Order Invariance on Decomposable Structures

Michael Elberfeld Marlin Frickenschmidt Martin Grohe

RWTH Aachen University, Germany

{elberfeld|frickenschmidt|grohe}@informatik.rwth-aachen.de

Abstract

Order-invariant formulas access an ordering on a structure's universe, but the model relation is independent of the used ordering. Order invariance is frequently used for logic-based approaches in computer science. Order-invariant formulas capture unordered problems of complexity classes and they model the independence of the answer to a database query from low-level aspects of databases. We study the expressive power of order-invariant monadic second-order (MSO) and first-order (FO) logic on restricted classes of structures that admit certain forms of tree decompositions (not necessarily of bounded width).

While order-invariant MSO is more expressive than MSO and, even, CMSO (MSO with modulo-counting predicates), we show that order-invariant MSO and CMSO are equally expressive on graphs of bounded tree width and on planar graphs. This extends an earlier result for trees due to Courcelle. Moreover, we show that all properties definable in order-invariant FO are also definable in MSO on these classes. These results are applications of a theorem that shows how to lift up definability results for order-invariant logics from the bags of a graph's tree decomposition to the graph itself.

Categories and Subject Descriptors F.4.1 [*Mathematical Logic*]: Model theory

Keywords finite model theory, first-order logic, monadic second-order logic, order-invariant logic, modulo-counting logic, bounded tree width, planarity

1. Introduction

A formula is *order-invariant* if it has access to an additional total ordering on the universe of a given structure, but its answer (that means, the model relation with respect to the structure that is expanded by the order) is invariant with respect to the given order. The concept of order invariance is used to formalize the observation that logical structures are often encoded in a form that implicitly depends on a linear order of the elements of the structure; think of the adjacency-matrix representation of a graph. Yet the properties of structures we are interested in should not depend on the encoding and hence the implicit linear order, but just on the abstract structure. Thus, we use formulas that access orderings, but define unordered properties. This approach can be prominently found in database theory where formulas from first-order (FO) and monadic second-order (MSO) logic are used to model query languages for relational databases and (hierarchical) XML documents, respectively. Being order-invariant means in this setting that the formula evaluation process is always independent of low-level aspects of databases like, for example, the encoding of elements as indices. Another example approach can be found in descriptive complexity theory where formulas whose evaluation is invariant with respect to specific encodings of the input structure capture unordered problems decidable by certain complexity classes. The famous open problem for a logic that captures all unordered properties decidable in deterministic polynomial time falls into this category. (See [13] for an introduction to these fields.)

Gurevich's construction (see [15] for details) shows that order-invariant FO ($<$-inv-FO) is more expressive than FO, which has only access to the relations of the structure. The same holds for order-invariant MSO ($<$-inv-MSO) and MSO with modulo-counting predicates (CMSO); Ganzow and Rubin showed that $<$-inv-MSO is able to express more properties than CMSO on general finite structures [12]. These inexpressibility results can be seen as "good news" since they show that the expressive power of logics and, hence, the range of definable properties, increases by using order invariance. On the other side, it is not possible to decide for a given FO-formula whether it is order-invariant or not. Thus, the standard syntax of FO-formulas does not provide us with an effective syntax for $<$-inv-FO. This opens up the question of whether we can find alternative logics that are equivalent to the order-invariant logics $<$-inv-FO and $<$-inv-MSO. Essentially, this is a question about comparing the expressive power of order-invariant logics with logics that have an effective syntax. This is the theme of the present paper.

While on general logical structures no logics that are equivalent to $<$-inv-FO or $<$-inv-MSO are known, this changes if we consider classes of structures that are well-behaved. Benedikt and Segoufin [1] showed that $<$-inv-FO and FO have the same expressive power on the class of all strings and the class of all trees (we write $<$-inv-FO $=$ FO *on* $\mathcal{C}$ to indicate that the properties definable in $<$-inv-FO equal the properties definable in FO when considering structures from a class $\mathcal{C}$). Considering $<$-inv-MSO, Courcelle [5] showed that it has the same expressive power as CMSO on the class of trees (that means, $<$-inv-MSO $=$ CMSO on trees). Recently it was shown that $<$-inv-FO $=$ FO($=$ MSO) and $<$-inv-MSO $=$ CFO($=$ CMSO) hold on classes of graphs of bounded tree depth [9]. More general results that apply to graphs of bounded tree width or planar graphs have not been obtained so far. This is due to the fact that, whenever we want to move from an order-invariant logic to another logic on a class of structures, we need to understand both (1) the expressive power of the order-invariant logic when restricted to these structures, and (2) the ability of the new logic to handle the structures in terms of, for example, definable decompositions.

Results. Our results address both of these issues to better understand the expressive power of order-invariant logics on decomposable structures. Addressing issue (1), we prove two general results, which show how to lift-up definability results for order-invariant logics from the bags of tree decompositions up to the whole decomposed structure. The corresponding theorems show that, whenever we are able to use MSO-formulas to define a tree decomposition whose adhesion is bounded (that means, bags have only bounded size intersections) and we can define total orderings on the vertices of each bag individually, then $<$-inv-MSO $=$ CMSO (Theorem 3.1) and $<$-inv-FO $\subseteq$ MSO (Theorem 3.2). Lifting theorems of this kind can be seen to be implicitly used earlier [1, 3, 4], but so far they only applied to the case where the defined tree decomposition has a bounded width. In this case, the whole structure can be easily transformed into an equivalent tree. Our theorems also handle the case where bags have an unbounded width: they merely assume the additional definability of a total ordering on bags, which is a much weaker assumption than having bounded width and covers larger graph classes. The proofs of the lifting theorems use type-composition methods to show how one can define the logical types of structures from the logical types of substructures. The main challenge lies in trading the power of the used types (in our case these are certain order-invariant types based on orderings that are compatible with the given decomposition) with the ability to prove the needed type-composition methods. The latter need to work with bags of unbounded size and, thus, are more general than the type-compositions methods that are commonly used for the case of bounded size bags.

Addressing issue (2), we study two types of classes of graphs where it is possible to meet the assumptions of the lifting theorems and, thus, show that $<$-inv-MSO $=$ CMSO and $<$-inv-FO $\subseteq$ MSO hold on these classes. The first two results apply to classes of graphs of bounded tree width. We show that one can define tree decompositions of bounded adhesion in MSO, where the bags admit MSO-definable total orderings. Then our lifting theorems apply to show that, on every class of graphs of bounded tree width, $<$-inv-MSO $=$ CMSO (Theorem 5.5) and (Theorem 5.6) $<$-inv-FO $\subseteq$ MSO hold. These two results generalize all the results mentioned above. Regarding these results, let us clarify their prior status: as discussed by Benedikt and Segoufin [1], an approach for proving $<$-inv-MSO $=$ CMSO and $<$-inv-FO $\subseteq$ MSO would be to combine and refine results of Courcelle [3, 4] with an approach for defining tree decompositions of bounded width using MSO-transductions, but the latter is not known [8, Section 7.6]. The missing MSO-definable tree decomposition of bounded width is not a problem for us since we do not use it. Our lifting theorems are able to handle a more general unbounded width case. Moreover, for the proof of our second set of results that apply to classes of graphs that do not contain $K_{3,\ell}$ for some $\ell \in \mathbb{N}$ as a minor, one definitely needs the unbounded width case. Using an MSO-definable tree decomposition into 3-connected components of Courcelle [6] along with proving that there are MSO-definable total orderings for the 3-connected bags of the decomposition, we are able to apply the lifting theorems to prove that $<$-inv-MSO $=$ CMSO (Theorem 5.9) and (Theorem 5.10) $<$-inv-FO $\subseteq$ MSO hold on every class of graphs that exclude $K_{3,\ell}$ as a minor for some $\ell \in \mathbb{N}$.

Organization of the paper. The paper starts with a preliminary section (Section 2) containing definitions related to graphs and logic. In Section 3, we formally state and prove the lifting theorems. Section 4 shows how to MSO-define tree decompositions along clique separators and reviews the known MSO-definable tree decomposition into 3-connected components. Section 5 picks up the decomposed graphs and shows how to define total orderings for bags. This is combined with the lifting theorems to prove the results about bounded tree width graphs and $K_{3,\ell}$-minor-free graphs stated above. Some details and proofs are moved to technical appendices due to space constraints.

2. Preliminaries

In the present section we introduce terms related to logical structures and graphs, monadic second-order logic and its order-invariant and modulo-counting variants, logical games and types, and transductions.

2.1 Structures and graphs

A *vocabulary* τ is a finite set of *relational symbols* where an *arity* $\mathrm{ar}(R) \geq 1$ is assigned to each symbol $R \in \tau$. A *structure* A over a vocabulary τ consists of a finite set $U(A)$, its *universe*, and a *relation* $R(A) \subseteq A^{\mathrm{ar}(R)}$ for every $R \in \tau$. We sometimes denote $R(A)$ by R^A, in particular if R is a symbol like $\leq$. *Graphs* G are structures over the vocabulary $\{E\}$ with $\mathrm{ar}(E) = 2$. When working with graphs, we write $V(G)$ for the graph's universe (its set of *vertices*) and $E(G)$ for its set of *edges*, respectively. The graphs we are working with are *undirected*. That means, for every two vertices v and w, we have $vw \in E(G)$ if, and only if, $(w, v) \in E(G)$. The *Gaifman graph* $G(A)$ of a τ-structure A has vertices $V(G(A)) = U(A)$ and for every pair of elements v and w that are part of a common relation in A, we insert the edge vw into $E(G(A))$.

A *tree decomposition* (T, β) of a structure A is a (rooted and directed) tree T together with a labeling function $\beta\colon V(T) \to \mathrm{pow}(A)$ that satisfies the following two properties: *(Connectedness condition)* For every element $v \in U(A)$, the induced subtree $T\big[\{t \in V(T) \mid v \in \beta(t)\}\big]$ is nonempty and connected. *(Cover condition)* For every tuple $(v_1, \ldots, v_r)$ of a relation in A, there is a $t \in V(T)$ with $\{v_1, \ldots, v_r\} \subseteq \beta(t)$. The sets $\beta(t)$ for every $t \in V(T)$ are the *bags* of the tree decomposition. Its *width* is $\max_{t \in V(T)} |\beta(t)| - 1$ and *adhesion* is $\max_{(t_1,t_2) \in E(T)} |\beta(t_1) \cap \beta(t_2)|$. The *tree width*, $\mathrm{tw}(A)$, of a structure A is the minimum width of a tree decomposition for it. We say that (T, β) is a tree decomposition *into* a class of structures $\mathcal{C}$ if for each $t \in V(T)$, $A[\beta(t)]$ (the restriction of A and all its relations to elements of $\beta(t)$) is in $\mathcal{C}$. Structures A and their Gaifman graphs $G(A)$ have the same tree decompositions. In particular $\mathrm{tw}(A) = \mathrm{tw}(G(A))$.

2.2 Monadic second-order logic and its variants

To define the *syntax* of *second-order logic* (SO-logic), we use *element variables* x_i for $i \in \mathbb{N}$ and *relation variables* X_i for $i \in \mathbb{N}$, which have an arity $\mathrm{ar}(X_i) \geq 1$. *Formulas of* SO-*logic* (SO-*formulas*) over a vocabulary τ are inductively defined as usual (see, for example, [13]). Such formulas are also called SO$[\tau]$-formulas to indicate the vocabulary along with the logic. The set of *free variables* of an SO-formula φ, denoted by $\mathrm{free}(\varphi)$, contains the variables of φ that are not used as part of a quantification. By renaming a formula's variables, we can always assume $\mathrm{free}(\varphi) = \{x_1, \ldots, x_k, X_1, \ldots, X_\ell\}$ for some $k, \ell \in \mathbb{N}$; we write $\varphi(x_1, \ldots, x_k, X_1, \ldots, X_\ell)$ to indicate that the free variables of φ are exactly x_1 to x_k and X_1 to X_ℓ. Given an SO-formula $\varphi(x_1, \ldots, x_k, X_1, \ldots, X_\ell)$, $A \models \varphi(a_1, \ldots, a_k, A_1, \ldots, A_\ell)$ indicates that A together with the assignment $x_i \mapsto a_i$, for $i \in \{1, \ldots, k\}$, and $X_i \mapsto A_i$, for $i \in \{1, \ldots, \ell\}$, to φ's free variables satisfies φ. A formula without free variables is also called a *sentence*.

Monadic second-order logic (MSO-logic) is defined by taking all SO-formulas without second-order quantifiers of arity 2 and higher. *Monadic second-order logic with modulo-counting* (CMSO-logic) extends MSO-logic with the ability to access (built-in) *modulo-counting atoms* $C_m(R)$ for every $m \in \mathbb{N}$ where R is a relation symbol. Given a structure A over a vocabulary that contains R, we have $A \models C_m(R)$ exactly if m divides $|R|$ (that means, $|R| \equiv 0 \mod m$). Atoms $C_m(X)$ where X is a relation variable are used in the same way.

Let τ be a vocabulary and $\leq$ a binary relation symbol not contained in τ. An MSO-sentence φ of vocabulary $\tau \cup \{\leq\}$ is *order-invariant* if for all τ-structures A and all linear orders $\leq_1, \leq_2$ of A we have $(A, \leq_1) \models \varphi \iff (A, \leq_2) \models \varphi$. We can now form a new logic, *order-invariant monadic second-order logic* ($<$-inv-MSO-logic), where the sentences of vocabulary τ are the order-invariant sentences of vocabulary $\tau \cup \{\leq\}$, and a τ-structure A satisfies an order-invariant sentence φ if $(A, \leq)$ satisfies φ in the usual sense for some (and hence for all) linear orders $\leq$ of A. There is a slight ambiguity in the definition of order-invariant sentences in which binary relation symbol $\leq$ we are referring to as our special "order symbol" (there may be several binary relation symbols in τ). But we always assume that $\leq$ is clear from the context. Alternatively, we could view $\leq$ as a "built-in" relation symbol that is fixed once and for all and is not part of any vocabulary. However, this would be inconvenient because we sometimes need to treat $\leq$ just as a ordinary relation symbol and the sentences of $<$-inv-MSO-logic of vocabulary τ just as ordinary MSO-sentences of vocabulary $\tau \cup \{\leq\}$.

First-order logic (FO-logic) and *order-invariant first-order logic* ($<$-inv-FO-logic) are defined by taking all formulas of MSO-logic and $<$-inv-MSO-logic, respectively, that do not contain second-order quantification. Moreover, a *modulo-counting* variant of first-order logic arises by using quantifier for counting the number of elements that satisfy a certain property modulo some $m \in \mathbb{N}$; we denote it by CFO in the below fact, but do not use it in the paper at other places.

Let P be a set of τ-structures for some vocabulary τ; we call P a *property* of structures. The property P is MSO-*definable* if there is an MSO-sentence φ that holds exactly for the structures of P. The class of properties definable in MSO-logic is denoted by MSO. The *definability* notion along with the class of definable properties is defined in the same way for all logics defined above.

2.3 Games and types

For structures A, B and $q \in \mathbb{N}$, we write $A \equiv_q^{\text{MSO}} B$ if A and B satisfy the same MSO-sentences of quantifier rank at most q. We write $A \equiv_q^{<\text{-inv-MSO}} B$ if A and B satisfy the same order-invariant MSO-sentences of quantifier rank at most q. For every $c \in \mathbb{N}$, we write $A \equiv_{q,c}^{\text{CMSO}} B$ if A and B satisfy the same CMSO-sentences of quantifier rank at most q and only numbers $\ell \leq c$ are used in the counting quantifiers. Alternatively, one can use first-order modulo-counting quantifier.

It will be convenient to use versions of MSO and CMSO without individual variables (see, for example, [11]). We assume that the reader is familiar with the characterizations of MSO-equivalence and CMSO-equivalence by Ehrenfeucht-Fraïssé games (see, for example, [12]). Corresponding to the versions of the logics without individual variables, we use a version of the games where the players only select sets and never elements, and a position *induces a partial isomorphism* if the mapping between the singleton sets of the position is a partial isomorphism. (The rules of the game require the Duplicator to answer to a singleton set with a singleton set and to preserve the subset relation.) Then a *position* of the game on structures A, B is a sequence $\Pi = (P_i, Q_i)_{i \in [p]}$ of pairs (P_i, Q_i) of subsets $P_i \subseteq U(A)$ and $Q_i \subseteq U(B)$. If Π is a position of the q-move game, then $0 \leq p \leq q$. The position is a *q-move winning position* for one of the players if this player has a winning strategy for the q-move game starting in this position.

We also use the concept of *types*. Let τ be a vocabulary and $q, p \in \mathbb{N}$. Then for all τ-structures A and sets $P_1, \ldots, P_p \subseteq U(A)$, the MSO-*type of* $(A, P_1, \ldots, P_p)$ *of quantifier rank* q is

$$\text{tp}_q^{\text{MSO}}(A, P_1, \ldots, P_p) :=$$
$$\{\varphi(X_1, \ldots, X_p) \mid \varphi \text{ is of rank } q \text{ and MSO}, A \models \varphi(P_1, \ldots, P_p)\}.$$

Moreover, the class of all types over τ with respect to rank q and p free set variables is $\text{TP}^{\text{MSO}}(\tau, q, p) :=$

$$\{\, \text{tp}_q^{\text{MSO}}(A, P_1, \ldots, P_p) \mid A \text{ is } \tau\text{-structure}, P_1, \ldots, P_p \subseteq U(A)\},$$

and we let $\text{TP}^{\text{MSO}}(\tau, q) := \text{TP}^{\text{MSO}}(\tau, q, 0)$. For $c, q \in \mathbb{N}$, we define the CMSO-type $\text{tp}_{q,c}^{\text{CMSO}}(A, P_1, \ldots, P_p)$ and sets $\text{TP}^{\text{CMSO}}(\tau, q, c, p)$ and $\text{TP}^{\text{CMSO}}(\tau, q, c)$ similarly.

Note that $\text{tp}_q^{\text{MSO}}(A, P_1, \ldots, P_p) = \text{tp}_q^{\text{MSO}}(B, Q_1, \ldots, Q_p)$ if, and only if, $(P_i, Q_i)_{i \in [p]}$ is a q-move winning position for the Duplicator in the MSO-game on A, B. Furthermore, for $p = 0$ we have $\text{tp}_q(A) = \text{tp}_q(B)$ if, and only if, $A \equiv_q^{\text{MSO}} B$. Similar remarks apply to the CMSO-types.

For a vocabulary τ and a binary relation symbol $\leq \notin \tau$, we say that a subset $I \subseteq \text{TP}^{\text{MSO}}(\tau \cup \{\leq\}, q)$ is *order-invariant* if for all τ-structures A and all linear orders $\leq, \leq'$ of A,

$$\text{tp}_q^{\text{MSO}}(A, \leq) \in I \iff \text{tp}_q^{\text{MSO}}(A, \leq') \in I\,.$$

If I is inclusion-wise minimal order-invariant, then we call it an *order-invariant type*. Note that every $\theta \in \text{TP}^{\text{MSO}}(\tau \cup \{\leq\}, q)$ is contained in exactly one order-invariant type, which we denote by $\langle\theta\rangle$. We let

$$\text{TP}^{<\text{-inv-MSO}}(\tau, q) := \{\, \langle\theta\rangle \mid \theta \in \text{TP}(\tau \cup \{\leq\}, q)\}$$

be the set of all order-invariant types. For a τ-structure A, we call the set $\text{tp}_q^{<\text{-inv-MSO}}(A) := \langle \text{tp}_q^{\text{MSO}}(A, \leq)\rangle$ for some and hence for all linear orders of A the *order-invariant* MSO-*type of A of quantifier rank q*. It may seem more natural to define the order-invariant type of a structure as the set of all order-invariant sentences it satisfies. The following proposition says that this would lead to an equivalent notion, but our version is easier to work with, because it makes the connection between types of ordered structures and order-invariant types more explicit. The following lemma is proved in the appendix.

Lemma 2.1. *For all τ-structure A, A', the following are equivalent.*

1. $\text{tp}_q^{<\text{-inv-MSO}}(A) = \text{tp}_q^{<\text{-inv-MSO}}(A')$.
2. $A \equiv_q^{<\text{-inv-MSO}} A'$.
3. *There is a sequence $A_0, \ldots, A_\ell$ of τ-structures and linear orders $\leq_i, \leq'_i$ with $A = A_0$, $A' = A_\ell$, and $(A_{i-1}, \leq_{i-1}) \equiv_q^{\text{MSO}} (A_i, \leq'_i)$ for all $i \in [\ell]$.*

If $A \equiv_q^{<\text{-inv-MSO}} A'$, we say that sequences $(A_i), (\leq_i), (\leq'_i)$ as in (3) *witness* $A \equiv_q^{<\text{-inv-MSO}} A'$.

2.4 Transductions

An MSO$[\tau, \tau']$-*transduction* is a finite set Λ of MSO-formulas over a vocabulary τ which transforms a structure A over τ into a (target) structure B of the vocabulary τ'. To achieve this, each of the formulas is either indexed with a symbol $R' \in \tau'$ and has $\text{ar}(R')$-many free variables; or the formula is indexed with U (standing for the universe $U(B)$ of B) and has just one free variable. paper it suffices to assume that these free variables. We write MSO-transductions as a tuple of formulas and mark each formula by the corresponding relation symbol from τ'. Formally, if $\tau' = \{R^1, \ldots, R^n\}$, then $\Lambda = (\lambda_U, \lambda_{R^1}, \ldots, \lambda_{R^n})$ and the resulting output structure B is obtained by interpreting the formulas of Λ with all possible assignments from the universe $U(A)$ of our input structure A. That means, $B = \Lambda[A] := (U_{\Lambda[A]}, R^1_{\Lambda[A]}, R^2_{\Lambda[A]}, \ldots R^n_{\Lambda[A]})$ with $B_{\Lambda[A]} := \{a \in U(A) \mid A \models \lambda_U(a)\}$ and $R^i_{\Lambda[A]} := \{\bar{a} \in U(A)^{\text{ar}(R^i)} \mid A \models \lambda_{R^i}(\bar{a})\}$ for each $i \in \{1, \ldots, n\}$. Because the universe and all relations of B are expressed using τ-formulas relative to A, we say for such a pair of structures that Λ *is a transduction from A to B*. If Λ is a transduction and B a structure, we define $\Lambda^{-1}[B] := \{A \mid \Lambda[A] = B\}$. MSO-transductions preserve MSO-definability (formally stated by) and they can be composed to new MSO-transductions. More details about linear transductions and parameters are given in Appendix A.1.

3. Lifting definability

An *ordered tree decomposition* of a structure A is a tree decomposition of A together with a linear order for each bag. We represent ordered tree decompositions by logical structures in the following way. An *ordered tree extension* (*otx* for short) of a τ-structure A is a structure $A^\star$ extending A by a tree decomposition (T^A, β^A) of A and a linear order $\preceq_t^A$ of $\beta^A(t)$ for each $t \in V(T^A)$. The *adhesion* of $A^\star$ is the adhesion of the tree decomposition (T^A, β^A). Formally, we view $A^\star$ as a structure over the vocabulary

$$\tau^\star := \tau \cup \{V_S, V_T, E_T, R_\beta, R_\preceq\},$$

where V_S, V_T are unary relation symbols, E_T and R_β are binary, and $R_\preceq$ is ternary. Of course we assume that none of these symbols appears in τ. In the $\tau^\star$-structure $A^\star$, these symbols are interpreted as follows:

$V_S(A^\star) := U(A)$,

$V_T(A^\star) := V(T^A)$,

$E_T(A^\star) := E(T^A)$,

$R_\beta(A^\star) := \{(t, v) \mid t \in V(T^A), v \in \beta^A(t)\}$, and

$R_\preceq(A^\star) := \{(t, v, w) \mid t \in V(T^A), v, w \in \beta^A(t) \text{ with } v \preceq_t^A w\}$.

An MSO$[\tau, \tau^\star]$-transduction $\Lambda^\star$ *defines an otx (of adhesion at most k)* of a τ-structure A if every $\widehat{A} \in \Lambda^\star(A)$ is an otx of A (of adhesion at most k). We say that $\Lambda^\star$ *defines otxs (of adhesion at most k)* on a class $\mathcal{C}$ of τ-structure if $\Lambda^\star$ defines an otx (of adhesion at most k) of every $A \in \mathcal{C}$. Moreover, $\mathcal{C}$ *admits* MSO*-definable ordered tree decompositions (of bounded adhesion)* if there is such a transduction $\Lambda^\star$ that defines otxs (of adhesion at most k) on $\mathcal{C}$. We make similar definitions for the logic CMSO.

We prove the following lemmas, which show how to use the tree decompositions and the bag orderings to define properties of order-invariant formulas without using order invariance.

Theorem 3.1 (Lifting theorem for $<$-inv-MSO)**.** *Let $\mathcal{C}$ be a class of structures that admits* CMSO*-definable ordered tree decompositions of bounded adhesion. Then* $<$-inv-MSO $=$ CMSO *on $\mathcal{C}$.*

Theorem 3.2 (Lifting theorem for $<$-inv-FO). *Let $\mathcal{C}$ be a class of structures that admits* MSO*-definable ordered tree decompositions of bounded adhesion. Then* $<$-inv-FO $\leq$ MSO

Theorem 3.1 is proved in three steps: First, in Section 3.1, we modify the given ordered tree extension, such that its tree decomposition follows a certain normal form that allows to partition its nodes into two different classes (called a-nodes and b-nodes). The partition of the nodes along with a global partial order that is based on the local orderings in the bags is then encoded as part of the structure, turning every otx into an expanded otx. Second, in Section 3.2, we prove type-composition lemmas for both the a-nodes and the b-nodes. They show how one can define the type of an expanded otx with respect to total orderings that respect the already existing partial order from the types of substructures that arise by adding such compatible orderings to them. Third, in Section 3.4, we apply the type compositions to prove Theorem 3.1. The proof of Theorem 3.2 proceeds in a similar way. The modifications that we need to apply to the proof of Theorem 3.1 in order to prove Theorem 3.2 are mentioned along the way.

3.1 Expanding ordered tree extensions

A tree decomposition (T, β) of a structure A is *segmented* if the set $V(T)$ can be partitioned into a set V_a of *adhesion nodes* and a set V_b of *bag nodes* (*a-nodes* and *b-nodes*, for short) satisfying the following conditions.

1. For all edges $tu \in E(T)$, either $t \in V_a$ and $u \in V_b$ or $u \in V_a$ and $t \in V_b$.
2. For all a-nodes $t \in V_a$ and all distinct neighbors $u_1, u_2 \in N(t)$, we have $\beta(t) = \beta(u_1) \cap \beta(u_2)$.
3. For all b-nodes $t \in V_b$ and all distinct neighbors and all distinct neighbors $u_1, u_2 \in N(t)$ we have $\beta(t) \cap \beta(u_1) \neq \beta(t) \cap \beta(u_2)$.
4. All leaves of T are b-nodes.

We can transform an arbitrary tree decomposition (T, β) into a segmented tree decomposition as follows. We first contract all edges $tu \in E(T)$ with $\beta(u) \subseteq \beta(t)$, resulting in a decomposition (T', β') where $\beta'(u) \not\subseteq \beta'(t)$ for all $tu \in E(T')$. Then, for all edges $tu \in E(T')$, we introduce a new node v_{tu}, where $v_{tu} = v_{ut}$, and edges from v_{tu} to t and u. Then we identify all nodes v_{tu} and $v_{tu'}$ such that $\beta'(t) \cap \beta'(u) = \beta'(t) \cap \beta'(u')$. We let T'' be the resulting tree and define β'' on $V(T'')$ by $\beta''(t) := \beta'(t)$ for $t \in V(T')$ and $\beta''(v_{tu}) := \beta'(t) \cap \beta'(u)$ for all $tu \in E(T')$. The resulting tree decomposition (T'', β'') is segmented. This transformation is MSO-definable and, thus, we may assume that the tree decompositions in ordered tree extensions are segmented. There is an MSO$[\tau^\star, \tau^\star]$-transduction Λ_{SEGMENT} that transforms every otx into an otx with a segmented tree decompositions.

It will be convenient to assume that the trees underlying our tree decompositions are directed. That means, all edges are directed away from a root. We denote the set of children of a node t in a directed tree T by $N_+^T(t)$, or just $N_+(t)$ if T is clear from the context.

For the rest of this section, we fix a vocabulary τ that does not contain the order symbol $\leq$ and a $k \in \mathbb{N}$. In the rest of this section, we only consider otxs of τ-structures. We assume that the adhesion of these otxs is at most k and their tree decomposition is segmented.

It will be convenient to introduce some additional notation. As before, whenever we denote an otx by $A^\star$, we denote the underlying structure by A and the tree decomposition by (T^A, β^A). We denote the descendant order in the tree T^A of an otx $A^{\star\star}$ by $\trianglelefteq^A$. For every node $t \in V(T^A)$, we let T_t^A be the subtree of T^A rooted in t, that is, $T_t^A := T^A[\{u \in V(T^A) \mid t \trianglelefteq^A u\}]$. We let $\gamma^A(t)$, called the *cone* of t, be the union of all bags $\beta^A(u)$ for $u \in V(T_t^A)$. If s is the parent of t we let $\sigma^A(t) := \beta^A(t) \cap \beta^A(s)$, called the *separator* of t. For the root r we let $\sigma^A(r) = \emptyset$. In all these notations we may omit the index ^{A} if A is clear from the context. Note that for all a-nodes t of T and all $u \in N_+(t)$ we have $\sigma(t) = \beta(t) = \sigma(u)$.

Let $A^\star$ be an otx. We define an expansion $A^{\star\star}$ of $A^\star$ to the vocabulary

$$\tau^{\star\star} := \tau^\star \cup \{V_a, V_b, R\gamma, R_\sigma, S_1, \ldots, S_k, \preceq\},$$

where V_a, V_b are unary and $R_\sigma, R_\gamma, S_1, \ldots, S_k, \preceq$ are binary relation symbols. We let $V_a(A^{\star\star})$ and $V_b(A^{\star\star})$ be the sets of a-nodes and b-nodes of the tree T^A, respectively, and

$$R_\sigma(A^{\star\star}) := \{(t, v) \mid t \in V(T^A), v \in \sigma^A(t)\},$$
$$R_\gamma(A^{\star\star}) := \{(t, v) \mid t \in V(T^A), v \in \gamma^A(t)\}.$$

We let $\preceq := \preceq^{A^{\star\star}}$ be the partial order on $U(A^{\star\star})$ defined as follows. We first define the restriction of $\preceq$ to $V(T)$. For all b-nodes

t, we let $\preceq'_t$ be the linear order on $N_+(t)$ defined by $u_1 \preceq'_t u_2$ if the set $\sigma(u_1) \subseteq \beta(t)$ is lexicographically smaller than or equal to the set $\sigma(u_2) \subseteq \beta(t)$ with respect to the linear order $\preceq_t$ on $\beta(t)$, for all children $u_1, u_2 \in N_+(t)$. This is indeed a linear order because $\preceq_t$ is a linear order of $\beta(t)$ and $\sigma(u_1) \neq \sigma(u_2)$ for all distinct $u_1, u_2 \in N_+(t)$. Then we let the restriction of $\preceq$ to $V(T)$ be the reflexive transitive closure of the "descendant order" $\trianglelefteq$ on T and all the relations $\preceq'_t$ for b-nodes $t \in V(T)$. To define the restriction of $\preceq$ to $U(A)$, for every $v \in U(A)$ we let $t(v)$ be the topmost (that is, $\trianglelefteq$-minimal) node $t \in V(T)$ such that $v \in \beta(t)$. Then we let $v \preceq w$ if, and only if, $t(v) \prec t(w)$ or $t(v) = t(w)$ and $v \preceq_{t(v)} w$. To complete the definition of $\preceq$, we let $t \preceq v$ for all $t \in V(T)$ and $v \in U(A)$.

Finally, we define the relations $S_1(A^{\star\star}), \ldots, S_1(A^{\star\star})$ by letting $S_i(A^{\star\star})$ be the set of all pairs (t, v), where $t \in V(T^A)$ and v is the ith element of $\sigma(t)$ with respect to the partial order $\preceq$, which is a linear order when restricted to $\sigma(t) \subseteq \beta(t)$. Note that we have $|\sigma(t)| \leq k$ by our general assumption that the adhesion of all otxs be at most k. This completes the definition of $A^{\star\star}$

It is easy to see that there is an $\mathrm{MSO}[\tau^\star, \tau^{\star\star}]$-transduction $\Lambda_{\mathrm{EXPAND}}$ that defines $A^{\star\star}$ in $A^\star$.

We call $A^{\star\star}$ an *expanded otx* (*otxx* for short) of A. More generally, we call a $\tau^{\star\star}$-structure A' an *expanded otx* if there is a τ-structure A such that A' is an otxx of A. Let $A^{\star\star}$ be an expanded otx. For every $t \in V(T)$, we let

$$A^{\star\star}_t := A^{\star\star}[\gamma(t) \cup V(T_t)], \text{ and}$$
$$A^{\star\star}_{(t)} := A^{\star\star}[\beta(t) \cup N_+(t)].$$

We call a $\tau^{\star\star}$-structure A' a *sub-otxx* if there is an otxx $A^{\star\star}$ and a node $t \in V(T^A)$ with $A' = A^{\star\star}_t$. The only difference between an otxx and a sub-otxx is that in an otxx the set $\sigma(r)$ is empty for the root r whereas in a sub-otxx it may be nonempty.

Lemma 3.3. *There are* MSO*-sentences* otxxs *and* $\mathsf{sub\text{-}otxx}$ *of vocabulary* $\tau^{\star\star}$ *defining the classes of all otxx and sub-otxx (satisfying our general assumptions: the tree decomposition is segmented and has adhesion at most* k*).*

Proof. Straightforward. □

We will later modify an otxx $A^{\star\star}$ by *replacing* a sub-otxx $A^{\star\star}_t$, for some $t \in V(T^A)$, by another sub-otxx $B^{\star\star}$. Let t' be the root node of the tree T^B. The replacement is possible if the induced substructures $A^{\star\star}[\{t\} \cup \sigma^A(t)]$ and $B^{\star\star}[\{t'\} \cup \sigma^B(t')]$ are isomorphic. If they are, there is a unique isomorphism, because $\{t\} \cup \sigma^A(t)$ and $\{t'\} \cup \sigma^B(t')$ are linearly ordered by the restrictions of $\preceq^{A^{\star\star}}$, $\preceq^{B^{\star\star}}$. Now replacing $A^{\star\star}_t$ by $B^{\star\star}$ in $A^{\star\star}$ just means deleting all vertices in $V(A^{\star\star}_t)$ except those in $\{t\} \cup \sigma^A(t)$, adding a disjoint copy of $B^{\star\star}$, and identifying the vertices in $\{t\} \cup \sigma^A(t)$ and $\{t'\} \cup \sigma^B(t')$ according to the unique isomorphism. Note that the substructures $A^{\star\star}[\{t\} \cup \sigma^A(t)]$ and $B^{\star\star}[\{t'\} \cup \sigma^B(t')]$ are isomorphic if the sub-otxxs $A^{\star\star}_t$ and $B^{\star\star}$ satisfy the same first-order sentences of quantifier rank $\mathrm{ar}(\tau) + 1$, where $\mathrm{ar}(\tau)$ denote the maximum arity of a relation symbol in the vocabulary τ. To express isomorphism, we use the relations $S_1, \ldots, S_k$ and the fact that the root of an otxx can be defined by a formula of quantifier rank 2. Thus in particular, if $\mathrm{tp}^{\mathrm{MSO}}_q(A^{\star\star}_t) = \mathrm{tp}^{\mathrm{MSO}}_q(B^{\star\star})$ for some $q \geq \mathrm{ar}(\tau) + 1$, we can replace $A^{\star\star}_t$ by $B^{\star\star}$.

Finally, we say that a linear order $\leq$ on an otxx or sub-otxx $A^{\star\star}$ is *compatible* if it extends the partial order $\preceq^{A^{\star\star}}$. If $\leq$ is a compatible linear order, then $(A^{\star\star}, \leq)$ denotes the $\tau^{\star\star} \cup \{\leq\}$-expansion of $A^{\star\star}$ by this order, and $(A^{\star\star}_t, \leq)$ denotes the induced substructure where $\leq$ is restricted to the sub-otxx $A^{\star\star}_t$. We can extend the replacement operation to such ordered expansions of otxxs; in the same way we replace a sub-otxx $A^{\star\star}_t$ by $B^{\star\star}$, we can replace a $(A^{\star\star}_t, \leq)$ by $(B^{\star\star}, \leq')$ for some compatible linear order $\leq'$ of $B^{\star\star}$.

3.2 Type-composition lemmas

As all structures we are working with in this section are otxxs and sub-otxx, we denote them by A rather than $A^{\star\star}$. Apart from that, we use the same notation as before. In particular, if A is an otxx then by T^A we denote the tree of its tree decomposition, and for a node $t \in V(T^A)$, by A_t we denote the sub-otxx rooted in t.

Throughout this subsection, we fix a $q \in \mathbb{N}$ such that $q \geq 2$ and $q \geq \mathrm{ar}(\tau) + 1$ and q is at least the quantifier rank of the formulas otxx and $\mathsf{sub\text{-}otxx}$ of Lemma 3.3. This means that if A is an otxx (or sub-otxx) and A' an arbitrary $\tau^{\star\star}$-structure such that $A \equiv^{\mathrm{MSO}}_q A'$ then A' is an otxx (a sub-otxx) as well. Furthermore, if t, t' are the root nodes of A, A', respectively, then induced substructures $A[\{t\} \cup \sigma^A(t)]$ and $A'[\{t'\} \cup \sigma^{A'}(t')]$ are isomorphic. Finally, if A, A' are otxxs and $\leq, \leq'$ are linear orders of A, A', respectively, such that $(A, \leq) \equiv^{\mathrm{MSO}}_q (A', \leq')$ then $\leq$ is compatible if and only if $\leq'$ is compatible.

We let $\Theta := \mathrm{TP}^{\mathrm{MSO}}(\tau^{\star\star} \cup \{\leq\}, q)$. Furthermore, we assume that $\Theta = \{\theta_1, \ldots, \theta_m\}$.

For an otxx A, a compatible linear order $\leq$ of A and a subset $N \subseteq V(T^A)$ (usually $N = N_+(t)$ for a node $t \in V(T^A)$). For all $i \in [m]$, let P_i be the set of all $u \in N$ such that $\mathrm{tp}^{\mathrm{MSO}}_q(A_u, \leq) = \theta_i$. We call $(P_1, \ldots, P_m)$ the *type partition* of N. (Note that some of the P_i may be empty. We always allow partitions to have empty parts.)

Lemma 3.4 (Ordered type composition at b-nodes)**.** *For every* $\theta \in \Theta$ *there is an* $\mathrm{MSO}[\tau^{\star\star}]$*-formula*

$$\mathsf{b\text{-}type}_\theta(X_1, \ldots, X_m)$$

such that for every otxx A, every b-node $t \in V(T^A)$*, and every compatible linear order* $\leq$ *of A, if* $(P_1, \ldots, P_m)$ *is the type partition of* $N_+(t)$*, then*

$$A_{(t)} \models \mathsf{b\text{-}type}_\theta(P_1, \ldots, P_m) \iff \mathrm{tp}^{\mathrm{MSO}}_q(A_t, \leq) = \theta.$$

The proof, which can be found in the appendix, uses (somewhat tedious) Ehrenfeucht-Fraïssé game arguments.

Note that the vocabulary of the formula $\mathsf{b\text{-}type}$ in the lemma is $\tau^{\star\star}$ and not $\tau^{\star\star} \cup \{\leq\}$. It will be important throughout the proofs of the lifting theorems to keep track of the vocabularies. The next lemma is a similar result for a-nodes, but there is one big difference: the formula $\mathsf{a\text{-}type}$ we obtain has vocabulary $\tau^{\star\star} \cup \{\leq\}$ and not just $\tau^{\star\star}$. This means that, at least a priori, the formula is not order-invariant. For b-nodes, the formula $\mathsf{b\text{-}type}_\theta$ does not depend on the order, because for b-nodes t every compatible linear order $\leq$ coincides with $\preceq$ on $V(A^{\star\star}_{(t)})$. The proof of the lemma is a straightforward adaptation of the proof of the previous lemma.

Lemma 3.5 (Ordered type composition at a-nodes)**.** *For every* $\theta \in \Theta$ *there is an* $\mathrm{MSO}[\{\leq\}]$*-formula*

$$\mathsf{a\text{-}type}_\theta(X_1, \ldots, X_m)$$

such that for every otxx $A^{\star\star}$*, every a-node* $t \in V(T^A)$*, and every compatible linear order* $\leq$ *of* $V(A^{\star\star})$ *the following holds. If for every* $i \in [m]$ *we let* P_i *be the set of all* $u \in N_+(t)$ *such that* $\mathrm{tp}^{\mathrm{MSO}}_q(A^{\star\star}_u, \leq) = \theta_i$*, then*

$$(N_+(t), \leq) \models \mathsf{a\text{-}type}_\theta(P_1, \ldots, P_m) \iff \mathrm{tp}^{\mathrm{MSO}}_q(A^{\star\star}_t, \leq) = \theta.$$

3.3 Order-invariant type composition

Recall from Section 2.3 the definition of order-invariant types and the characterization of order-invariant equivalence that we gave in Lemma 2.1. We continue to adhere to the assumptions made in the previous subsections (otxx have segmented tree decompositions of adhesion at most k, q is sufficiently large, $\Theta = \{\theta_1, \dots, \theta_m\} = \mathrm{TP}^{\text{MSO}}(\tau^{\star\star}, q)$) and use the same notation.

Recall that, since q is sufficiently large and the class of otxxs is MSO-definable, if A is an otxx and $A' \equiv_q^{\text{MSO}} A$ then A' is an otxx. This implies that if $A \equiv_q^{<\text{-inv-MSO}} A'$ then all structures appearing in a sequence witnessing this equivalence (cf. Lemma 2.1(3)) are otxxs. The same is true for sub-otxxs. However, it is not clear that all linear orders appearing in such a witnessing sequence are compatible. In other words, it is not clear that order invariance on otxxs coincides with invariance with respect to all compatible orders. For this reason, we need to introduce a finer equivalence relation $\equiv_{co}$, *compatible-order equivalence*. For two sub-otxx A, A', we let $A \equiv_{co} A'$ if there is a sequence $A_0, \dots, A_\ell$ of sub-otxxs and compatible linear orders $\leq_i, \leq_i'$ of A_i such that $A = A_0$ and $A' = A_\ell$ and $(A_{i-1}, \leq_{i-1}) \equiv_q^{\text{MSO}} (A_i, \leq_i')$ for all $i \in [\ell]$.

Then clearly $A \equiv_{co} A'$ implies $A \equiv_q^{<\text{-inv-MSO}} A'$. We do not know whether the converse holds.

Let us call a type $\theta \in \Theta$ *realizable* if there is a sub-otxx A and a compatible linear order $\leq$ of A such that $\mathrm{tp}_q^{\text{MSO}}(A, \leq) = \theta$. We call $(A, \leq)$ a *realization* of θ. Two types $\theta, \theta' \in \Theta$ are *compatible-order equivalent* (we write $\theta \equiv_{co} \theta'$) if there are realizations $(A, \leq)$ of θ and $(A', \leq')$ of θ' such that $A \equiv_{co} A'$. Then $\equiv_{co}$ is an equivalence relation on the set of realizable types. We denote the equivalence class of a type $\theta \in \Theta$ by $\langle\theta\rangle_{co}$. Clearly, we have $\langle\theta\rangle_{co} \subseteq \langle\theta\rangle$.

Now let A be an otxx and $t \in V(T^A)$. We call a set $\Theta' \subseteq \Theta$ *compatible at* t if there is a compatible linear order $\leq$ of $V(A_t)$ such that $\theta := \mathrm{tp}_q^{\text{MSO}}(A_t, \leq) \in \Theta'$ and $\Theta' \subseteq \langle\theta\rangle_{co}$. Note that this implies that all $\theta' \in \Theta'$ are realizable.

A *cover* of a set N is a sequence $(P_1, \dots, P_m)$ of subsets of N such that $\bigcup_{i=1}^m P_i = N$. For an otxx A and node $t \in V(T^A)$, we call a cover $(P_1, \dots, P_m)$ of $N_+(t)$ *compatible* if for all $u \in N_+(t)$ the set $\{\theta_i \mid i \in [m] \text{ such that } u \in P_i\}$ is compatible at u. Observe that if $(P_1, \dots, P_m)$ is the type partition of $N_+(t)$ with respect to some compatible linear order then $(P_1, \dots, P_m)$ is a compatible cover.

Lemma 3.6 (Order-invariant type composition at b-nodes)**.** *For every $\theta \in \Theta$ there is an $\text{MSO}[\tau^{\star\star}]$-formula*

$$\mathsf{oi\text{-}b\text{-}type}_\theta(X_1, \dots, X_m)$$

such that for every otxx A, every b-node $t \in V(T^A)$, and every compatible cover $(P_1, \dots, P_m)$ of $N_+(t)$, the set of all $\theta \in \Theta$ such that $A_{(t)} \models \mathsf{oi\text{-}b\text{-}type}_\theta(P_1, \dots, P_m)$ is compatible at t.

The proof can be found in the appendix. The idea is that within the structure $A_{(t)}$ we can quantify over the possible type partitions to the children (they are just collections of sets) and then apply Lemma 3.4 to each of them individually.

Lemma 3.7 (Order-invariant type composition at a-nodes)**.** *For every $\theta \in \Theta$ there is an $\text{CMSO}[\emptyset]$-formula*

$$\mathsf{oi\text{-}a\text{-}type}_\theta(X_1, \dots, X_m)$$

such that for every otxx A, every a-node $t \in V(T^A)$, and every compatible cover $(P_1, \dots, P_m)$ of $N_+(t)$, the set of all $\theta \in \Theta$ such that $(N_+) \models \mathsf{oi\text{-}a\text{-}type}_\theta(P_1, \dots, P_m)$ is compatible at t.

The proof, which can be found in the appendix, builds on the ideas developed in the previous proofs, and in addition, crucially depends on the fact that ordered invariant MSO on word structures coincides with CMSO.

3.4 Proofs of the lifting theorems

Proof of Theorem 3.1. Let $\mathcal{C}$ be a class of structures over some vocabulary τ that admit CMSO-definable ordered tree decompositions and let φ be an order-invariant MSO-formula over τ. We show that there exists a CMSO-formula ψ, such that for every structure A from $\mathcal{C}$ we have $A \models \varphi \iff A \models \psi$.

First of all, we turn every structure A into an otx $A^\star$ with bounded adhesion, which is possible using a CMSO-transduction by the theorem's precondition. Using the transformations discussed in Section 3.1, we continue to turn $A^\star$ into an otx whose tree decomposition is segmented and, then, expand it into an otxx $A^{\star\star}$. Both transductions preserve the bounded adhesion property. Since A's relations are still present in $A^{\star\star}$ and we can distinguish the elements in $A^{\star\star}$ that are also in the original structure A from the elements that are added to $A^{\star\star}$ by the transductions, we can rewrite φ to a formula $\varphi^{\star\star}$ with $A \models \varphi \iff A^{\star\star} \models \varphi^{\star\star}$ for each $A \in \mathcal{C}$. In particular, $\varphi^{\star\star}$ is still an order-invariant MSO-formula.

Now we are ready to apply the Lemmas 3.6 and 3.7, which handle compatible sets of MSO-types. The compatible sets of types are used as substitutes for (full) order-invariant types. They contain enough information to decide whether the $<$-inv-MSO-formula $\varphi^{\star\star}$ is satisfied. Formally speaking, $A^{\star\star} \models \varphi^{\star\star}$ implies that for every Θ' that is compatible at the decomposition's root, there is a $\theta \in \Theta'$ with $\varphi^{\star\star} \in \theta$ (to test this element relation, $\varphi^{\star\star}$ is viewed as an MSO-formula over $\tau^{\star\star} \cup \{\leq\}$ since θ is a type over this vocabulary). In standard applications of composition theorems, we would now argue that we can first guess the types of all substructures of $A^{\star\star}$ using existential set quantifier and, then, check whether the types are guessed in a correct way at the leaves and in a consistent way at all intermediate nodes. Using this approach, we are normally able to define the unique type of the whole structure. It is the type that labels the root. Our approach works in a similar way, but it is not the whole order-invariant type of the structure that we define. Instead, we existentially guess compatible types for all nodes and, then, check whether the choice of types is consistent for each node. In order to do that, we use the order-invariant composition lemma for a-nodes (which uses CMSO-formulas) or b-nodes (which uses MSO-formulas) based on the fact whether the currently considered node is an a-node or a b-node, respectively. Since the formula $\varphi^{\star\star}$ is order-invariant, $A^{\star\star} \models \varphi^{\star\star}$ holds exactly if $\varphi^{\star\star} \in \theta$ for some $\theta \in \Theta'$ where Θ' is the compatible type of the root of $A^{\star\star}$. This results in a CMSO-formula $\psi^{\star\star}$ that is equivalent to $\varphi^{\star\star}$ on the expanded otxs. Since $\varphi^{\star\star}$ on A is equivalent to φ on A and CMSO-transductions preserve CMSO-definability (Fact A.2), we know that there exists a formula ψ on τ that is equivalent to φ on all structures from $\mathcal{C}$. ☐

Proof of Theorem 3.2. The arguments are the same as in the proof of Theorem 3.1, except that we need to avoid the use of CMSO-formulas. First of all, this is possible for the initial transduction that produces the otx $A^\star$ from A since the theorem only talks about MSO-definable ordered tree decompositions, not CMSO-definable ones. Second, we need to avoid the use of CMSO-formulas in the order-invariant compositions for a-nodes. During the proof of Lemma 3.7, we translate an $<$-inv-MSO-formula on colored sets into an equivalent CMSO-formula. If we start with an $<$-inv-FO-formula instead, then we are able to translate it into an equivalent MSO-formula at this point in the proof. This follows from the fact that FO has the same expressive power as $<$-inv-FO on the class of trees [1]. The resulting proof of Theorem 3.2 produces an MSO-formula instead of a CMSO-formula. ☐

4. Defining decompositions

From the beginning of Section 3 we know how ordered tree decompositions can be viewed as an extension of a structure's vocabulary by the new relations V_S, V_T, E_T, R_β, and $R_\preceq$. The goal of this section is to show that the relations defining the rooted tree decomposition can be defined via the means of a linear transduction with parameters. The following Section 5 then explains how such an (unordered) *tree extension* can be extended further with the relation $R_{\preceq}$ by using another transduction.

In this section, when we say that we can define a certain property, this is meant with respect to MSO-definability. In Section 4.1 we show a general transduction template (called *decomposition scheme*) for obtaining tree extensions of input graphs. Section 4.2 applies this to graphs of bounded tree width and shows in detail how an MSO-transduction into bags without clique separators can be defined. Section 4.3 shows how we obtain an MSO-transduction that extends a graph with a tree decomposition into 3-connected components, based on known MSO-definability results.

4.1 Defining tree decompositions via transductions

To extend graph structures $G = (V, E)$ with tree decompositions, we use a specific template of linear transductions with parameters that define tree extensions. We fix the output vocabulary $\{E, V_S, V_T, E_T, R_\beta\}$ and the width w of the linear decomposition. Let us list the formulas of the decomposition scheme and their semantics regarding the defined tree extension. We only describe the scheme here, the concrete formulas are then defined in the remainder of Section 4.

First we have a λ_{VALID} which has no free variables aside from the parameter of the transduction. It checks whether it satisfies the properties that we require for our decomposition. The decomposition is only defined for input structure and choices of parameters that satisfy this formula.

We have two formulas λ_V and λ_E which define which of the vertices (and edges, respectively) of the original graph are present in the resulting extended graph. Since our tree extensions must retain the original graph without any modifications, we always set $\lambda_V := \mathsf{true}$ and $\lambda_E := \mathsf{true}$.

In order to define our decomposition based on a graph without such an extension, we have to use elements from the universe of our input graph G as *representatives* for the vertices of the tree decomposition (which we shall henceforth call *nodes*). Since a decomposition might contain many more nodes than the original graph has vertices, we use a linear decomposition with some constant w many levels. Hence we have w-many formulas λ_N^i and a total of w^2-many formulas $\lambda_A^{i,j}$ (with $i, j \in [w]$): the N stands for the *nodes* of the tree and the corresponding formulas have one free variable aside from the parameter. The edges of the tree are called *arcs* (hence the index A) and have two free variables aside from the parameter.

As a reminder, the linear decomposition works by copying the original vertex set w times and defining a subset of each of these vertex sets via the formula $\lambda_N^i(v)$: It is satisfied for an input vertex v if, and only if, the node (v, i) exists in the tree decomposition. The nodes of the defined tree decomposition are the union over all the levels. In order to allow arbitrary arcs between all the levels of nodes, we need to specify $w \times w$-many formulas $\lambda_A^{i,j}$.

Finally, the formulas λ_β^i define the bags of the decomposition by assigning to each node (v, i) the set of vertices in G that get placed in its respective bag. They have two free variables aside from the parameter: a vertex v that is the representative of the node (v, i), and a vertex of the original graph x. The formula is satisfied if x is in the bag of (v, i).

We do not want to be limited by having only graph structures as valid input structures, but also be able to apply a tree extension. This plays a vital role in decomposing graphs step-by-step in Section 4.2. We thus say that our decomposition scheme has output vocabulary $\{E, V_S, V_T, E_T, R_\beta\}$ and an input vocabulary $\tau = \{E, V_S\} \cup \tau'$, where τ' might consist of other unused symbols. This gives us the flexibility that our input structure can be both just a graph $G = (V, E, V_S)$ (where V_S is a unary relation containing all elements of the universe V) and a graph that was already extended by a tree decomposition.

For a τ-input structure G, if $G \models \lambda_{\text{VALID}}$, then the resulting output structure of our transduction is given by $\Lambda[A] := (U, E, V_S, V_T, E_T, R_\beta)$ with

$$\begin{aligned}
V_S = V_{\Lambda[G]} &:= \{v \in V_S \mid G \models \lambda_V(v)\} = V_S(A),\\
E = E_{\Lambda[G]} &:= \{(u, v) \in E(G) \mid A \models \lambda_E(u, v)\} = E(A),\\
V_T = N_{\Lambda[G]} &:= \{(v, i) \in V_S \times [w] \mid G \models \lambda_N^i(v)\},\\
E_T = A_{\Lambda[G]} &:= \{((u, i), (v, j)) \in (V_S \times [w])^2\\
&\qquad \mid G \models \lambda_A^{i,j}(u, v), G \models \lambda_N^i(u) \text{ and } G \models \lambda_N^j(v)\},\\
R_\beta = \beta_{\Lambda[G]} &:= \{((u, i), v) \in (V_S \times [w]) \times V_S\\
&\qquad \mid G \models \lambda_\beta^i(u, v) \text{ and } G \models \lambda_N^i(u)\}.
\end{aligned}$$

4.2 Tree decompositions along clique separators

For a graph G, a *separator* is a set $S \in V(G)$ such that there exist two vertices $x, y \in V(G) \setminus S$ where all paths from x to y contain a vertex of S. A k-separator is a separator S of size $k = |S|$. A *clique* is a graph with an edge between every two vertices. We call a separator S that is a clique in G a *clique separator*. An *atom* is a connected graph that doesn't contain any clique separators.

Let $c \in \mathbb{N}$, which we use as an upper bound on the size of clique separators we consider. A *c-atom* is therefore a connected graph that does not contain clique separators of size at most c. A *maximal c-atom* of a graph G is a maximal induced subgraph $G[A]$ for some $A \subseteq V(G)$ that is a c-atom (that means, every extension of it contains a clique separator of size at most c). A *maximal atom* in a graph G is a maximal $|V(G)|$-atom. For every $c \in \mathbb{N}$, c-atoms are nonempty.

In this section we show that tree extensions into atoms are MSO-definable. We utilize an inductive construction of such a decomposition from [10], whose basic construction and properties we first cover briefly. Then we show that this exact decomposition is MSO-definable via a series of transductions (using the above decomposition scheme). The main result of this section is the following lemma:

Lemma 4.1. *Let $\mathcal{C}$ be a class of graphs of bounded tree width. There is an* MSO*-transduction Λ_{tx} that defines for every graph G from $\mathcal{C}$ a tree extension G_{tx} whose*

1. *subgraphs induced be the bags are atoms, and*
2. *adhesion sets are cliques.*

We use an inductive approach here: as intermediate steps, we define *tree extensions into c-atoms* (we write such a graph as G_{tx}^c), which have the above properties but the bags only induce c-atoms. On graphs of bounded tree width k, the clique size is at most $k+1$ and thus clearly $G_{tx}^{k+1} = G_{tx}$.

The main difficulty in proving Lemma 4.1 consists of properly defining a transduction that moves from G_T^c to G_T^{c+1} for some $c \in \mathbb{N}$. To achieve this, we need to guess new *representatives* of the graph that can be used as nodes of the tree decomposition. This happens not just a single time, but once before each of these inductive transduction steps. We require certain properties about the transduction's parameters, so we cover how these properties are also MSO-definable (via the formula λ_{VALID} from our decomposition scheme).

A tree decomposition into atoms along growing separator sizes. We use a variant of the tree decompositions from [10] into bags that are c-atoms; all of the following facts are proven there.

Fact 4.2. *For every $k \in \mathbb{N}$, there is a mapping that turns a graph G with tree width at most k into a tree decomposition (T, β) for G in which*

1. *subgraphs induced be the bags are atoms, and*
2. *adhesion sets are cliques.*

The proof of this fact is constructive. For any $c \in \mathbb{N}$ and graph G, the decomposition defines the tree decomposition (T_c, β) in the following way. The node set of T_c consists of all c-atoms of G and all *minimum clique separators* of size at most c (a minimum clique separators is an inclusion-wise minimal separator that is also a clique). We call the former the *atom nodes* and the latter the *clique nodes*.

An edge is inserted in T_c between every c-atom $G[A]$ and minimum clique separator C with $C \subseteq A$. For an atom node t, its bag $\beta(t)$ is the vertex set of the corresponding atom and for a clique node t, its bag $\beta(t)$ is the vertex set of the corresponding clique separator. The following fact shows that this indeed defines a tree decomposition into atoms of one larger size.

Fact 4.3. *For every positive $c \in \mathbb{N}$ and $(c-1)$-atom G, (T_c, β) is a tree decomposition for G.*

In this tree decomposition, atom nodes are only connected to clique nodes and vice versa, and all leafs are atom nodes. When decomposing a graph step by step like this, the clique nodes are never changed again and the atom nodes are potentially decomposed further (because a $(c-1)$-atom might have to be split up into multiple c-atoms). This is done by replacing each $(c-1)$-atom node t with the partial tree decomposition (T_c, β) on $G[\beta(t)]$, which results in a tree decomposition into c-atoms on all of G. In Section 4.1 we explain in detail how these partial decompositions are inserted into T_{c-1}.

To prove Lemma 4.1, we show how for each c, (T_{c+1}, β) can be defined via an MSO-transduction from G^c_T. The required properties of our decomposition then follow from Fact 4.2.

Rooting the decomposition using a parameter.

In the following, we define some auxiliary MSO-formulas that help express more complex properties. Clearly, there is an MSO-formula $\varphi_{\text{SEP}}(S)$ with $G \models \varphi_{\text{SEP}}(S)$ if and only if S is a separator in the graph G; and also an MSO-formula $\varphi_{\text{CLIQUE}}(C)$ which is satisfied if and only if C is a clique in G. By combining those two, it is easy to define a formula $\varphi_{\text{CLIQUE-SEP}}(S)$; and for each $i \in \mathbb{N}$, a formula $\varphi^i_{\text{CLIQUE-SEP}}(S)$ which is satisfied if and only if S is a clique separator of size exactly i. This, in turn, allows us to define the formulas $\varphi^i_{\text{ATOM}}(A)$ and $\varphi_{\text{ATOM}}(A)$, respectively, which express the similar property of A being an $(i$-)atom. All of these formulas can also be formulated relative to a subgraph of G.

Let us assume we have a graph G that is a c-atom for some c (possibly 0). It is useful to root the tree of the decomposition in one node since this lets us define a partial order on the tree nodes. To root one of the nodes, we may pick any vertex of the original graph that can be uniquely assigned to one of the $(c+1)$-atoms. We hence pick a root vertex r which is not in any $(c+1)$-clique separator. Such a vertex always exists because each $(c+1)$-separator S induces at least two connected components in $G[V(G) \setminus S]$, and iterating this separation process on any of the arising components always leaves us with a non-empty component without any more $(c+1)$-clique separators in the end. With the above auxiliary formulas, it is easy to express that a vertex r has the desired properties via a formula $\varphi^i_{\text{ROOT}}(r)$.

This process of defining a unique root node is extended to pick representatives for our decomposition. In the proofs of the below lemmas we want to pick a root node for each c-atom of a larger graph. For this it is important that we may pick a vertex from one of the $(c+1)$-atoms that contain a given clique $C \subseteq G$ of size c. By the cover and connectedness conditions in any tree decomposition, the induced subtree of nodes whose bags contain the full clique C is nonempty and connected. Thus the above arguments hold for this subtree as well, and thus given a clique C, we can find a vertex r from a $(c+1)$-atom that contains all of C. We can express this via a formula $\varphi^i_{\text{ROOT-WITH-CLIQUE}}(r, C)$.

A transduction that defines a tree decomposition into atoms. We prove Lemma 4.1 constructively. We first decompose an input graph G into its 0-atoms (connected components) via a basic MSO-transduction. The we successively decompose the c-atoms into $c+1$-atoms until we reach a decomposition into $(k+1)$-atoms, where k is the tree width of G. Each of these decomposition steps is defined via an MSO-transduction that follows the proof of Fact 4.3. The proofs of these lemmas are found in the appendix.

Lemma 4.4. *There is an MSO$[\{E\}, \tau]$-transduction Λ^0_{tx} that defines a tree extension into 0-atoms for an input graph G.*

Lemma 4.5. *Let $c \in \mathbb{N}$. If we have an MSO$[\{E\}, \tau]$-transduction Λ^c_{tx} that defines the tree extension into c-atoms for an input graph G, we can also define an MSO$[\{E\}, \tau]$-transduction Λ^{c+1}_{tx} into $(c+1)$-atoms.*

Proof of Lemma 4.1. Our Lemmas 4.4 and 4.5 together show that for every $c \in \mathbb{N}$, a transduction Λ^c_{tx} which defines the tree extension of a graph into its c-atoms is definable. For a graph of tree width k, a decomposition into $(k+1)$-atoms is equivalent to a decomposition into atoms (of arbitrary size).

Moreover, the decomposition was defined in a way that always alternates atom nodes and clique nodes. Hence each adhesion set is the subset of a clique node's bag, which is naturally still a clique. Thus all of the required properties of Lemma 4.1 are satisfied. (This is no surprise, since we defined precisely the decomposition from Fact 4.2.) □

4.3 Tree decompositions into 3-connected components

A graph is *k-connected* if it contains no separator of size up to $k-1$. A *k-connected component* of a graph is a maximal induced subgraph that is k-connected. It is long-known that tree decompositions into 2- and 3-connected components are definable in MSO (see for instance [6]). A *torso* of a node t in a tree decomposition is the induced subgraph in its bag extended by edges between all adjacent adhesion sets. That means, for each neighbor $u \in N(t)$ we add the edges of the complete graph on the intersection of bags $\beta(t) \cap \beta(u)$ to $G[\beta(t)]$ and receive the torso of t.

Fact 4.6. *There is an MSO-transduction $\Lambda_{3\text{CC}}$ that defines for every graph G a tree extension G_{3cc} whose*

1. *torsos are 3-connected or cycles, and*
2. *adhesion sets have size at most 2.*

5. Defining orderings

In the previous section, we proved how to define tree decompositions along clique separators and discussed how to define tree decompositions into 3-connected components. In the present section we further define total orders for the bags of these decompositions whenever our graphs have bounded tree width or exclude a $K_{3,\ell}$-minor for some ℓ. The latter covers planar graphs since they exclude the minor $K_{3,3}$.

5.1 Orderings definable in monadic second-order logic

Our bag orderings are based on applying the following result of Blumensath and Courcelle [2], which says that one can define in GSO a total ordering for graphs whose separability is bounded and that exclude certain minors.

A class $\mathcal{C}$ of graphs has the *bounded separability* property if there is a function $s\colon \mathbb{N} \to \mathbb{N}$, such that for all graphs $G \in \mathcal{C}$ and vertex sets $S \subseteq V(G)$, the number of components of $G[V(G) \setminus S$ is bounded by $f(|S|)$.

Fact 5.1. *Let $\mathcal{C}$ be a class of graphs with bounded separability that excludes $K_{\ell,\ell}$ as a minor for some $\ell \in \mathbb{N}$. There is a* GSO*-transduction $\Lambda'_{\text{ORDER-SEP}}$ that defines a total ordering for every $G \in \mathcal{C}$.*

Since GSO- collapses to MSO-logic on every class of graphs that exclude a fixed minor [7] (in fact, this applies to the more general class of uniformly k-sparse graphs, but we do not need them for our proofs), and neither bounded tree width graphs nor the $K_{3,\ell}$-minor-free graphs contain all complete bipartite minors, the previous fact provides us with a helpful ordering method in the case of an additional bounded separability.

5.2 Defining orderings in the bounded tree width case

In general, it is not possible to totally order atoms of bounded tree width in MSO or, even, CMSO. An example being a graph made up by n cycles of length n each connected to two universal vertices u_1 and u_2, but without an edge between u_1 and u_2. Graphs of this kind have constant tree width, are atom, but CMSO is not able to define a total ordering on the graph's vertices. In the following we show how to preprocess given graphs, such that the resulting atoms cannot be of the above kind. In particular, the preprocessing ensures that the two universal vertices in the above example have an edge between them and, thus, the considered graph is no longer an atom and, thus, already decomposed further.

Given a graph G, its *improved version* G' is the graph with vertex set $V(G') := V(G)$ and $vw \in E(G')$ holds for every two vertices $vw \in V(G')$ if, and only if, $vw \in E(G)$ or there are $\text{tw}(G) + 1$ internally disjoint paths between v and w in G. Computing the improved version of a graph is commonly part of algorithms that construct tree decompositions [14]. Pairs of vertices with $\text{tw}(G) + 1$ internally-disjoint paths between them always lie in a common bag in every tree decomposition. Thus, connecting pairs with this property with an edge does not change the tree decompositions of the graph and, moreover, it simplifies the task of constructing tree decompositions by producing a graph that is closer to embeddings into k-trees for $k = \text{tw}(G)$ than the original graph.

The MSO-transduction of the below proposition is based on the alternative characterization of connectivity in terms of separators from Menger's Theorem.

Proposition 5.2. *Let $k \in \mathbb{N}$. There is an* MSO*-transduction Λ_{IMPROVE} that defines the improved version for every graph of tree width at most k.*

Since MSO-transductions are closed under composition, we continue to work with the improved version of the graph instead of the original input graph.

The main reason behind the non-definability of total orderings in the above example lies in the fact that there is an unbounded number of subgraphs connected to each other via a small separator. This is not possible if we look at the bags of decomposed improved graphs.

Lemma 5.3. *Let $\mathcal{C}$ be a class of graphs of bounded tree width that are improved and atoms. Then $\mathcal{C}$ has the bounded separability property.*

Proof. Let $G \in \mathcal{C}$, $S \subseteq V(G)$, and let $G_1, \dots, G_n$ be the components of $G[V(G) \setminus S]$. If there is a component whose neighborhood in S is a single vertex or an edge, then $n = 1$ since G is an atom. In this case, the function f witnessing the bounded separability property needs to be at least 1. We are left with the case that every component is connected to a non-edge in S. Let $\{u, v\} \notin E(G[S])$ be a non-edge in S. Since G is improved, u and v are connected to at most $\text{tw}(G)$ components G_i, each containing a path connecting u and v that is internally-disjoint to the paths in the other components. Since there are $\binom{|S|}{2}$ candidate non-edges in S, we can upper-bound the number of components via $n \leq \binom{|S|}{2} \cdot \text{tw}(G)$. Since G's tree width is bounded by a constant $k \in \mathbb{N}$, the number of components is bounded in terms of the size of the separator via the function $f(|S|) := \binom{|S|}{2} \cdot k + 1$. ☐

We get the following from combining Lemma 5.3 with Fact 5.1.

Corollary 5.4. *Let $\mathcal{C}$ be a class of graphs of bounded tree width that are improved and atoms. There is an* MSO*-transduction $\Lambda_{\text{ORDER-TW}}$ that defines a total ordering for every $G \in \mathcal{C}$.*

Using the definable decompositions from the previous section and the just developed definable orderings, we can prove the results about bounded tree width and $<$-inv-MSO as well as $<$-inv-FO.

Theorem 5.5. *Let $\mathcal{C}$ be a class of graphs with bounded tree width. Then* $<$-inv-MSO $=$ CMSO *on $\mathcal{C}$.*

Proof. We show that $\mathcal{C}$ admits MSO-definable (hence CMSO-definable) ordered tree decompositions of bounded adhesion. This proves the theorem by applying Theorem 3.1, the lifting theorem for $<$-inv-MSO. Let k be a tree width bound for the graphs from $\mathcal{C}$. Instead of directly working with the structure A, we work with its Gaifman graph $G' = G(A)$, which has the same tree decompositions and is MSO-definable in A. We start to define the improved version G' in G using the MSO-transduction Λ_{IMPROVE} from Proposition 5.2. Next, we apply the transduction Λ_{tx} of Lemma 4.1 to G', which defines a tree extension G'_{tx}. The bags of the tree decomposition underlying the tree extension induce subgraphs that are atoms, and all adhesion sets are cliques. Since G and, hence, also G' has tree width k and graphs of tree width at most k only contain cliques of size at most $k + 1$, this implies a bounded adhesion (the adhesion is bounded by $k + 1$). In order to obtain an otx, we need to add total orderings for each bag. The bags of the tree decomposition obtained so far induce atoms and, since G' is an improved graph, these atoms are improved, too. That means, we can now use the transduction $\Lambda_{\text{ORDER-TW}}$ from Corollary 5.4 to obtain a total ordering for a given bag. In order to define orderings for all bags at the same time, we utilize the decomposition's bounded adhesion in the following way. Transduction $\Lambda_{\text{ORDER-TW}}$ orders a single bag by using a collection of set parameters, which are vertex colorings from which we can define the ordering. If we now want to order different neighboring bags at the same time, these vertex colorings might interfere in a way that makes it impossible to reconstruct an ordering.

We can do the following: as our (improved) graph has tree width at most k, it has coloring number at most $k+1$, and thus we can first guess a proper $(k+1)$-coloring where no two adjacent vertices have the same color. In particular, this implies that for each adhesion set S that occurs, all elements of S have different colors, because they are cliques. This gives us a way to simultaneously get a linear order of all adhesion sets by just fixing an order on the $(k+1)$ colors. Let us call the $(k + 1)$-colors we used this way our *adhesion colors*.

Now we guess a collection of colors that we would like to use to order the bags at the atom nodes. (The bags at clique nodes are just adhesion sets and thus already ordered by the adhesion colors.) We globally guess a suitable collection of colors. Let us call them *bag colors*. Within each bag B of the tree, we ignore the colors in the adhesion (upward) adhesion set S and instead consider all extensions of the coloring of the remaining nodes that lead to a linear order of the bag. There is only a bounded number of such extensions, and as the adhesion set S is linearly ordered, we can use the lexicographically smallest of these extensions to define the order. ☐

Theorem 5.6. *Let $\mathcal{C}$ be a class of graphs with bounded tree width. Then* $<$-inv-FO $\subseteq$ MSO *on* $\mathcal{C}$.

Proof. We use the proof of Theorem 5.5, but apply Theorem 3.2, the lifting theorem for $<$-inv-FO, instead of Theorem 3.1, the lifting theorem for $<$-inv-MSO. □

5.3 Defining orderings in the $K_{3,\ell}$-minor-free case

Like in the previous section, we want to apply Fact 5.1 to define total orderings, but this time use it for graphs that are 3-connected and do not contain $K_{3,\ell}$ as a minor for some $\ell \in \mathbb{N}$.

Lemma 5.7. *Let $\mathcal{C}$ be a class of graphs of 3-connected graphs that exclude a $K_{3,\ell}$-minor for some $\ell \in \mathbb{N}$. Then $\mathcal{C}$ has the bounded separability property.*

Proof. Let G be a 3-connected graph that does not contain $K_{3,\ell}$ for some $\ell \in \mathbb{N}$ as a minor and $S \subseteq V(G)$ with $k = |S|$. Now let $G_1, \dots, G_n$ be the components of $G[V(G) \setminus S]$. If $k \leq 2$, then $n \leq 1$ since G is 3-connected. If $k \geq 3$, 3-connectedness implies that every component is connected to at least 3 vertices in S. For the sake of contradiction, assume $n \geq \ell\binom{k}{3}$. Then there exists a subset T of S with $T = 3$ that is connected to at least ℓ components. By deleting everything except T and these components as well as contracting the components we produce the minor $K_{3,\ell}$. Since this is not possible, we have $n < \ell\binom{k}{3}$ and bounded separability via the function $f(|S|) := \ell|S|^3 + 1$. □

Corollary 5.8. *Let $\mathcal{C}$ be a class of 3-connected graphs that exclude a $K_{3,\ell}$-minor for some $\ell \in \mathbb{N}$. There is an* MSO*-transduction $\Lambda_{\text{ORDER-MINOR}}$ that defines a total ordering for every $G \in \mathcal{C}$.*

Combining the decompositions from the previous section with the ordering from Corollary 5.8, we can prove the following.

Theorem 5.9. *Let $\mathcal{C}$ be a class of graphs that exclude $K_{3,\ell}$ as a minor for some $\ell \in \mathbb{N}$. Then* $<$-inv-MSO $=$ CMSO *on* $\mathcal{C}$.

Proof. The proof is similar to the proof of Theorem 5.5, except that we need to use different transduction to define the tree decomposition and the ordering for the bags. Everything else remains the same since we still work with tree decompositions that have a bounded adhesion (in this case, the maximum adhesion is 2) and apply the lifting theorem for $<$-inv-MSO. For constructing a tree decomposition of bounded adhesion, we use Fact 4.6. For constructing the bag orderings, we follow the arguments from Theorem 5.5, but apply Corollary 5.8 to the torsos of the decomposition combined with the observation that graphs that exclude a minor can be properly colored with a bounded number of colors. □

Theorem 5.10. *Let $\mathcal{C}$ be a class of graphs that exclude $K_{3,\ell}$ as a minor for some $\ell \in \mathbb{N}$. Then* $<$-inv-FO $\subseteq$ MSO *on* $\mathcal{C}$.

Proof. Similar to the idea in the proof of Theorem 5.6. We take the proof of Theorem 5.9, but use the lifting theorem for $<$-inv-FO instead of the lifting theorem for $<$-inv-MSO. □

6. Conclusion

We proved two lifting definability theorems, which show that if a class $\mathcal{C}$ of structures admits MSO-definable ordered tree extensions, then $<$-inv-MSO $=$ CMSO and $<$-inv-FO $\subseteq$ MSO on $\mathcal{C}$. Using the lifting theorems in conjunction with definable tree decompositions and definable bag orderings, we were able to show that $<$-inv-MSO $=$ CMSO and $<$-inv-FO $\subseteq$ MSO hold for every class of graphs (and structures) of bounded tree width and every class of graphs (and structures) that exclude $K_{3,\ell}$ for some $\ell \in \mathbb{N}$ as a minor. The latter covers planar graphs.

Seeing the wide range of applications of the lifting theorems, it seems promising to apply or extend them in order to handle every graph class defined by excluding minors in future works. Moreover, an interesting question is whether the $<$-inv-FO $\subseteq$ MSO in Theorem 3.2 can be turned into an equality; possibly by using a logic more restrictive than MSO.

Acknowledgments

We thank Pascal Schweitzer for the idea of Lemma 5.3.

References

[1] M. Benedikt and L. Segoufin. Towards a characterization of order-invariant queries over tame graphs. *The Journal of Symbolic Logic*, 74 (1):168–186, 2009. URL http://www.jstor.org/stable/40378400.

[2] A. Blumensath and B. Courcelle. Monadic second-order definable graph orderings. *Logical Methods in Computer Science*, 10(1), 2014. doi: 10.2168/LMCS-10(1:2)2014.

[3] B. Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi: 10.1016/0890-5401(90)90043-H.

[4] B. Courcelle. The monadic second-order logic of graphs V.: on closing the gap between definability and recognizability. *Theoretical Computer Science*, 80(2):153–202, 1991. doi: 10.1016/0304-3975(91) 90387-H.

[5] B. Courcelle. The monadic second-order logic of graphs X: Linear orderings. *Theoretical Computer Science*, 160(1-2):87–143, 1996. ISSN 0304-3975. doi: 10.1016/0304-3975(95)00083-6.

[6] B. Courcelle. The monadic second-order logic of graphs XI: Hierarchical decompositions of connected graphs. *Theoretical Computer Science*, 224(1–2):35–58, 1999. doi: 10.1016/S0304-3975(98) 00306-5.

[7] B. Courcelle. The monadic second-order logic of graphs XIV: uniformly sparse graphs and edge set quantifications. *Theoretical Computer Science*, 299(1–3):1–36, 2003. doi: 10.1016/S0304-3975(02) 00578-9.

[8] B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, Cambridge, 2012. ISBN 978-0-521-89833-1.

[9] K. Eickmeyer, M. Elberfeld, and F. Harwath. Expressivity and succinctness of order-invariant logics on depth-bounded structures. In *Proceedings of MFCS 2014, Part I*, LNCS, pages 256–266, 2014. doi: 10.1007/978-3-662-44522-8_22.

[10] M. Elberfeld and P. Schweitzer. Canonizing graphs of bounded tree width in logspace. In *Proceedings of STACS 2016*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. to appear.

[11] M. Elberfeld, M. Grohe, and T. Tantau. Where first-order and monadic second-order logic coincide. In *Proceedings of the 27th Annual IEEE/ACM Symposium on Logic in Computer Science (LICS 2012)*, pages 265–274. IEEE Computer Society, 2012. doi: 10.1109/LICS. 2012.37.

[12] T. Ganzow and S. Rubin. Order-invariant MSO is stronger than counting MSO in the finite. In *Proceedings of 25th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2008)*, pages 313–324, 2008. doi: 10.4230/LIPIcs.STACS.2008.1353.

[13] L. Libkin. *Elements Of Finite Model Theory*. Springer, Heidelberg, 2004.

[14] D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Fixed-parameter tractable canonization and isomorphism test for graphs of bounded treewidth. In *Proceedings of the 55th IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, pages 186–195. IEEE Computer Society, 2014. doi: 10.1109/FOCS.2014.28.

[15] N. Schweikardt. A short tutorial on order-invariant first-order logic. In *Proc. CSR 2013*, pages 112–126, 2013. doi: 10.1007/ 978-3-642-38536-0_10.

A. Technical appendix for Section 2

Proof of Lemma 2.1. We prove (1) $\Longrightarrow$ (3) $\Longrightarrow$ (2) $\Longrightarrow$ (1).

For (1) $\Longrightarrow$ (3), suppose $\mathrm{tp}_q^{<\text{-inv-MSO}}(A) = \mathrm{tp}_q^{<\text{-inv-MSO}}(A')$. Let $\theta := \mathrm{tp}_q^{\mathrm{MSO}}(A, \leq)$ for some linear order $\leq$ of A and $\theta' := \mathrm{tp}_q^{<\text{-inv-MSO}}(A', \leq')$ for some linear order $\leq'$ of A'. Let $[A]$ be the class of all ordered τ-structures $(A'', \leq'')$ such that there is a sequence $A_0, \ldots, A_\ell$ of τ-structures and linear orders $\leq_i, \leq_i'$ such that $A = A_0$ and $A'' = A_\ell$ and $(A_{i-1}, \leq_{i-1}) \equiv_q^{\mathrm{MSO}} (A_i, \leq_i')$ for all $i \in [\ell]$, and let $[\theta]$ the set of types $\mathrm{tp}_q^{\mathrm{MSO}}(A'', \leq'')$ for $(A'', \leq'') \in [A]$. An easy induction on the length ℓ of the witnessing sequence shows that $[\theta] \subseteq \langle\theta\rangle$. Moreover, $[\theta]$ is order-invariant, and thus $[\theta] = \langle\theta\rangle$. Similarly, we define $[\theta']$ and prove that $[\theta'] = \langle\theta'\rangle$. Thus $[\theta] = [\theta']$, and this implies (3).

To prove (3) $\Longrightarrow$ (2), just note that all structures in a witnessing sequence satisfy the same order-invariant formulas.

Finally, to prove (2) $\Longrightarrow$ (1), suppose that $A \equiv_q^{<\text{-inv-MSO}} A'$. Let $\theta := \mathrm{tp}(A, \leq)$ for some linear order $\leq$ of A. Then $\mathrm{tp}_q^{<\text{-inv-MSO}}(A) = \langle\theta\rangle$. Let

$$\varphi_\theta := \bigvee_{\theta' \in \langle\theta\rangle} \bigwedge_{\psi \in \theta'} \psi.$$

Then φ_θ is an order-invariant MSO-sentence of quantifier rank q. As $(A, \leq) \models \bigwedge_{\psi\in\theta} \psi$, we have $(A, \leq) \models \varphi_\theta$, and thus A satisfies φ_θ as a sentence of $<$-inv-MSO. Hence A' satisfies φ_θ as a sentence of $<$-inv-MSO, and thus $(A', \leq') \models \varphi_\theta$ for some linear order $\leq'$ of A'. Thus there is a $\theta' \in \langle\theta\rangle$ such that $(A', \leq') \models \bigwedge_{\psi\in\theta'} \psi$, which implies $\mathrm{tp}_q^{\mathrm{MSO}}(A', \leq') = \theta'$. Hence $\mathrm{tp}_q^{<\text{-inv-MSO}}(A') = \langle\theta'\rangle = \langle\theta\rangle$. ☐

A.1 Technical appendix for Section 2.4

Fact A.1 follows from the idea that if there is an MSO$[\tau, \tau']$-transduction Λ from A of vocabulary τ to B of vocabulary τ' and an MSO-formula φ speaking over τ' that speaks about B, then $B \models \varphi$ if, and only if, $A \models \varphi'$, where φ' is the τ-formula in which any mention of the vocabulary τ' has been replaced by using the corresponding τ-formula from Λ. A formal proof of this fact is given in [8]. The transduction Λ from Fact A.2 can be achieved with a similar approach as described above, which plugs the formulas of Λ_1 into those that define Λ_2 and, thus, receiving an equivalent transduction to the composition that directly speaks about A.

Fact A.1 (MSO is closed under MSO-transductions)**.** *Let P be an MSO-definable property and Λ an MSO-transduction. Then the property $P' := \bigcup_{B\in P} \Lambda^{-1}[B]$ is MSO-definable.*

Fact A.2 (MSO-transductions are closed under composition)**.** *Let Λ_1 and Λ_2 be MSO-transductions where the output vocabulary of Λ_1 equals the input vocabulary of Λ_2. Then there is an MSO-transduction Λ with $\Lambda[A] = \Lambda_2[\Lambda_1[A]]$ for every structure A.*

The above transductions do not allow to introduce unary relations with more elements than the original structure's universe. Therefore we also consider *linear transductions* that slightly extend the above: We scale the universe by a constant factor w, obtaining additional elements of the form (a, i) where a is an element of the input structure, and $i \in \{1, \ldots, w\}$ (where we call i the *level* of the element (a, i)) Consequently, the relations R^i of B are no longer defined by a single formula λ_{R^i}, but by up to $w^{\mathrm{ar}(R^i)}$-many formulas $\lambda^j_{R^i}$ and set $R^i_{\Lambda[A]} := \{\overline{a} \in (U(A)\times[n])^{\mathrm{ar}(R^i)} \mid A \models \lambda^j_{R^i}(\overline{a})\}$ for each $i \in \{1, \ldots, n\}$ and $j \in w^{\mathrm{ar}(R^i)}$. So essentially, we obtain R^i by taking the union over all the satisfying interpretations of the formulas corresponding to R^i. The proofs of the above facts can easily be amended to also hold for linear transductions. We call the factor w the *width* of the transduction.

Transductions can have *parameters*. Formally, these appear as additional free element or set variables in each of the formulas λ of the decomposition. For example, an MSO-transduction that defines a rooted tree decomposition might require an element variable r that marks a specific element as belonging to the root node. We write $\Lambda[A, P]$ to denote the unique output structure resulting from the input structure A and the assignment of the free parameter variables given by P.

MSO-definability is maintained under transductions even if parameters are required. This is because when translating an MSO-sentence to a different vocabulary in Fact A.1, the parameters can simply be existentially quantified. Of course, if the parameters need to satisfy certain conditions to lead to a valid output structure for each input structure, these conditions have to be MSO-definable as well to restrict the existentially qualified variables. For this reason we introduce an additional formula λ_{VALID} into every transduction Λ with parameters and define $\Lambda(A) := \{\Lambda[A, P] \mid A \models \lambda_{\text{VALID}}(P)$ for some assignment of parameters $P\}$. Hence $B \in \Lambda(A)$ if any valid choice of parameters (as defined by the formula λ_{VALID}) leads to the output structure B. This will be important in Section 4 for choosing elements of the input structure which represent the nodes of the decomposition.

B. Technical appendix for Section 3

Proof of Lemma 3.4. For $0 \leq i \leq q$, let

$$\Theta_i := \mathrm{TP}^{\mathrm{MSO}}(\tau^{\star\star} \cup \{\leq\}, q - i, i),$$

and suppose that $\Theta_i = \{\theta_{i1}, \ldots, \theta_{im_i}\}$. Then $\Theta_0 = \Theta$ and $m_0 = m$, and we may assume that $\theta_{0j} = \theta_j$ for all $j \in [m]$. Let

$$q' := 1 + \sum_{i=1}^{q} (1 + m_i).$$

The core of the proof is the following claim.

Claim. *Let A, B be otxxs and $\leq^A, \leq^B$ compatible linear orders of A, B, respectively. Let $t \in V(T^A)$ and $t' \in V(T^B)$. Let $(P_{01}, \ldots, P_{0m_0})$and $(Q_{01}, \ldots, Q_{0m_0})$ be the type partitions of $N_+(t)$ and $N_+(t')$, respectively. Suppose that*

$$\mathrm{tp}_{q'}^{\mathrm{MSO}}(A_{(t)}, P_{01}, \ldots, P_{0m_0}) = \mathrm{tp}_{q'}^{\mathrm{MSO}}(B_{(t')}, Q_{01}, \ldots, Q_{0m_0}).$$
(1)

Then

$$(A_t, \leq^A) \equiv_q^{\mathrm{MSO}} (B_{t'}, \leq^B).$$

Proof. We shall prove that Duplicator has a winning strategy for the q-move MSO-game on $(A_t, \leq^A)$, $(B_{t'}, \leq^B)$.

It is crucial to note that the compatible linear orders $\leq^A, \leq^B$ coincide with the partial orders $\preceq^A, \preceq^B$ of the structures A, B when restricted to $V(A_{(t)}), V(B_{(t)})$, respectively. The reason for this is that the restrictions of $\preceq^A, \preceq^B$ to $V(A_{(t)}), V(B_{(t)})$, respectively, are linear orders, because t and t' are b-nodes. This means that the games on $(A_{(t)}, \leq^A), (B_{(t')}, \leq^B)$ and on $A_{(t)}, B_{(t')}$ are the same.

With every sequence $\overline{P} = (P_1, \ldots, P_p)$ of subsets of $V(A_t)$ we associate a sequence $P^+ := (P_{01}, \ldots, P_{0m_0}, P_{10}, P_{11}, \ldots, P_{1m_1}, P_{20}, \ldots , P_{(p-1)m_{p-1}} P_{p0}, P_{p1}, \ldots, P_{pm_p})$ of subsets of $V(A_{(t)})$ as follows:

- $P_{i0} := P_i \cap V(A_{(t)})$, for all $i \in [p]$;
- P_{ij} is the set of all $u \in N_+(t)$ such that
$$\mathrm{tp}_{q-i}^{\mathrm{MSO}}\left(A_u, \leq, P_1 \cap V(A_u), \ldots, P_i \cap V(A_u)\right) = \theta_{ij},$$
for all $i \in [p]$ and $j \in [m_i]$.

For every sequence $\overline{Q} = (Q_1, \ldots, Q_p)$ of subsets of $V(B_{t'})$ we define $\overline{Q}^+$ similarly, and for every position $\Pi = (P_i, Q_i)_{i\in[p]}$ of the MSO-game on $(A_t, \leq^A), (B_{t'}, \leq^B)$ we let Π^+ be the position of the MSO-game on $A_{(t)}, B_{(t)}$ consisting of $\overline{P}^+$ and $\overline{Q}^+$.

Our goal is to define a strategy for Duplicator in the q-move game on $(A_t, \leq^A), (B_{t'}, \leq^B)$ such that for every reachable position Π of length p the position Π^+ is a $1+\sum_{i=p+1}^{q}(1+m_i)$-move winning position for Duplicator in the MSO-game on $A_{(t)}, B_{(t')}$. Such a strategy will clearly be a winning strategy. We define the strategy inductively. For the initial empty position Π_0 we have

$$\Pi_0^+ = (P_{0j}, Q_{0j})_{j\in[m_0]},$$

and it follows from (1) that is is a q'-move winning position for Duplicator in the MSO-game on $A_{(t)}, B_{(t')}$.

So suppose now we are in a position $\Pi = (P_i, Q_i)_{i\in[p]}$ and the corresponding position Π^+ is a $1+\sum_{i=p+1}^{q}(1+m_i)$-move winning position for Duplicator in the MSO-game on $A_{(t)}, B_{(t')}$. Without loss of generality, we assume that in the $(p+1)$st move of the game on $(A_t, \leq^A), (B_{t'}, \leq^B)$, Spoiler chooses a set $P_{p+1} \subseteq V(A_t)$. (The case that he chooses a set $Q_{p+1} \subseteq V(B_{t'})$ is symmetric.)

We define the sets P_{ij} for $i \in [p+1]$ and $j \in \{0, \ldots, m_i\}$ as above. Suppose that, starting in position Π^+, in the game on $A_{(t)}, B_{(t')}$ Spoiler selects the sets $P_{(p+1)0}, \ldots, P_{(p+1)m_{p+1}}$ in the next $m_{p+1}+1$ moves. Let $Q_{(p+1)0}, \ldots, Q_{(p+1)m_{p+1}}$ be Duplicator's answers according to some winning strategy. Let $(\Pi^+)'$ be the resulting position of the MSO-game on $A_{(t)}, B_{(t')}$; this is a $1+\sum_{i=p+2}^{q}(1+m_i)$-move winning position for Duplicator.

As the sets $P_{(p+1)0}, \ldots, P_{(p+1)m_{p+1}}$ form a partition of $N_+(t)$, the sets $Q_{(p+1)1}, \ldots, Q_{(p+1)m_{p+1}}$ form a partition of $N_+(t')$, because otherwise Spoiler wins in the next round of the game (this explains the '1+' in the the number of moves of the game). Let $u' \in N_+(t')$ and $j = j(u')$ such that $u' \in Q_{(p+1)j}$. Then there is at least one $u \in P_{(p+1)j}$; otherwise Spoiler wins in the next round of the game. Let $j' \in [m_p]$ such that $u \in P_{pj'}$. Then

$$\mathrm{tp}_{q-p}(A_u, \leq, P_1 \cap V(A_u), \ldots, P_p \cap V(A_u)) = \theta_{pj'}, \quad (2)$$

$$\mathrm{tp}_{q-p-1}(A_u, \leq, P_1 \cap V(A_u), \ldots, P_{p+1} \cap V(A_u)) = \theta_{(p+1)j}. \quad (3)$$

Hence the type $\theta_{pj'}$ is the unique "restriction" of $\theta_{(p+1)j}$, and for all $u'' \in P_{(p+1)j}$ we have $u'' \in P_{pj'}$. This implies that $u' \in Q_{pj'}$, because otherwise Spoiler wins in the next round of the game. It follows that

$$\mathrm{tp}_{q-p}(B_{u'}, \leq, Q_1 \cap V(B_{u'}), \ldots, Q_p \cap V(B_{u'})) = \theta_{pj'}. \quad (4)$$

This implies that there is a $Q^{u'}_{(p+1)} \subseteq V(B_{u'})$ such that $\theta_{(p+1)j} =$

$$\mathrm{tp}_{q-p-1}(B_{u'}, \leq, Q_1 \cap V(B_{u'}), \ldots, Q_p \cap V(B_{u'}), Q^{u'}_{(p+1)})$$

We let $Q_{p+1} := Q_{(p+1)0} \cup \bigcup_{u'\in N_+(t')} Q^{u'}_{(p+1)}$. The new position is $\Pi' := (P_i, Q_i)_{i\in[p+1]}$. Then $(\Pi')^+ = (\Pi^+)'$, which is a $1+\sum_{i=p+2}^{q}(1+m_i)$-move winning position for Duplicator in the MSO-game on $A_{(t)}, B_{(t')}$. ⌟

The claim implies that $\mathrm{tp}_q^{\text{MSO}}(A, \leq^A)$ only depends on the type on $\mathrm{tp}_{q'}^{\text{MSO}}(A_{(t)}, P_1, \ldots, P_m)$. Let $\theta \in \Theta$. To define $\mathsf{b\text{-}type}_\theta$, let $\theta'_1, \ldots, \theta'_\ell$ be the list of all types $\theta' \in \mathrm{TP}^{\text{MSO}}(\tau, q', m)$ such that $\mathrm{tp}_{q'}^{\text{MSO}}(A_{(t)}, P_1, \ldots, P_m) = \theta'$ implies $\mathrm{tp}_q^{\text{MSO}}(A, \leq^A) = \theta$. Then $\mathrm{tp}_q^{\text{MSO}}(A, \leq^A) = \theta$ if and only if

$$A_{(t)} \models \bigvee_{i=1}^{\ell} \bigwedge_{\psi(X_1,\ldots,X_m)\in\theta'_i} \psi(P_1, \ldots, P_m).$$

□

Proof of Lemma 2.1. Let $\varphi(X_1, \ldots, X_m, Y_1, \ldots, Y_m)$ be an MSO–formula stating that $Y_i \subseteq X_i$ for all i, that the Y_i are mutually disjoint, and that $\bigcup_i Y_i = \bigcup_i X_i$. We let $\mathsf{oi\text{-}b\text{-}type}_\theta(X_1, \ldots, X_m) :=$

$$\exists Y_1 \ldots \exists Y_m \big(\varphi(X_1, \ldots, X_m, Y_1, \ldots, Y_m) \wedge \mathsf{b\text{-}type}_\theta(Y_1, \ldots, Y_m)\big).$$

Let A be an otxx, $t \in V(T^A)$ a b-node, and $(P_1, \ldots, P_m)$ a compatible cover of $N_+(t)$. Let Θ^t be the set of all θ such that $A_{(t)} \models \mathsf{oi\text{-}b\text{-}type}_\theta(P_1, \ldots, P_m)$. We need to prove that Θ^t is compatible at t.

For every $u \in N_+(t)$, let $\Theta^u := \{\theta_i \mid i \in [m]$ such that $u \in P_i\}$. As the cover $(P_1, \ldots, P_m)$ is compatible, for all u the set Θ^u is compatible at u. Thus there is a $\theta^u \in \Theta^u$ and a compatible linear order $\leq_u$ of A_u such that $\theta_u = \mathrm{tp}_q^{\text{MSO}}(A_u, \leq_u)$ and $\Theta^u \subseteq \langle\theta^u\rangle_{co}$. Let $\leq$ be the (unique) compatible linear order of A_t such that for all $u \in N_+(t)$, the restriction of $\leq$ to $V(A_u)$ is $\leq_u$. For every $i \in [m]$, let Q_i be the set of all $u \in N_+(t)$ such that $\theta^u = \theta_i$. Then $(Q_1, \ldots, Q_m)$ is a partition of $N_+(t)$ that refines the cover $(P_1, \ldots, P_m)$

Let $\theta_t := \mathrm{tp}_q^{\text{MSO}}(A_t, \leq)$. By Lemma 3.4, we have $A_{(t)} \models \mathsf{b\text{-}type}_{\theta^t}(Q_1, \ldots, Q_m)$, thus $A_{(t)} \models \mathsf{oi\text{-}b\text{-}type}_{\theta^t}(Q_1, \ldots, Q_m)$. Hence $\theta_t \in \Theta^t$.

We claim that $\Theta^t \subseteq \langle\theta_t\rangle_{co}$. Let $\theta \in \Theta^t$. We first prove that θ is realizable. As $A_{(t)} \models \mathsf{oi\text{-}b\text{-}type}_\theta(P_1, \ldots, P_m)$, there is a partition $(Q'_1, \ldots, Q'_m)$ of $N_+(t)$ that refines the cover $(P_1, \ldots, P_m)$ such that $A_{(t)} \models \mathsf{b\text{-}type}_\theta(Q'_1, \ldots, Q'_m)$. For each $u \in N_+(t)$, let $\theta'_u := \theta_i$ for the unique i such that $u \in Q'_i$. Then $\theta'_u \in \Theta^u$, and thus θ'_u is realizable. Let $(A'_u, \leq'_u)$ be a realization of θ'_u.

Let A' be the sub-otxx obtained from A_t by simultaneously replacing the sub-otxx A_u by the sub-otxx A'_u for all $u \in N_+(t)$ (see page 5 for a description of the replacement operation). As $\theta'_u \in \langle\theta_u\rangle$, we have $A_u \equiv_q^{\text{MSO}} A'_u$ and thus the induced substructures $A[\{u\} \cup \sigma^A(u)]$ and $A'_u[\{u'\} \cup \sigma^{A'_u}(t')]$, where u' is the root of A'_u, are isomorphic, and the replacement is possible. (We will use similar arguments about replacements below without mentioning them explicitly.) Let $\leq'$ be the (unique) compatible linear order of A' such that for all $u \in N_+(t)$, the restriction of $\leq'$ to $V(A'_u)$ is $\leq'_u$. Note that $(A'_{(t)}, \leq') = (A_{(t)}, \leq)$, because the linear orders $\leq$ and $\leq'$ both coincide with $\preceq^A$ on $V(A_{(t)})$. Thus $A'_{(t)} \models \mathsf{b\text{-}type}_\theta(Q'_1, \ldots, Q'_m)$, and by Lemma 3.4, $\mathrm{tp}_q^{\text{MSO}}(A', \leq') = \theta$. Thus θ is realizable.

It remains to prove that $\theta_t \equiv_{co} \theta$. For each $u \in N_+(t)$, we have $\mathrm{tp}_q^{\text{MSO}}(A_u, \leq_u) = \theta_u \equiv_{co} \theta'_u = \mathrm{tp}_q^{\text{MSO}}(A'_u, \leq'_u)$. Thus there is a sequence $A_{u0}, \ldots, A_{u\ell}$ of sub-otxxs and for each i two compatible linear orders $\leq_{ui}, \leq'_{ui}$ of A_{ui} such that $(A_{u0}, \leq_{u0}) = (A_u, \leq_u)$ and $(A_{u\ell}, \leq_{u\ell}) = (A'_u, \leq'_u)$ and $\mathrm{tp}_q^{\text{MSO}}(A_{u(i-1)}, \leq'_{u(i-1)}) = \mathrm{tp}_q^{\text{MSO}}(A_{ui}, \leq_{ui})$ for all $i \in [\ell]$. As we do not require the A_{ui} and the orders $\leq_{ui}, \leq'_{ui}$ to be distinct, we may assume without loss of generality that the sequences have the same length ℓ for all u. Let A_i be the structure obtained from A_t by simultaneously replacing A_u by A_{ui} for all $u \in N_+(t)$. Define linear orders $\leq_i, \leq'_i$ of A_i from the orders $\leq'_{ui}, \leq_{ui}$ and $\preceq^A$ in the usual way. The resulting sequence of structures and orders witnesses $\theta_t = \mathrm{tp}_q^{\text{MSO}}(A_t, \leq) \equiv_{co} \mathrm{tp}_q^{\text{MSO}}(A', \leq') = \theta$. To prove this, we apply Lemma 3.4 at every step. □

Proof of Lemma 3.7. Let $\theta \in \Theta$. Without loss of generality we may assume that θ is realizable.

We may view the MSO-formula $\mathsf{a\text{-}type}_\theta(X_1, \ldots, X_m)$ as an MSO-sentence of vocabulary $\sigma := \{\leq, X_1, \ldots, X_m\}$ where we interpret the X_i as unary relation symbols. Let χ^1_θ be the conjunction of this sentence with a sentence saying that $\leq$ is a linear order and the X_i partition the universe. Then all models of χ^1_θ are proper word structures. Let q_1 be an upper bound for the quantifier rank of the formulas χ^1_θ for $\theta \in \Theta$. Let $\Xi := \mathrm{TP}^{\text{MSO}}(\sigma, q_1)$, and for each

$\xi \in \Xi$, let $\langle\xi\rangle$ be the order-invariant type that contains ξ (that is, the inclusion-wise minimal order-invariant subset of Ξ that contains ξ). Now let $\xi_1, \dots, \xi_\ell$ be all $\xi \in \Xi$ that contain χ^1_θ, and let

$$\chi^2_\theta := \bigvee_{i=1}^{m} \bigvee_{\xi \in \langle\xi_i\rangle} \bigwedge_{\varphi \in \xi} \varphi.$$

Then χ^2_θ is order-invariant; we may view it has the "best order-invariant approximation" of χ^1_θ. The sentence χ^2_θ is over the vocabulary of words, but is invariant with respect to the ordering underlying the word. In other words, it is an order-invariant formula of vocabulary $\{X_1, \dots, X_m\}$ and, thus, equivalent to a CMSO-sentence χ^3_θ over the same vocabulary [5, Corollary 4.3].

We view $\chi^3_\theta = \chi^3_\theta(X_1, \dots, X_m)$ as a CMSO-formula of empty vocabulary with free variables $X_1, \dots, X_m$.

Let Θ_θ be the set of all $\theta' \in \Theta$ such that the following holds: there is an otxx A', an a-node $t' \in V(T^{A'})$, and a compatible linear order $\le'$ such that $(N_+(t')) \models \chi^3_\theta(P'_1, \dots, P'_m)$ for the type partition $(P'_1, \dots, P'_m)$ of $N_+(t')$ and $\mathrm{tp}^{\mathrm{MSO}}_q(A'_{t'}, \le') = \theta'$.

Then trivially, all $\theta' \in \Theta_\theta$ are realizable. Furthermore, $\theta \in \Theta_\theta$. To see this, recall that we assumed θ to be realizable and note that $\mathsf{a\text{-}type}_\theta(X_1, \dots, X_m)$ implies $\chi^3_\theta(X_1, \dots, X_m)$.

Claim 1. $\Theta_\theta \subseteq \langle\theta\rangle_{co}$.

Proof. Let $\theta' \in \Theta_\theta$. Let A' be an otxx, $t' \in V(T^{A'})$ an a-node, $\le'$ a compatible linear order of A', and $(P'_1, \dots, P'_m)$ the type partition of $N_+(t')$ such that $(N_+(t')) \models \chi^3_\theta(P'_1, \dots, P'_m)$ and $\mathrm{tp}^{\mathrm{MSO}}_q(A'_{t'}, \le') = \theta'$. Then $(N_+(t'), \le') \models \chi^2_\theta(P'_1, \dots, P'_m)$. Then there is a $(N, \le)$ and a partition $P_1, \dots, P_m$ of N such that

$$(N, \le, P_1, \dots, P_m) \equiv^{<\text{-inv-MSO}}_{q_1} (N_+(t'), \le', P'_1, \dots, P'_m)$$

and $(N, \le, P_1, \dots, P_m) \models \chi^1_\theta$. Equivalently, we have $(N, \le) \models \mathsf{a\text{-}type}_\theta(P_1, \dots, P_m)$.

By Lemma 2.1, there is an $\ell \in \mathbb{N}$ and for $0 \le i \le \ell$ sets N_i, partitions $(P_{i1}, \dots, P_{im})$ of N_i, and linear orders $\le_i, \le'_i$ of N_i such that $(N_1, \le_1, P_{11}, \dots, P_{1m}) = (N, \le, P_1, \dots, P_m)$ and $(N_\ell, \le'_\ell, P_{\ell 1}, \dots, P_{\ell m}) = (N_+(t'), \le', P'_1, \dots, P'_m)$ and $(N_{i-1}, \le'_{i-1}, P_{(i-1)1}, \dots, P_{(i-1)m}) \equiv^{\mathrm{MSO}}_{q_1} (N_i, \le_i, P_{i1}, \dots, P_{im})$.

We let $A^\ell := A'_{t'}$ and $t_\ell := t'$, and for $0 \le i < \ell$ we build a sub-otxx A^i as follows: we take a fresh node t_i, which will be the root of the tree T^{A^i}. We make $N_+(t_i) := N_i$ the set of children of t_i. The node t_i will be an a-node in A^i. We let $\beta^{A^i}(t_i) := \beta^{A'}(t')$. For each $u \in N_i$, say, with $u \in P_{ij}$, we take some $u' \in P'_j$. Note that P'_j is nonempty, because P_{ij} is nonempty and $(N_i, P_{i1}, \dots, P_{im}) \equiv^{\mathrm{MSO}}_{q_1} (N_+(t'), P'_1, \dots, P'_m)$. Then we take a copy A^i_u of $A'_{u'}$ and identify the copy of u' with u and the copy of $\sigma^{A'}(u')$ with the corresponding elements in $\beta^{A^i}(t_i) = \beta^{A'}(t')$. We define two compatible orders $\le_i, \le'_i$ on A^i that extend the corresponding orders on N_i and coincide with the linear order induced by $\le'$ on the copies of the sub-otxxs $A'_{u'}$ that we used to build A^i.

Then for $0 \le i < \ell$, all $j \in [m]$, and all $u \in N_i$, if $u \in P_{ij}$ then $(A^i_u, \le_i)$ and $(A^i_u, \le'_i)$ are copies of $(A'_{u'}, \le')$ for some $u' \in P'_j$, and hence

$$\mathrm{tp}^{\mathrm{MSO}}_q(A^i_u, \le_i) = \mathrm{tp}^{\mathrm{MSO}}_q(A^i_u, \le'_i) = \mathrm{tp}^{\mathrm{MSO}}_q(A'_{u'}, \le') = \theta_j\,.$$

Since

$$(N_{i-1}, \le'_{i-1}, P_{(i-1)1}, \dots, P_{(i-1)m}) \equiv^{\mathrm{MSO}}_{q_1} (N_i, \le_i, P_{i1}, \dots, P_{im})\,,$$

it follows from Lemma 3.5 that we have $\mathrm{tp}^{\mathrm{MSO}}_q(A^{i-1}, \le'_{i-1}) = \mathrm{tp}^{\mathrm{MSO}}_q(A^i, \le_i)$ for all i. Moreover, as we have $(N_0, \le_0) \models \mathsf{a\text{-}type}_\theta(P_{01}, \dots, P_{0m})$, again by Lemma 3.5 we have $\mathrm{tp}^{\mathrm{MSO}}_q(A^0, \le_0) = \theta$. Thus $\theta \equiv_{co} \theta'$. ⌟

Claim 2. *Let A be an otxx, $t \in V(T^A)$ an a-node, $\le$ a compatible linear order of A, and $(P_1, \dots, P_m)$ the type partition of $N_+(t)$. Then the set Θ_t of all $\theta \in \Theta$ such that*

$$(N_+(t)) \models \chi^3_\theta(P_1, \dots, P_m)$$

is compatible at t.

Proof. Let $\theta_t := \mathrm{tp}^{\mathrm{MSO}}_q(A_t, \le)$. Then by Claim 1, for all $\theta \in \Theta_t$ we have $\theta_t \in \Theta_\theta \subseteq \langle\theta\rangle_{co}$. As $\equiv_{co}$ is an equivalence relation, it follows that $\langle\theta_t\rangle_{co} = \langle\theta\rangle_{co}$. Thus $\Theta_t \subseteq \langle\theta_t\rangle_{co}$, and this shows that Θ_t is compatible at t. ⌟

The rest of the proof is very similar to the proof of Lemma 3.6. Again, we let $\varphi(X_1, \dots, X_m, Y_1, \dots, Y_m)$ be an MSO-formula stating that $Y_i \subseteq X_i$ for all i, that the Y_i are mutually disjoint, and that $\bigcup_i Y_i = \bigcup_i X_i$. We let $\mathsf{oi\text{-}a\text{-}type}_\theta(X_1, \dots, X_m) :=$

$$\exists Y_1 \dots \exists Y_m \big(\varphi(X_1, \dots, X_m, Y_1, \dots, Y_m) \wedge \chi^3_\theta(Y_1, \dots, Y_m)\big).$$

Let A be an otxx, $t \in V(A^T)$ an a-node, and $(P_1, \dots, P_m)$ a compatible cover of $N_+(t)$. Let Θ^t be the set of all $\theta \in \Theta$ such that $(N_+) \models \mathsf{oi\text{-}a\text{-}type}_\theta(P_1, \dots, P_m)$. We need to prove that Θ^t is compatible at t.

For every $u \in N_+(t)$, let $\Theta^u := \{\theta_i \mid i \in [m]$ such that $u \in P_i\}$. As the cover $(P_1, \dots, P_m)$ is compatible, for all u the set Θ^u is compatible at u. In particular, there is a $\theta^u \in \Theta^u$ and a compatible linear order $\le_u$ of A_u such that $\theta_u = \mathrm{tp}^{\mathrm{MSO}}_q(A_u, \le_u)$ and $\Theta^u \subseteq \langle\theta^u\rangle_{co}$. Let $\le^1$ be the (unique) compatible linear order of A_t such that for all $u \in N_+(t)$, the restriction of $\le^1$ to $V(A_u)$ is $\le_u$. For every $i \in [m]$, let Q_i be the set of all $u \in N_+(t)$ such that $\theta^u = \theta_i$. Then $(Q_1, \dots, Q_m)$ is the type partition of $N_+(t)$ in $(A_t, \le^1)$, and it refines the cover $(P_1, \dots, P_m)$.

By Claim 2, the set $\Theta_t(Q_1, \dots, Q_m)$ of all $\theta \in \Theta$ such that $(N_+(t)) \models \chi^3_\theta(Q_1, \dots, Q_m)$ is compatible at t. Thus there is a type $\theta_t \in \Theta_t(Q_1, \dots, Q_m)$ and a linear order $\le^2$ of A such that $\mathrm{tp}^{\mathrm{MSO}}_q(A_t, \le^2) = \theta_t$ and $\Theta_t(Q_1, \dots, Q_m) \subseteq \langle\theta_t\rangle_{co}$. As $\theta_t \in \Theta_t(Q_1, \dots, Q_m)$ we have $(N_+(t)) \models \chi^3_{\theta_t}(Q_1, \dots, Q_m)$ and thus $(N_+(t), \le) \models \mathsf{oi\text{-}a\text{-}type}_{\theta_t}(P_1, \dots, P_m)$. Thus $\theta_t \in \Theta^t$.

We need to prove that $\Theta^t \subseteq \langle\theta_t\rangle_{co}$. Let $\theta \in \Theta^t$. Then $A_{(t)} \models \mathsf{oi\text{-}a\text{-}type}_\theta(P_1, \dots, P_m)$, and thus there is a partition $(Q'_1, \dots, Q'_m)$ of $N_+(t)$ that refines the cover $(P_1, \dots, P_m)$ such that $(N_+(t)) \models \chi^3_\theta(Q'_1, \dots, Q'_m)$. Let $\Theta_t(Q'_1, \dots, Q'_m)$ be the set of all $\theta' \in \Theta$ such that $(N_+(t)) \models \chi^3_{\theta'}(Q'_1, \dots, Q'_m)$. Then $\theta \in \Theta_t(Q'_1, \dots, Q'_m)$. By Claim 2, the set $\Theta_t(Q'_1, \dots, Q'_m)$ is compatible at t. Thus there is a $\theta'_t \in \Theta_t(Q'_1, \dots, Q'_m)$ and a compatible linear order $\le^3$ of A such that $\mathrm{tp}^{\mathrm{MSO}}_q(A_t, \le^3) = \theta'_t$ and $\Theta_t(Q'_1, \dots, Q'_m) \subseteq \langle\theta'_t\rangle_{co}$.

It remains to prove that $\theta_t \equiv_{co} \theta'_t$. For each $u \in N_+(t)$, let $\theta_u := \mathrm{tp}^{\mathrm{MSO}}_q(A_u, \le^2)$ and $\theta'_u := \mathrm{tp}^{\mathrm{MSO}}_q(A_u, \le^3)$. Then $\theta_u = \theta_i$ for the unique i such that $u \in Q_i$ and $\theta'_u = \theta_{i'}$ for the unique i' such that $u \in Q_{i'}$. As both $(Q_1, \dots, Q_m)$ and $(Q'_1, \dots, Q'_m)$ refine the cover $(P_1, \dots, P_m)$ and the set Θ^u is compatible at u, we have $\theta_u \equiv_{co} \theta'_u$. Now we can form a sequence witnessing $\theta_t \equiv_{co} \theta'_t$ from sequences witnessing $\theta_u \equiv_{co} \theta'_u$ for the $u \in N_+(t)$ as in the proof of Lemma 3.6 (when we showed $\theta_t \equiv_{co} \theta$). □

C. Technical appendix for Section 4

Proof of Lemma 4.4. This is just a preprocessing step. We define a simple transduction to extend our input graph G with the tree decomposition that consists of a single bag for each connected component, which are the 0-atoms of the graph. In each connected component, we guess an arbitrary vertex v as a representative of the 0-atom and use $(v, 1)$ as a node in the tree. We also guess a single global root vertex r among the atom representatives as the root node representative. For each representative v, we define $\beta((v, 1))$ as the connected component in which v resides, which is trivially a 0-atom. We also connect every non-root node directly to the root node $(r, 1)$. Since the adhesion is empty, it is trivially a minimal clique. Thus this decomposition satisfies all conditions of a tree extension into 0-atoms. □

Proof of Lemma 4.5. We are given the transduction Λ^c_{tx} of width w with output vocabulary $\tau := \{E, V_S, V_T, E_T, R_\beta\}$, so we define a $[\tau, \tau]$-transduction Λ_{c+1} that leaves $G = (V_S, E)$ unchanged and only modifies the tree decomposition $D := (T, R_\beta)$ (with $T := (V_T, E_T)$). The $[\{E\}, \tau]$-transduction Λ^{c+1}_{tx} is then obtained using Fact A.1. In order to represent the old nodes, arcs and bags, we just copy the corresponding formulas of Λ^c_{tx} for all of its w-many levels. For all new nodes that Λ_{c+1} defines, we use two fresh levels l and $l + 1$ with $l = w + 1$ so there are no collisions with existing representatives.

Since Λ^c_{tx} defines the tree extension into c-atoms, each node $t \in V_T$ is either a clique node or an atom node. This can be expressed by an MSO-formula: all leaf nodes are defined as atom nodes, and in the tree T each atom node only has an arc to clique nodes and vice versa.

The only modification defined by Λ_{c+1} happens to atom nodes t where the graph induced on $R_\beta(t)$ contains a $(c + 1)$-clique separator; they get deleted and are replaced with a tree that is reconnected to all their previously incident clique nodes. All other atom nodes t_a are kept in T with the same connectivity and bags. All clique nodes t_c are kept in T with the same bags as well, but due to the above replacement every incident arc to a deleted atom node t will be replaced with an arc to a new $(c + 1)$-atom node. We thus check for each atom node $t = (v, i)$ whether the graph $C_t := G[\beta(t)]$ induced on its bag is already a $(c+1)$-atom. If this is not the case, we have to decompose it further as described below. To delete the node t, for every $i \in [w]$, we modify the formula $\lambda^i_N(x)$ such that it evaluates to false when $\beta((x, i))$ is not a $(c + 1)$-atom (using the formula $\varphi^{c+1}_{\text{ATOM}}$).

We now define the partial tree decomposition of C_t into $(c + 1)$-atoms, and then show how D' can be reinserted into the D by replacing the deleted node t. Keep in mind that C_t is a c-atom, so it is free of any clique separators up to size c.

Claim. *For every atom node t such that $C_t := G[R_\beta(t)]$ contains a $(c + 1)$-clique separator, the partial tree decomposition $D_t = (T_t, b_t)$ of C_t into $(c + 1)$-atoms is MSO-definable.*

Proof. We follow the the previously described tree decomposition from [10]. The decomposition is straightforward, but in order to define it in MSO, we have to pick representatives that can be used as nodes of our decomposition. For this we will use three parameters that allow us to guess (and verify) the required representatives.

As explained in Section 4.2, we can use a parameter $Root^{c+1}$ to guess a single vertex r that will be the representative of the root node of T_t. The bag (r, i) is precisely the unique $(c + 1)$-atom in which r resides. Note that we take care to pick one of the atoms that contain the full clique $R_\beta(s)$, where s is the parent node of t (this is needed for the later reinsertion of D_t into the larger tree). A valid r is thus expressible via the earlier formula $\varphi^i_{\text{ROOT-WITH-CLIQUE}}(r, R_\beta(s))$.

For every other $(c + 1)$-atom, we also have to pick a unique representative, but here we no longer know whether we have a single vertex that appears only in this atom. We thus have to find another define representatives a bit differently. Among all vertices of an atom, there is at least one vertex v that does not appear in the adhesion of the emerging atom node. This adhesion is precisely the unique $(c + 1)$-clique separator *closest to* v that separates r and v. We can define another auxiliary MSO-formula $\varphi^{c+1}_{\text{CLOSEST CLIQUE SEPARATOR}}(v, S)$ which is satisfied if and only if S is a $(c + 1)$-clique separator that separates v from r, and there is no $(c + 1)$-clique separator that separates a vertex of S from v (thus S is closest to v).

So for each $(c+1)$-atom C' of C_t, we can define the (nonempty) set $Z_{C'}$ of vertices of this atom which are not in the closest $(c + 1)$-clique-separator. For different $(c + 1)$-atoms, these sets are clearly distinct since otherwise, the vertex would have to appear in the adhesion of one of the atom nodes. Via a parameter $Atom^{c+1}$, we can guess a single vertex of this set for each $(c + 1)$-atom. An MSO-formula can verify that conversely, no two vertices of $Atom^{c+1}$ are in the same set $Z_{C'}$. This establishes the one-to-one correspondence of every $a \in Atom^{c+1}$ to the sets $Z_{C'}$ and thus, the atoms C'.

The transduction can thus use these representatives from $Root^{c+1}$ and $Atom^{c+1}$ to define the new atom nodes of T_t. We use the level $l = w+1$ of the transduction for this. It remains to define representatives for the clique nodes of T_t. For this we make the following observation: in the tree decomposition, each separator node has at least one atom node as its child. An easy way for finding a representative for the separator is thus to just re-use the representative of one of the child atom nodes as the representative of their shared closest $(c + 1)$-clique separator towards the root vertex r. We thus use a third parameter Sep^{c+1} to guess these representatives, and have to only verify $Sep^{c+1} \subseteq Atom^{c+1}$ and that no two vertices in Sep^{c+1} have the same closest $(c + 1)$-clique separator. We use the level $l+1$ for defining the clique nodes of T_t via the representatives Sep^{c+1}.

We have established a one-to-one-correspondence between our guessed representative vertices and the new nodes of the decomposition. We see that the validity of these parameters can be defined in MSO, and this is carried out in the formula λ_{VALID} of our transduction. Defining the bags is now simple: for an $(c+1)$-atom-representative, it is precisely the corresponding $(c + 1)$-atom and for a $(c + 1)$-clique-representative, it is precisely the closest $(c + 1)$-clique separator towards the root r.

We use the formulas $\lambda^{l,l+1}_A$ and $\lambda^{l+1,l}_A$ to define the arcs of T_t: There is an arc from a clique node $(u, l + 1)$ to an atom node (v, l) precisely u and v have the same (unique) closest $(c + 1)$-clique separator (to see this, remember how we defined the clique representatives). And there is an arc from an atom node t to a clique node t' precisely if $\beta(t') \subseteq \beta(t)$.

Note that the above defines the bags and connectivity of the tree in precisely the same way as the previously described tree decomposition from [10]. Fact 4.3 shows that this construction is indeed a tree decomposition into $(c + 1)$-atoms on the graph C_t. The partial tree decomposition D' is thus definable in MSO as part of the transduction Λ_{c+1}. ⌟

We now move from the view of the single c-atom C_t and its tree decomposition T_t to the global view on all of G. We can have arbitrarily many c-atoms in G, so we first have to ensure that we do not need fresh parameters for each of them. Instead of a single vertex, we can say that $Root^{c+1}$ guesses a whole set of root vertices: one vertex for each c-atom of G. Clearly, the one-to-one correspondence is definable just as easily as for the $(c + 1)$-atoms above; we can even pick precisely the atom representatives of Λ^c_{tx} which have a c-atom as its bag. $Atom^{c+1}$ is already a set, but since the sets $Z_{C'}$ can not overlap globally on G either, the parameter maintains the required properties in the global view as well; and of course the same holds for Sep^{c+1}. So the transduction that we defined so far correctly constructs all the individual partial tree decompositions. If we stopped defining the rest of the transduction here, the decomposition graph would now be a forest of these partial tree decompositions D_t, and of the connected components of $T[V_T \setminus R]$, where R is the set of removed c-atom nodes. So it only remains to define how this forest is merged back together into a single tree.

Let $t \in R$ be a deleted c-atom node and T_t the newly constructed tree of the partial decomposition D_t into $(c+1)$-atoms on the bag $C_t \coloneqq G[\beta(t)]$. We connect the nodes in the neighborhood $N(t)$ by defining the formulas $\lambda_A^{i,l}$ and $\lambda_A^{l,i}$ (for $i \in [w]$), respectively. Note that no arcs between levels $i \leq w$ and level $l+1$ are necessary, since $N(t)$ consists entirely of clique nodes and we want those to be connected only to the atom nodes (defined above to lie exclusively on level l) within T_t.

We define the formulas $\lambda_A^{i,l}(u,v)$ such that it checks if the node (u,i) has a deleted child node, and if additionally $v \in \mathit{Root}^{c+1}$. Clearly this creates an arc from the parent clique node of each deleted node t to precisely the root node of the corresponding tree T_t. This definition suffices because there can be no other new incoming arcs to the new nodes defined in each T_t.

For the formulas $\lambda_A^{l,i}$, we have to consider all former child nodes $s_1, \ldots, s_n$ of t. Each of their bags is a clique, and we would thus receive a valid tree decomposition if we connected each s_j to *any* atom node t_j of T_t such that $\beta(s_j) \subseteq \beta(s_j)$ for all $j \in [n]$. Among these potential choices of the compatible atom nodes for s_j, there exists a unique atom node t_j^* which lies closest to the root of T_t. Moreover, because the set of nodes that include the clique $\beta(s_j)$ is connected in T_t, this node can be found in MSO by asking for a node whose bag includes $\beta(s_j)$, but whose parent node does not include $\beta(s_j)$. We can thus define the arcs to the s_j for precisely these nodes t_j^* in an MSO-definable way. This concludes the reintegration of the trees T_t and clearly, the transduction Λ_{c+1} now defines a tree.

Since we only reinsert partial tree decompositions D_t into an existing tree decomposition D, we know that the connectedness and cover conditions still hold on $T[V_T \setminus R]$ and each new T_t. The cover condition is maintained globally in the newly constructed decomposition because we only replaced each single bag of a deleted node t with a tree decomposition D_t on precisely this bag.

We have defined the root node of each T_t to contain the full clique that is the bag of the parent of the deleted node t. On the other hand, the arcs between the new atom nodes and old child clique nodes s_j has been defined precisely such that the bag of the s_j lie completely within the atom nodes that link to them. So the connectedness condition is also maintained, which concludes the proof that Λ_{c+1} indeed defines a tree decomposition into $(c+1)$-atoms.

□