{pmatrix} a \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} c \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} b \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} d \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} a \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix}$$

Note that the language $L_1 \otimes L_2$ is not a context-free language[3].

The key observation is that the structure associated with the product $L_1 \otimes L_2$ of two deterministic context-free languages L_1 and L_2 can be defined in first-order logic in some context-free word.

Proposition 1. *Let L_1 and L_2 be two deterministic context-free languages. There exists a language L over an ordered alphabet which is the disjoint union of deterministic context-free languages not containing the empty word such that the structure $S_{L_1 \otimes L_2}$ can be first-order interpreted in W_L.*

Proof. Let L_1 and L_2 be two deterministic context-free languages. Using a standard binary encoding, we can assume that L_1 and L_2 are on the binary alphabet $A = \{a, b\}$ ordered by $a < b$.

Consider the alphabet $B = \{\triangleright, \bar{a}, \bar{b}, 0_1, 1_1, 0_2, 1_2, \triangleleft, \#, a, b\}$ with $\triangleright < \bar{a} < \bar{b} < 0_1 < 1_1 < 0_2 < 1_2 < \triangleleft < \# < a < b$ and the language L which is the (disjoint) union of the following languages:

$$\begin{array}{ll} \{u\#\triangleright & \mid u \in A^*\} \\ \{u\#u^R\triangleleft & \mid u \in A^*\} \\ \{u\#vx\bar{x} & \mid u \in A^*, v \in A^*, x \in A \text{ and } vx \leq_p u^R\} \\ \{u\#v0_i & \mid u \in A^*, v \in A^+ \text{ and } v \notin L_i\}, \quad i = 1, 2 \\ \{u\#v1_i & \mid u \in A^*, v \in A^+ \text{ and } v \in L_i\}, \quad i = 1, 2. \end{array}$$

We now define a first-order interpretation transforming W_L into $S_{L_1 \otimes L_2}$. The interpretation only keeps the vertices labelled by the predicate $\bar{a}$ or $\bar{b}$:

$$\delta(x) = P_{\bar{a}}(x) \vee P_{\bar{b}}(x).$$

The order relation $<$ coincides with the order relation of W_L but restricted to vertices lying between a vertex labelled by $\triangleright$ and a vertex labelled by $\triangleleft$ with no vertex labelled by $\triangleleft$ in between:

$$\begin{aligned} \varphi_<(x, y) = \exists z_1 \exists z_2 \ \ & (z_1 < x < y < z_2 \wedge P_\triangleright(z_1) \wedge P_\triangleleft(z_2) \\ & \wedge \forall z' \ (z_1 < z' < z_2 \rightarrow \neg P_\triangleleft(z'))). \end{aligned}$$

[3] Indeed, by taking the intersection of $L_1 \otimes L_2$ with the regular language $(A \times \mathbf{2}^2)^*(A \times \{1\} \times \{1\})$ and then projecting on the first component, we can obtain the language $\{w \in A^* \mid w \neq \epsilon,\ |w|_a = |w_b| \text{ and } |w|_c = |w|_d\}$, which is known not to be context-free.

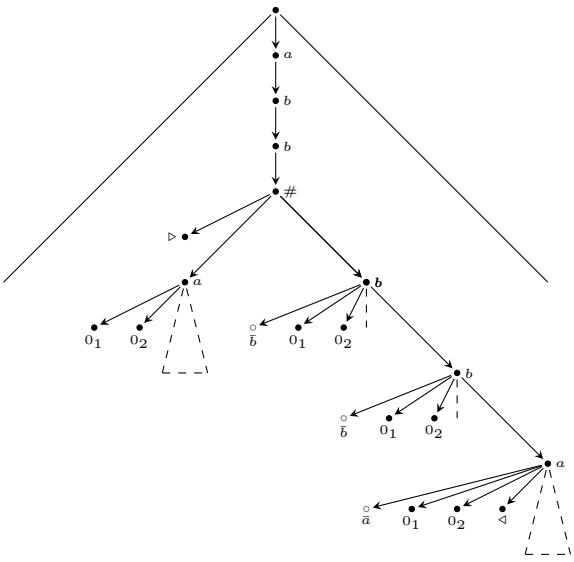


Fig. 3. Assuming that $L_1 = A^*b$ and $L_2 = A^*ba$, we depict the part of the tree T_L corresponding to the subset of the language producing after interpretation the word $(b,1,0)(b,1,0)(a,0,1)$. The white nodes correspond to words that are kept by the interpretation.

For $(a,b,c) \in A \times \mathbf{2} \times \mathbf{2}$, the predicate $P_{(a,b,c)}$ is defined by

$$\varphi_{P_{(a,b,c)}}(x) = P_{\bar{a}}(x) \wedge \exists y \exists z \ (\mathrm{Succ}(x,y) \ \wedge \ \mathrm{Succ}(y,z) \wedge P_{b_1}(y) \wedge P_{c_2}(z))$$

where b_1 is 0_1 if $b = 0$ and 1_1 otherwise, and similarly, c_2 is 0_2 if $c = 0$ and 1_2 otherwise. □

To conclude the proof, it remains to show that there exists a product of two deterministic context-free languages whose structure has an undecidable first-order theory.

Proposition 2. *There exist two deterministic context-free languages L_1 and L_2 such that $S_{L_1 \otimes L_2}$ has an undecidable first-order theory.*

Proof. Let A be the alphabet containing the symbols $+_{\mathbf{1}}, +_{\mathbf{2}}, -_{\mathbf{1}}, -_{\mathbf{2}}, =$ and \$. Consider the following deterministic context-free languages:

$$L_1 = \{w \in A^* \mid |w|_{+_{\mathbf{1}}} = |w|_{-_{\mathbf{1}}}\}$$
$$L_2 = \{w \in A^* \mid |w|_{+_{\mathbf{2}}} = |w|_{-_{\mathbf{2}}}\}.$$

In our argument, we will use reduction from the halting problem of 2-counter machines. A program for a 2-counter machine is a nonempty sequence $I_1; \ldots; I_n$ of instructions, where I_n is a halt instruction and all other instructions I_i are of

the form Inc_j, Dec_j or $\mathrm{Test}_j(k)$, $j = 1, 2$, $k \in [n]$. Here, Inc_j increments the value of the jth counter by 1, Dec_j decrements it by 1 – provided that the current value is not 0. If the current value is 0, then Dec_j corresponds to a skip instruction. A conditional branch instruction $\mathrm{Test}_j(k)$, where $k \in [n]$, tests the current value of the jth counter and transfers the control to the kth instruction if this value is 0. Initially, the values of the counters are 0. The instructions are executed sequentially, except for the effect of the conditional branch instructions. The machine halts when I_n is executed. Without loss of generality we will consider machines that do not try to decrease the value of a counter whose value is 0. This condition can be syntactically enforced by prefacing each decrease operation with a test.

Formally, a computation sequence for the program is a sequence $i_1, \ldots, i_m$ of instruction numbers in $[n]$ such that one can define a valuation mapping $v : [m] \mapsto \mathbb{N} \times \mathbb{N}$ associating to every index $\ell \in [m]$, the value of the two counters before executing the ℓth instruction. The computation sequence and the valuation must satisfy the following properties:

- $i_1 = 1$ and $v(1) = (0, 0)$
- for all $\ell \in [m-1]$, if $I_{i_\ell} = \mathrm{Inc}_1$ (resp. $I_{i_\ell} = \mathrm{Inc}_2$), then $i_{\ell+1} = i_\ell + 1$ and $v(\ell+1) = v(\ell) + (1, 0)$ (resp. $v(\ell+1) = v(\ell) + (0, 1)$),
- for all $\ell \in [m-1]$, if $I_{i_\ell} = \mathrm{Dec}_1$ (resp. $I_{i_\ell} = \mathrm{Dec}_2$), then $i_{\ell+1} = i_\ell + 1$ and $v(\ell+1) = v(\ell) + (-1, 0)$ (resp. $v(\ell+1) = v(\ell) + (0, -1)$),
- for all $\ell \in [m-1]$, if $I_{i_\ell} = \mathrm{Test}_1(k)$ (resp. $I_{i_\ell} = \mathrm{Test}_2(k)$), then $v(\ell+1) = v(\ell)$, and $i_{\ell+1} = k$ if the first (resp. second) component of $v(\ell)$ is equal to zero and $i_{\ell+1} = i_\ell + 1$ otherwise.

Furthermore, a computation sequence is halting if $i_m = n$.

A word w over A is said to represent a computation sequence if it satisfies the following conditions:

1. it is of the form $\$^{i_1} x_1 \cdots x_{m-1} \$^{i_m}$ with $x_1, \ldots, x_{m-1} \in \{+_{\mathbf{1}}, +_{\mathbf{2}}, -_{\mathbf{1}}, -_{\mathbf{2}}, =\}$ and $i_\ell \in [n]$ for $\ell \in [m]$;
2. for all $\ell \in [m-1]$, x_ℓ is $+_{\mathbf{1}}$ if $I_{i_\ell} = \mathrm{Inc}_1$, $+_{\mathbf{2}}$ if $I_{i_\ell} = \mathrm{Inc}_2$, $-_{\mathbf{1}}$ if $I_{i_\ell} = \mathrm{Dec}_1$, $-_{\mathbf{2}}$ if $I_{i_\ell} = \mathrm{Dec}_2$, and $=$ otherwise;
3. $i_1, \ldots, i_m$ is a computation sequence.

Claim. The exists a first-order formula φ_{Halt} such that for every word $w \in L_1 \otimes L_2$, $S_w \models \varphi_{\mathrm{Halt}}$ if and only if w projected on A represents a halting computation sequence.

Proof. It is straightforward to write a first-order formula ensuring that w projected on A satisfies the conditions 1 and 2 above. To be able to express condition 3 in first-order, the only difficulty consists in testing if before the ℓth instruction the value of a given counter is 0, where $\ell \in [n]$. For this it is enough to notice that if the value of the two counters before the ℓth instruction is given by $(|w_\ell|_{+_{\mathbf{1}}} - |w_\ell|_{-_{\mathbf{1}}}, |w_\ell|_{+_{\mathbf{2}}} - |w_\ell|_{-_{\mathbf{2}}})$, where $w_\ell = \$^{i_1} x_1 \cdots x_{\ell-1} \$^{i_\ell}$ – recall that we do not consider machines that can decrease the value of a counter when its value is zero –, then by the definition of $L_1 \otimes L_2$, the fact that the jth counter has value 0 can be tested by reading the jth bit of the attached flag.

To conclude the proof, we construct a formula ϕ_P such that $S_{L_1 \otimes L_2} \models \phi_P$ if and only if $L_1 \otimes L_2$ contains at least one word satisfying φ_{Halt} (and hence if and only if P is halting). The formula ϕ_P is equal to $\exists x\, \varphi'_{\text{Halt}}(x)$, where $\varphi'_{\text{Halt}}(x)$ is the formula obtained from φ_{Halt} by relativizing all quantifications to elements that are comparable to x with respect to $<$ or $=$. □

We can now prove Theorem 1.

Proof (Proof of Theorem 1). By Proposition 2, there exist two deterministic context-free languages L_1 and L_2 such that $S_{L_1 \otimes L_2}$ has an undecidable first-order theory. By Proposition 1, there exists a language L over an ordered alphabet which is the disjoint union of deterministic context-free languages (not containing the empty word), such that $S_{L_1 \otimes L_2}$ can be first-order interpreted in W_L. Thus W_L has an undecidable first-order theory. □

5 Corollaries of Theorem 1

Using Lemma 3, Theorem 2 can be transferred to context-free linear orderings.

Corollary 1. *There exists a context-free linear ordering with an undecidable first-order theory. Furthermore such a context-free linear ordering can be defined by a finite disjoint union of deterministic context-free languages.*

Proof. By Theorem 1, there exists a finite union L of deterministic context-free languages such that W_L has an undecidable first-order theory. By Lemma 3, there exists a context-free language L' and a first-order interpretation $\mathcal{I}$ of W_L in $O_{L'}$. Since the first-order theory of W_L is undecidable, it follows that the first-order theory of $O_{L'}$ is also undecidable.

Observe that as L is a finite disjoint union of deterministic context-free languages, the language L' constructed in Lemma 3 can also be chosen to be a finite disjoint union of deterministic context-free languages. □

As all deterministic context-free linear orderings have a decidable monadic second-order theory, this result provides an example of a context-free linear ordering that is not deterministic context-free.

Corollary 2. *There exists a context-free linear ordering that is not a deterministic context-free linear ordering.*

Moving our focus to trees, we obtain a simple proof of a result first proved in [26].

Corollary 3. *There exists a finite disjoint union of deterministic context-free languages such that*

1. *the associated tree has an undecidable first-order theory over the signature σ_{anc} (which includes the ancestor relation),*
2. *it cannot be accepted by any deterministic collapsible automaton.*

Proof. The first claim is a direct consequence of Corollary 1 and Lemma 4. The second claim then follows form the fact that any language accepted by a deterministic collapsible automaton has a tree with a decidable MSO-theory [25], and hence a decidable first-order theory over the signature σ_{anc}. □

In a draft of this article, we asked if there exists a context-free language whose associated tree has an undecidable first-order theory over the signature σ_{suc}. This question was positively answered by Markus Lohrey [23].

Proposition 3 (M. Lohrey). *There exists a context-free language whose associated tree has an undecidable first-order theory over the signature σ_{suc}.*

Proof. The proof starts by establishing that there exists a context-free language L_0 over an alphabet A such that the following problem is undecidable : "Given a word $w \in A^+$, decide if L_0 contains all words ending with w".

To construct L_0, we consider a universal Turing machine M with a set Q of states and a set Γ of tape symbols. It is well-known that the set L_M of words representing ill-formed or non-terminating computations of a Turing machine is a context-free language [21]. More precisely, a configuration is represented by a word in $\Gamma^* Q \Gamma^*$ and a computation $c_0 \vdash \cdots \vdash c_n$ is represented by the word $w = c_0 \sharp c_1^R \sharp c_2 \cdots$. For the language L_0, we take $L_M^R = \{w^R \mid w \in L_M\}$. For a word w representing an initial configuration c_0 of M, it is clear that L_M^R contains all words ending with w^R if and only if M does not have a halting computation starting from c_0. Hence the aforementioned problem is undecidable for L_0.

Let \$ be a fresh symbol. We show that the first-order theory over σ_{suc} of $T_{L_0\$}$ is undecidable.

Let $w = a_1 \cdots a_n$ with $n > 0$ be a word over A. The formula φ_w over σ_{suc},

$$\varphi_w = \forall x \, \exists x_1 \cdots \exists x_{n+1} \quad \text{Succ}(x, x_1) \wedge \textstyle\bigwedge_{i \in [n]} \text{Succ}(x_i, x_{i+1}) \\ \wedge \textstyle\bigwedge_{i \in [n]} P_{a_i}(x_i) \wedge P_{\$}(x_{n+1}),$$

expresses that for every word u, the word $uw\$$ is in the domain of $T_{L_0\$}$ and hence that uw belongs to L_0. Hence for all $w \in A^+$, $T_{L_0\$} \models \varphi_w$ if and only if L_0 contains all words ending in w. We conclude that the undecidability of the first-order theory of $T_{L_0\$}$ follows from the undecidability of the above problem. □

As observed by M. Lohrey, the above proposition can be stated for the more restrictive signature {Succ} in which labels are omitted. Indeed, this follows from the following lemma that shows that a context-free tree over σ_{suc} can be first-order interpreted in a context-free trees over {Succ}.

Lemma 5. *Let L be a context-free language. There exists a context-free language L' such that T_L over the signature σ_{suc} can be first-order interpreted in $T_{L'}$ over the signature* {Succ}.

Proof. Let L be a context-free language over the alphabet $A = \{a_1, \ldots, a_n\}$. Without loss of generality we may assume that L contains a nonempty word. Consider the context-free language

$$L' = \{ua_j i \mid i \leq j \text{ and } \exists w \in L, ua_j \leq_p w\}$$

over the alphabet $A \cup [n]$. The first-order interpretation of T_L over σ_{suc} in $T_{L'}$ over $\{\mathrm{Succ}\}$ only keeps non-leaf nodes. The Succ relation is inherited. For $i \in [n]$, the predicate P_{a_i} holds at those non-leaf nodes which have exactly i sons which are leaves. Formally the interpretation is defined by

$$\begin{aligned}
&\delta(x) = \neg\mathrm{Leaf}(x)\\
&\varphi_{\mathrm{Succ}}(x,y) = \mathrm{Succ}(x,y)\\
&\varphi_{P_{a_i}}(x) = \exists^{=i} y\ \mathrm{Succ}(x,y) \wedge \mathrm{Leaf}(y)
\end{aligned}$$

where $\mathrm{Leaf}(x)$ is a formula expressing that x is a leaf. □

It would be interesting to know if Property 3 remains true when only considering context-free languages which are finite unions of deterministic context-free languages.

6 Discussion

In this article, we have established that the linear orderings and trees associated with context-free languages are more complex than those associated to deterministic context-free languages. This result even holds for finite disjoint unions of deterministic context-free languages and hence for unambiguous context-free languages. It would be interesting to investigate whether such a separation result exists for context-free scattered linear orderings.

Acknowledgements

The authors would like to thank Markus Lohrey for his remarks on a previous draft of this article as well as for the proof of Proposition 3. The authors are also grateful for the remarks of the anonymous referees.

References

1. S. L. Bloom and C. Choffrut, Long words: the theory of concatenation and omega-power, *Theoretical Computer Science*, 259(2001), 533–548.
2. S. L. Bloom and Z. Ésik, Deciding whether the frontier of a regular tree is scattered, *Fundamenta Informaticae*, 55(2003), 1–21.
3. S. L. Bloom and Z. Ésik, The equational theory of regular words, *Information and Computation*, 197(2005), 55–89.
4. S. L. Bloom and Z. Ésik, Regular and algebraic words and ordinals, in: *CALCO 2007*, Bergen, LNCS 4624, Springer, 2007, 1–15.
5. S. L. Bloom and Z. Ésik, Algebraic ordinals, *Fundamenta Informaticae*, 99(2010), 383–407.
6. S. L. Bloom and Z. Ésik, Algebraic linear orderings, *Int. J. Foundations of Computer Science*, 22(2011), 491–515.
7. L. Braud and A. Carayol, Linear orders in the pushdown hierarchy, in: *ICALP 2010*, LNCS 6199, Springer, 2010, 88–99.

8. A. Carayol and Z. Ésik, The FC-rank of a context-free language, to appear.
9. A. Carayol and S. Wöhrle. The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata, in: *FSTTCS 2003*, LNCS 2914, pp. 112–123, Springer, 2003.
10. D. Caucal. On infinite terms having a decidable monadic theory, in: *MFCS 2002*, LNCS 2420, 165–176, Springer, 2002.
11. D. Caucal. On infinite transition graphs having a decidable monadic theory. *Theoretical Computer Science* 290(2003), 79–115.
12. B. Courcelle, Frontiers of infinite trees. *Theoretical Informatics and Applications*, 12(1978), 319–337.
13. B. Courcelle, Fundamental properties of infinite trees, *Theoretical Computer Science*, 25(1983), 95–169.
14. C. Delhommé, Automaticité des ordinaux et des graphes homogènes, *C. R. Acad. Sci. Paris, Ser. I*, 339(2004) 5–10.
15. Z. Ésik, Representing small ordinals by finite automata, in: *12th Workshop Descriptional Complexity of Formal Systems*, Saskatoon, Canada, 2010, EPTCS, vol. 31, 2010, 78–87.
16. Z. Ésik, An undecidable property of context-free linear orders, *Information Processing Letters*, 111(2010), 107–109.
17. Z. Ésik, Scattered context-free linear orders, in: *DLT 2011*, Milan, LNCS 6795, Springer-Verlag, 2011, 216–227.
18. Z. Ésik and S. Iván, Hausdorff rank of scattered context-free linear orders, in: *LATIN 2012*, Arequipa, Peru, LNCS 7256, Springer-Verlag, 2012, 291–302.
19. M. Hague, A. Murawski, C.-H. L. Ong and Olivier Serre, Collapsible Pushdown Automata and Recursion Schemes, in: *LICS 2008*, IEEE, 2008, 452–461.
20. S. Heilbrunner, An algorithm for the solution of fixed-point equations for infinite words, *Theoretical Informatics and Applications*, 14(1980), 131–141.
21. J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, Languages and Computation. *Addison-Wesley*, 1979
22. B. Khoussainov, S. Rubin and F. Stephan, Automatic linear orders and trees, *ACM Trans. Comput. Log.*, 6(2005), 625–700.
23. M. Lohrey. Private communication. 2012
24. M. Lohrey and Ch. Mathiessen, Isomorphism of regular words and trees, in: *ICALP 2011*, Zurich, Switzerland, 2011, LNCS 6756, 210–221.
25. C.-H. Luke Ong, On model-checking trees generated by higher-order recursion schemes. in: *LICS 2006*, IEEE Press, 2006, 81–90.
26. P. Parys, Higher-order stacks can not replace nondeterminism. Note published on the author's webpage (3 pages), February 2010.
27. J. G. Rosenstein, *Linear Orderings*, Pure and Applied Mathematics, Vol. 98, Academic Press, 1982.
28. W. Thomas, On frontiers of regular trees, *Theoretical Informatics and Applications*, vol. 20, 1986, 371–381.