Regular tree languages, cardinality predicates, and addition-invariant FO

Frederik Harwath and Nicole Schweikardt

Institut für Informatik
Goethe-Universität Frankfurt am Main, Germany
Email: `{harwath,schweika}@cs.uni-frankfurt.de`
URL: `http://www.tks.cs.uni-frankfurt.de/{harwath,schweika}`

Abstract

This paper considers the logic $\mathrm{FO}_{\mathrm{card}}$, i.e., first-order logic with *cardinality predicates* that can specify the size of a structure modulo some number. We study the expressive power of $\mathrm{FO}_{\mathrm{card}}$ on the class of languages of ranked, finite, labelled trees with successor relations.

Our first main result characterises the class of $\mathrm{FO}_{\mathrm{card}}$-definable tree languages in terms of algebraic closure properties of the tree languages. As it can be effectively checked whether the language of a given tree automaton satisfies these closure properties, we obtain a decidable characterisation of the class of regular tree languages definable in $\mathrm{FO}_{\mathrm{card}}$.

Our second main result considers first-order logic with unary relations, successor relations, and two additional designated symbols $<$ and $+$ that must be interpreted as a linear order and its associated addition. Such a formula is called *addition-invariant* if, for each fixed interpretation of the unary relations and successor relations, its result is independent of the particular interpretation of $<$ and $+$. We show that the $\mathrm{FO}_{\mathrm{card}}$-definable tree languages are exactly the regular tree languages definable in addition-invariant first-order logic.

Our proof techniques involve tools from algebraic automata theory, reasoning with locality arguments, and the use of logical interpretations. We combine and extend methods developed by Benedikt and Segoufin (ACM ToCL, 2009) and Schweikardt and Segoufin (LICS, 2010).

1998 ACM Subject Classification F.4.1 Mathematical Logic, F.4.3 Formal Languages

Keywords and phrases regular tree languages, algebraic closure properties, decidable characterisations, addition-invariant first-order logic, logical interpretations

Digital Object Identifier 10.4230/LIPIcs.STACS.2012.489

1 Introduction

The search for decidable characterisations of certain classes of languages has a long tradition in logic and automata theory. For the case of *word languages* definable by first-order logic FO and extensions thereof, the situation is quite well-understood by now. For example, the languages definable by FO over linearly ordered word structures are exactly the *aperiodic* languages [10], and the languages definable by FO on word structures with successor relation (but without order) are precisely the aperiodic languages *closed under idempotent-guarded swaps* [1]. Similar results are known for extensions of FO such as, e.g., the logics $\mathrm{FO}_{\mathrm{mod}}$ and $\mathrm{FO}_{\mathrm{card}}$ that enrich FO by quantifiers that count modulo some integer, respectively, by predicates that specify the size of the word modulo some integer [11, 9]. All these characterisations lead to effective procedures for deciding whether a given regular language is definable by the respective logic. We refer to [11] for a detailed overview.

Transferring such characterisations from word languages to *tree* languages is usually quite a challenge. In particular, it is a longstanding open problem to find a decidable characterisation

29th Symposium on Theoretical Aspects of Computer Science (STACS'12).
Editors: Christoph Dürr, Thomas Wilke; pp. 489–500
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

of the regular tree languages definable by FO on tree structures with prefix-order (i.e., the transitive closure of the parent-child relation). For trees with successor relations (and without prefix-order), according results for FO and FO_{mod} have been achieved in [2], using a new notion of *closure under guarded swaps*.

The present paper transfers techniques of [2] from FO to FO_{card}, generalising results of [9] from word languages to tree languages. We consider languages of ranked, finite, labelled trees with successor relations (and without prefix-order) definable in FO_{card}. Our first main result identifies a new property of tree languages called *closure under transfer* and shows that the FO_{card}-definable tree languages coincide with the regular tree languages that are closed under transfer and under guarded swaps. This leads to a decidable characterisation of the FO_{card}-definable regular tree languages.

Our second main result considers first-order logic with unary relations, successor relations, and two additional designated symbols $<$ and $+$ that must be interpreted as a linear order and its associated addition. Such a formula is called *addition-invariant* if, for each fixed interpretation of the unary relations and successor relations, its result is independent of the particular interpretation of $<$ and $+$. For some background on addition-invariant first-order logic we refer to [7, 9]. The present paper's second main result shows that the FO_{card}-definable tree languages are exactly the regular tree languages definable in addition-invariant first-order logic. Our proof techniques involve tools from algebraic automata theory [11, 2], reasoning with locality arguments [7, 6], and the use of logical interpretations (cf., e.g., [7, 5]). In particular, we combine and extend methods developed in [2, 9].

Structure of the paper. We start by fixing the necessary notations in section 2. In section 3, we state and prove our algebraic characterisation of the FO_{card}-definable tree languages. Section 4 shows that it can be effectively decided whether a given regular tree language has the closure properties associated with FO_{card}-definability. In section 5, we consider addition-invariant FO and show that the FO_{card}-definable tree languages are exactly the regular tree languages definable in addition-invariant FO.

Due to space limitations, many technical details of our proofs are deferred to the full version of this paper, available at the authors' websites.

2 Preliminaries

We write $\mathbb{N}$ for the set of natural numbers starting with 0, and $\mathbb{N}_{\geq 1}$ for $\mathbb{N} \setminus \{0\}$. The notation $[n, m]$ is used for the closed interval of natural numbers between n and m. We use the abbreviations $[n] := [0, n]$, $(n] := [1, n]$ and $[n) := [0, n-1]$. By $x \text{ MOD } m$ we denote the non-negative remainder when dividing x by m.

We consider *tree languages* of finite trees that are labelled with symbols of a finite alphabet Σ (fixed for the course of the paper). We assume that each node has at most two children, called left child and right child, respectively. This is done for the ease of exposition; all our results easily generalise to arbitrary ranked finite labelled trees. Words are identified with trees where every node has at most one child. We write $\lhd$ (resp. $\unlhd$) for the transitive (resp. reflexive-transitive) closure of the parent-child relation. On each tree, there exists a canonical linear order of the nodes of the tree according to the order in which they are visited by a breadth-first-traversal, where the left child of a node is visited before the right child. We refer to this ordering as the *bf-order* of a tree. The *size* of a tree t, denoted $|t|$, is the number of nodes of t.

A tree is identified with a logical structure, whose universe consists of all nodes of the tree. For each $a \in \Sigma$, it contains unary relations P_a for the set of nodes with label a, and binary relations S_1 resp. S_2 for the left resp. right child relation. The set of all formulae of first-order logic with these relation symbols is denoted by FO.

We now introduce some basic concepts that will be used throughout this article to talk about the shape of trees. Let t be a tree. For a node v of t, we denote the *subtree rooted at* v by $t_{|v}$. The *k-spill* of v in t, denoted by $t^k_{|v}$, is the restriction of $t_{|v}$ to all vertices with distance at most k from v. The equivalence class of $t^k_{|v}$ under isomorphism is called the *k-type* of v in t. We say that v *realises* its k-type in t. Two nodes (in, potentially, distinct trees) are k-similar, if they realise the same k-type. Two trees are k-similar if their roots are k-similar. For each k-type τ, $|t|_\tau$ is the number of nodes of t that realise τ. If $|t|_\tau > 0$, then τ *occurs* in t. For a tree s, we write $s \leq_k t$ if $|s|_\tau \leq |t|_\tau$ holds for all k-types τ. We use $s =_k t$ and $s <_k t$ analogously. These notations are extended to finite sequences $(t_i)_{i\in(n]}$ of trees by the definition $|(t_i)_{i\in(n]}|_\tau = |t_1|_\tau + \cdots + |t_n|_\tau$.

An *n-context* C, for $n \in \mathbb{N}_{\geq 1}$, is a tree with distinguished leaves $h_1, \ldots, h_n$, called *holes*. If $n = 1$, C is plainly called a *context*. The *inner tree* of C is the tree obtained from C by removing its holes. The size $|C|$ of C is the size of its inner tree. By replacing a hole h_i of C by a tree t (resp. context) one obtains an $(n-1)$-context (resp. n-context). Given trees $t_1, \ldots, t_n$, let $C[t_1, \ldots, t_n]$ be the tree obtained from C by replacing the hole h_i by t_i, for all $i \in (n]$. For a context C and a tree t, $Ct := C \cdot t := C[t]$ is the *concatenation* of C with t. We mostly use contexts as means to decompose given trees. For a tree t with a node u and nodes $v_1, \ldots, v_n$ below u, let $t[u, v_1, \ldots, v_n)$ be the n-context obtained from $t_{|u}$ by removing all nodes strictly below $v_1, \ldots, v_n$ and making $v_1, \ldots, v_n$ holes. Usually, the k-type of a hole's parent in this n-context will not equal its k-type in t. For this reason, we introduce the following concepts.

Let C be a context with a hole h. A k-type-labelling of C is a labelling λ of the nodes of C that assigns $(k+1)$-types to the nodes of the inner tree of C, and a k-type to h. A tree t is *compatible* with λ, if the $(k+1)$-type of t induces $\lambda(h)$. If there exists such a tree t and $\lambda(v)$ is the $(k+1)$-type of v in $C \cdot t$, for each $v \in C$ with $v \neq h$, then λ is *consistent*. A context C together with a *consistent* k-type-labelling λ of C is called a *k-abstract context*. All concepts introduced for trees will be used for abstract contexts as well. When we refer to the types of nodes in abstract contexts, we always mean the types given by λ. (C, λ) is *compatible* with a tree t, if t is compatible with λ. If (C, λ) is compatible with another k-abstract context (C', λ'), then $C \cdot C'$ is also a k-abstract context. A *k-abstract loop* is a k-abstract-context where the $(k+1)$-type of the root induces the k-type of its hole. Notice that for a k-abstract loop C the set of $(k+1)$-types realised by nodes of C, and that realised by nodes of C^n is the same, for any $n \in \mathbb{N}_{\geq 1}$.

Let L be a tree language. Two trees s, t *agree on* L if either $s, t \in L$ or $s, t \notin L$. Two contexts C_1, C_2 are *congruent modulo* L, written $C_1 \cong_L C_2$, if for all contexts C and trees t, the trees $C \cdot C_1 \cdot t$ and $C \cdot C_2 \cdot t$ agree on L. A context C is *idempotent* if $C^2 \cong_L C$. A tree language is *regular*, if it is recognised by a *(bottom-up) tree automaton* (for a reference on tree automata, see e.g. [4]). The set of all contexts with the operation of concatenation forms a monoid. The quotient of this monoid by $\cong_L$ is called, in analogy to the word case, the *syntactic monoid* of L. Just as in the word case, a tree language is regular iff its syntactic monoid is finite. Therefore, with each regular tree language L come two associated constants: ω_L is the least number such that for each context C, C^{ω_L} is idempotent; κ_L is the least number such that, for each context C there exists a context C' of size at most κ_L with $C' \cong_L C$. In both cases, we usually omit the index L.

3 First-order logic with cardinality predicates

In this section, we consider an extension of first-order logic by *cardinality predicates*, and we characterise regular tree languages definable by this logic. Let $\mathrm{FO}_{\mathrm{card}}$ (resp. $\mathrm{FO}^{m}_{\mathrm{card}}$) denote the set of formulae of first-order logic that, in addition to the common rules for the formation of formulae of first-order logic, may use relation symbols from the set $\{\mathrm{C}_{a,m} : m \in \mathbb{N}_{\geq 1}, a \in [m)\}$, (resp. $\{\mathrm{C}_{a,m} : a \in [m)\}$) where each $C_{a,m}$ is a nullary relation symbol. The formula $\mathrm{C}_{a,m}$ shall be satisfied in a structure iff the size of the structure's universe is congruent a modulo m. A tree language L is $\mathrm{FO}_{\mathrm{card}}$-definable iff there exists an $\mathrm{FO}_{\mathrm{card}}$-sentence φ such that $t \in L$ iff $t \models \varphi$, for all trees t. For trees s, t, we write $s \approx^{m}_{q} t$ to denote that s and t agree on all tree languages definable by $\mathrm{FO}^{m}_{\mathrm{card}}$-sentences of quantifier depth at most q.

Our aim for this section will be a characterisation of the $\mathrm{FO}_{\mathrm{card}}$-definable tree languages in terms of their closure properties. To achieve this goal, we combine and extend the techniques developed in [2] and [9]. In [2], necessary and sufficient conditions for the FO-definability of a regular tree language were shown. To state the characterisation, we need to introduce the following notions. A tree language L is *aperiodic* iff there exists a constant $\ell \in \mathbb{N}$, such that $C^{\ell} \cong_L C^{\ell+1}$, for all contexts C.

▶ **Definition 3.1** (Guarded Swaps, [2]). Let t be a tree with root w.

Let $u \trianglelefteq v \trianglelefteq u' \trianglelefteq v'$ be nodes of t. Let $C := t[w, u), C_1 := t[u, v), D := t[v, u'), C_2 := t[u', v')$ be contexts and let $s := t_{|v'}$. The *vertical swap* of the tree $t = C \cdot C_1 \cdot D \cdot C_2 \cdot s$ between C_1 and C_2 is the tree $C \cdot C_2 \cdot D \cdot C_1 \cdot s$. If u and u' as well as v and v' are k-similar, for some $k \in \mathbb{N}$, then we say that the vertical swap is *k-guarded*.

Let u and v be incomparable nodes of t (i.e., neither $u \trianglelefteq v$ nor $v \trianglelefteq u$ holds). Let $C := t[w, u, v)$, and let $s_1 := t_{|u}$ and $s_2 := t_{|v}$. The *horizontal swap* of $t = C[s_1, s_2]$ between u and v is the tree $C[s_2, s_1]$. If u and v are k-similar, for some $k \in \mathbb{N}$, then we say that the horizontal swap is k-guarded.

A tree t' is a *k-guarded swap* of t iff it is either a k-guarded vertical swap or a k-guarded horizontal swap of t. A tree language L is *closed under k-guarded swaps* iff each tree t agrees on L with all its k-guarded swaps. L is closed under guarded swaps, if there exists a k such that L is closed under k-guarded swaps.

The characterisation of FO-definable tree languages by Benedikt and Segoufin reads as follows:

▶ **Theorem 3.2** ([2]). *A tree language is* FO*-definable iff it is regular, aperiodic, and closed under guarded swaps.*

For the special case of regular *word* languages it was shown in [9] that $\mathrm{FO}_{\mathrm{card}}$-definability of a regular language is characterised by certain closure properties as well. Let L be a regular word language over an alphabet Σ. The language L is said to be *closed under idempotent-guarded swaps* if for all words $\underline{p}, \underline{q}, r, e, f \in \Sigma^*$, such that e, f are idempotent it holds that $e\,\underline{p}\,f\,r\,e\,\underline{q}\,f \cong_L e\,\underline{q}\,f\,r\,e\,\underline{p}\,f$. A regular word language L is *closed under transfer* iff $x^{\omega+1}yz^{\omega} \cong_L x^{\omega}yz^{\omega+1}$, for all words x, y, z with $|x| = |z|$. The following was proved in [9]:

▶ **Theorem 3.3** ([9]). *A word language L is $\mathrm{FO}_{\mathrm{card}}$-definable iff it is regular, closed under idempotent-guarded swaps, and closed under transfer.*

The present section's goal is to show that Theorem 3.3 can be generalised to regular tree languages. To this end, we introduce a generalisation of the notion of closure under transfer to tree languages. Similarly to guarded swaps, it consists of a "vertical" and a "horizontal" property. In this case, the vertical property is a direct translation of the notion of transfer from the syntactic monoid of word languages to the syntactic monoid of tree languages.

▶ **Definition 3.4** (Transfer)**.** A regular tree language L is closed under *vertical transfer* if $C_1^{\omega+1} \cdot D \cdot C_2^{\omega} \cong_L C_1^{\omega} \cdot D \cdot C_2^{\omega+1}$ holds for all contexts C_1, D, C_2 with $|C_1| = |C_2|$. L is closed under *horizontal transfer* if the trees $C[C_1^{\omega+1} \cdot s_1,\, C_2^{\omega} \cdot s_2]$ and $C[C_1^{\omega} \cdot s_1,\, C_2^{\omega+1} \cdot s_2]$ agree on L, for all 2-contexts C, contexts C_1, C_2 with $|C_1| = |C_2|$, and trees s_1 and s_2. If L is closed under vertical and under horizontal transfer, then L is called *closed under transfer*.

The remainder of this section is devoted to the proof of the following theorem:

▶ **Theorem 3.5** (Characterisation of the $\mathrm{FO}_{\mathrm{card}}$-definable tree languages)**.** *A tree language L is $\mathrm{FO}_{\mathrm{card}}$-definable iff it is regular, closed under guarded swaps, and closed under transfer.*

The transfer property, as stated in Definition 3.4, makes the connection with the corresponding property of word languages clear and will be useful when considering decidability questions in section 4. For the proof of Theorem 3.5, however, another formulation of transfer in terms of the following notion will be convenient:

▶ **Definition 3.6** (Growing a tree by a context; n-Template)**.** Let t be a tree with root w, and let Δ be a context. Let p be a node of t, and let $C := t[w, p)$ and $s := t_{|p}$ (i.e. $t = Cs$). We say that the tree $C\Delta s$ is obtained from t by *letting t grow by Δ at p*.

For any $n \in \mathbb{N}$, an *n-template* is a tree T with n *expansion points*, i.e. n distinct distinguished nodes $p_1, \ldots, p_n$. We define $T\langle\rangle := T$ and, given a sequence of contexts $\Delta_1, \ldots, \Delta_\ell$, for an $\ell \le n$, we let $T\langle\Delta_1, \ldots, \Delta_\ell\rangle$ be the tree obtained by letting $T\langle\Delta_1, \ldots, \Delta_{\ell-1}\rangle$ grow by Δ_ℓ at p_ℓ.

The following lemma gives an alternative formulation of transfer in terms of templates and is easily seen to be true:

▶ **Lemma 3.7** (Alternative formulation of transfer)**.** *Let L be a regular tree language. L is* closed under transfer *iff for all 2-templates T and all contexts C_1, C_2 with $|C_1| = |C_2|$, the trees $T\langle C_1^{\omega+1}, C_2^{\omega}\rangle$ and $T\langle C_1^{\omega}, C_2^{\omega+1}\rangle$ agree on L.*

The outline of our proof of Theorem 3.5 is similar to the proof of Theorem 3.2 given in [2], in that a major part of it consists in the proof of the following lemma:

▶ **Lemma 3.8** (Main lemma)**.** *Let L be a regular tree language that is closed under guarded swaps and closed under transfer. There exist $m, q \in \mathbb{N}$, such that L is a union of $\approx_q^m$-equivalence classes.*

Before we turn to the proof of this lemma, we show how to prove Theorem 3.5 with its help.

Proof of Theorem 3.5 using Lemma 3.8: For the "if-direction", let L be a regular tree language closed under transfer and guarded swaps. We want to show that L is $\mathrm{FO}_{\mathrm{card}}$-definable. By Lemma 3.8, we know that there exist $m, q \in \mathbb{N}$ such that L is a union of $\approx_q^m$-equivalence classes. It easy to see that each such class is definable by an $\mathrm{FO}_{\mathrm{card}}$-sentence, and the number of these classes is finite. Hence L can be defined by the disjunction of such sentences.

For the "only-if" direction, let L be an $\mathrm{FO}_{\mathrm{card}}$-definable tree language. For all $m \in \mathbb{N}_{\ge 1}$ and all $a \in [m)$, let $T_{a,m}$ denote the language of all trees of size a modulo m. We make use of the following easy observation:

▶ Claim 3.9. There exists an $m \in \mathbb{N}_{\ge 1}$ and FO-definable tree languages $L_0, \ldots, L_{m-1}$, such that $L = \bigcup_{a \in [m)} (L_a \cap T_{a,m})$.

Every FO-definable tree language is regular, as is each of the languages $T_{a,m}$. Hence Claim 3.9 immediately implies that L is regular, too. By Theorem 3.2, for each $a \in [m)$ there is a $k_a \in \mathbb{N}$ such that the language L_a is closed under k_a-guarded swaps. Let $k := \max\{k_0, \ldots, k_{m-1}\}$. Each language L_a is obviously closed under k-guarded swaps, too. As guarded-swaps do not change the size of a tree, every language $L_a \cap T_{a,m}$ is closed under k-guarded-swaps, so their union L is so as well.

It remains to show closure of L under transfer. By Lemma 3.7, it suffices to show that for arbitrary 2-templates T, and contexts C_1 and C_2 with $|C_1| = |C_2|$, the trees $s := T\langle C_1^{\omega+1}, C_2^{\omega}\rangle$ and $t := T\langle C_1^{\omega}, C_2^{\omega+1}\rangle$ agree on L. For each $a \in [m)$, let φ_a be the FO-sentence defining L_a, and let q_a denote its quantifier depth. Let φ denote the $\mathrm{FO}_{\mathrm{card}}$-sentence defining L. If s, t agree on their size modulo m and on all sentences $\varphi_0, \ldots, \varphi_{m-1}$, they must, by Claim 3.9, agree on φ as well. Let $q := \max\{q_0, \ldots, q_{m-1}\}$. Because of the idempotency of C_1^{ω} and C_2^{ω}, the trees s and t agree on L iff for some $n \in \mathbb{N}_{\geq 1}$ the trees $s' := T\langle C_1^{n\omega+1}, C_2^{n\omega}\rangle$ and $t' := T\langle C_1^{n\omega}, C_2^{n\omega+1}\rangle$ agree on L. Note that, for any $\ell \in \mathbb{N}$, the number of occurrences of each ℓ-neighbourhood-type in the trees s' and t' is either the same (this is the case for the ℓ-neighbourhood-types of nodes whose ℓ-neighbourhood is neither strictly contained in the sequence of repetitions of C_1 nor in that of C_2) or can be made arbitrarily large by the choice of n (for ℓ-neighbourhood-types of nodes whose ℓ-neighbourhood is strictly contained in a long sequence of repetitions of C_1 or C_2). Thus we may deduce by an application of Hanf's Theorem (see e.g. [5]) that for some n, the trees s' and t' agree on all FO-sentences of quantifier depth at most q. By what was said above, this implies that the trees agree on φ. Therefore they agree on L, so L is closed under transfer. ◀

The remainder of section 3 is dedicated to the proof of the main lemma (Lemma 3.8).

Let L be a regular tree language that is closed under transfer and under k-guarded swaps, for a $k \in \mathbb{N}$. We want to show that there exist $q \in \mathbb{N}$ and $m \in \mathbb{N}_{\geq 1}$ such that two trees s, t that agree on all $\mathrm{FO}^m_{\mathrm{card}}$-sentences of quantifier depth q agree on L. We let m be given by the following lemma, which is an adaptation of a lemma proved in [9] for regular word languages. Its proof is a simple restatement of the proof given in [9] in terms of templates and contexts.

▶ **Lemma 3.10.** *Let L be a regular tree language. L is closed under transfer iff there exists an $m \in \mathbb{N}_{\geq 1}$, such that for all $\ell \in \mathbb{N}_{\geq 1}$, all contexts $\Delta_1, \ldots, \Delta_\ell$, all ℓ-templates T and all $\delta_1, \ldots, \delta_\ell \in \mathbb{N}$, if $\delta_1|\Delta_1| + \delta_2|\Delta_2| + \cdots + \delta_\ell|\Delta_\ell| \equiv 0 \pmod{m}$, then the trees $T\langle \Delta_1^{\omega}, \ldots, \Delta_\ell^{\omega}\rangle$ and $T\langle \Delta_1^{\omega+\delta_1}, \ldots, \Delta_\ell^{\omega+\delta_\ell}\rangle$ agree on L.*

As an intermediate step towards our goal, we show that $s \approx^m_q t$ implies that either t has the same number of occurrences of each $(k+1)$-type as s, and s and t agree on L (in this case we are done with the proof of the main lemma), or the number of occurrences of some type differs between s and t and, in this case, t agrees on L with "s with some additional contexts added". This is basically done as in the proof of Theorem 2 of [2], with only minor modifications of the lemmas used therein. Note, however, that in contrast to [2], we also have to care about the size of the trees modulo m. The following lemma gives a precise formulation of what we show:

▶ **Lemma 3.11.** *Let L be a regular tree language that is closed under transfer and k-guarded swaps, for a $k \in \mathbb{N}$. Let s, t be trees.*
(a) If $s =_{k+1} t$ and s, t are $(k+1)$-similar, then both trees agree on L.
(b) For all $d, m \in \mathbb{N}$ there exists a $q \in \mathbb{N}$ such that, if $s \approx^m_q t$ and not $s =_{k+1} t$, then there exists an $n \in \mathbb{N}_{\geq 1}$ and a sequence of k-abstract loops $(S_i)_{i\in(n]}$ and expansion points $p_1, \ldots, p_n$ in s such that (i) $s\langle S_1, \ldots, S_n\rangle$ agrees with t on L, (ii) $|s\langle S_1, \ldots, S_n\rangle| \equiv |s| \pmod{m}$, (iii) $|s|_\tau > d$, for all $(k+1)$-types τ occurring in $(S_i)_{i\in(n]}$.

We use Lemma 3.11(b) with $d := m\omega b$, where b is fixed according to Lemma 3.12 below. Now choose q according to Lemma 3.11(b), and let s, t be trees such that $s \approx_q^m t$. Our aim is to prove that s and t agree on L. If $s =_{k+1} t$, we are done due to Lemma 3.11(a). For the remainder of this section, assume that $s =_{k+1} t$ does not hold. Let $(S_i)_{i\in(n]}$ be given by Lemma 3.11(b). Let $t' := s\langle S_1, \dots, S_n\rangle$. We know that t' and t agree on L, both trees have the same size modulo m, and each $(k+1)$-type occurring in $(S_i)_{i\in(n]}$ occurs strictly more than d times in s. We will construct a new tree s', $(k+1)$-similar to s, agreeing with s on L, and with all the $(k+1)$-types from the loops that distinguish s from t' added. I.e., we want to achieve $|s'|_\tau = |s|_\tau + |(S_i)_{i\in(n]}|_\tau = |t'|_\tau$ for all $(k+1)$-types τ. Then we are assured by Lemma 3.11(a) that t' and s' agree on L. Therefore, because t' and t as well as s' and s agree on L, we know that s and t agree on L, which is the conclusion that we are aiming at.

As a first step to construct s', we replace each loop S_i by a loop congruent to it modulo L, the size of which is bounded by a constant b depending only on L. The existence of such a loop is guaranteed by the following lemma, whose proof uses a standard pumping argument, where we have to ensure that the size of the given tree remains unchanged modulo m.

▶ **Lemma 3.12** (Loop bound). *Let $k \in \mathbb{N}$, $m \in \mathbb{N}_{\geq 1}$, and let L be a regular tree language. There exists a (computable) bound $b \in \mathbb{N}$ such that for all k-abstract loops Δ there exists a loop Δ' satisfying: (1) $\Delta' \cong_L \Delta$, (2) $|\Delta'| \leq b$, (3) $|\Delta'| \equiv |\Delta| \,(\mathrm{mod}\, m)$, (4) $\Delta' \leq_{k+1} \Delta$.*

For each $i \in (n]$, let S'_i be the loop of size at most b given by the lemma for the loop S_i. Let $I := \{i_1, \dots, i_\ell\} \subseteq [n]$ be a non-empty set of size at most m such that $|S'_{i_1}| + \cdots + |S'_{i_\ell}| \equiv 0 \,(\mathrm{mod}\, m)$. Such a set exists by a simple application of the pigeonhole principle, because, as a consequence of Lemma 3.11(b), the summed size of the loops $(S_i)_{i\in(n]}$ is 0 modulo m. The next lemma tells us that there exists a tree, obtained from s by a sequence of k-guarded swaps, which contains (disjoint) copies of the loops $(S_i'^{\omega})_{i\in I}$. The proof of the lemma uses Lemma 4 of [2] to include one loop after another into a tree obtained from s by k-guarded swaps (note that these change neither the $(k+1)$-type of the root nor the number of occurrences of $(k+1)$-types in a tree [2]), and then removes intersections between the images of the individual loops under the inclusion mappings by k-guarded swaps.

▶ **Lemma 3.13.** *Let L be a regular tree language closed under k-guarded swaps, for $k \in \mathbb{N}$. Let $(\Delta_i)_{i\in(\ell]}$, for $\ell \in \mathbb{N}_{\geq 1}$, be a sequence of k-abstract loops. For all trees s such that $(\Delta_i)_{i\in(\ell]} <_{k+1} s$, there exists an ℓ-template T such that $T\langle \Delta_1, \dots, \Delta_\ell\rangle$ agrees with s on L, $T\langle \Delta_1, \dots, \Delta_\ell\rangle =_{k+1} s$, and $T\langle \Delta_1, \dots, \Delta_\ell\rangle$ is $(k+1)$-similar to s.*

Recall that we know, by Lemma 3.11 and 3.12, that each $(k+1)$-type occurring in one of the loops $(S'_i)_{i\in(n]}$ occurs more than d times in s. The contexts $(S'_i)_{i\in(n]}$ being k-abstract loops, we do not introduce any *new* $(k+1)$-types when taking their ω-powers. Hence, each $(k+1)$-type of $(S_i'^{\omega})_{i\in I}$ occurs at least d times in s.

We want to apply Lemma 3.13 for $(\Delta_i)_{i\in(\ell]} := (S_i'^{\omega})_{i\in I}$ and s. To do this, we need to make sure that $(S'_i)_{i\in I} <_{k+1} s$. This is assured by taking $d := m\omega b$ as, obviously, there cannot be more occurrences of any particular $(k+1)$-type in $(S'_i)_{i\in I}$ than there are nodes in $(S'_i)_{i\in I}$ altogether. Let T be given by Lemma 3.13. By Lemma 3.10, we know that $T\langle S_{i_1}'^{\omega}, \dots, S_{i_\ell}'^{\omega}\rangle$ agrees with $T\langle S_{i_1}'^{\omega} \cdot S'_{i_1}, \dots, S_{i_\ell}'^{\omega} \cdot S'_{i_\ell}\rangle$ on L. This tree, in turn, agrees with $T\langle S_{i_1}'^{\omega} \cdot S_{i_1}, \dots, S_{i_\ell}'^{\omega} \cdot S_{i_\ell}\rangle$ on L, as each context S'_i is congruent S_i modulo L by Lemma 3.12. By this reasoning, we have added a copy of $(S_i)_{i\in I}$ (and hence, especially, a copy of every $(k+1)$-type therein) to a tree that agrees with s on L.

Now we may apply the same argument successively on the tree just obtained to add the remaining $(k+1)$-types from $\{S_1, \dots, S_n\} \setminus \{S_{i_1}, \dots, S_{i_\ell}\}$. Finally this yields the desired tree s'. This finishes the proof of Lemma 3.8. ◀

4 Decidability

In this section we show that it is possible to decide if a given regular tree language is closed under transfer. Combined with the decidability of closure under guarded swaps (see [2]) and our results from section 3 this implies that $\mathrm{FO}_{\mathrm{card}}$-definability of a given regular tree language is decidable.

▶ **Theorem 4.1.** *It is decidable whether the language L recognised by a given tree automaton $\mathcal{A}$ is closed under transfer.*

Proof. We assume w.l.o.g. that $\mathcal{A}$ is a minimal deterministic tree automaton with state set Q. In order to decide whether L is closed under horizontal transfer, we will check all possible counter examples to this property. If L does *not* posses the closure property, there exists a 2-context C and contexts Δ_1, Δ_2 with $|\Delta_1| = |\Delta_2|$ and trees s_1, s_2 such that the trees $t_1 := C[\Delta_1^{\omega+1} s_1, \Delta_2^{\omega} s_2]$ and $t_2 := C[\Delta_1^{\omega} s_1, \Delta_2^{\omega+1} s_2]$ do *not* agree on L. Let $f_{\Delta_1}, f_{\Delta_2} : Q \to Q$ and $f_C : Q \times Q \to Q$ denote the transition functions induced by $\mathcal{A}$ and, respectively, Δ_1, Δ_2, and C on Q. The trees t_1 and t_2 do not agree on L iff there exist states p_1, p_2 and q_1^+,q_1,q_2, q_2^+ such that: (1) $f_{\Delta_1}^{\omega}(p_1) = q_1$ and $f_{\Delta_1}^{\omega+1}(p_1) = q_1^+$, (2) $f_{\Delta_2}^{\omega}(p_2) = q_2$ and $f_{\Delta_2}^{\omega+1}(p_2) = q_2^+$, (3) $f_C(q_1^+, q_2) \neq f_C(q_1, q_2^+)$.

Let $R \subseteq Q^6$ be a relation such that a tuple of states $\vec{q} := (p_1, p_2, q_1^+, q_1, q_2, q_2^+)$ belongs to R, iff there are contexts Δ_1, Δ_2 with $|\Delta_1| = |\Delta_2|$ satisfying conditions (1) and (2) above.

For all $i \in \mathbb{N}$, let M_i denote the set of transition functions induced by contexts of size i on Q. The set M_i can be computed by simply enumerating all of the (finitely many) contexts of size i and computing the transition function of each such context in turn. Hence, we can recursively enumerate R by iterating through all the sets M_i and comparing the behaviour of the transition functions therein upon all combinations of states. By a pumping argument one sees that there exists a computable bound n such that, if there are contexts Δ_1, Δ_2 witnessing conditions (1) and (2) for a tuple $\vec{q}$, then there have to be such witnesses of size at most n. Hence, R is decidable.

Now we can decide closure under horizontal transfer by checking all possible counter examples: For all 6-tuples $\vec{q}$ as above with $\vec{q} \in R$, we compute all possible transition functions $f : Q \times Q \to Q$ induced by $\mathcal{A}$ and check if $f(q_1^+, q_2) \neq f(q_1, q_2^+)$. If such an f is found, condition (3) is satisfied and L cannot be closed under horizontal transfer. On the other hand, if the check fails for all functions f, we know that L is closed under horizontal transfer.

The decidability of closure under vertical transfer follows using an analogous argument. ◀

Combining Theorem 3.5 with Theorem 4.1 and the decidability of closure under guarded swaps obtained in [2], immediately leads to:

▶ **Corollary 4.2.** *It is decidable whether the language L recognised by a given tree automaton $\mathcal{A}$ is $\mathrm{FO}_{\mathrm{card}}$-definable.*

5 Addition-invariant FO

The set of all first-order formulae that may use the additional binary relation symbol $<$ and a ternary relation symbol $+$ is denoted by $\mathrm{FO}[<,+]$. A $\{<,+\}$-expansion of a tree t is a structure that keeps the interpretation of P_a (for all $a \in \Sigma$) and $\mathrm{S}_1, \mathrm{S}_2$ given by t, and interprets $<$ as a linear order on t and $+$ as the addition relation induced by $<$. A $\mathrm{FO}[<,+]$-formula φ is *addition-invariant*, if for all $\{<,+\}$-expansions of s, s' of a tree, $s \models \varphi$ iff $s' \models \varphi$. Let $+$–inv–FO denote the set of addition-invariant formulae. This section's main result is the following theorem, generalising a result of [9] from words to trees:

▶ **Theorem 5.1.** *Let L be a regular tree language. The following statements are equivalent: (1) L is $+$–inv–FO-definable, (2) L is closed under transfer and guarded swaps, (3) L is FO_{card}-definable.*

The equivalence of statements (3) and (2) was proved in Theorem 3.5. It is easily seen that any regular tree language definable by an FO_{card}-sentence φ is definable by an $+$–inv–FO-sentence. For example, the following $+$–inv–FO-sentence defines $C_{1,2}$ (where we assume that the least element with respect to $<$ has index 0):

$$\exists x\, \exists z\, \big(z = x + x \ \wedge\ \neg\exists y\, (y < z)\big).$$

For the remainder of this section, we will be occupied by the proof that $+$–inv–FO-definability implies closure under guarded swaps and transfer. The following proofs make extensive use of *first-order interpretations*; see e.g. [5] for an exposition of this technique.

Closure under guarded swaps. To prove the closure of $+$–inv–FO-definable regular tree languages under guarded swaps, we use the following Lemma 5.2, which is an immediate consequence of [8, Proposition 6.11] (which lies at the heart of the results from [9], too). To state the lemma, we need the following notations: Let σ be a relational signature. For each σ-structure A, we write σ^A for the set of relations of A. Let A, B be σ-structures. Let α be a mapping from the universe of A to the universe of B. For a relation $R \in \sigma^A$ of arity m, we define $\alpha(R) := \{(\alpha(a_1), \ldots, \alpha(a_m)) : (a_1, \ldots, a_m) \in R\}$. For $\sigma^A = \{R_1, \ldots, R_n\}$, let $\alpha(\sigma^A) := \{\alpha(R_1), \ldots, \alpha(R_n)\}$. We write $A \approx_q B$ to indicate that A and B satisfy the same first-order-sentences of quantifier depth at most q.

▶ **Lemma 5.2** ([8]). *Let $q'', h, e \in \mathbb{N}_{\geq 1}$ and let σ be a signature. There exists an infinite set $P := \{p_1 < p_2 < p_3 \ldots\} \subseteq \mathbb{N}$ with $p_1 > h$ and $p_i \equiv h \,(\text{mod}\, e)$, for all $i \in \mathbb{N}_{\geq 1}$, and a number q' such that the following is true for all finite σ-structures M and all linear orders $<_1$ and $<_2$ on M's universe: if $\langle M, <_1\rangle \approx_{q'} \langle M, <_2\rangle$, then $\langle \mathbb{Z}, +, P, \alpha_1(\sigma^M)\rangle \approx_{q''} \langle \mathbb{Z}, +, P, \alpha_2(\sigma^M)\rangle$, where α_i is a map taking the j-th node of M according to $<_i$ to p_j, for $i \in (2]$ and $j \in \mathbb{N}_{\geq 1}$.*

The second ingredient to our proof of the closure of $+$–inv–FO-definable tree languages under guarded swaps is a lemma of [6] which was used in [3] to prove closure under guarded swaps of order-invariantly definable tree languages:

▶ **Lemma 5.3** (implicit in [6]). *Let $x, q' \in \mathbb{N}$, and let σ be a signature. There exists $k' \in \mathbb{N}$ such that for each finite σ-structure M and all x-tuples $\bar{a}$ and $\bar{b}$ of M with isomorphic k'-neighbourhoods, there exist linear orders $<_1$ and $<_2$ of the universe of M, whose initial elements are respectively $\bar{a}\bar{b}$ and $\bar{b}\bar{a}$, such that $\langle M, <_1\rangle \approx_{q'} \langle M, <_2\rangle$.*

We use the lemmas 5.2 and 5.3 together with an interpretation argument, to prove:

▶ **Lemma 5.4.** *Let L be a regular tree language. If L is definable by an $+$–inv–FO-sentence, then L is closed under guarded horizontal swaps.*

Proof sketch. Let φ be an $+$–inv–FO-sentence defining L. Let Q be the state set of a minimal deterministic tree automaton recognising L. We want to show that L is closed under k-guarded horizontal swaps, for a $k \in \mathbb{N}$ that will be fixed later on. Consider a tree t with incomparable k-similar nodes u and v. Let $t_1 := t_{|u}$ and $t_2 := t_{|v}$, i.e. $t = C[t_1, t_2]$ for a 2-context C. Let $t' := C[t_2, t_1]$. We may assume that the trees t_1 and t_2 have height at least k. Taking $k > \kappa_L k' + |Q^Q|$ (with κ_L as defined at the end of section 2), where k' will be fixed later on by our application of Lemma 5.3, a standard pumping argument shows that we may assume that $t_1 = DEt'_1$, for a tree t'_1 and contexts E, D such that E is idempotent, and D is $\kappa_L k'$-similar to t_2. Let $e := |E|$. Without loss of generality, $e \leq \kappa_L$ (if not, we can replace E by a congruent context of that size) and $|t'_1| \geq e$ (if not, we can prepend a copy of E to it).

For $i \in \{1, 2\}$, we decompose t_i into *blocks* of size e, plus a *residual block* of size $n_i := |t_i| \text{ MOD } e$, if $|t_i|$ is not divisible by e: A block consists of e consecutive nodes of t_i, ordered according to the *bf-order* of the tree (cf., section 2). We let M be a structure using the set of blocks of size e of t_1 and t_2 as universe, with relations which encode the following information about t_1 and t_2: the successor relations between the nodes of the different blocks (resp., between the blocks and the residual blocks not in M), the position of E, and the labels of the nodes in each block. Let b_1 and b_2 be the blocks containing the roots of t_1 and t_2, respectively. Since t_1 and t_2 are k-similar, the k/e-neighbourhoods of b_1 and b_2 in M are isomorphic. We let k' be given by Lemma 5.3 for $x := 1$ and a q' to be fixed later on. By our choice of k we have $k' \leq k/e \leq k/\kappa_L$. By Lemma 5.3 we obtain two linear orders $<_1$ and $<_2$ on M such that, according to $<_1$, b_1 comes first and b_2 comes second, and according to $<_2$ it is just the other way round. Lemma 5.3 guarantees that $\langle M, <_1\rangle$ and $\langle M, <_2\rangle$ agree on all FO$[<, +]$-sentences of quantifier depth $\leq q'$. Thus, by Lemma 5.2, we obtain structures M_1 and M_2 over the integers which contain "stretched copies" of $\langle M, <_1\rangle$ and $\langle M, <_2\rangle$, respectively. I.e. some elements of M_1 and M_2, marked by a unary predicate P, correspond to the original structures, and other positions in between do not. The number h of Lemma 5.2 is set to be $|C| + n_1 + n_2$, and q'' will be fixed later on. We choose q' as given by Lemma 5.2 and obtain that M_1 and M_2 agree on all FO$[<, +]$-sentences of quantifier depth at most q''.

Now we specify an FO$[<, +]$-interpretation that transforms M_1 into a tree agreeing with t on L, and M_2 into a tree agreeing with t' on L: The set of nodes of the trees consist of all non-negative integers before the least position in P that is not included in any "stretched relation". The successor relations and labels of the first e nodes starting at a node p where P holds are interpreted in such a way that t_1 and t_2 are simulated on these nodes; for this, the "stretched copies" of the relations from $\langle M, <_1\rangle$ resp. $\langle M, <_2\rangle$ are used. The nodes between $p + e$ and the next number p' in P, are interpreted as copies of the idempotent context E; the same is done for the nodes between the positions h and $p_1 - 1$. All these copies of E are inserted at the original position of E in the simulated tree t_1. In the first h nodes, the (inner tree of) the 2-context C and the two residual blocks of size n_1 and n_2 are simulated. The simulated parent of the first hole of C is linked to the node that is simulated at the first position in P; the parent of the second hole of C is linked to the node at the second position in P. This way, for $\widetilde{t_1} := DE^i t'_1$, for a suitable $i \in \mathbb{N}_{\geq 1}$, the interpretation turns M_1 into the tree $C[\widetilde{t_1}, t_2]$ (which, as E is idempotent, agrees with t on L), and M_2 into $C[t_2, \widetilde{t_1}]$ (which agrees with t' on L). By choosing q'' larger than the sum of the maximal quantifier depth of the formulae of this interpretation, and the quantifier depth of the formula φ defining the language L, we ensure that t and t' agree on φ, finishing the proof. ◀

Our next goal is to prove that every +–inv–FO-definable regular tree language is closed under guarded vertical swaps as well. To achieve this, we first prove the closure under a variant of guarded vertical swaps, where the guardedness assumptions are somewhat strengthened: A language L is said to be *closed under strongly-k-guarded* vertical swaps, for $k \in \mathbb{N}$, if each tree t containing nodes $u \lhd v \lhd u' \lhd v'$, such that u and u' are k-similar, v and v' have *isomorphic k-neighbourhoods*, and the k-neighbourhoods of v and v' and k-spills of u and u' are all mutually disjoint, agrees on L with its vertical swap between $t[u, v)$ and $t[u', v')$.

▶ **Lemma 5.5.** *Let L be a regular tree language. If L is* $+$–inv–FO*-definable, then L is closed under strongly-k-guarded vertical swaps, for some $k \in \mathbb{N}$.*

The proof of Lemma 5.5 proceeds similarly to the proof of Lemma 5.4 (the *strongly*-guarded swaps being necessary to apply Lemma 5.3). We continue by showing that being closed under strongly-guarded vertical swaps is actually equivalent to being closed under guarded vertical swaps, if the language under consideration is closed under guarded horizontal swaps, too.

▶ **Lemma 5.6.** *Let L be a tree language. L is closed under guarded swaps iff it is closed under strongly-guarded vertical swaps and guarded horizontal swaps.*

Proof idea. Closure under guarded swaps immediately implies closure under strongly-guarded swaps and guarded horizontal swaps. Let L be a tree language that is closed under strongly-k'-guarded vertical swaps and k'-guarded horizontal swaps. We show that L is closed under k-guarded vertical swaps, for a suitable number $k > k'$. Let $t := C\Delta_1\Delta\Delta_2 s$ be given as in the definition of vertical guarded-swaps, and let $t' := C\Delta_2\Delta\Delta_1 s$. The proof proceeds by distinguishing cases depending on the root-hole-distance of the contexts Δ_1, Δ, Δ_2. We show how to find nodes $\tilde{u}$ and $\tilde{u}'$ in the k-spills of the root of Δ_1 resp. Δ_2, and $\tilde{v}$ and $\tilde{v}'$ in the k-spills of the hole of Δ_1 resp. Δ_2 in t, fulfilling the preconditions for a strongly-k'-guarded vertical swap between $t[\tilde{u}, \tilde{v})$ and $t[\tilde{u}', \tilde{v}')$. After this swap, we have either swapped "too much" or "not enough" of Δ_1 and Δ_2. In these cases, we are able to repair the remaining parts by a series of k'-guarded horizontal swaps between the (incomparable) nodes "around" the nodes swapped in the strongly-guarded vertical swap. ◀

Closure under transfer. We now show that a regular tree language definable by an $+$–inv–FO-sentence is closed under transfer. This is the easier part of the proof of Theorem 5.1, as we are able to build directly on results proved in [9]. To state the according result, we need some further notation: Given words $w, x \in \Sigma^*$, with $|w| \geq |x|$, we denote by $|w|_x$ the number of non-overlapping occurrences of x as a factor in w. Furthermore we say that a sentence φ *separates* languages L_1 and L_2 iff every word in L_1, but no word in L_2, satisfies φ.

▶ **Lemma 5.7** (Proposition 3.3 of [9])**.** *Let $n \in \mathbb{N}$ with $n \geq 2$, $y \in \Sigma^*$, $\bar{x} \in (\Sigma \times \{1\})^*$ and $\bar{z} \in (\Sigma \times \{2\})^*$. For all $a, b \in \mathbb{N}$, let*

$$L_{n,a,b} \ := \ \{w \in y\bar{x}(\bar{x}\bar{z}|\bar{z}\bar{z})^* : |w|_{\bar{x}}, |w|_{\bar{z}} \geq n, |w|_{\bar{x}} \equiv a \,(\mathrm{mod}\, n), |w|_{\bar{z}} \equiv b \,(\mathrm{mod}\, n)\}.$$

There exists no $\mathrm{FO}[<,+]$*-sentence that separates $L_{n,1,0}$ from $L_{n,0,1}$.*

We use Lemma 5.7 and proceed with a proof by contradiction: If a regular tree language L is not closed under transfer, we show by an interpretation argument that there exists a $\mathrm{FO}[<,+]$-sentence separating a suitable word language $L_{n,1,0}$ from a word language $L_{n,0,1}$ for $n := \omega_L$. This is akin to what is done in [9] to prove that every regular *word* language is closed under transfer, the difference being that we have to simulate trees in words.

▶ **Lemma 5.8.** *Let L be a regular tree language. If L is definable by an* $+$–inv–FO*-sentence, then L is closed under transfer.*

Proof sketch. Assume that L is not closed under transfer. This means, there are contexts Δ_1, Δ_2 with $|\Delta_1| = |\Delta_2|$ and a 2-template T, such that $t := T\langle\Delta_1^{\omega+1}, \Delta_2^{\omega}\rangle \in L$, but $t' := T\langle\Delta_1^{\omega}, \Delta_2^{\omega+1}\rangle \notin L$. Let $t_{i,j} := T\langle\Delta_1^{i}, \Delta_2^{j}\rangle$, for all $i, j \in \mathbb{N}_{\geq 1}$. Because Δ_1^{ω} and Δ_2^{ω} are idempotent, we may repeat both contexts in the trees t and t' without affecting membership in L. Hence, for all $i, j \geq \omega$, (1) if $i \equiv 1 \,(\mathrm{mod}\, \omega), j \equiv 0 \,(\mathrm{mod}\, \omega)$, then $t_{i,j} \in L$, and (2) if $i \equiv 0 \,(\mathrm{mod}\, \omega), j \equiv 1 \,(\mathrm{mod}\, \omega)$, then $t_{i,j} \notin L$. As we are aiming at a contradiction to Lemma 5.7, we fix the numbers and words therein: We take n to be ω. To each tree we assign the word of its labels, ordered according to the bf-order on the nodes of the tree. Let y be the word obtained from the template T in that way. Let x and z be the words obtained from the inner tree of the contexts Δ_1 and Δ_2, respectively. Let $\bar{x}$ be the word x with each symbol $a \in \Sigma$ that occurs in x replaced by the tuple $(a, 1)$, and let $\bar{z}$ be obtained from z by tagging each symbol of z accordingly by 2. Let $L_T := \{t_{i,j} : i, j \in \mathbb{N}\}$. We define an FO$[<, +]$-interpretation that interprets trees from L_T in words from the language $y\bar{x}(\bar{x}\bar{z}|\bar{z}\bar{z})^*$ such that, given a word $w \in y\bar{x}(\bar{x}\bar{z}|\bar{z}\bar{z})^*$ with $i := |w|_{\bar{x}}$ and $j := |w|_{\bar{z}}$, this interpretation constructs the tree $t_{i,j}$. It is possible to do this by FO$[<, +]$-formulae, because Δ_1, Δ_2, and T are fixed, and we can use the tags 1 and 2 in the words $\bar{x}$ and $\bar{z}$ to identify the positions in a word where subwords corresponding to Δ_1 resp. Δ_2 start. Now consider the languages $L_{n,1,0}$ and $L_{n,0,1}$ (for $n := \omega$) of Lemma 5.7: If $w \in L_{n,1,0}$, then, by (1), $t_{i,j} \in L$. On the other hand, if $w \in L_{n,0,1}$, then $t_{i,j} \notin L$. Let φ be the $+$–inv–FO-sentence defining L. We alter φ according to our interpretation to obtain an FO$[<, +]$-sentence φ'. By the addition-invariance of φ, the choice of the addition relation (here, the one induced by the linear order on the word) is immaterial for the satisfaction of φ by $t_{i,j}$. Therefore, $w \models \varphi'$ iff $t_{i,j} \models \varphi$ iff $t_{i,j} \in L$. Thus, φ' separates $L_{n,0,1}$ from $L_{n,1,0}$, contradicting Lemma 5.7 and finishing the proof of Lemma 5.8. ◀

From the lemmas 5.8, 5.4, 5.5, 5.6, we obtain that every $+$–inv–FO-definable regular tree language is closed under transfer and guarded swaps, concluding the proof of Theorem 5.1. ◀

Acknowledgement. We thank Luc Segoufin for helpful discussions on the subject of this paper.

References

1. D. Beauquier and J.-E. Pin. Factors of words. In *Proc. ICALP'89*, volume 372 of *Lecture Notes in Computer Science*, pages 63–79. Springer-Verlag, 1989.
2. M. Benedikt and L. Segoufin. Regular tree languages definable in FO and in FO$_{mod}$. *ACM Transactions on Computational Logic*, 11(1), 2009.
3. M. Benedikt and L. Segoufin. Towards a characterization of order-invariant queries over tame graphs. *Journal of Symbolic Logic*, 74(1):168–186, 2009.
4. H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available at `http://tata.gforge.inria.fr/`. Release: October, 12th 2007.
5. H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer-Verlag, 2nd edition, 1999.
6. M. Grohe and T. Schwentick. Locality of order-invariant first-order formulas. *ACM Transactions on Computational Logic*, 1(1):112–130, 2000.
7. L. Libkin. *Elements of Finite Model Theory.* Springer-Verlag, 2004.
8. N. Schweikardt. An Ehrenfeucht-Fraïssé game approach to collapse results in database theory. *Information and Computation*, 205(3):311–379, 2007.
9. N. Schweikardt and L. Segoufin. Addition-invariant FO and regularity. In *Proc. 25th IEEE Symposium on Logic in Computer Science (LICS'10)*, pages 285–294. IEEE, 2010.
10. M.P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.
11. H. Straubing. *Finite automata, formal logic, and circuit complexity.* Birkäuser, 1994.