

MSO-definable Properties of Muller Context-Free Languages Are Decidable

Zoltán Ésik, Szabolcs Iván

► **To cite this version:**

Zoltán Ésik, Szabolcs Iván. MSO-definable Properties of Muller Context-Free Languages Are Decidable. 18th International Workshop on Descriptive Complexity of Formal Systems (DCFS), Jul 2016, Bucharest, Romania. pp.87-97, 10.1007/978-3-319-41114-9_7. hal-01633953

HAL Id: hal-01633953

<https://hal.inria.fr/hal-01633953>

Submitted on 13 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



MSO-definable properties of Muller context-free languages are decidable^{*}

Zoltán Ésik and Szabolcs Iván

University of Szeged

Abstract. We show that it is decidable given an MSO-definable property P of countable words and a Muller context-free grammar G , whether every word in the language generated by G satisfies P .

1 Introduction

A word, called ‘arrangement’ in [9], is an isomorphism type of a countable labeled linear order. Such words form a generalization of the classic notions of finite and ω -words.

Finite automata on ω -words have by now a vast literature, see [13] for a comprehensive treatment. Finite automata acting on well-ordered words longer than ω have been investigated in [1, 6, 7, 16, 17], to mention a few references. In the last decade, the theory of automata on well-ordered words has been extended to automata on all countable words, including scattered and dense words. In [2, 3, 5], both operational and logical characterizations of the class of languages of countable words recognized by finite automata were obtained.

Context-free grammars generating ω -words were introduced in [8] and subsequently studied in [4, 12]. Context-free grammars generating arbitrary countable words were defined in [10, 11]. Actually, two types of grammars were defined, context-free grammars with Büchi acceptance condition (BCFG), and context-free grammars with Muller acceptance condition (MCFG). These grammars generate the Büchi and the Muller context-free languages of countable words, abbreviated as BCFLs and MCFLs. Every BCFL is clearly an MCFL, but there exists an MCFL of well-ordered words that is not a BCFL, for example the set of all countable well-ordered words over some alphabet. In contrast, the set of all countable words over an alphabet is a BCFL.

In [11], it was shown that it is decidable (in polynomial time) whether a given MCFG generates well-ordered (or scattered) words only. This result was obtained by analysing the structure of a finite graph canonically associated with the grammar. In this note we establish a generic decidability result to the effect that whenever P is some property of countable words definable in monadic second-order logic (MSO), e.g., being well-ordered or scattered, then it is decidable whether an MCFG generates only words satisfying P . Of course for such

^{*} This research was supported by NKFI grant number K108448.

a general setting one cannot hope for efficient algorithms since model checking MSO is nonelementary.

The main idea of the proof is that one can associate with each MCFG a regular tree in which every derivation tree of the grammar can be represented and moreover, the set of all derivation trees is MSO-definable.

2 Notation

Countable words and Muller context-free grammars. An *alphabet* is a finite nonempty set Σ of symbols, usually called *letters*. A *word* over Σ is a strict linear ordering $(I, <)$ equipped with a *labeling function* $\lambda : I \rightarrow \Sigma$. The *empty word*, denoted ε , is the unique word over the empty ordering. It is assumed that no alphabet contains ε . When Σ is an alphabet, Σ_ε stands for $\Sigma \cup \{\varepsilon\}$. An *embedding of words* is a mapping between the respective underlying linear orderings that preserves the order and the labeling; a surjective embedding is an *isomorphism*. We usually identify isomorphic words and denote by $\Sigma^\#$ the set of all countable words over the alphabet Σ . As usual, we denote the collections of finite and ω -words over Σ by Σ^* and Σ^ω , respectively. Sometimes we will also use the same notation for infinite sets.

Let \mathbb{N} denote the set of positive integers. When $u \in \mathbb{N}^*$ and $i \in \mathbb{N}$, we usually write ui as $u \cdot i$. A *tree domain* D is a prefix- and left-sibling closed nonempty (but possibly infinite) subset of \mathbb{N}^* . Thus, whenever $u \cdot (i + 1)$ is in D , where $i \in \mathbb{N}$, then $u \cdot i$ is also in D , and $u \cdot i \in D$ implies $u \in D$ as well. Elements of a tree domain D are also called the *nodes* of D . When u and $u \cdot i$ are nodes of D , where $u \in \mathbb{N}^*$ and $i \in \mathbb{N}$, then $u \cdot i$ is called a *child* of u . A *descendant* of a node u is a node of the form uv , or $u \cdot v$, where $v \in \mathbb{N}^*$. Nodes of D having no child are the *leaves* of D . The leaves, equipped with the order inherited from the lexicographic ordering \prec_ℓ of \mathbb{N}^* (that is, $u \prec_\ell v$ iff $u = wiw'$ and $v = wjw''$ for some $i < j \in \mathbb{N}$, $w, w', w'' \in \mathbb{N}^*$) form the *frontier* of D , denoted $\text{fr}(D)$. An *inner node* of D is a non-leaf node. A *path* of a tree domain D is a (finite or infinite) prefix-closed subset π of D such that each node of π has at most one child in π . Given a tree domain D and some node $u \in D$, the *sub-tree domain* of D rooted at u is the tree domain $D|_u = \{v : uv \in D\}$.

A *tree* over some alphabet Δ , or a Δ -tree for short, is a mapping $t : \text{dom}(t) \rightarrow \Delta_\varepsilon$, where $\text{dom}(t)$ is a tree domain, such that inner vertices are mapped to letters in Δ . Notions such as nodes, paths etc. of tree domains are lifted to trees. When π is a path of the tree t , then $\text{labels}(\pi) = \{t(u) : u \in \pi\}$ is the set of labels occurring on π and $\text{infLabels}(\pi) = \bigcap_{u \in \pi} \{t(v) : uv \in \pi\} \subseteq \text{labels}(\pi)$ is the set of labels occurring infinitely often. Given a tree t and some node $u \in \text{dom}(t)$, the *subtree* of t rooted at u is the tree $t|_u$ with domain $\text{dom}(t|_u) = \text{dom}(t)|_u$ and labeling $t|_u(v) = t(uv)$. A tree is *regular* if it has finitely many subtrees.

The *frontier word* $\text{lfr}(t)$ of a tree t is determined by the leaves *not* labeled by ε , which is equipped with the lexicographic ordering of \mathbb{N}^* and the labeling function inherited from t . The *root symbol* of t is $t(\varepsilon)$.

A *Muller context-free grammar* [11], or MCFG for short, is a system $G = (V, \Sigma, R, S, \mathcal{F})$, where V and Σ are the pairwise disjoint alphabets of *nonterminals* and *terminals* respectively, R is the finite set of *productions* of the form $A \rightarrow \alpha$ with $A \in V$ and $\alpha \in (\Sigma \cup V)^*$, $S \in V$ is the *start symbol* and $\mathcal{F} \subseteq P(V)$ is the set of *accepting sets*.

A $(V \cup \Sigma)$ -tree t is *locally consistent* with the above grammar G if it satisfies the following conditions:

1. The root symbol of t is S .
2. For each inner node u of t there exists a production $A \rightarrow X_1 \dots X_n$ in R with $t(u) = A$, $X_i \in V \cup \Sigma$ such that:
 - (a) either $n > 0$, the children of u are exactly $u \cdot 1, \dots, u \cdot n$ and for each $1 \leq i \leq n$, $t(u \cdot i) = X_i$;
 - (b) or $n = 0$ and u has a single child $u \cdot 1$ labeled ε .
3. The leaves of t are labeled in Σ_ε .

A *derivation tree* of the above grammar G is a locally consistent tree t satisfying the additional condition that for each infinite path π of t , $\text{inflLabels}(\pi)$ is an accepting set of G .

The language $L(G) \subseteq \Sigma^\#$ *generated* by G is the set of frontier words of derivation trees. A *Muller context-free language*, or MCFL for short, is a language generated by some MCFG.

Example 1. If $G = (\{S, I\}, \{a, b\}, R, S, \{\{I\}\})$, with

$$R = \{S \rightarrow a, S \rightarrow b, S \rightarrow \varepsilon, S \rightarrow I, I \rightarrow SI\},$$

then $L(G)$ consists of all the well-ordered words over $\{a, b\}$.

Indeed, assume t_1, t_2, \dots are derivation trees. Then so is the tree t depicted in Figure 1 with frontier word $\text{lfr}(t_1)\text{lfr}(t_2)\dots$. Thus, $L(G)$ contains the empty word (by $S \rightarrow \varepsilon$), the words of length 1 (by $S \rightarrow a$ and $S \rightarrow b$), and is closed under taking “ ω -products”. Since the least class of order types which contains $\mathbf{0}$, $\mathbf{1}$ and which is closed under ω -sums is the class of all countable ordinals (see e.g. [15]), $L(G)$ contains all the well-ordered words over $\{a, b\}$.

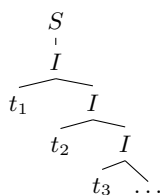


Fig. 1. Derivation tree corresponding to Example 1

For the other direction, assume t is a derivation tree having a frontier word containing an infinite descending chain $u_1 \succ_\ell u_2 \succ_\ell \dots$. Then let us define the

path v_0, v_1, \dots in t : $v_0 = \varepsilon$ and v_{i+1} is $v_i \cdot 1$ if this node is an ancestor of infinitely many u_j and $v_i \cdot 2$ otherwise (which happens if v_i corresponds to the production $I \rightarrow SI$ and the node $v_i \cdot 1$ (which is labeled S) has no descendant of the form u_j at all). Note that for each u_j there exists a unique v_{i_j} such that v_{i_j} is an ancestor of u_j and v_{i_j+1} is not, since the length of the words v_i grows without a bound. Now these nodes v_{i_j} correspond to the production $I \rightarrow SI$ and $v_{i_j+1} = v_{i_j} \cdot 1$, so that the successor of v_{i_j} along the path is labeled by S . Hence v_0, v_1, \dots , is a path π in t such that $\text{infLabels}(\pi)$ contains S , which is a contradiction since the only accepting set is $\{I\}$.

MSO on trees and words. Let \mathcal{X}_1 and \mathcal{X}_2 be fixed, countably infinite, disjoint sets of *first-order* and *second-order* variables, respectively. It is assumed that \mathcal{X}_1 and \mathcal{X}_2 are disjoint from alphabets, they do not contain ε , etc.

Given an alphabet Δ , the set of *monadic second-order*, or MSO-formulas (for *trees* over Δ) is the least set satisfying the following conditions:

1. When \mathbf{x} is a first-order variable and $\delta \in \Delta_\varepsilon$ is a symbol, then $\delta(\mathbf{x})$ is an MSO-formula.
2. When \mathbf{x} and \mathbf{y} are first-order variables, then $\mathbf{y} = \mathbf{x} \cdot 1$ and $\text{sibling}(\mathbf{x}, \mathbf{y})$ are MSO-formulas.
3. When \mathbf{x} is a first-order and \mathbf{X} is a second-order variable, then $\mathbf{X}(\mathbf{x})$, also written $\mathbf{x} \in \mathbf{X}$ is an MSO-formula.
4. When φ and ψ are MSO-formulas, then so are $(\varphi \vee \psi)$ and $(\neg\varphi)$.
5. When \mathbf{x} (\mathbf{X} , resp.) is a first-order (second-order, resp.) variable and φ is an MSO-formula, then so is $(\exists \mathbf{x}\varphi)$ ($(\exists \mathbf{X}\varphi)$, resp.).

We also use the standard abbreviations of $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi = (\neg\varphi) \vee \psi$, $\forall \mathbf{x}\varphi = \neg \exists \mathbf{x} \neg \varphi$ etc., and omit some parentheses for the sake of readability. Formulas over Δ are interpreted on Δ -trees in the expected way. A *structure* is a triple (t, Π_1, Π_2) where t is a Δ -tree, $\Pi_1 : \mathcal{X}_1 \rightarrow \text{dom}(t)$ assigns a node of t to each first-order variable, and $\Pi_2 : \mathcal{X}_2 \rightarrow P(\text{dom}(t))$ assigns a set of nodes of t to each second-order variable. Then, the above structure *satisfies* the formula φ , denoted $(t, \Pi_1, \Pi_2) \models \varphi$, if and only if one of the following conditions holds:

1. $\varphi = \delta(\mathbf{x})$ for $\delta \in \Delta_\varepsilon$ and $\mathbf{x} \in \mathcal{X}_1$, and $t(\Pi_1(\mathbf{x})) = \delta$.
2. $\varphi = (\mathbf{y} = \mathbf{x} \cdot 1)$ for $\mathbf{x}, \mathbf{y} \in \mathcal{X}_1$ and $\Pi_1(\mathbf{y}) = \Pi_1(\mathbf{x}) \cdot 1$.
3. $\varphi = \text{sibling}(\mathbf{x}, \mathbf{y})$ for $\mathbf{x}, \mathbf{y} \in \mathcal{X}_1$ and there exist $u \in \mathbb{N}^*$, $i \in \mathbb{N}$ with $\Pi_1(\mathbf{x}) = u \cdot i$, $\Pi_1(\mathbf{y}) = u \cdot (i + 1)$.
4. $\varphi = \mathbf{X}(\mathbf{x})$ for $\mathbf{x} \in \mathcal{X}_1$, $\mathbf{X} \in \mathcal{X}_2$ and $\Pi_1(\mathbf{x}) \in \Pi_2(\mathbf{X})$.
5. $\varphi = (\varphi_1 \vee \varphi_2)$ and (t, Π_1, Π_2) satisfies φ_1 or φ_2 (or both).
6. $\varphi = (\neg\varphi_1)$ and it is not the case that the structure satisfies φ_1 .
7. $\varphi = (\exists \mathbf{x}\varphi_1)$ and there is a structure (t, Π'_1, Π_2) satisfying φ_1 such that $\Pi_1(\mathbf{y}) = \Pi'_1(\mathbf{y})$ for each first-order variable $\mathbf{y} \neq \mathbf{x}$.
8. $\varphi = (\exists \mathbf{X}\varphi_1)$ and there is a structure (t, Π_1, Π'_2) satisfying φ_1 such that $\Pi_2(\mathbf{Y}) = \Pi'_2(\mathbf{Y})$ for each second-order variable $\mathbf{Y} \neq \mathbf{X}$.

It is clear that satisfaction depends on $\Pi_1(\mathbf{x})$ or $\Pi_2(\mathbf{X})$ only if the appropriate variable occurs *freely* in the formula (i.e. not within the scope of some \exists quantifier). Hence when φ is a *sentence*, a formula without free variable occurrences, it makes sense to write $t \models \varphi$ instead of $(t, \Pi_1, \Pi_2) \models \varphi$.

In order to ease notation, when Π_1 and Π_2 are clear from the context, we write $\underline{\mathbf{x}}$ and $\underline{\mathbf{X}}$ for $\Pi_1(\mathbf{x})$ and $\Pi_2(\mathbf{X})$.

Example 2. One can define the i -th child relation $\mathbf{y} = \mathbf{x} \cdot i$ for $i \in \mathbb{N}$ inductively as $\exists \mathbf{z}(\mathbf{z} = \mathbf{x} \cdot (i - 1) \wedge \text{sibling}(\mathbf{z}, \mathbf{y}))$ (which is satisfied by a structure if and only if $\underline{\mathbf{y}} = \underline{\mathbf{x}} \cdot i$).

Consider the formula

$$\begin{aligned} \text{child}(\mathbf{x}, \mathbf{y}) = & \exists \mathbf{z}(\mathbf{z} = \mathbf{x} \cdot 1) \wedge \\ & \forall \mathbf{X} \left((\forall \mathbf{z}((\mathbf{z} = \mathbf{x} \cdot 1) \rightarrow \mathbf{z} \in \mathbf{X})) \wedge \right. \\ & \left. \forall \mathbf{z} \forall \mathbf{w}(\mathbf{z} \in \mathbf{X} \wedge \text{sibling}(\mathbf{z}, \mathbf{w}) \rightarrow \mathbf{w} \in \mathbf{X}) \rightarrow \mathbf{y} \in \mathbf{X} \right). \end{aligned}$$

Then, $\text{child}(\mathbf{x}, \mathbf{y})$ holds in the structure iff $\underline{\mathbf{x}}$ has a first child and if whenever a set $\underline{\mathbf{X}}$ contains the first child of $\underline{\mathbf{x}}$ and is closed under taking right siblings, then $\underline{\mathbf{X}}$ contains $\underline{\mathbf{y}}$ as well, that is, if and only if $\underline{\mathbf{y}} = \underline{\mathbf{x}} \cdot i$ for some i .

As another example, the formula

$$\exists \mathbf{x}(\mathbf{x} \in \mathbf{X}) \wedge \forall \mathbf{x} \forall \mathbf{y}(\mathbf{x} \in \mathbf{X} \wedge \text{child}(\mathbf{y}, \mathbf{x}) \rightarrow \mathbf{y} \in \mathbf{X})$$

holds in a structure if $\underline{\mathbf{X}}$ is a nonempty, prefix-closed subset of the nodes.

It is well-known [14] that given any *regular* tree t and MSO sentence φ , it is decidable whether $t \models \varphi$ holds.

For countable Σ -words, the syntax and semantics of MSO are slightly changed due to the differing relational structure: the atomic formulas are of the form $a(\mathbf{x})$ for $a \in \Sigma$ and $\mathbf{x} \in \mathcal{X}_1$ and $\mathbf{x} < \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathcal{X}_1$, interpreted in the expected way. A property P of countable Σ -words is called *MSO-definable* if there exists an MSO sentence φ_P which is satisfied exactly by those Σ -words having property P .

3 Result

Let us fix a Muller context-free grammar $G = (V, \Sigma, R, S, \mathcal{F})$ for this section, with R being disjoint from $V \cup \Sigma$. Without loss of generality we assume that each $A \in V$ is the left-hand side of at least one production. We define the *grammar tree* associated with G as the unique derivation tree \mathcal{T} of the following grammar $G' = (V \cup R, \Sigma, R', S, P(V \cup R))$ with R' consisting of productions of the following form:

1. When $A \rightarrow \alpha_1, \dots, A \rightarrow \alpha_k$ are all the productions of G having A on their left side in some fixed ordering of the productions, then $A \rightarrow (A \rightarrow \alpha_1)(A \rightarrow \alpha_2) \dots (A \rightarrow \alpha_k)$ is a production of G' (the right-hand side of this single production is in R^* while the left-hand side is in V).

2. For each production $A \rightarrow X_1 \dots X_k$, the production $(A \rightarrow X_1 \dots X_k) \rightarrow X_1 \dots X_k$. (which is a production of the form $r \rightarrow \alpha$ for some $r \in R$ and $\alpha \in (V \cup \Sigma)^*$) is a production of G' .

Note that each element of $V \cup R$ is the left-hand side of exactly one production in R' , hence there exists exactly one locally consistent tree of G' . Also, since the acceptance condition is $P(V \cup R)$, this tree is a valid derivation tree of the grammar, thus \mathcal{T} is well-defined and has at most $|V| + |R| + |\Sigma_\varepsilon|$ subtrees up to isomorphism. Hence it is a regular tree.

Example 3. For the MCFG of Example 1, this tree \mathcal{T} is depicted in Figure 2.

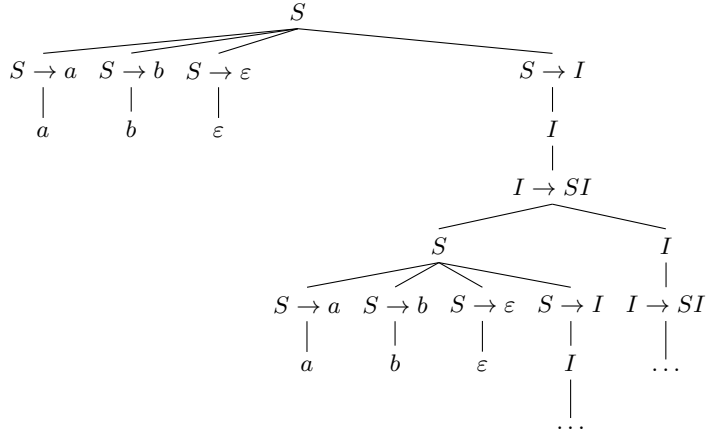


Fig. 2. Grammar tree of the MCFG of Example 1

Moreover, each locally consistent tree t of G can be *embedded* into \mathcal{T} in the following sense: there exists a mapping $h_t : \text{dom}(t) \rightarrow \text{dom}(\mathcal{T})$ with $h_t(\varepsilon) = \varepsilon$ and $t(u) = \mathcal{T}(h_t(u))$ for each $u \in \text{dom}(t)$, moreover $h_t(u \cdot i)$ is a descendant (in particular, a grandchild) of $h_t(u)$ in \mathcal{T} for each $u \cdot i \in \text{dom}(t)$, moreover, when u and v are siblings in t , then so are $h_t(u)$ and $h_t(v)$ in \mathcal{T} . Indeed, assume $u \cdot i \in \text{dom}(t)$ and that $u' = h_t(u)$ is already defined. Then since u is an inner node of t , we have $t(u) = A \in V$. By $\mathcal{T}(u') = t(u) = A$, each production $r = A \rightarrow \alpha$ occurs as $\mathcal{T}(u' \cdot k_r)$ for some $k_r \in \mathbb{N}$. In particular, let $r = A \rightarrow X_1 \dots X_n$ be the production corresponding to u , so that u has n children and $t(u \cdot j) = X_j$ for each $j = 1, \dots, n$ (subsuming the case when $n = 0$ as $t(u \cdot 1) = \varepsilon$). Then, we define $h_t(u \cdot j)$ as $u' \cdot k_r \cdot j$.

We call a prefix-closed nonempty set $T \subseteq \text{dom}(\mathcal{T})$ *derivation-like* iff it satisfies the following conditions:

1. For each $u \in T$ with $\mathcal{T}(u) \in R$, each child of u is in T .
2. For each $u \in T$ with $\mathcal{T}(u) \in V$, exactly one child of u is in T .

It is clear that derivation-like subsets of $\text{dom}(\mathcal{T})$ are in one-to-one correspondence with the locally consistent trees of G : with any locally consistent tree t of G we associate the derivation-like set $T \subseteq \text{dom}(\mathcal{T})$ which is the closure of $\text{im}(h_t)$ with respect to the prefix relation. Given a derivation-like set T , the corresponding locally consistent tree is denoted t .

Example 4. Figure 3 shows a (part of a) derivation tree t of the grammar of Example 1 and the corresponding derivation-like subset T of $\text{dom}(\mathcal{T})$ (as nodes in boldface).

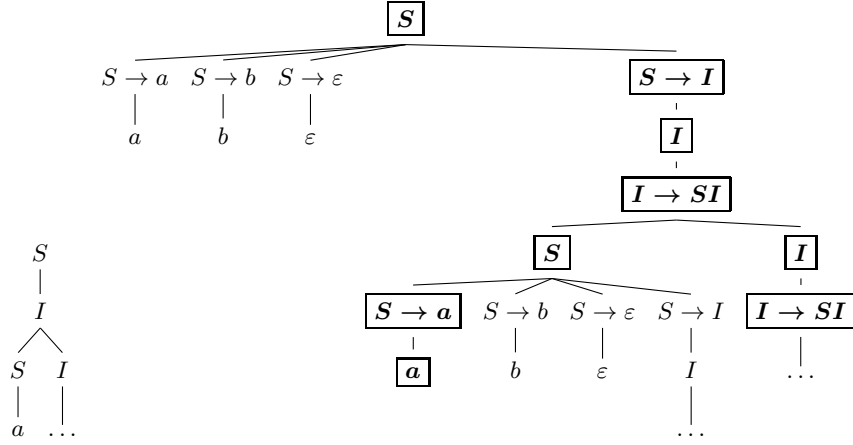


Fig. 3. A part of a derivation tree t of Example 1 and the corresponding derivation-like subset T of \mathcal{T}

Proposition 1. *There is an MSO formula $d(\mathbf{X})$ with the free variable $\mathbf{X} \in \mathcal{X}_2$ such that $(\mathcal{T}, \Pi_1, \Pi_2) \models d(\mathbf{X})$ if and only if $\underline{\mathbf{X}}$ is a derivation-like set.*

(In short, it is MSO-definable whether some set $\underline{\mathbf{X}}$ is derivation-like.)

Proof. We can define $d(\mathbf{X})$ as the conjunction of the formulas stating that $\underline{\mathbf{X}}$ is nonempty and prefix-closed (see Example 2), the formula

$$\forall \mathbf{x} \left((\mathbf{x} \in \mathbf{X} \wedge \bigvee_{r \in R} r(\mathbf{x})) \rightarrow \forall \mathbf{y} (\text{child}(\mathbf{x}, \mathbf{y}) \rightarrow \mathbf{y} \in \mathbf{X}) \right)$$

stating that all the children of the nodes labeled by productions are members of $\underline{\mathbf{X}}$, and the formula

$$\forall \mathbf{x} \left((\mathbf{x} \in \mathbf{X} \wedge \bigvee_{A \in V} A(\mathbf{x})) \rightarrow \exists \mathbf{y} (\text{child}(\mathbf{x}, \mathbf{y}) \wedge \forall \mathbf{z} (\text{child}(\mathbf{x}, \mathbf{z}) \wedge \mathbf{z} \in \mathbf{X} \leftrightarrow \mathbf{z} = \mathbf{y})) \right)$$

stating that exactly one child of the nodes labeled by nonterminals is in $\underline{\mathbf{X}}$. \square

Also, from a derivation-like set T determining a locally consistent tree t of G , one can select a subset of nodes corresponding to a path of t :

Proposition 2. *There exists an MSO-formula $p(\mathbf{X}, \mathbf{Y})$ with the free second-order variables \mathbf{X}, \mathbf{Y} such that $(\mathcal{T}, \Pi_1, \Pi_2) \models p(\mathbf{X}, \mathbf{Y})$ if and only if $\underline{\mathbf{X}} = T$ is a derivation-like subset of \mathcal{T} with corresponding locally consistent tree t and $\underline{\mathbf{Y}} = h_t(\pi)$ ¹ for some infinite path π of t .*

Proof. $p(\mathbf{X}, \mathbf{Y})$ expresses the following:

- i) $\underline{\mathbf{X}}$ is a derivation-like subset of \mathcal{T} , i.e. $d(\mathbf{X})$ holds,
- ii) $\underline{\mathbf{Y}} \subseteq \underline{\mathbf{X}}$,
- iii) each of the nodes of $\underline{\mathbf{Y}}$ is labeled by some member of V ,
- iv) whenever v is a grandparent of some node $u \in \underline{\mathbf{Y}}$, then $v \in \underline{\mathbf{Y}}$ as well, and
- v) each $u \in \underline{\mathbf{Y}}$ has exactly one grandchild in $\underline{\mathbf{Y}}$.

These properties can clearly be defined in MSO. □

Proposition 3. *There is an MSO formula $d'(\mathbf{X})$ expressing that $\underline{\mathbf{X}}$ is a derivation-like set corresponding to an actual derivation tree of G .*

Proof. The descendant relation $\mathbf{x} \preceq \mathbf{y}$ can be defined in MSO by a formula expressing that whenever $\underline{\mathbf{Y}}$ is a prefix-closed set containing \mathbf{y} , then $\underline{\mathbf{Y}}$ contains \mathbf{x} as well.

Then, for each $F \in \mathcal{F}$ we can construct a formula $m_F(\mathbf{Y})$ stating that if $\underline{\mathbf{Y}} = h_t(\pi)$ for some path π of some locally consistent tree t of G , then $\text{infLabels}(\pi) = F$:

$$m_F(\mathbf{Y}) = \bigwedge_{A \in F} i_A(\mathbf{Y}) \wedge \bigwedge_{A \notin F} \neg i_A(\mathbf{Y})$$

where $i_A(\mathbf{Y})$ is the formula

$$\forall \mathbf{x}(\mathbf{x} \in \mathbf{Y} \rightarrow \exists \mathbf{y}(\mathbf{x} \preceq \mathbf{y} \wedge \mathbf{y} \in \mathbf{Y} \wedge A(\mathbf{y})))$$

stating that A occurs infinitely many times on the path given by $\underline{\mathbf{Y}}$.

Now, we can define $d'(\mathbf{X})$ as

$$d(\mathbf{X}) \wedge \forall \mathbf{Y}(p(\mathbf{X}, \mathbf{Y}) \rightarrow \bigvee_{F \in \mathcal{F}} m_F(\mathbf{Y}))$$

expressing that $\underline{\mathbf{X}}$ is a derivation-like set corresponding to some locally consistent tree t of G such that any infinite path π of t satisfies the Muller acceptance condition. □

Now a set $\underline{\mathbf{Y}} \subseteq \text{dom}(\mathcal{T})$ corresponds to a frontier word of some derivation tree of G (i.e. belongs to $L(G)$) if and only if there exists some $\underline{\mathbf{X}} \subseteq \text{dom}(\mathcal{T})$ satisfying

¹ Here, $h_t(\pi)$ denotes the set of images of the nodes of π with respect to the embedding h_t .

$d'(\mathbf{X})$ such that a node $v \in \text{dom}(\mathcal{T})$ is in $\underline{\mathbf{Y}}$ if and only if $v \in \underline{\mathbf{X}}$ and is a Σ -labeled node of \mathcal{T} , which is an MSO-definable property:

$$\exists \mathbf{X} (d'(\mathbf{X}) \wedge \forall y (y \in \mathbf{Y} \leftrightarrow (y \in \mathbf{X} \wedge \bigvee_{a \in \Sigma} a(y))))).$$

Moreover, the lexicographic ordering on these leaves can also be defined in MSO as $u \prec_\ell v$ iff there exists some common ancestor w of u and v such that $w \cdot i$ and $w \cdot j$ are respectively ancestors of u and v for some $i < j$:

$$\mathbf{x} \prec_\ell \mathbf{y} = \exists z_1 \exists z_2 (z_1 \preceq \mathbf{x} \wedge z_2 \preceq \mathbf{y} \wedge \text{sibling}^+(z_1, z_2))$$

where $\text{sibling}^+(\mathbf{x}, \mathbf{y})$ is the transitive closure of sibling :

$$\mathbf{x} \neq \mathbf{y} \wedge \forall \mathbf{X} (\mathbf{x} \in \mathbf{X} \wedge \forall z_1 \forall z_2 (z_1 \in \mathbf{X} \wedge \text{sibling}(z_1, z_2) \rightarrow z_2 \in \mathbf{X})) \rightarrow \mathbf{y} \in \mathbf{X}$$

Hence we have shown:

Proposition 4. *For any MCFG G , there exists an effectively constructible MSO formula $w(\mathbf{Y})$ such that $(\mathcal{T}, \Pi_1, \Pi_2) \models w(\mathbf{Y})$ if and only if $\underline{\mathbf{Y}}$ is the set of Σ -labeled leaves of some derivation-like subset of \mathcal{T} corresponding to a derivation tree of G .*

As a corollary, we obtain the main result of this note.

Theorem 1. *It is decidable for a given MCFL L and an MSO-definable property φ of words whether every member of L satisfies φ .*

Proof. The question can be reduced to checking whether \mathcal{T} satisfies the formula $\forall \mathbf{Y} (w(\mathbf{Y}) \rightarrow \varphi(\mathbf{Y}))$ where $\varphi(\mathbf{Y})$ is obtained from φ by replacing all the subformulas of the form $\exists \mathbf{x} \varphi'$ and $\exists \mathbf{X} \varphi'$ respectively to $\exists \mathbf{x} (\mathbf{x} \in \mathbf{Y} \wedge \varphi')$ and $\exists \mathbf{X} (\forall \mathbf{x} (\mathbf{x} \in \mathbf{X} \rightarrow \varphi')) \wedge \varphi'$ and substituting the formula defining the lexicographic ordering $\mathbf{x} \prec_\ell \mathbf{y}$ in place of the atomic formulas $\mathbf{x} < \mathbf{y}$. Since \mathcal{T} is regular, model checking the resulting formula on \mathcal{T} is decidable. \square

(We remark that thus it is also decidable whether *there exists* a word in L satisfying an MSO formula φ since such a word exists if and only if *not* all members of L satisfy $\neg \varphi$.)

In particular, our former decidability results (without complexity bounds) regarding whether an MCFG generates scattered (or well-ordered) words are corollaries of this general theorem. However, since model-checking MSO formulas on regular trees has a high complexity in general ($\text{tower}(n)$ when n is the alternation depth of the second-order quantifiers), no polytime decision procedures follow from the present theorem. Nevertheless several interesting properties are decidable in polynomial time, including whether every word generated by an MCFG is scattered or well-ordered, cf. [11].

As another example, let $\Sigma = \{a, b\}$ be an alphabet. The following formula $\text{segment}(\mathbf{X})$ expresses that $\underline{\mathbf{X}}$ is a nonempty interval, or *segment* of a given word:

$$(\exists \mathbf{x} \mathbf{x} \in \mathbf{X}) \wedge (\forall \mathbf{x} \forall \mathbf{y} \forall \mathbf{z} (\mathbf{x} < \mathbf{y} \wedge \mathbf{y} < \mathbf{z} \wedge \mathbf{x} \in \mathbf{X} \wedge \mathbf{z} \in \mathbf{X}) \rightarrow \mathbf{y} \in \mathbf{X}).$$

The following formula $\text{dense}(\mathbf{X})$ expresses that $\underline{\mathbf{X}}$ is a dense subset (containing at least two elements) of the word:

$$\begin{aligned} & \exists x \exists y (x < y \wedge x \in \mathbf{X} \wedge y \in \mathbf{X}) \wedge \\ & \forall x \forall y (x < y \wedge x \in \mathbf{X} \wedge y \in \mathbf{X} \rightarrow \exists z (x < z \wedge z < y \wedge z \in \mathbf{X})). \end{aligned}$$

Thus, the property “there exists a dense segment X of the word such that for all $x < y$ in X there exists z, z' with $x < z, z' < y$ and z is labeled by a , z' is labeled by b ” is also expressible in MSO as

$$\begin{aligned} & \exists \mathbf{X} \left(\text{dense}(\mathbf{X}) \wedge \text{segment}(\mathbf{X}) \wedge \right. \\ & \quad \forall x \forall y (x < y \wedge x \in \mathbf{X} \wedge y \in \mathbf{X} \\ & \quad \left. \rightarrow \exists z \exists z' (x < z \wedge z < y \wedge x < z' \wedge z' < y \wedge a(z) \wedge b(z')) \right) \end{aligned}$$

In other words, $w \in \{a, b\}^\#$ satisfies the above formula iff $w = u\{a, b\}^n v$ for some words $u, v \in \{a, b\}^\#$ where $\{a, b\}^n$ is the so-called *shuffle* of a and b . Thus, it is also decidable for a MCFL L whether every word in L is of the form $u\{a, b\}^n v$.

Another expressible property is that whether a word is the *shuffle product* of, say, a dense word and a scattered word consisting only of a 's, that is, whether the underlying linear order can be partitioned into two subsets such that the two subwords determined by the partitions satisfy the appropriate property:

$$\exists \mathbf{X} \left(\text{dense}(\mathbf{X}) \wedge \forall x (x \notin \mathbf{X} \rightarrow a(x)) \wedge \neg \exists \mathbf{Y} (\forall x (x \in \mathbf{Y} \rightarrow x \notin \mathbf{X}) \wedge \text{dense}(\mathbf{Y})) \right).$$

Thus, it is also decidable for a given MCFL L whether every member of L is a shuffle product of a dense word and a scattered one consisting only of a 's.

4 Conclusion

We have proved that there is an algorithm to decide for a Muller context-free language L generated by an MCFG and an MSO-definable property P of words whether every word in L has property P . We obtained this result by assigning a regular tree t to an MCFG such that the derivation trees of the grammar have an MSO-interpretation in t . We then used the fact that the MSO-theory of a regular tree is decidable.

There is an alternative method. First, we can prove that the MCFLs are exactly the frontier languages of the tree languages recognizable by Muller tree automata. This is similar to the well-known fact that ordinary context-free languages are the frontier languages of the languages of finite trees recognizable by finite tree automata. Also, there is an algorithm to decide, for a Muller tree automaton and an MSO-definable property of trees whether every tree in the language L recognized by the automaton has property P . This follows using the fact that every Muller tree automaton can be converted to an MSO-formula φ , and if P is definable by the formula ψ , then it holds that every tree in L satisfies P iff there is no tree satisfying $\varphi \wedge \neg\psi$, which is decidable.

References

1. Nicolas Bedon. Finite automata and ordinals. *Theoretical Computer Science*, 156(1-2):119–144, 1996.
2. Nicolas Bedon, Alexis Bès, Olivier Carton, and Chloe Rispal. Logic and rational languages of words indexed by linear orderings. In *Proceedings of the third International Computer Science Symposium in Russia (CSR'08)*, volume 5010 of *Lecture Notes in Computer Science*, pages 76–85. Springer, 2008.
3. Alexis Bès and Olivier Carton. A kleene theorem for languages of words indexed by linear orderings. In *Proceedings of Developments in Language Theory 2005*, volume 3572 of *Lecture Notes in Computer Science*, pages 158–167. Springer-Verlag, 2005.
4. L. Boasson. Context-free sets of infinite words. In K. Weihrauch, editor, *Theoretical Computer Science 4th GI Conference*, volume 67 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin Heidelberg, 1979.
5. Véronique Bruyère and Olivier Carton. Automata on linear orderings. *Journal of Computer and System Sciences*, 73(1):1 – 24, 2007.
6. J. Richard Büchi. The monadic second order theory of ω_1 . *Lecture Notes in Mathematics*, 328:1–127, 1973.
7. Yaacov Choueka. Finite automata, definable sets, and regular expressions over ω^n -tapes. *Journal of Computer and System Sciences*, 17(1):81 – 97, 1978.
8. R. S. Cohen and A. Y. Gold. Theory of ω -languages, parts one and two. *Journal of Computer and System Sciences*, 15:169–208, 1977.
9. Bruno Courcelle. Frontiers of infinite trees. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 12(4):319–337, 1978.
10. Zoltán Ésik and Szabolcs Iván. Büchi context-free languages. *Theoretical Computer Science*, 412(8–10):805 – 821, 2011.
11. Zoltán Ésik and Szabolcs Iván. On Müller context-free grammars. *Theoretical Computer Science*, 416:17 – 32, 2012.
12. Maurice Nivat. Sur les ensembles de mots infinis engendrés par une grammaire algébrique. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 12(3):259–278, 1978.
13. D. Perrin and J.É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics. Elsevier Science, 2004.
14. Michael O. Rabin. Decidability of second order theories and automata on infinite trees. *Transactions of the AMS*, 141:1–35, 1969.
15. Joseph G. Rosenstein. *Linear orderings*. Pure and Applied Mathematics. Academic Press, 1982.
16. Jerzy Wojciechowski. Classes of transfinite sequences accepted by finite automata. *Fundamenta Informaticae*, 7(2):191–223, 1984.
17. Jerzy Wojciechowski. Finite automata on transfinite sequences and regular expressions. *Fundamenta Informaticae*, 8(3–4):379–396, 1985.