TOWARDS A CHARACTERIZATION OF ORDER-INVARIANT QUERIES OVER TAME GRAPHS

MICHAEL BENEDIKT AND LUC SEGOUFIN

Abstract. This work deals with the expressive power of logics on finite graphs with access to an additional "arbitrary" linear order. The queries that can be expressed this way are the *order-invariant queries* for the logic. For the standard logics used in computer science, such as first-order logic, it is known that access to an arbitrary linear order increases the expressiveness of the logic. However, when we look at the separating examples, we find that they have satisfying models whose Gaifman Graph is complex – unbounded in valence and in treewidth. We thus explore the expressiveness of order-invariant queries over well-behaved graphs. We prove that first-order order-invariant queries over strings and trees have no additional expressiveness over first-order logic in the original signature. We also prove new upper bounds on order-invariant queries over bounded treewidth and bounded valence graphs. Our results make use of a new technique of independent interest: the application of algebraic characterizations of definability to show collapse results.

§1. Introduction. In classical finite model theory [15, 9], a logic $\mathcal{L}(\sigma)$ for models over the relational signature σ, associates words of a grammar (the syntax) to relations of the model (the semantics). One generally requires that the logic is *closed under isomorphisms*: that is if A and B are finite models over σ and h is an isomorphism between A and B then for all $q \in \mathcal{L}(\sigma)$, $q \circ h$ and $h \circ q$ give the same answer. This is the case for all standard logics: first-order logic, monadic second-order logic, fixed point logic etc.

In practice, for instance in the database context where logics correspond to query languages, one can refer in the syntax to a predicate which is not necessarily in the signature σ of the input: a linear order which corresponds to the order in which the elements of the universe are stored on disk. Sentences of the query language can then make use of this predicate to perform operations over all elements of the universe separately. We reflect this by denoting $\sigma_<$ the extension of σ with an additional binary relation symbol $<$ which is assumed to be interpreted by a linear order over the domain, then one actually has $\mathcal{L}(\sigma_<)$ available instead of $\mathcal{L}(\sigma)$.

Of course it is preferable to restrict the use that a query language can make of the extra predicate $<$. One would not wish to allow queries that use $<$ in order to return the smallest, according to $<$, element of the universe; the answer would depend on how the universe is stored on disk, which in turn may vary with time (depending, for instance, on the presence of indexes). To be *meaningful* a formula in $\mathcal{L}(\sigma_<)$ should be closed under isomorphism. A sentence $\phi \in \mathcal{L}(\sigma_<)$ is closed under isomorphisms iff it is *order-invariant*: For every finite σ-structure A, for every two expansions A_1 and A_2 of A to an $\sigma_<$-structure in which $<$ is interpreted by a linear order, $A_1 \models \phi \leftrightarrow A_2 \models \phi$. We denote by Inv-$\mathcal{L}(\sigma_<)$ (Inv-$\mathcal{L}(<)$ if σ is understood) the fragment of $\mathcal{L}(\sigma_<)$ ($\mathcal{L}(<)$ if σ is understood) containing all order-invariant sentences.

There are two questions that immediately arise. The first one is whether there exists an effective syntax for Inv-$\mathcal{L}(<)$. That is whether there exists a logic $\mathcal{L}'$ with an effective syntax and the same expressive power as Inv-$\mathcal{L}(<)$. The second one is finding the expressive power of Inv-$\mathcal{L}(<)$, in particular whether it is strictly more that $\mathcal{L}$ or not (the converse inclusion being obvious). The questions are connected, since if Inv-$\mathcal{L}(<)$ is no more expressive than $\mathcal{L}$ then Inv-$\mathcal{L}(<)$ has effective syntax.

These questions were first considered in the case of fixed-point logics. Indeed a rephrasing of the Immerman-Vardi Theorem (see e.g. [15]) says that Inv-IFP$(<)$ is PTIME and that Inv-FP$(<)$ is PSPACE. Here IFP stands for the inflationary fixed-point semantics while FP is the non-inflationary semantics. Note that this immediately implies that Inv-IFP$(<)$ (resp. Inv-FP$(<)$) is strictly more expressive than IFP (FP) as IFP (FP) fails to express all of PTIME (PSPACE). The question of the existence of a logic with effective syntax for Inv-IFP$(<)$ is open: From the result above this is identical to the question whether there is a logic for PTIME, a longstanding open question in finite model theory [1].

Another example is Monadic Second Order Logic (MSO). It is easy to see that Inv-MSO$(<)$ allows one to express every query in the extension of MSO with counting quantifiers (CMSO), which is strictly more expressive than MSO.

The example above shows that access to an arbitrary ordering increases expressiveness when one deals with powerful logics that can express recursive operators. What about weaker logics, such as first-order logic (FO)? A famous example due to Gurevich (see exercise 17.27 in [2]) shows that Inv-FO$(<)$ is more expressive than FO for any σ including at least one binary predicate. Extensions of this result due to Otto [18] give examples of Inv-FO$(<)$ sentences that are quite complex: in particular, Otto shows that there are Inv-FO$(<)$ sentences not expressible in infinitary logic formed over first-order logic with a bounded number of variables and quantifiers of the form $\exists^{!i} x\ \phi(x, \vec{y})$. However, the example queries of Gurevich and Otto each have satisfying models that include a binary predicate which becomes graph-theoretically very complex as the models vary. Thus it is natural to conjecture that if one restricts the structures to be well-behaved, Inv-FO$(<)$ cannot express complex queries.

In this paper we investigate the expressiveness of Inv-FO$(<)$ sentences over well-behaved graphs, specifically graphs of bounded treewidth and graphs of bounded valence. We will show that Inv-FO$(<)$ collapses to FO on trees. We then show that Inv-FO$(<)$ collapses to MSO on graphs of bounded valence and on graphs of bounded treewidth.

We use algebraic tools in order to analyze order-invariant queries. Suppose one wants to show bound of the form Inv-$\mathcal{L}(<) \subseteq \mathcal{L}'$ for some logic $\mathcal{L}'$. The most common method (e.g. [12]) is to show that two sufficiently $\mathcal{L}'$-equivalent structures can be ordered so that they agree on $\mathcal{L}$ sentences of a given quantifier rank. But ordering two such arbitrary equivalent structures is difficult. In addition, since ordering structures is a priori stronger than what is required, this technique may not be sufficient to show tight bounds. We show that algebraic characterizations of first-order definability, such as those in [3, 4], can be utilized to give new results on collapse of order-invariant queries over logics on restricted graphs. These characterization theorems show that if a query is not definable in the logic without the order, then there are witness graphs that are similar and of a very restricted form which cannot be distinguished by the query. Only these special witness graphs need to be ordered, and the restricted form of these graphs makes the ordering arguments much more tractable.

Related Work. Following the initial example of Gurevich, order-invariant queries over first-order logic were investigated in [18, 12, 19]. It is clear that Inv-FO($<$) queries are expressible in both existential second order (ESO) and universal second order (USO) logic, and also that they are computable in NLOGSPACE. To our knowledge, there is no result giving containment of Inv-FO($<$) queries in a sublogic of ESO $\cap$ USO. While Otto [18] shows that there are Inv-FO($<$) queries that are not in infinitary logic with counting quantifiers, Grohe and Schwentick [12] show that all Inv-FO($<$) open formulas are local (cannot distinguish points with similar local neighborhoods). Rossman [19] shows there are Inv-FO($<$) queries that are not first-order which only make use of the successor relation in the order. All of the published examples of Inv-FO($<$) sentences are expressible in CMSO.

Courcelle [8] studies order-invariant MSO queries, showing that over trees, Inv-MSO($<$) has exactly the same expressiveness as CMSO. The results of Lapoire [14] combined with those of Courcelle [6, 7] show that the same equality holds for graphs of bounded treewidth. It was proved recently that the inclusion of Inv-MSO($<$) in CMSO is strict over arbitrary graphs [11].

The algebraic tools we use here derives from our previous work [4]. However, the bounds we give on Inv-FO($<$) queries on trees have been announced independently by Niemistö [17]. The result of [17] relies on the locality of Inv-FO($<$) proved in [12], while the technique given here relies only on a Ramsey-like lemma, also proved in [12], and may be applicable to logics that are not local.

Note that the classical Craig Interpolation Theorem [5] implies that a first-order query that is invariant over *all* structures must be in FO. The interpolation theorem is known not to hold over finite structures (even for trees) [9]. Our results can be seen as showing that one can reclaim some consequences of interpolation by restricting to well-behaved classes of structures.

Organization: Section 2 gives the basic definitions of the paper and reviews results about regular languages that will be used here. Section 3 gives characterizations of invariant FO queries on strings. Section 4 extends these characterizations to trees. Section 5 gives bounds for graphs of bounded valence and bounded treewidth. Section 6 gives conclusions.

§2. Background.

By a *query*, we refer to any boolean function on finite relational structures in some vocabulary σ. We will generally have σ consist of at least a binary relation S, and Σ a finite set of unary predicates. We will refer to structures for such a σ as colored graphs. Given some class C of colored graphs (strings, trees, etc.) a query ϕ over the signature $\sigma_< = \sigma \cup \{<\}$ is *order-invariant over C* if for every finite σ-structure $G \in C$, for every two expansions G_1 and G_2 of G to a $\sigma_<$ structure, $G_1 \models \phi \leftrightarrow G_2 \models \phi$. Such queries can clearly also be considered as queries over σ. The queries we consider here will always be defined by logics (FO, Inv-FO(<), etc.). Two logics are said to have the same expressiveness (over class C) if the set of queries they define (resp. set of restrictions of queries to domain C) are the same.

Our definition of first-order logic (FO) and Monadic Second Order Logic (MSO) over a vocabulary σ is standard. We will sometimes abuse notation and refer to an "Inv-FO(<) query over C", to mean an FO($\sigma_<$) query that is order-invariant over C, and similarly for other logics. The logic CMSO is the extension of MSO with a predicate counting the cardinality of sets modulo some integer. For $\mathcal{L}$ any one of MSO, FO we write $G \equiv_r^{\mathcal{L}} G'$ if G and G' agree on $\mathcal{L}$ sentences of quantifier rank at most r. We assume familiarity with Ehrenfeucht-Fraïssé games (see e.g. [15]), which characterize $\equiv_r^{\mathrm{FO}}$.

The most restricted set of structures we consider are strings. Let σ^S consist of exactly $\Sigma \cup \{S\}$. We will use the terms string and word interchangeably to mean any σ^S-structure in which the domain with S is isomorphic to an initial segment of the integers with the successor relation. We will also assume (as part of the definition of string and word) that every element in the structure satisfies exactly one of the predicates in Σ: this assumption is only to simplify the presentation. The set of strings over a fixed Σ as above will be denoted Σ^*.

By a tree we mean a connected directed graph S that is acyclic, where every element has at most one predecessor and has a single root. We will also sometimes use "tree" to mean an expansion of a tree in the above sense by a set of unary predicates in Σ. Let $\sigma^{S,S'}$ be the signature extending σ^S with a new binary predicate S'. A sibling ordering on a tree is any binary relation that compares only elements with a common parent, and which is a linear order on the set of children of any node. By a "siblinged tree" we mean a $\sigma^{S,S'}$-structure, where S is a tree and S' is the successor relation corresponding to some sibling ordering. We distinguish between the set of unordered ranked trees RT_n for $n \in N$, (where n is the bound on the number of children of any node), siblinged ranked trees SRT_n (which we consider as siblinged trees where there is a bound on the number of children) , unranked trees UT, and siblinged unranked trees SUT. For any of these domains D, a collection $C \subseteq D$ is regular if it is MSO definable over the appropriate vocabulary.

A *language* is a set of strings or a set of trees. Any query over strings or trees defines a language, the set of string or trees satisfying the formula. If Q is a query, we denote by $L(Q)$ the corresponding language.

Note that in this work, when we consider queries over strings or unordered trees, we deal with sentences that contain only the successor-relation or parent/child relation, unary predicates for the labels, and the additional order-invariant binary predicate. We do not consider queries that include also a symbol for the transitive-closure of the successor or parent/child relation. The problems that we consider become very simple when the transitive closure is available, since then a linear order is already definable in the structure. It is easy to see that for any class of structures that already contains a definable linear-ordering, the ability to use an additional order-invariant ordering symbol in a first-order sentence adds no expressive power; one can simply replace the additional predicate by the definition of the linear order. In the case of siblinged unranked trees, we likewise do not allow the sibling relation itself, only the local successor relation. For order-invariant first-order queries over trees in the vocabulary that includes a sibling relation, the only bound we know of is MSO.

§3. Order-Invariant Queries on Strings. We first deal with the decidability of membership in Inv-FO($<$). If one could decide membership in Inv-FO($<$), one would immediately have an effective syntax for Inv-FO($<$) queries. However, it is well-known [2] that one cannot decide whether or not an FO($<$) query is order-invariant: this follows easily from the undecidability of the satisfiability problem for first-order logic. Over strings, satisfiability is decidable, hence it is a priori feasible that membership in Inv-FO($<$) over strings is decidable. We show that this is not the case:

PROPOSITION 3.1. *The problem of deciding, given a sentence $\phi \in$ FO($<$), whether or not it is order-invariant over Σ^*, is undecidable.*

PROOF. Let ϕ_0 be the formula of FO($<$), $\forall x[(\forall y\, y \leq x) \rightarrow (\exists y\, S(x, y))]$, which expresses the fact that the last element relative to $\leq$ has a successor relative to S. This formula is obviously not order-invariant, even when we restrict to structures with a fixed domain.

Consider now the function that takes a $\phi \in$ FO($<$) and returns the conjunction of ϕ with ϕ_0. This function reduces membership in the complement of the order-invariant queries to satisfiability of an FO($<$) sentence over expansions of structures in Σ^* by a linear order, and hence it suffices to show that this satisfiability problem is undecidable.

This is done by using the two available successors, the one given by the string structure and the one induced by the linear order, for coding a grid. This grid is then used to code a run of a Turing Machine in a classical way.

More precisely, from an input-free Turing machine M we construct a FO($<$) formula φ_M such that M halts iff φ_M has a model in Σ^*. Assume WLOG that all Turing machines work over the binary alphabet $\{0, 1\}$. Let Σ consist of the unary predicates $P_0, P_1, P_\sharp, P_\Box$. We consider strings from this alphabet.

A configuration c of a Turing machine using memory of size k can be described using a string of length k, where unused cells are colored with $\Box$. Therefore a string of the form $\sharp c_1 \sharp c_2 \cdots \sharp c_n$ can code a set of configurations of a Turing Machine. We want to use S and $<$ in order to show that such a word codes a run of a particular Turing Machine.

To do this we restrict the linear orders considered as follows. Let succ$_<$ be the successor relation corresponding to $<$ (note that succ$_<$ is definable in FO from $<$).

- All nodes labeled by $\sharp$ come first in the ordering $<$:
  $\forall x, y\ P_\sharp(x) \wedge \neg P_\sharp(y) \longrightarrow x < y$
- The successor of the last (according to $<$) node labeled by $\sharp$ is the successor (according to S) of the first (according to $<$) node labeled by $\sharp$:
  $\forall x, y, z, u\ \big[(P_\sharp(x) \wedge P_\sharp(u) \longrightarrow u < x) \wedge \mathrm{succ}_<(x, y) \wedge S(z, y)\big] \longrightarrow (\forall u\ P_\sharp(z) \wedge P_\sharp(u) \longrightarrow z < u)$
- The remaining nodes are ordered so that the following property holds: $\forall x, y, u, v\ \neg P_\sharp(x) \wedge S(y, x) \wedge \mathrm{succ}_<(y, u) \wedge S(u, v) \longrightarrow \mathrm{succ}_<(x, v)$.

3	6	9	12	1	4	7	10	2	5	8	11
♯	a	b	c	♯	d	e	f	♯	g	h	i

FIGURE 1. A model (bottom line) with a possible order $<$ (top line).

In words, this says that once the order on the symbols $\sharp$ is fixed, then $<$ is completely defined by induction on strings of the form $\sharp c_1 \sharp c_2 \cdots \sharp c_n$. The order on $\sharp$ symbols induces an order $\prec$ on the $(c_i)_{1 \leq i \leq n}$. From this ordering we derive $<$, which orders the remaining symbols lexicographically based first on their position in one of the c_i, using $\prec$ to break ties. Note that the property given above is definable in FO($<$) by a formula that we denote by $\psi_<$. One can verify that all strings that are models of $\psi_<$ are of the form $\sharp c_1 \sharp c_2 \cdots \sharp c_n$ where the size of each c_i is the same. An example of such a model is given in Figure 1. Given a model of $\psi_<$, let $\alpha_<$ be the bijection of $\{1, \cdots, n\}$ such that $\alpha_<(i) = j$ where c_j is the string following the i^{th} symbol $\sharp$ according to $<$. Each model of $\psi_<$ can thus be seen as a sequence of configurations $c_{\alpha(1)} \cdots c_{\alpha(n)}$.

We now fix M and construct a formula ψ_M which, assuming $\psi_<$, checks that the sequence $c_{\alpha(1)} \cdots c_{\alpha(n)}$ is an accepting run of M. For this notice that $\mathrm{succ}_<$ associates cells located at the same place on the tape of M and at two successive steps of M. Using this relation it is a classical technique (see e.g. Chap. 9 in [15]) to code in first-order logic the fact that two successive configurations are valid according to M.

From the discussion above it is now easy to see that the formula $\varphi_M = \psi_M \wedge \psi_<$ has the desired property. $\dashv$

The proof technique can easily be modified to show undecidability for the other classes considered in this paper.

We now turn to the expressiveness of Inv-FO($<$) over strings. We show that this is as low as it can possibly be (recall that every FO query is in Inv-FO($<$)). This result will follow from our results on trees in Section 4. We prove this directly here for two reasons. First, it exhibits the main technique of the paper within a simple framework. Second, it uses a simple argument while the proof of the tree cases uses a technical lemma of [12].

THEOREM 3.2. Inv-FO($<$) = FO *over strings.*

The proof of Theorem 3.2 is based on an algebraic characterization of FO within the set of regular languages. Before stating this result we first review some of the known connections between definability and algebraic properties of string languages. All the results below can be found in [20]. The MSO(σ^S) sentences define exactly regular languages. With any language L, one can associate the equivalence relation $\equiv_L$ on Σ^*: $x \equiv_L y \leftrightarrow (\forall u \in \Sigma^* \; \forall v \in \Sigma^* \; uxv \in L \leftrightarrow uyv \in L)$. Regularity of L is equivalent to the fact that the set of equivalence classes is finite. The set of classes of $\equiv_L$ equipped with the concatenation operation forms a monoid, called the *syntactic monoid* of L, denoted η_L. An element e of the syntactic monoid is an *idempotent* if $e^2 = e$. We can consider a word to be idempotent if its class is; translating the above, we have that a word e is idempotent iff $uev \in L \leftrightarrow ue^2v \in L$. The theorems of MacNaughton and Papert and of Schützenberger characterize when a regular language is definable in first-order logic over σ^S augmented with an additional binary predicate $\prec$, where $\prec$ is interpreted as the transitive closure of S: this occurs exactly when the syntactic monoid of L is *aperiodic* (e.g. see [20] Theorem VI.1.1): there is $l \in \mathbb{N}$ such that the monoid satisfies $\forall m \in \eta_L \; m^l = m^{l+1}$. Translated back to words, this means that $\forall u, v, w \in \Sigma^* \; uv^l w \in L \leftrightarrow uv^{l+1}w \in L$.

The logic FO only has the successor relation and is transitive closure is not definable in FO. Hence we will use the following theorem of Beauquier and Pin that characterizes when a regular language L is FO definable (see also [20], Thm VI.3.1):

Theorem 3.3 ([3]). *A regular language L is* FO *definable iff its syntactic monoid is aperiodic, and additionally for any $e, f, u, v, w \in \eta_L$ with e, f idempotent, $eufvewf = ewfveuf$.*

Proof of Theorem 3.2. Let $\phi \in$ Inv-FO($<$). Let M be a string model of ϕ. Then M is of the form $(\omega_M, <_M)$ where ω_M is a string and $<_M$ a linear order on the universe of ω_M. Let $L(\phi) = \{\omega_M \mid M \models \phi\}$. A model M is *obvious* if $<_M$ is exactly the transitive closure of the successor relation in ω_M. Let $L'(\phi) = \{\omega_M \mid M \models \phi \text{ and } M \text{ is obvious}\}$. Because $L'(\phi)$ is definable in FO($<$), it is regular and its syntactic monoid is aperiodic (see Section 2). By order-invariance $L(\phi) = L'(\phi)$ thus $L(\phi)$ is aperiodic. To show that $L(\phi)$ is definable in FO, by Theorem 3.3, it suffices to show that for any $e, f, u, v, w \in \eta_{L(\phi)}$ with e, f idempotent, $eufvewf = ewfveuf$.

Recall that elements of $\eta_{L(\phi)}$ are equivalence classes of words. Replacing each of e, f, u, v, w with a word which is a representative of the corresponding equivalence class, and recalling what it means for the word $eufvewf$ to be equivalent to $ewfveuf$, we have the following equivalent condition: For all words e, f which represent idempotents in $\eta_{L(\phi)}$, for all words u, v, w and all words a, b:

$$aeufvewfb \in L(\phi) \leftrightarrow aewfveufb \in L(\phi)$$

Note that if e and f are idempotent, then $e^n = e$ and $f^n = f$ for any integer n. Therefore for any word c, d and any n we have $ced \in L(\phi)$ iff $ce^n d \in L(\phi)$. Hence we can conclude $L(\phi)$ is FO-definable if for some n we can show that for all words a, e, u, f, w, v, b we have the following:

$$ae^n uf^n ve^n wf^n b \in L(\phi) \text{ iff } ae^n wf^n ve^n uf^n b \in L(\phi)$$

Let r be the quantifier rank of ϕ. We produce an integer n and two orders $<_1$ and $<_2$ such that $\langle ae^n uf^n ve^n wf^n b, <_1\rangle \equiv_r \langle ae^n wf^n ve^n uf^n b, <_2\rangle$. This will conclude the proof of the theorem, because ϕ is order-invariant.

The two orders are obtained using the technique of FO interpretation (see, e.g. [13]). An interpretation of a σ-structure over a σ'-structure is a collection μ of FO(σ') formulas ϕ_d, $(\phi_R)_{R\in\sigma}$ such that if d is the number of free variables of ϕ_d then ϕ_R has $d \cdot x$ free variables where x is the arity of R. On a σ'-structure I, an interpretation μ defines a σ-structure $\mu(I) = \langle \phi_d(u), (\phi_R(u))_{R\in\sigma}\rangle$ where $\phi(u)$ stands for $\{\bar{x} \mid u \models \phi(\bar{x})\}$.

The properties of interpretations that we use are that: a) interpretations can be composed and b) elementary equivalence of σ'-structures can be lifted to their interpretations. More formally, let q be the maximum quantifier of formulas occurring in μ, then for any two σ'-structures I and I' and any number s, we have $I \equiv_{s+q} I'$ implies that $\mu(I) \equiv_s \mu(I')$.

Our starting point is given by the following lemma, based of Ramsey's theorem, proved in [12]. An *ordered string* is a string structure expended with a linear order corresponding to the transitive closure of the successor relation.

Lemma 3.4 ([12]). *$\forall s\ \exists m, n$ such that for every ordered string w over the alphabet consisting of a single letter x, if $|w| \geq n$, then there exists unary predicates P and P' such that:*

- $|P| = m$
- $|P'| = m - 1$
- $\langle w, P\rangle \equiv_s \langle w, P'\rangle$

Let a, e, u, v, w, f, b be arbitrary words.

Let s be big relative to r (the precise value of s will be apparent during the proof). Let m and n be given as in Lemma 3.4 for this s. Let ω be an ordered string of length n using only the letter x. Let ω_1 be the structure $\langle \omega, P\rangle$ and ω_2 be the structure $\langle \omega, P'\rangle$, where P, P' are the positions in ω given by Lemma 3.4. Thus we have: $\omega_1 \equiv_s \omega_2$.

We show that there is an interpretation μ such that $\mu(\omega_1)$ is isomorphic to $\langle ae^n uf^n ve^n wf^n b, <_1\rangle$ while $\mu(\omega_2)$ is isomorphic to $\langle ae^n wf^n ve^n uf^n b, <_2\rangle$. This implies that $\langle ae^n uf^n ve^n wf^n b, <_1\rangle \equiv_r \langle ae^n wf^n ve^n uf^n b, <_2\rangle$ assuming $s = r + q$, where q bounds the quantifier rank of all formulas in the interpretation.

The interpretation is obtained in two stages.

The first stage is simple. It it straightforward to define an interpretation μ_1 such that $\mu_1(\omega_1) = \langle abv(efef)^n uw, Q, A, B, V, E, F, U, W\rangle$ and $\mu_1(\omega_2) = \langle abv(efef)^n uw, Q', A, B, E, F, U, V, W\rangle$ where $abv(efef)^n uw$ is viewed as an ordered string, A (resp. B, ...) is a unary relation corresponding to the first position of each occurrence of a (resp. b ...), and Q is a unary relation which is true for all positions of the i^{th} occurrence of $efef$ iff $P(i)$ holds (similarly for Q'). μ_1 is constructed in the obvious way by interpreting each symbol x of ω as the substring $efef$ while adding the prefix abv and the suffix uw and transferring the string structure and marked positions of ω to the interpreted model.

Next we define a second interpretation μ_2 over strings as above, which, when composed with μ_1, yields the desired interpretation μ. The formula for the domain and the unary symbols are trivial; they simply copy the information from the input structure. A formula for the linear order is obtained by using the available transitive closure of the string successor (recall that our strings were ordered).

It remains to provide a formula specifying the successor relation. This is done using the properties listed below, each of them being definable in FO$(S, \prec, P)$. Inside any subword of one of the forms a, b, e, f, u, v, w (where the beginning of the subword is denoted by the corresponding unary predicate), the new successor relation is taken consistent with S. Hence WLOG we can assume that a, b, e, f, u, v, w are strings of length 1 and are treated as letters in the specification below. Finally, in order to avoid ambiguity, the successor which is specified is denoted by S' while S and $\prec$ are the successor and its transitive closure in the structure. Moreover, each time we use a navigational adjective (like *left*, *successor*, *first* ...) we add a prefix to it that clarifies which order is referred to when using this notion. We refer to Figure 2 for an illustration of the successor relation S' as specified by the properties listed below.

- The S'-first element is a.
- The S'-second element is the $\prec$-first occurrence of e.
- The S'-successor of a position labeled e not in P is the $\prec$-second e to its $\prec$-right.
- The S'-successor of a position labeled e in P is the $\prec$-third e to its $\prec$-right or the $\prec$-first e to its $\prec$-right depending on whether the initial e has a S-predecessor in P or not.
- At the $\prec$-end of the string $(efef)^n$, the S'-successor of the $\prec$-last e is w while the S'-successor of the e $\prec$-before is u.
- The S'-successor of w is the $\prec$-last f while the S'-successor of u is the f $\prec$-before the $\prec$-last.
- The S'-successor of a f not in P is the $\prec$-second f to its $\prec$-left.
- The S'-successor of a f in P is the $\prec$-third f to its $\prec$-left or the $\prec$-first f to its $\prec$-left depending on whether the initial f has a S-successor in P or not.
- At the $\prec$-beginning of the string $(efef)^n$, the S'-successor of the $\prec$-first f is v while the S'-successor of the $\prec$-second f is b.
- The S'-successor of v is the $\prec$-second e of the string and b is the S'-last element.

1	17	9	2	8	10	16	3	15	11	7	12	14	4	6	13	5
a	b	v	e	f	e	f	e+P	f+P	e+P	f+P	e	f	e	f	u	w

$$aeeewfffveeeufffb$$

1	17	9	2	8	10	16	3	7	11	15	4	6	12	14	5	13
a	b	v	e	f	e	f	e	f	e	f	e	f	e	f	u	w

$$aeeeufffveeewfffb$$

FIGURE 2. Two models and their corresponding successor as specified by the interpretation formula. Below each model the string according to the new successor relation is depicted. Notice how the presence of P induces a switch in the order of w and u between the top and the bottom case.

An inspection of the items above shows that this specification is $\mathrm{FO}(S, \prec, P)$ definable. Notice also that if $|P|$ is even then the string resulting from the new successor relation is $ae^nuf^nve^nwf^nb$ while if $|P|$ is odd the resulting string is $ae^nwf^nve^nuf^nb$ (see also Figure 2).

This concludes the proof of the theorem. $\dashv$

§4. Order-invariant Queries over Trees. We now deal with extending the above results to trees. We will give results for ranked trees, with or without a sibling ordering, and for unranked trees with no sibling ordering. For siblinged unranked trees, we have only an MSO upper bound on the expressiveness of order-invariant queries. To generalize the results for strings to trees, we use analogous algebraic machinery to characterize first-order definability over trees. We will then show that any query that fails this characterization must not be order-invariant.

As with strings, MSO definable sets of (siblinged or unordered) trees are exactly the regular sets of trees. Every such set is thus the acceptance set of a bottom-up tree automaton, which can be taken to be deterministic, see the monograph [10] for a survey on tree automata and their link with MSO in the ranked and unranked tree case. Note that in this section, we will use the following convention: an "unordered tree" will mean a tree without a sibling relation, a "siblinged tree" will mean a tree supplemented with a sibling relation, and a "tree" will mean either an unordered tree or a siblinged tree. When we refer to multiple trees (e.g. "let t, t' be trees ... ") we will naturally mean that all trees are of the same type (e.g. both t, t' unordered, or both siblinged).

Given a tree t and a node x of t the subtree of t rooted at x, consisting of all the nodes of t which are descendants of x, is denoted by $t|_x$.

A *context* is a tree with a distinguished leaf node, called its *port*. For any tree automata A, a context C induces a function C_A from the states of A to the states of A such that $C_A(q)$ is the state q' reached by A at the root of C when the port of C is assigned the state q and all other leaves of C are assigned the initial state of A. Given contexts C and C', their concatenation $C \cdot C'$ is the context formed by identifying the root of C' with the port of C. Concatenation of context C and tree T is defined similarly. Given a tree t and two nodes x, y of t such that $x \leq y$, the context $C_t[x, y)$ is defined from $t_1 = t|_x$ by replacing $t_1|_y$ by a port.

Let t be a tree, and x, x' be two nodes of t such that x and x' are not related by the descendant relationship. The *horizontal swap* of t at nodes x and x' is the tree t' constructed from t by replacing $t|_x$ with $t|_{x'}$ and vice-versa.

Let t be a tree of root r, and x, y, x', y' be four nodes of t such that $x \leq y \leq x' < y'$. The *vertical swap* of t between $[x, y)$ and $[x', y')$ is the tree t' constructed from t as depicted in Figure 3. More formally let $C = C_t[r, x)$, $\Delta_1 = C_t[x, y)$, $\Delta_2 = C_t[x', y')$, $\Delta = C_t[y, x')$, $T = t|_{y'}$. Then notice that $t = C \cdot \Delta_1 \cdot \Delta \cdot \Delta_2 \cdot T$. The tree t' is defined as $t' = C \cdot \Delta_2 \cdot \Delta \cdot \Delta_1 \cdot T$.

Let t be a tree and x be a node of t, the *k-spill* of x is the restriction of $t|_x$ to the set of nodes of t at distance at most k from x.

Two nodes x, y are said to be depth-k similar if their k-spill are isomorphic. The *k-type* of a node x is its equivalence class under the relation depth-k similar. Two trees t and t' are depth-k similar if their roots are depth-k similar.

A horizontal swap is said to be *k-guarded* if x and x' are depth-k similar. A vertical swap is said to be *k-guarded* if x and x' are depth-k similar and y and y' are depth-k similar.

We say that L is *closed under k-guarded horizontal swaps* if for all tree $t \in L$ and all tree t' constructed from t by a horizontal k-guarded swap then t' is in L.

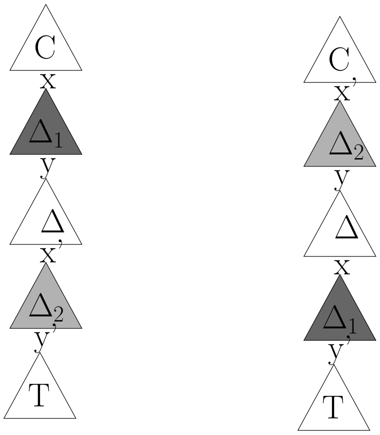

FIGURE 3. Illustration of the vertical swap

Closure under k-guarded vertical swaps is defined similarly. It should be clear that being closed under k-guarded swaps implies being closed under k'-guarded swaps for $k' \geq k$.

A regular tree language L is said to be *aperiodic* if there exists $l \in \mathbb{N}$, the *aperiodicity number* of L, such that for all contexts C, D, and all tree t, $D \cdot C^l \cdot t$ is in L iff $D \cdot C^{l+1} \cdot t$ is in L. If the set of contexts is seen as a monoid, this is the classical notion of aperiodicity in a monoid.

Let L be a regular tree language. We say that L satisfies (†) if the following holds for some k:

1. L is closed under k-guarded vertical swaps
2. L is closed under k-guarded horizontal swaps
3. L is aperiodic

In [4], the following algebraic characterization is proved:

THEOREM 4.1 ([4]). *Let L be a regular tree language over* SRT*,* RT *or* UT*. Then L is definable in* FO *iff L satisfies* (†)*.*

Using this, we prove the following:

THEOREM 4.2. Inv-FO(<) = FO *over* RT*,* SRT*, and* UT*.*

The first step of the proof is to show that:

THEOREM 4.3. Inv-FO(<) $\subseteq$ MSO *over* RT*,* SRT*,* UT*, and* SUT*.*

PROOF. This is rather immediate for ranked and/or ordered trees (SRT, SUT and RT), as a linear order can be defined in MSO. In the case of ordered trees the “lexicographical order” can be defined in MSO. In this order $x < y$ if x is an ancestor of y or x has an ancestor z which is a sibling of an ancestor z' of y and $z \prec z'$ using the sibling order. In the ranked (unordered) case, say the rank is r, an order can be obtained by first guessing r sets $S_1, S_2, \cdots, S_r$, then making sure that each node has exactly one child in each of the sets. These sets induces an order among the siblings and a lexicographical order is defined as above. In the case of UT, it is no longer possible to define a linear order in MSO. In this case, we use Ehrenfeucht-Fraïssé techniques. Let ϕ be an Inv-FO($<$) sentence, and $<_s$ be a sibling ordering. Then there is a canonical linear order on t which is constructed by a lexicographical process from $<_s$. Linear orders obtained this way are called *natural* in the rest of the proof. Let $L'(\phi) = \{t \mid$ there exists a natural linear order $<$ such that $\langle t, <\rangle \models \phi\}$. Because ϕ is order-invariant we have $L'(\phi) = L(\phi)$. Therefore the next lemma, which shows that sibling-ordering-invariant-FO($<$) collapses to MSO, concludes the proof of the theorem.

Lemma 4.4. *$L'(\phi)$ is definable in* MSO*.*

Proof. Recall that σ^S is our signature for unordered trees and $\sigma^{S,S'}$ is the extension of σ^S with an extra binary predicate S' which is interpreted as a successor among siblings. From the SUT case we know that $L'(\phi)$ is definable in MSO($\sigma^{S,S'}$). We want to show that it is definable in MSO(σ^S).

Because $L'(\phi)$ is definable in MSO($\sigma^{S,S'}$), there is a deterministic unranked bottom-up tree automaton $A = \langle \Sigma, Q, \delta, q_0, F\rangle$ that computes $L'(\phi)$ [10]. Using classical minimization and completion techniques we can further assume that A satisfies the following properties:

- For every $q \in Q$ there is a tree t^q such that when A runs on t^q it reaches state q at the root.
- For every $q \neq q' \in Q$ there exists a context $\Delta^{q,q'}$ such that the set $\{\Delta_A^{q,q'}(q), \Delta_A^{q,q'}(q')\}$ contains one accepting and one non-accepting state.

Recall from [10] that $\delta(q, a)$, for each $q \in Q$ and each $a \in \Sigma$, is a regular expression over Q, with the meaning that a node label a gets state q if the sequence (according to S') of states already computed at its children forms a word in $\delta(q, a)$. Let Q^* be the signature with a unary predicate for every state $q \in Q$. A string in Q^* is considered as a structure for Q^* in the obvious way. We first show that the regular expression is actually expressible in FO(Q^*), in particular it does not refer to S'.

Claim 4.5. *For each $q \in Q$ and each $a \in \Sigma$, $\delta(q, a)$ is definable by a formula of* FO(Q^*)*; that is, a formula using only unary predicates from Q.*

Proof of the claim. Take an arbitrary $q \in Q$ and $a \in \Sigma$, and let $L = \delta(q, a)$. Let r be the quantifier rank of ϕ and fix s such that for all $n, m \geq s$ the linearly ordered sequence $1^n \equiv_r^{\mathrm{FO}(<)} 1^m$ [15]. Take two words w and w' in Q^* such that $w \equiv_s^{\mathrm{FO}(Q^*)} w'$. We show that $w \in L$ iff $w' \in L$. This immediately implies that L is in FO(Q^*).

Fix an arbitrary order $<_Q$ on Q. Reorder w so that whenever $p <_Q p'$, elements in w labeled with p come before elements labeled with p', and similarly for w'. This yields two new strings $\bar{w} = q_1 \cdots q_u$ and $\bar{w}' = q'_1 \cdots q'_v$.

We first claim that $w \in L$ iff $\bar{w} \in L$. Write $w = p_1 \cdots p_u$ and assume, for a contradiction, that $w \in L$ but $\bar{w} \notin L$. Let t_1 and t_2 be the two trees $t_1 = a[t^{p_1} \cdots t^{p_u}]$ and $t_2 = a[t^{q_1} \cdots t^{q_u}]$. Let p and p' be the states reached by A on t_1 and t_2. By determinism of A and because $w \in L$ and $\bar{w} \notin L$ we have $p' \neq p$. Let $t = \Delta^{p,p'} \cdot t_1$ and $t' = \Delta^{p,p'} \cdot t_2$. By construction t is accepted by A but t' is not. This contradicts sibling-invariance of ϕ. By symmetry we also have $w' \in L$ iff $\bar{w}' \in L$. Therefore it suffices to show that $\bar{w} \in L$ iff $\bar{w}' \in L$.

Assume by way of contradiction that $\bar{w} \in L$ but $\bar{w}' \notin L$. Consider now the trees $t_1 = a[t^{q_1} \cdots t^{q_u}]$ and $t_2 = a[t^{q'_1} \cdots t^{q'_v}]$. We can see that q is the state reached by A on t_1 while the state reached by A on t_2 is some $q' \neq q$. Let $t = \Delta^{q,q'} \cdot t_1$ and $t' = \Delta^{q,q'} \cdot t_2$. By construction one is accepted by A and the other is rejected. By symmetry we assume that t is accepted by A but t' is not. Let $<$ (resp. $<'$) be the natural sibling order on t (resp. t'). We now claim that $\langle t, <\rangle \equiv_r \langle t', <'\rangle$, which implies that t and t' agree on ϕ, the desired contradiction.

We show this by giving a winning strategy for the corresponding Ehrenfeucht-Fraïssé game. On $\Delta^{q,q'}$ and the roots of t_1 and t_2, Duplicator plays using the identity map. On a move where Spoiler plays a node x in tree t^{q_i} for $i \geq 1$, Duplicator always responds by an identical y in a tree t^{q_j} such that $q_i = q_j$. There might be several possible choices of q_j. Let n be the number of occurrences of state q_i in $\bar{w}$ and m be the number of occurrences of the same state in $\bar{w}'$. Because $w \equiv_s^{\mathrm{FO}(Q^*)} w'$ we have $n = m$ or $n, m \geq s$. In both cases Duplicator picks one appropriate p_j using its strategy in the $\equiv_r^{\mathrm{FO}(<)}$ game between 1^n and 1^m. It is easy to verify that this strategy works. ⊣

Now by the claim, each transition is given by a $\mathrm{FO}(Q^*)$ formula. Hence, we know that A is given by an automaton that uses unordered first-order transitions. It is now immediate to see that such an automaton can be simulated in MSO. This completes the proof of Lemma 4.4, and hence the proof of Theorem 4.3. ⊣

⊣

Proof of Theorem 4.2: Let $\phi \in$ Inv-FO($<$), we wish to show that $\phi \in$ FO. From Theorem 4.1 and Theorem 4.3 it suffices now to show that $L(\phi)$ satisfies (†).

Part 3) of (†) (aperiodicity) is simple: if aperiodicity fails, then we know that for arbitrarily large l we can find s, u, t such that L can distinguish $s \cdot u^l \cdot t$ and $s \cdot u^{l+1} \cdot t$. Choosing l large with respect to the quantifier-rank r of ϕ, we fix an arbitrary sibling order on s, u, t and expand both $s \cdot u^l \cdot t$ and $s \cdot u^{l+1} \cdot t$ by the corresponding lexicographic ordering. If l is big enough, it is straightforward to show that the expanded structures are indistinguishable by formulas of rank r, hence we have again contradicted order-invariance.

For part 1) and 2) of (†) we will make use of the following lemma which is implicit in the proof of the main theorem of [12] (locality of Inv-FO($<$)).

Lemma 4.6. *[12] Let x be a number. For all r there exists k such that for each structure M and all x-tuples $\bar{a}$ and $\bar{b}$ of M such that the k-neighborhood of $\bar{a}$ is isomorphic to the k-neighborhood of $\bar{b}$, it is possible to construct linear orders $<_1$ and $<_2$ on M whose initial $2x$ elements are respectively $\bar{a}\bar{b}$ and $\bar{b}\bar{a}$ and such that $\langle M, \bar{a}\bar{b}, <_1\rangle \equiv_r \langle M, \bar{b}\bar{a}, <_2\rangle$.*

Intuitively the orders of Lemma 4.6 are obtained by first comparing nodes according to their distance from $\bar{a}\bar{b}$ and breaking the ties by alternating between nodes "on the side of $\bar{a}$" and those "on the side of $\bar{b}$". In the end there are too many alternations, and hence their parity cannot be detected. This construction, and hence the lemma, is easy to perform on trees, which is our setting here. Over arbitrary structures (as needed in the setting of [12]), this is much more complicated.

Let r be the quantifier-rank of ϕ.

Consider now part 2) of (†). Let k be given by Lemma 4.6 for $x = 1$ and r as above. Let t be a tree with two incomparable nodes x and y that are depth-k similar. Let $t_1 = t|_x$ and $t_2 = t|_y$. Let D be the tree constructed from t by removing all descendants of x and of y. Let $D[t_2, t_1]$ be the result of horizontally swapping t_2 by t_1 in t. Let $D[t_1, t_2]$ denote t itself. We wish to show that if $D[t_1, t_2]$ is in L then $D[t_2, t_1]$ is also in L. For this we construct two linear orders $\prec_1$ and $\prec_2$ such that $\langle D[t_1, t_2], \prec_1\rangle \equiv_r \langle D[t_2, t_1], \prec_2\rangle$. This would prove part 2) of (†) as ϕ has quantifier-rank r and is order-invariant.

Let M be the structure formed by the disjoint union of t_1 and t_2. Because x and y are depth-k similar, the k-neighborhood of x is isomorphic to the k-neighborhood of y. By Lemma 4.6 there exists orders $<_1$ and $<_2$ of M such that: with respect to $<_1$ x is minimal and y is its successor, with respect to $<_2$ y is minimal and x is its successor, and $\langle M, <_1\rangle \equiv_r \langle M, <_2\rangle$.

Fix a linear order $<$ on D. Define $\prec_1$ on $D[t_1, t_2]$ by concatenating $<$ and $<_1$ and define $\prec_2$ on $D[t_2, t_1]$ by concatenating $<$ and $<_2$.

It is now easy to verify that $\langle D[t_1, t_2], \prec_1\rangle \equiv_r \langle D[t_2, t_1], \prec_2\rangle$ by combining the identity strategy on D and the strategy given by Lemma 4.6 on $t_1 + t_2$.

Consider now part 3) of (†). Let k be given by Lemma 4.6 for $x = 2$ and r. Let t be a tree and x, y, x', y' be four nodes of t such that $x \leq y \leq x' \leq y'$, x and x' are depth-k similar, and y and y' are depth-k similar. Let $\Delta_1 = C_t[x, y)$ and $\Delta_2 = C_t[x', y')$. We proceed exactly as in the case of part 2) of †, using Lemma 4.6 to obtain orders $<_1$ and $<_2$ on $\Delta_1 + \Delta_2$, and combining these two orders with a fixed one on the remaining part of the tree. We then argue as in part 2) of (†). ⊣

§5. Order Invariance on Tree-like and Bounded Valence Structures.

We now consider how to extend the bounds given in the previous sections to graphs that may have cycles, but which are still well-behaved. We concentrate on two well-behaved classes: the bounded treewidth structures [16], and the bounded valence structures. We show that over such structures Inv-FO(<) ⊆ MSO. Note that this inclusion is not true in general as Gurevich's example, separating FO from Inv-FO(<), is not definable in MSO.

A *tree decomposition* of a graph G consists of a tree T and a function d mapping nodes of T to sets of vertices of G, satisfying:

- For every edge $(v_1, v_2) \in G$, there is $t \in T$ with $v_1, v_2 \in d(t)$.
- For every vertex v of G, $\{n \in T : v \in d(n)\}$ is a connected subset of T.

The *width* of a decomposition (T, d) is $max_{t\in T}|d(t)| - 1$. The *treewidth* of a graph G is the minimal width of a tree decomposition of G.

THEOREM 5.1. *For every b,* Inv-FO(<) ⊆ MSO *over graphs of treewidth b.*

PROOF. The idea of the proof is as follows. Courcelle shows that the set of tree decompositions of a family of graphs definable in Inv-MSO(<) is definable in CMSO. We show that CMSO can be replaced by MSO if we start with an Inv-FO(<) definable set of graphs. This is done by analyzing the automaton equivalent to the CMSO formula and showing that any counting can be avoided. The later is shown by reordering the siblings in an appropriate way without violating the truthness of the initial Inv-FO(<) formula. Once definability in MSO is achieved we conclude using a result of Lapoire, which shows that over graphs of bounded treewidth a tree decomposition of a graph can be defined in MSO. The combination of the two formula above yield the desired result.

Fix b and assume that all the graphs discussed in this proof have treewidth bounded by b. Let $\phi \in$ Inv-FO(<) and assume that ϕ has quantifier rank r. Let $L(\phi) = \{G \mid$ there exists a linear order $<$ such that $\langle G, <\rangle \models \phi\}$ be the set of graphs defined by ϕ.

We introduce some terminology based on the work of Courcelle and Lapoire (e.g. see [14], page 34). We describe how a tree decomposition $D = (T, d)$ of G can be considered as a colored tree T_D. The underlying tree of T_D is T and predicates for the colors are defined as follows: Fix for each node x of T a graph G_x with vertices in $\{1 \dots b+1\}$ that is isomorphic to the restriction of G to $d(x)$, and an isomorphism μ_x taking G_x onto this restriction. Let ν_x be the partial function from $\{1 \dots b+1\}$ to $\{1 \dots b+1\}$ that maps i to j exactly when $\mu_x(i) = \mu_y(j)$, where y is the parent node of x. For each graph τ with vertices in $\{1 \dots b+1\}$, we have a predicate P_τ that holds at a node x of T_D iff $G_x = \tau$, and for each h from $\{1 \dots b+1\}$ to $\{1 \dots b+1\}$, we have a a predicate P_h that holds at x iff $\nu_x = h$. That is, the predicates specify which graphs are associated with a node, and how the graph at a node is linked to the graph of its parent. Let $\mathcal{T}_b$ be the set of trees for this signature, and let f be the evaluation map taking a colored tree in $\mathcal{T}_b$ to the corresponding graph.

The following result was proved by Courcelle for any Inv-MSO(<) formula. We only use it for formulas $\phi \in$ Inv-FO(<). Let $T(\phi)$ be the set of tree decompositions of graphs defined by ϕ. More formally, $T(\phi) = f^{-1}(L(\phi))$. Courcelle showed:

Theorem 5.2. *[6, 7] For all $\varphi \in Inv - MSO(<)$, $T(\phi)$ is CMSO definable*

We show that, when restricted to first-order formulas, the result of Theorem 5.2 can be improved in order to obtain MSO definability.

Lemma 5.3. *For all $\varphi \in$* Inv-FO(<)*, $T(\phi)$ is* MSO *definable*

Before proving Lemma 5.3, we first show how we conclude the proof of the theorem. By a result of Lapoire [14], there exists a MSO transduction g, which, given a graph G of treewidth b, computes a structure $D \in \mathcal{T}_b$ such that $f(D) = G$. The exact definition of MSO transduction will not be needed here; the key fact about such transductions is that the pre-image of an MSO definable set under a transduction is again MSO definable [7]. The theorem now follows because $L(\phi) = g^{-1}(T(\phi))$, and by Lemma 5.3, $T(\phi)$ is definable in MSO. The closure of MSO definability under inverse MSO transduction mentioned above implies that $L(\phi) \in$ MSO and the theorem is proved.

We now turn to the proof of Lemma 5.3.

We will again make use of $\sigma^{S,S'}$, the signature for unordered trees with an extra binary predicate S' which is interpreted as a successor among siblings. Let $T'(\phi) = \{(t, \prec_s) \mid t \in T(\phi), \prec_s$ a sibling ordering on $T\}$. Since $T(\phi)$ is CMSO definable by Theorem 5.2, $T'(\phi)$ is definable in MSO($\sigma^{S,S'}$), since counting quantifiers are definable in the presence of a linear order. We claim that

$T'(\phi)$ (hence $T(\phi)$) is $\mathrm{MSO}(\sigma^S)$ definable. As discussed above we view $T'(\phi)$ as a set of unranked trees labeled with a finite alphabet Σ. Because $T'(\phi)$ is definable in $\mathrm{MSO}(\sigma^{S,S'})$, let A be an unranked tree automaton for $T'(\phi)$ with state set Q and transition function δ. Recall from the proof of Lemma 4.4 that δ maps every pair $(q, a) \in Q \times \Sigma$ to a regular expression over Q. We also assume that A is minimal (see the assumptions on the automaton A listed in the proof of Lemma 4.4). As in Lemma 4.4 we show that each transition of the automaton is actually given by a very simple regular expression.

Claim 5.4. *For each $q \in Q$ and each $a \in \Sigma$, $\delta(q, a)$ is definable by a formula of* $\mathrm{FO}(Q^*)$*, that is a formula using only unary predicates from Q.*

From this claim each transition of A is given by a $\mathrm{FO}(Q^*)$ formula. Hence, we know that A is given by an automaton that uses unordered first-order transitions. It is now immediate to see that such an automaton can be simulated in $\mathrm{MSO}(\sigma^S)$. Therefore $T'(\phi)$ (hence $T(\phi)$) is $\mathrm{MSO}(\sigma^S)$ definable and the lemma is proved.

Proof of the claim. The proof follows along the lines of the proof of Claim 4.5. In particular recall the definition of t^q and $\Delta^{q,q'}$ given in the proof of Lemma 4.4 for any $q, q' \in Q$.

For strings s, s' in Q^*, say $s \simeq_k s'$ if they have the same number of occurrences of each q up to threshold k.

Take an arbitrary $q \in Q$ and $a \in \Sigma$, and let $L = \delta(q, a)$. Let $k = 3^r$ (recall that r is the quantifier rank of ϕ)

Take $s, s' \in Q^*$ such that $s \simeq_k s'$. We show that $s \in L$ iff $s' \in L$. This would imply the claim. Fix an order on Q and let ω and ω' be the strings computed from s and s' by making the letters appear in the order of Q. Notice now that $s \in L$ iff $\omega \in L$. This is by definition of $T'(\phi)$, which is invariant under sibling reorderings. Therefore it suffices to show that $\omega \in L$ iff $\omega' \in L$.

Write $\omega = q_1 \cdots q_u$ and $\omega' = q'_1 \cdots q'_v$. Assume, for a contradiction, that $\omega \in L$ but $\omega' \notin L$. Let t_1 and t_2 be the two trees $t_1 = a[t^{q_1} \cdots t^{q_u}]$ and $t_2 = a[t^{q'_1} \cdots t^{q'_v}]$. By construction q is the state reached by A on t_1 and let q' be the state reached by A on t_2. By determinism of A and because $\omega \in L$ and $\omega' \notin L$ we have $q' \neq q$. Let $t = \Delta^{q,q'} \cdot t_1$ and $t' = \Delta^{q,q'} \cdot t_2$. By construction t is accepted by A but t' is not. Therefore $f(t) \models \phi$ but $f(t') \models \neg\phi$.

We now create orderings $<$ on $G = f(t)$ and $<'$ on $G' = f(t')$ such that $\langle G, < \rangle \equiv_r \langle G', <' \rangle$, a contradiction. For any tree T, we let $\nu(T)$ be the union of all nodes of G that are in $f(T)$. We first choose fixed orderings on $\nu(\Delta^{q,q'})$, on $\nu(t^q)$ for all $q \in Q$. In G we begin with the fixed ordering on $\nu(\Delta^{q,q'})$, then proceed with the fixed ordering for the nodes $\nu(t^{q_1}) - \nu(\Delta^{q,q'}), \cdots, \nu(t^{q_u}) - \nu(\Delta^{q,q'})$. Note that by definition of tree decompositions, the fact that t^{q_i} are all subtrees below $\Delta^{q,q'}$ in the tree decomposition t implies that the sets $\nu(t^{q_i}) - \nu(\Delta^{q,q'})$ are pairwise disjoint. In G' we proceed similarly, using $\nu(t^{q_i}) - \nu(\Delta^{q,q'})$. Let $<$ and $<'$ be these orderings. We show how to play the r-Ehrenfeucht-Fraïssé game between $\langle G, < \rangle$ and $\langle G', <' \rangle$. Given a play of the game, let h be the function taking each pebble x in $G - \nu(\Delta^{q,q'})$ to the unique l such that $x \in \nu(t^{q_l}) - \nu(\Delta^{q,q'})$, and let h' be the similar function on G'. We show that the Duplicator can play maintaining the following properties on the pebbles $x_1 \cdots x_i$ and $y_1 \cdots y_i$ at any step i:

(i) the play is the identity for moves in $\nu(\Delta^{q,q'})$,

(ii) $\langle \omega, h(x_1), \cdots, h(x_i)\rangle \equiv_{r-i}^{\mathrm{FO}(<)} \langle \omega', h'(y_1), \cdots, h'(y_i)\rangle$,

(iii) $\langle t^{h(x_i)}, x_i\rangle$ is isomorphic to $\langle t^{h'(y_i)}, y_i\rangle$.

This is easy to show by induction. For the initial case notice that by the choice of k, it follows from $s \simeq_k s'$ that $\langle \omega, <\rangle \equiv_r \langle \omega', <'\rangle$, where $<$ and $<'$ are the obvious orders on ω and ω' [15]. During the game, Duplicator mimics Spoiler's move when Spolier plays in $\nu(\Delta^{q,q'})$. If Spoiler plays pebble x_{i+1} on G outside of $\nu(\Delta^{q,q'})$, then Duplicator computes $j = h(x_{i+1})$, and uses her winning strategy for $\langle \omega, h(x_1), \cdots, h(x_i)\rangle \equiv_{r-i}^{\mathrm{FO}(<)} \langle \omega', h'(y_1), \cdots, h'(y_i)\rangle$ to derive the value j' that $h'(y_{i+1})$ must take. Note that necessarily $q_j = q'_{j'}$ and hence t^j and $t^{j'}$ are isomorphic and y_{j+1} can be safely placed at the same position as x_{i+1} inside the image by ν of the subtree pointed by j'.

This completes the proof of the claim and hence of Lemma 5.3 and Theorem 5.1. ⊣

⊣

We now turn to proving the analogous result for the set of colored graphs of valence less than b.

THEOREM 5.5. *For every b,* Inv-FO($<$) $\subseteq$ MSO *over graphs of valence less than b.*

PROOF. Fix b, and let $\phi \in$ Inv-FO($<$), and let r be the quantifier rank of ϕ.

Our goal is to find $s \in \mathbb{N}$ such that for all graphs G and G' of valence less than b, $G \equiv_s^{\mathrm{MSO}} G'$ implies the existence of linear orders $<_G$ and $<_{G'}$ such that $\langle G, <_G\rangle \equiv_r^{\mathrm{FO}} \langle G', <_{G'}\rangle$. The theorem then follows by order-invariance of ϕ.

We first consider the case where both G and G' are connected graphs of valence at most b. Because they have bounded valence there exists MSO formulas $\psi(x, y, S_1 \dots S_{b+1})$ and $\gamma(S_1 \dots S_{b+1})$ such that, given a connected graph G of valence less than b, we have: *(i)* G has at least one expansion to $S_1 \dots S_b$ such that $\gamma(S_1 \dots S_b)$ holds, *(ii)* for any $S_1 \dots S_b$ satisfying γ, $\rho(x, y) = \psi(x, y, S_1 \dots S_b)$ defines a linear-ordering on G. The formula γ says that S_i suffices to give a local ordering of G (i.e. a linear order on the successors of any given node), and ψ says that S_{b+1} is a singleton $\{p\}$, and x comes before y in a depth-first traversal of G starting from p using the local ordering definable from the S_i (see [8] for the detailed argument on the construction of γ and ψ). Let q be the maximum quantifier rank of ψ and γ above. Then let $s_0 = rq + b + 1$ and assume that $G \equiv_{s_0}^{\mathrm{MSO}} G'$. Pick any $S_1 \dots S_{b+1}$ in G such that $\gamma(S_1 \dots S_{b+1})$ holds and choose $S'_1 \dots S'_{b+1}$ in G' according to the winning strategy of Duplicator in the MSO-s_0-game on G and G'. This strategy guarantees that the two expansions agree on MSO-formulas of quantifier rank at most rq. Let $\prec_G$ (resp. $\prec_{G'}$) be the linear order on G (resp. G') defined by $S_1 \dots S_{b+1}$ (resp. $S'_1 \dots S'_{b+1}$) according to γ and ψ. Since γ and ψ are definable by formulas with quantifier rank at most q, these further expansions agree on MSO formulas of rank at most r. Hence there is a winning strategy for Duplicator in the r-move FO-game on $\langle G, \prec_G\rangle$ and $\langle G', \prec_{G'}\rangle$. Hence we have achieved our goal in the case of connected graphs.

For an integer k an MSO-k-type is an equivalence class of $\equiv_k^{\mathrm{MSO}}$, or equivalently a maximal consistent collection of MSO sentences of quantifier rank at most k.

Assume now that G and G' are arbitrary graphs of valence at most b. Fix $l \in \mathbb{N}$ such that for all $n, m \geq l$ the linearly ordered sequence $1^n \equiv_r^{\mathrm{FO}(<)} 1^m$. Let $s \in \mathbb{N}$ be such that whenever $G \equiv_s^{\mathrm{MSO}} G'$ then G and G' have the same number of connected components of any MSO-s_0-type (where s_0 is defined above), up to threshold l, (since being a component is MSO definable, one can guarantee this with a suitably large s). Now fix a linear order $<_\tau$ on the MSO-s_0-types of connected graphs. Assume now that $G \equiv_s^{\mathrm{MSO}} G'$. We define $<_G$ as follows. We use $<_\tau$ to order the connected components of G according to their MSO-s_0-type, breaking ties arbitrarily; within any connected component, we proceed as in the connected case above and obtain a linear order $\prec$. Form the analogous ordering $<_{G'}$ on G'.

We can now verify that $\langle G, <_G\rangle \equiv_r^{\mathrm{FO}} \langle G', <_{G'}\rangle$. When a pebble is placed in a connected component, Duplicator always responds in a connected component having the same MSO-s_0-type, and then the strategy inside a connected component is as shown in the connected case. For dealing with the arbitrary number of connected components, we note that both G and G' agree on the number of occurrences of each MSO-s_0-type up to threshold l. Hence if n is the number of occurrences of a given MSO-s_0-type in G and m is this same number in G', then we know $1^n \equiv_r^{\mathrm{FO}(<)} 1^m$ by the definition of l, and hence we can play according to the strategy given by $1^n \equiv_r^{\mathrm{FO}(<)} 1^m$. ⊣

The main tool used in the proof above is the possibility of defining linear orders in a graph using MSO formulas. These linear orders were definable because it was sufficient to guess an order among the finitely many successors of any given node, by guessing the corresponding number of unary predicates. This can be generalized as follows. A graph is said to be locally-ordered if a local order (i.e. on the successors of any given node) is definable. In this case Courcelle showed that within a component, an ordering can always be defined in MSO_2, Monadic Second Order Logic where quantification is over edges rather than nodes, with parameters (see [8]). It is then possible to show, as in the proof of Theorem 5.5, that over locally-ordered graphs we have Inv-FO($<$) $\subseteq \mathrm{MSO}_2$.

We do not know yet whether Inv-FO($<$) is contained in CMSO over arbitrary structures.

§6. Conclusions. Our aim is to show that over well-behaved classes of graphs, order-invariant queries over first-order logic in any given signature σ collapse to first-order logic over the signature without the order. Thus far we have shown this for strings and trees. One method of extending this to first-order logic over bounded treewidth graphs is to prove that if two graphs of treewidth b agree on first-order sentences of sufficiently large quantifier-rank, one can find tree decompositions of each graph that agree on fixed quantifier-rank; this would allow the characterizations of definability to be pushed from trees to graphs. We do not yet know of interesting classes for which a transfer of first-order equivalence from graphs to trees can be performed.

Another intersting open question is whether the results presented here for graphs would extend to structures such that the corresponding Gaifman graph is well behaved.

In [4] we have obtained an algebraic characterization of the logic FO_{mod} (the extension of FO with modulo counting quantifiers) on trees, by removing the aperiodicity condition. This extends the characterization of FO_{mod} on strings of Straubing (VII.3.1 of [20]). We think that this characterization could be used to prove that $\text{Inv-FO}_{mod}(<) = \mathrm{FO}_{mod}$ on trees, but we were not yet able to play the corresponding games.

REFERENCES

[1] *Open Problems in Finite Model Theory*, http://www-mgi.informatik.rwth-aachen.de/FMT.

[2] Serge Abiteboul, Richard Hull, and Victor Vianu, **Foundations of Databases**, Addison-Wesley, 1995.

[3] Daniele Beauquier and Jean-Eric Pin, *Factors of words*, ***Proc. of Intl. Coll. on Automata, Languages and Programming (ICALP)***, 1989.

[4] Michael Benedikt and Luc Segoufin, *Regular Tree Languages Definable in FO and FOmod*, 2008, To appear in Transactions of Computational Logic.

[5] C.C. Chang and H.Jerome Keisler, ***Model theory***, North-Holland Elsevier, 1990.

[6] Bruno Courcelle, *The Monadic Second Order Logic of Graphs I: Recognizable Sets of Finite Graphs*, ***Information and Computation***, (1990).

[7] ———, *The Monadic Second Order Logic of Graphs V: On Closing the Gap Between Definability and Recognizability*, ***Theoretical Computer Science***, (1991).

[8] ———, *The Monadic Second Order Logic of Graphs X: Linear Orders*, ***Theoretical Computer Science***, vol. 160 (1996), pp. 87–143.

[9] Hans-Dieter Ebbinghaus and Jörg Flum, ***Finite Model Theory***, Springer Verlag, 1995.

[10] Hubert Comon et al., *Tree Automata: Techniques and Applications*, Available at `http://tata.gforge.inria.fr/`.

[11] Tobias Ganzow and Sasha Rubin, *Order-invariant mso is stronger than counting mso in the finite*, ***Symposium on Theoretical Aspects of Computer Science (STACS)***, 2008, pp. 313–324.

[12] Martin Grohe and Thomas Schwentick, *Locality of Order-Invariant First-Order Queries*, ***ACM Transactions of Computational Logic***, (2000).

[13] Yuri Gurevich and Saharon Shelah, *"On the Strength of the Interpretation Method"*, ***The Journal of Symbolic Logic***, vol. **54** (1989), no. 2, pp. 305–323.

[14] Denis Lapoire, *Recognizability Equals Monadic Second-Order Definability for Sets of Graphs of Bounded Tree-Width*, ***Symposium on Theoretical Aspects of Computer Science (STACS)***, 1998.

[15] Leonid Libkin, ***Elements of finite model theory***, Springer, 2004.

[16] Neil Robertson and Paul D. Seymour, *Graph Minors III: planar tree-width*, ***J. Combin. Theory Ser. B***, vol. 36 (1984), pp. 49–64.

[17] Hannu Niemistö, *On Locality and Uniform Reduction*, ***Proc. IEEE Conf. on Logic in Computer Science (LICS)***, 2005.

[18] Martin Otto, *Epsilon-logic is more expressive than first-order logic on finite structures*, ***Journal of Symbolic Logic***, vol. 65 (2000), no. 4, pp. 749–757.

[19] Benjamin Rossman, *Successor-invariance in the finite*, ***Proc. IEEE Conf. on Logic in Computer Science (LICS)***, 2003.

[20] Howard Straubing, ***Finite Automata, Formal Logic, and Circuit Complexity***, Birkhäuser, 1994.

WOLFSON BUILDING, OXFORD, OX1 3QD, USA.
E-mail: Michael.Benedikt@comlab.ox.ac.uk

INRIA, LSV- ENS CACHAN, 61 AV. DU PRÉSIDENT WILSON, 94235 CACHAN CEDEX, FRANCE.
E-mail: `http://www-rocq.inria.fr/~segoufin`