arXiv:1606.06557v1 [cs.LO] 21 Jun 2016

# Order Invariance on Decomposable Structures

Michael Elberfeld Marlin Frickenschmidt Martin Grohe

RWTH Aachen University
Aachen, Germany
{elberfeld|frickenschmidt|grohe}@informatik.rwth-aachen.de

October 15, 2018

**Abstract**

Order-invariant formulas access an ordering on a structure's universe, but the model relation is independent of the used ordering. They are 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) in general, 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.

*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 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 of whether there is a logic that captures all unordered properties decidable in polynomial time falls into this category.

Gurevich [17] proved that order-invariant FO (<-inv-FO) is more expressive than FO (also see [24] for details). 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 [15]. Since it is not possible to decide, for a given FO-formula, whether it is order-invariant or not, 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. 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 [7] 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 [12]. 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 other 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. We 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, 5, 6], 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, possibly using arbitrary parameters (which may be sets in the case of MSO-definability). This is a much weaker assumption than having bounded width, and it 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-composition 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 (formally stated as Theorems 5.6 and 5.7) apply to classes of graphs of bounded tree width. For the proof, we show that one can define tree decompositions of bounded adhesion in MSO, where the bags admit MSO-definable total orderings. Let us remark that in proving these results we do not rely on the MSO-definability of width-bounded tree decompositions, a result announced by Lapoire [18], but only proved recently (and independently of our work) by Bojańczyk and Pilipczuk [3] [4]. Benedikt and Segoufin [1] had shown earlier how to prove these results using the MSO-definability of width-bounded tree decompositions. Our second application of the lifting theorem is concerned with classes of graphs that, for some $\ell \in \mathbb{N}$, do not contain $K_{3,\ell}$ as a minor. This includes the class of planar graphs and all classes of graphs embedabble in a fixed surface [22, 23]. Using an

MSO-definable tree decomposition into 3-connected components due to Courcelle [8] 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 5.10) and $<$-inv-FO $\subseteq$ MSO (Theorem 5.11) 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.

# 2 Background

In the present section, we introduce the necessary background related to logical structures and graphs (Section 2.1), monadic second-order logic and its variants (Section 2.2), logical games and types (Section 2.3), and transductions (Section 2.4).

## 2.1 Structures and Graphs

A *vocabulary* $\tau$ is a finite set of *relational symbols* where an *arity* $\mathrm{ar}(R) \geq 1$ is assigned to each $R \in \tau$. A *structure* $A$ over a vocabulary $\tau$ consists of a finite set $U(A)$, its *universe*, and a *relation* $R(A) \subseteq U(A)^{\mathrm{ar}(R)}$ for every $R \in \tau$. We sometimes write $R(A)$ by $R^A$, in particular if $R$ is a symbol like $\leq$.

An *expansion* of a $\tau$-structure $A$ is a $\tau'$-structure $A'$ for some vocabulary $\tau' \supseteq \tau$ such that $U(A) = U(A')$ and $R(A) = R(A')$ for all $R \in \tau$. If $A$ is a $\tau$-structure and $V \subseteq U(A)$, then the *induced substructure* $A[V]$ is the $\tau$-structure with universe $U(A[V]) = V$ and relations $R(A[V]) := R(A) \cap V^{\mathrm{ar}(R)}$ for all $R \in \tau$. Furthermore, we let $A \setminus V := A[U(A) \setminus V]$.

*Graphs* $G$ are structures over the vocabulary $\{E\}$ with $\mathrm{ar}(E) = 2$. When working with graphs, we also write $V(G)$ for the graph's universe (its set of *vertices*) and call $E(G)$ its set of *edges*. The graphs we are working with are *undirected*. That means, for every two vertices $v$ and $w$, we have $(v,w) \in E(G)$ if, and only if, $(w,v) \in E(G)$ and $(v,v) \notin E(G)$. The *Gaifman graph* $G(A)$ of a structure $A$ has vertices $V(G(A)) = U(A)$ and for every pair of distinct elements $v$ and $w$ that are part of a common tuple in $A$, we insert the edge $(v,w)$ into $E(G(A))$; thus, $G(A)$ is always undirected.

A *tree decomposition* $(T,\beta)$ of a structure $A$ is a tree $T$ together with a labeling function $\beta\colon V(T) \to 2^{U(A)}$ satisfying the following two conditions. (*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 $(v_1,\dots,v_r)$ of a relation in $A$, there is a $t \in V(T)$ with $\{v_1,\dots,v_r\} \subseteq \beta(t)$. 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. The set $N^T(t)$ of *neighbors* of a node $t$ in a directed tree $T$ consists of its children (if $t$ is not a leaf) and its parent (if $t$ is not the root). The set of children of a node $t$ in a directed tree $T$ is denoted by $N_+^T(t)$. We omit $^T$ from $N^T(t)$ and $N_+^T(t)$ if it is clear from the context. The sets $\beta(t)$ for every $t \in V(T)$ are the *bags* of the tree decomposition. The *width* of the tree decomposition is $\max_{t \in V(T)} |\beta(t)| - 1$ and its *adhesion* is $\max_{(t,u) \in E(T)} |\beta(t) \cap \beta(u)|$. The *tree width*, $\mathrm{tw}(A)$, of a structure $A$ is the minimum width of a tree decomposition for it. Structures $A$ and their Gaifman graphs $G(A)$ have the same tree decompositions. In particular $\mathrm{tw}(A) = \mathrm{tw}(G(A))$. The *torso* of a node $t \in V(T)$ in a tree decomposition $D = (T,\beta)$ for a structure $A$ with Gaifman graph $G = G(A)$ is $G[\beta(t)]$ together with edges between all pairs $v,w \in \beta(t) \cap \beta(u)$ for $u \in N(t)$.

## 2.2 Monadic Second-Order Logic and its Variants

*Monadic second-order logic* (MSO-logic) is defined by taking all second-order formulas without second-order quantifiers of arity 2 and higher. More specifically, to define its syntax, we use *element variables* $x_i$ for $i \in \mathbb{N}$ and *set variables* $X_i$ for $i \in \mathbb{N}$. *Formulas of* MSO*-logic* (MSO*-formulas*) over a vocabulary $\tau$ are inductively defined as usual (see, for example, [19]). Such formulas are also called MSO$[\tau]$-formulas to indicate the vocabulary along with the logic. The set of *free variables* of an MSO-formula $\varphi$, denoted by free$(\varphi)$, contains the variables of $\varphi$ that are not used as part of a quantification. By renaming a formula's variables, we can always assume free$(\varphi) = \{x_1,\dots,x_k,X_1,\dots,X_\ell\}$ for some $k,\ell \in \mathbb{N}$; we write $\varphi(x_1,\dots,x_k,X_1,\dots,X_\ell)$ to indicate that the free variables of $\varphi$ are exactly $x_1$ to $x_k$ and $X_1$ to $X_\ell$. Given an MSO-formula $\varphi(x_1,\dots,x_k,X_1,\dots,X_\ell)$, $A \models \varphi(a_1,\dots,a_k,A_1,\dots,A_\ell)$ indicates that $A$ together with the assignment $x_i \mapsto a_i$, for $i \in \{1,\dots,k\}$, and $X_i \mapsto A_i$, for $i \in \{1,\dots,\ell\}$, to $\varphi$'s free variables satisfies $\varphi$. A formula without free variables is also called a *sentence*.

*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 set variable are used in the same way.

Let $\tau$ be a vocabulary and $\leq$ a binary relation symbol not contained in $\tau$. An MSO-sentence $\varphi$ of vocabulary $\tau \cup \{\leq\}$ is *order-invariant* if for all $\tau$-structures $A$ and all linear orders $\leq_1, \leq_2$ of $U(A)$ we have $(A,\leq_1) \models \varphi$ if, and only if, $(A,\leq_2) \models \varphi$. We can now form a new logic, *order-invariant monadic second-order logic* ($<$-inv-MSO-logic), where the sentences of vocabulary $\tau$ are the order-invariant sentences of vocabulary $\tau \cup \{\leq\}$, and a $\tau$-structure $A$ satisfies an order-invariant sentence $\varphi$ if $(A,\leq)$ satisfies $\varphi$ in the usual sense for some (and hence for all) linear orders $\leq$ of $U(A)$. 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 $\tau$). 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 an ordinary relation symbol and the sentences of $<$-inv-MSO-logic of vocabulary $\tau$ 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 sentences of MSO-logic and $<$-inv-MSO-logic, respectively, that do not contain set variables.

## 2.3 Games and Types

The *quantifier rank* of an MSO-formula $\varphi$, denoted by qr$(\varphi)$, is the maximum number of nested quantifiers in $\varphi$. 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 $m \leq c$ are used in the modulo-counting atoms.

It will sometimes be convenient to use versions of MSO and CMSO without element variables (see, for example, [25]). In particular, in the context of Ehrenfeucht-Fraïssé games. We will freely do so. We assume that the reader is familiar with the characterizations of MSO-equivalence and CMSO-equivalence by Ehrenfeucht-Fraïssé games (see, for example, [11, 15]). Corresponding to the versions of the logics without element 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 $Q_i \subseteq U(B)$. 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 $\tau$ be a vocabulary and $q, p \in \mathbb{N}$. Then for all $\tau$-structures $A$ and sets $P_1, \dots, P_p \subseteq U(A)$, the MSO-*type of* $(A, P_1, \dots, P_p)$ *of quantifier rank* $q$ is

$$\mathrm{tp}_q^{\mathrm{MSO}}(A, P_1, \dots, P_p) := \big\{ \varphi(X_1, \dots, X_p) \mid \varphi \text{ is MSO-formula with } \mathrm{qr}(\varphi) \leq q \text{ and } A \models \varphi(P_1, \dots, P_p) \big\}.$$

Moreover, the class of all types over $\tau$ with respect to rank $q$ and $p$ free set variables is

$$\mathrm{TP}^{\mathrm{MSO}}(\tau, q, p) := \big\{ \mathrm{tp}_q^{\mathrm{MSO}}(A, P_1, \dots, P_p) \mid A \text{ is } \tau\text{-structure}, P_1, \dots, P_p \subseteq U(A) \big\},$$

and we let $\mathrm{TP}^{\mathrm{MSO}}(\tau, q) := \mathrm{TP}^{\mathrm{MSO}}(\tau, q, 0)$. For $q, c \in \mathbb{N}$, we say that a CMSO-formula has *rank* at most $(q, c)$ if it has quantifier rank at most $q$ and only contains modulo-counting atoms $C_m(X)$ with $m \leq c$. Based on this notion of rank, we define the CMSO-type $\mathrm{tp}_{q,c}^{\mathrm{CMSO}}(A, P_1, \dots, P_p)$, and sets $\mathrm{TP}^{\mathrm{CMSO}}(\tau, q, c, p)$ and $\mathrm{TP}^{\mathrm{CMSO}}(\tau, q, c)$.

Note that $\mathrm{tp}_q^{\mathrm{MSO}}(A, P_1, \dots, P_p) = \mathrm{tp}_q^{\mathrm{MSO}}(B, Q_1, \dots, Q_p)$ 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 $\mathrm{tp}_q(A) = \mathrm{tp}_q(B)$ if, and only if, $A \equiv_q^{\mathrm{MSO}} B$. Similar remarks apply to CMSO-types.

For a vocabulary $\tau$ and a binary relation symbol $\leq \notin \tau$, we say that a subset $I \subseteq \mathrm{TP}^{\mathrm{MSO}}(\tau \cup \{\leq\}, q)$ is *order-invariant* if for all $\tau$-structures $A$ and all linear orders $\leq, \leq'$ of $A$ we have $\mathrm{tp}_q^{\mathrm{MSO}}(A, \leq) \in I$ if, and only if, $\mathrm{tp}_q^{\mathrm{MSO}}(A, \leq') \in I$. If $I$ is inclusion-wise minimal order-invariant, then we call it an *order-invariant type*. Note that every $\theta \in \mathrm{TP}^{\mathrm{MSO}}(\tau \cup \{\leq\}, q)$ is contained in exactly one order-invariant type, which we denote by $\langle \theta \rangle$. We set $\mathrm{TP}^{<\text{-inv-MSO}}(\tau, q) := \big\{ \langle \theta \rangle \bigm| \theta \in \mathrm{TP}^{\mathrm{MSO}}(\tau \cup \{\leq\}, q) \big\}$, the set of all order-invariant types. For a $\tau$-structure $A$, we call the set $\mathrm{tp}_q^{<\text{-inv-MSO}}(A) := \big\langle \mathrm{tp}_q^{\mathrm{MSO}}(A, \leq) \big\rangle$ 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.

**Lemma 2.1.** *For all $\tau$-structure $A, A'$, the following statements are equivalent.*

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

If $A \equiv_q^{<\text{-inv-MSO}} A'$, we say that sequences $(A_i)$, $(\leq_i)$, and $(\leq_i')$ as in statement 3 of Lemma 2.1 *witness* $A \equiv_q^{<\text{-inv-MSO}} A'$.

*Proof of Lemma 2.1.* We prove each of the implications from the chain (1) $\Longrightarrow$ (3) $\Longrightarrow$ (2) $\Longrightarrow$ (1).

For proving (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 $\tau$-structures $(A'', \leq'')$ such that there is a sequence $A_0, \dots, A_\ell$ of $\tau$-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 class of types $\mathrm{tp}_q^{\mathrm{MSO}}(A'', \leq'')$ for $(A'', \leq'') \in [A]$. An easy induction on the length $\ell$ 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 $\theta := \mathrm{tp}_q^{\mathrm{MSO}}(A, \leq)$ for some linear order $\leq$ of $A$. Then $\mathrm{tp}_q^{<\text{-inv-MSO}}(A) = \langle \theta \rangle$. Let $\varphi_{\langle \theta \rangle} := \bigvee_{\theta' \in \langle \theta \rangle} \varphi_{\theta'}$ with $\varphi_{\theta'} := \bigwedge_{\psi \in \theta'} \psi$. Then $\varphi_{\langle \theta \rangle}$ is an order-invariant MSO-sentence of quantifier rank $q$. As $(A,\leq) \models \varphi_\theta$, we have $(A,\leq) \models \varphi_{\langle\theta\rangle}$, and thus $A$ satisfies $\varphi_{\langle\theta\rangle}$ as a sentence of $<$-inv-MSO. Hence $A'$ satisfies $\varphi_{\langle\theta\rangle}$ as a sentence of $<$-inv-MSO, and thus $(A',\leq') \models \varphi_{\langle\theta\rangle}$ for some linear order $\leq'$ of $A'$. Thus there is a $\theta' \in \langle\theta\rangle$ such that $(A',\leq') \models \varphi_{\theta'}$, which implies $\mathrm{tp}_q^{\mathrm{MSO}}(A',\leq') = \theta'$. Hence $\mathrm{tp}_q^{<\text{-inv-MSO}}(A') = \langle\theta'\rangle = \langle\theta\rangle$. □

### 2.4 Transductions

Transductions define new structures out of a given structure. We use $w$-copying MSO-transductions as defined in [10], but based on the below terminology. They are able to (1) enlarge the universe of a given structure by establishing $w$ copies of each element, (2) define relations over the new universe from the given structure, and (3) not only define a single structure, but a set of new structures parameterized by adding monadic relations to the given structure.

An MSO$[\tau,\tau']$-*transduction of width $w$ with $p$ parameters* for some $w,p \in \mathbb{N}$ is defined via a finite collection $\Lambda$ of MSO-formulas over $\tau \cup \{P_1,\dots,P_p\}$ where the relation symbols $P_j$ are monadic and not part of $\tau$. $\Lambda$ consists of a group of $w$ MSO-formulas $\lambda_U^1(x),\dots,\lambda_U^w(x)$ for defining the universe of a new structure and for each $R \in \tau'$ with some arity $r = \mathrm{ar}(R)$ a group of $w^r$ formulas $\lambda_R^{(i_1,\dots,i_r)}(x_1,\dots,x_r)$ for $(i_1,\dots,i_r) \in \{1,\dots,w\}^r$. Given a $\tau$-structure $A$ and $P_1,\dots,P_p \subseteq U(A)$, they define the universe of a $\tau'$-structure $\Lambda[A,P_1,\dots,P_p]$ via

$$U(\Lambda[A,P_1,\dots,P_p]) := \{(a,i) \in U(A) \times \{1,\dots,w\} \mid (A,P_1,\dots,P_p) \models \lambda_U^i(a)\}$$

and for each relation symbol $R \in \tau'$ the relation

$$R(\Lambda[A,P_1,\dots,P_p]) := \{((a_1,i_1),\dots,(a_r,i_r)) \in (U(A) \times \{1,\dots,w\})^r \mid A \models \lambda_R^{(i_1,\dots,i_r)}(a_1,\dots,a_r)\}\,.$$

Finally, by ranging over all possible parameters, $\Lambda$ defines the set

$$\Lambda[A] := \{\Lambda[A,P_1,\dots,P_p] \mid P_1,\dots,P_p \subseteq U(A) \wedge (A,P_1,\dots,P_p) \models \lambda_{\mathrm{VALID}}\}$$

for a given structure $A$ where $\lambda_{\mathrm{VALID}}$ is a formula that is also part of the transduction, which singles out the valid combinations of the given structure and parameters. Moreover, for a $\tau'$-structure $B$, we set $\Lambda^{-1}[B] := \{\tau\text{-structure } A \mid B \in \Lambda[A]\}$. For an element $(a,i)$, we call $i$ its *level*.

MSO-transductions preserve MSO-definability (formally stated by Fact 2.2) and they can be composed to form new transductions (formally stated by Fact 2.3). For a formal proof of Fact 2.3, which implies Fact 2.2, see [10]. The facts also hold if we replace all occurrences of MSO by CMSO.

**Fact 2.2** (MSO is closed under MSO-transductions)**.** *Let $\mathcal{P}$ be an MSO-definable property of $\tau'$-structures and $\Lambda$ an MSO$[\tau,\tau']$-transduction. Then the property of $\tau$-structures $\mathcal{P}' := \bigcup_{B\in\mathcal{P}} \Lambda^{-1}[B]$ is MSO-definable.*

**Fact 2.3** (MSO-transductions are closed under composition)**.** *Let $\Lambda_1$ be an MSO$[\tau,\tau']$-transduction and $\Lambda_2$ be an MSO$[\tau',\tau'']$ for some vocabularies $\tau,\tau',\tau''$. Then there is an MSO$[\tau,\tau'']$-transduction $\Lambda$ with $\Lambda[A] = \bigcup_{B\in\Lambda_1[A]} \Lambda_2[B]$ for every $\tau$-structure $A$.*

## 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 $\tau$-structure $A$ is a structure $A^\star$ that extends $A$ by a tree decomposition $(T^A,\beta^A)$ 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 $(T^A,\beta^A)$. 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 $V_S$ and $V_T$ are unary, $E_T$ and $R_\beta$ are binary, and $R_\preceq$ is ternary. Of course we assume that none of these symbols appears in $\tau$. In the $\tau^\star$-structure $A^\star$, these symbols are interpreted as follows:

$$
\begin{aligned}
&V_S(A^\star) := U(A),\\
&V_T(A^\star) := V(T^A),\\
&E_T(A^\star) := E(T^A),\\
&R_\beta(A^\star) := \big\{(t,v) \mid t \in V(T^A), v \in \beta^A(t)\big\}, \text{and}\\
&R_\preceq(A^\star) := \big\{(t,v,w) \mid t \in V(T^A) \text{ and } v,w \in \beta^A(t) \text{ with } v \preceq_t^A w\big\}.
\end{aligned}
$$

An MSO$[\tau,\tau^\star]$-transduction $\Lambda^\star$ *defines an otx (of adhesion at most $k$)* of a $\tau$-structure $A$ if every $B \in \Lambda^\star(A)$ is isomorphic to an otx of $A$ (of adhesion at most $k$) and $\Lambda^\star(A)$ is nonempty. We say that $\Lambda^\star$ *defines otxs (of adhesion at most $k$)* on a class $\mathcal{C}$ of $\tau$-structures 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 $k$ for some constant $k \in \mathbb{N}$) on $\mathcal{C}$. We make similar definitions for the logic CMSO.

We prove the following theorems, 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 $\subseteq$ MSO *on $\mathcal{C}$.*

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, Section 3.3 shows how these type-composition lemmas can be used in the context of order-invariance. Finally, Section 3.4 applies 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 Segmented Ordered Tree Extensions

Recall that we view the tree in a tree decomposition as directed. A tree decomposition $(T,\beta)$ 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 $u_1, u_2 \in N(t)$ 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 $(T,\beta)$ into a segmented tree decomposition $(T'',\beta'')$ as follows. In the construction, we view $T$ as an undirected tree. We will have $V(T) \subseteq V(T'')$. Thus we can direct the edges of $T''$ away from the root of $T$, which will remain the root of $T''$. We first contract all edges $tu \in E(T)$ with $\beta(u) \subseteq \beta(t)$, resulting in a decomposition $(T',\beta')$ 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. The nodes from the original tree $T$ are the b-nodes, and the nodes $v_{tu}$ are the a-nodes. We define $\beta''$ 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 $(T'', \beta'')$ is segmented. This transformation is definable by an MSO-transduction. Thus we may assume that the tree decompositions in ordered tree extensions are segmented, because there is an $\mathrm{MSO}[\tau^\star, \tau^\star]$-transduction $\Lambda_{\text{SEGMENT}}$ that transforms every otx into an otx where the tree decomposition is segmented.

For the rest of this section, we fix a vocabulary $\tau$ 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 $\tau$-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 $(T^A, \beta^A)$. We denote the descendant order in the tree $T^A$ of an otx $A^\star$ by $\trianglelefteq^A$. For every node $t \in V(T^A)$, we let $T^A_t$ be the subtree of $T^A$ rooted in $t$, that is, $T^A_t := T^A[\{u \in V(T^A) \mid t \trianglelefteq^A u\}]$. We let $\gamma^A(t)$, called the *cone* of $t$, be the union of all bags $\beta^A(u)$ for $u \in V(T^A_t)$. If $s$ is the parent of $t$ we let $\sigma^A(t) := \beta^A(t) \cap \beta^A(s)$; this is the *separator at* $t$. 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 $\sigma(t) = \beta(t) = \sigma(u)$.

We expand an otx $A^\star$ to a structure $A^{\star\star}$ over 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 that do not appear in $\tau$. 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

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

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 $S_1(A^{\star\star}), \ldots, S_k(A^{\star\star})$ 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 $i$th element of $\sigma(t)$ with respect to the partial order $\preceq$, which is a linear order when restricted to $\sigma(t) \subseteq \beta(t)$. Recall that we have $|\sigma(t)| \leq k$ by our general assumption that the adhesion of all otxs is 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 $\Lambda_{\text{EXPAND}}$ 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 $\tau$-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* sub-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 $A_t^{\star\star}$, 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 $A_t^{\star\star}$ by $B^{\star\star}$ in $A^{\star\star}$ just means deleting all elements in $U(A_t^{\star\star})$ except those in $\{t\} \cup \sigma^A(t)$, adding a disjoint copy of $B^{\star\star}$, and identifying the elements 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 $A_t^{\star\star}$ 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 $\tau$. To express isomorphism, we use the relations $S_1,\dots,S_k$ and the fact that the root of an otxx can be defined by a formula of quantifier rank 2. Thus in particular, if $\mathrm{tp}_q^{\mathrm{MSO}}(A_t^{\star\star}) = \mathrm{tp}_q^{\mathrm{MSO}}(B^{\star\star})$ for some $q \geq \mathrm{ar}(\tau)+1$, we can replace $A_t^{\star\star}$ 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 $(A_t^{\star\star}, \leq)$ denotes the induced substructure where $\leq$ is restricted to the sub-otxx $A_t^{\star\star}$. We can extend the replacement operation to such ordered expansions of otxxs; in the same way we replace a sub-otxx $A_t^{\star\star}$ by $B^{\star\star}$, we can replace a $(A_t^{\star\star}, \leq)$ by $(B^{\star\star}, \leq')$ for some compatible linear order $\leq'$ of $B^{\star\star}$.

## 3.2 Ordered Type Compositions

As all structures we are working with in this subsection 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 $t$, and we let $A_{(t)} = A[\beta(t) \cup N_+(t)]$.

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 sub-otxx of Lemma 3.3. This means that if $A$ is an otxx (or sub-otxx) and $A'$ an arbitrary $\tau^{\star\star}$-structure with $A \equiv_q^{\mathrm{MSO}} A'$, then $A'$ is an otxx (a sub-otxx) as well. Furthermore, if $t,t'$ are the root nodes of $A, A'$, respectively, then the 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 $(A, \leq) \equiv_q^{\mathrm{MSO}} (A', \leq')$ 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 $\Theta = \{\theta_1, \dots, \theta_m\}$.

Let $A$ be an otxx, $\leq$ a compatible linear order of $A$, and $N \subseteq V(T^A)$ (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 $\mathrm{tp}_q^{\mathrm{MSO}}(A_u, \leq) = \theta_i$. We call $(P_1, \dots, P_m)$ the *type partition* of $N$. (Note that some of the $P_i$ may be empty. We always allow partitions to have empty parts.) The following lemma extends classical type-composition theorems [21, 14] to our situation, where substructures are combined through b-nodes.

**Lemma 3.4** (Ordered type composition at b-nodes)**.** *For every $\theta \in \Theta$ there is an* $\mathrm{MSO}[\tau^{\star\star}]$*-formula* $\text{b-type}_\theta(X_1, \dots, X_m)$ *such that for every otxx $A$, every b-node $t \in V(T^A)$, and every compatible linear order $\leq$ of $A$, if $(P_1, \dots, P_m)$ is the type partition of $N_+(t)$, then*

$$A_{(t)} \models \text{b-type}_\theta(P_1, \dots, P_m) \text{ if, and only if, } \mathrm{tp}_q^{\mathrm{MSO}}(A_t, \leq) = \theta.$$

*Proof.* For $0 \leq i \leq q$, let $\Theta_i := \mathrm{TP}^{\mathrm{MSO}}(\tau^{\star\star} \cup \{\leq\}, q-i, i)$, and suppose that $\Theta_i = \{\theta_{i1}, \dots, \theta_{im_i}\}$. 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. If*

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

*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 $U(A_{(t)}), U(B_{(t)})$, respectively. The reason for this is that the restrictions of $\preceq^A, \preceq^B$ to $U(A_{(t)}), U(B_{(t)})$, 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 $U(A_t)$ we associate a sequence

$$\overline{P}^+ := (P_{01}, \ldots, P_{0m_0}, P_{10}, P_{11}, \ldots, P_{1m_1}, P_{20}, \ \ldots \ , P_{(p-1)m_{p-1}} P_{p0}, P_{p1}, \ldots, P_{pm_p})$$

of subsets of $U(A_{(t)})$ as follows:

– $P_{i0} := P_i \cap U(A_{(t)})$, for all $i \in [p]$;

– $P_{ij}$ is the set of $u \in N_+(t)$ with $\theta_{ij} = \mathrm{tp}_{q-i}^{\mathrm{MSO}}\big(A_u, \leq, P_1 \cap U(A_u), \ldots, P_i \cap U(A_u)\big)$ for all $i \in [p]$, $j \in [m_i]$.

For every sequence $\overline{Q} = (Q_1, \ldots, Q_p)$ of subsets of $U(B_{t'})$ 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 $\Pi^+$ 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 $\Pi$ of length $p$ the position $\Pi^+$ 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 $\Pi_0$ 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 $\Pi^+$ 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 $P_{p+1} \subseteq U(A_t)$. (The case that he chooses a set $Q_{p+1} \subseteq U(B_{t'})$ 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 $\Pi^+$, 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

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

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

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

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

This implies that there is a $Q^{u'}_{(p+1)} \subseteq U(B_{u'})$ with

$$\theta_{(p+1)j} = \mathrm{tp}_{q-p-1}(B_{u'}, \leq, Q_1 \cap U(B_{u'}), \dots, Q_p \cap U(B_{u'}), Q^{u'}_{(p+1)}).$$

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

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

□

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 $\{\leq\}$ and not $\tau^{\star\star}$. 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 $U(A_{(t)})$. 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* MSO$[\{\leq\}]$*-formula* $\mathsf{a\text{-}type}_\theta(X_1, \dots, X_m)$ *such that for every otxx $A$, every a-node $t \in V(T^A)$, and every compatible linear order $\leq$ of $A$, if $(P_1, \dots, P_m)$ is the type partition of $N_+(t)$, then*

$$(N_+(t), \leq) \models \mathsf{a\text{-}type}_\theta(P_1, \dots, P_m) \textit{ if, and only if, } \mathrm{tp}^{\mathrm{MSO}}_q(A_t, \leq) = \theta.$$

### 3.3 Order-Invariant Type Compositions

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, and $\mathrm{TP}^{\mathrm{MSO}}(\tau^{\star\star} \cup \{\leq\}, q) = \Theta = \{\theta_1, \dots, \theta_m\}$) 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 $A' \equiv^{\mathrm{MSO}}_q A$ then $A'$ is an otxx. This implies that if $A \equiv^{<\text{-inv-MSO}}_q A'$, 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 $(A_{i-1}, \leq_{i-1}) \equiv^{\mathrm{MSO}}_q (A_i, \leq'_i)$ for all $i \in [\ell]$. Then clearly $A \equiv_{co} A'$ implies $A \equiv^{<\text{-inv-MSO}}_q A'$. The converse holds as well, because from an arbitrary linear order we can define a compatible linear order, but this is not important for us.

Let us call a type $\theta \in \Theta$ *realizable* if there is a sub-otxx $A$ and a compatible linear order $\leq$ of $A$ with $\mathrm{tp}^{\mathrm{MSO}}_q(A, \leq) = \theta$. We call $(A, \leq)$ a *realization* of $\theta$. Two types $\theta, \theta' \in \Theta$ are *compatible-order equivalent* (we write $\theta \equiv_{co} \theta'$) if there are realizations $(A, \leq)$ of $\theta$ and $(A', \leq')$ of $\theta'$ 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 $U(A_t)$ such that $\theta := \mathrm{tp}_q^{\mathrm{MSO}}(A_t, \leq) \in \Theta'$ 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 $\{\theta_i \mid i \in [m]$ 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* 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 $\theta \in \Theta$ with $A_{(t)} \models \mathsf{oi\text{-}b\text{-}type}_\theta(P_1,\dots,P_m)$ is compatible at $t$.*

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

*Proof of Lemma 3.6.* 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

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

Let $A$ be an otxx, $t \in V(T^A)$ a b-node, and $(P_1,\dots,P_m)$ a compatible cover of $N_+(t)$. Let $\Theta^t$ be the set of all $\theta$ such that $A_{(t)} \models \mathsf{oi\text{-}b\text{-}type}_\theta(P_1,\dots,P_m)$. We need to prove that $\Theta^t$ 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 $\Theta^u$ 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 $\theta_u = \mathrm{tp}_q^{\mathrm{MSO}}(A_u, \leq_u)$ 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 $U(A_u)$ 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 $(Q_1,\dots,Q_m)$ is a partition of $N_+(t)$ that refines the cover $(P_1,\dots,P_m)$.

Let $\theta_t := \mathrm{tp}_q^{\mathrm{MSO}}(A_t, \leq)$. By Lemma 3.4, we have $A_{(t)} \models \mathsf{b\text{-}type}_{\theta^t}(Q_1,\dots,Q_m)$ and, thus, $A_{(t)} \models \mathsf{oi\text{-}b\text{-}type}_{\theta^t}(Q_1,\dots,Q_m)$. 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 $\theta$ is realizable. Since we have $A_{(t)} \models \mathsf{oi\text{-}b\text{-}type}_\theta(P_1,\dots,P_m)$, there is a partition $(Q'_1,\dots,Q'_m)$ of $N_+(t)$ that refines the cover $(P_1,\dots,P_m)$ such that

$$A_{(t)} \models \mathsf{b\text{-}type}_\theta(Q'_1,\dots,Q'_m). \tag{5}$$

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 $\theta'_u$ is realizable. Let $(A'_u, \leq'_u)$ be a realization of $\theta'_u$.

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 9 for a description of the replacement operation). As $\theta'_u \in \Theta^u \subseteq \langle\theta_u\rangle_{co} \subseteq \langle\theta_u\rangle$, we have $A_u \equiv_q^{\mathrm{MSO}} A'_u$ 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 $U(A'_u)$ is $\leq'_u$. Note that $(A'_{(t)}, \leq') = (A_{(t)}, \leq)$, because the linear orders $\leq$ and $\leq'$ both coincide with $\preceq^A$ on $U(A_{(t)})$. Thus by (5), $A'_{(t)} \models \mathsf{b\text{-}type}_\theta(Q'_1,\dots,Q'_m)$, and by Lemma 3.4, $\mathrm{tp}_q^{\mathrm{MSO}}(A', \leq') = \theta$. Thus $\theta$ is realizable.

It remains to prove that $\theta_t \equiv_{co} \theta$. For each $u \in N_+(t)$, we have $\mathrm{tp}_q^{\mathrm{MSO}}(A_u, \leq_u) = \theta_u \equiv_{co} \theta'_u = \mathrm{tp}_q^{\mathrm{MSO}}(A'_u, \leq'_u)$. Thus there is a sequence $A_{u0},\dots,A_{u\ell}$ 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

$$\mathrm{tp}_q^{\mathrm{MSO}}(A_{u(i-1)}, \leq'_{u(i-1)}) = \mathrm{tp}_q^{\mathrm{MSO}}(A_{ui}, \leq_{ui})$$

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 $\ell$ 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. $\square$

**Lemma 3.7** (Order-invariant type composition at a-nodes)**.** *For every $\theta \in \Theta$ there is an* CMSO$[\emptyset]$*-formula* $\textsf{oi-a-type}_\theta(X_1, \dots, X_m)$ *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 $\theta \in \Theta$ with $(N_+(t)) \models \textsf{oi-a-type}_\theta(P_1, \dots, P_m)$ is compatible at $t$.*

Here $(N_+(t))$ denotes the $\emptyset$-structure with universe $N_+(t)$. Note that, as opposed to the formula $\textsf{a-type}_\theta$ of Lemma 3.5, the formula $\textsf{oi-a-type}_\theta$ has an empty vocabulary. Thus, the condition expressed by this formula no longer depends on the arbitrarily chosen compatible linear order. The proof builds on the ideas developed in the previous proofs and, in addition, crucially depends on the fact that $<$-inv-MSO coincides with CMSO on set structures, which only have monadic relations.

*Proof of Lemma 3.7.* Let $\theta \in \Theta$. We may view the MSO-formula $\textsf{a-type}_\theta(X_1, \dots, X_m)$ as an MSO-sentence of vocabulary $\sigma := \{\leq, X_1, \dots, X_m\}$, where we interpret the $X_i$ as unary relation symbols. Let $\chi^1_\theta$ 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 $\chi^1_\theta$ are proper word structures. Let $q_1$ be an upper bound for the quantifier rank of the formulas $\chi^1_{\theta'}$ for $\theta' \in \Theta$. Let $\Xi := \text{TP}^{\text{MSO}}(\sigma, q_1)$, and for each $\xi \in \Xi$, let $\langle \xi \rangle$ be the order-invariant type that contains $\xi$. Now let $\xi_1, \dots, \xi_\ell$ be all $\xi \in \Xi$ that contain $\chi^1_\theta$, and let

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

Then $\chi^2_\theta$ is order-invariant; we may view it has the "best order-invariant approximation" of $\chi^1_\theta$. The sentence $\chi^2_\theta$ 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 $\chi^3_\theta$ over the same vocabulary [7, 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 $\Theta_\theta$ 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 $\leq'$ of $A'$ 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 $\mathrm{tp}_q^{\text{MSO}}(A'_{t'}, \leq') = \theta'$. Then trivially, all $\theta' \in \Theta_\theta$ are realizable.

**Claim 1.** *If $\Theta_\theta \neq \emptyset$, then $\theta$ is realizable and $\theta \in \Theta_\theta$ and $\Theta_\theta \subseteq \langle \theta \rangle_{co}$.*

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

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

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

By Lemma 2.1, there is an $\ell \in \mathbb{N}$ and for $0 \leq i \leq \ell$ sets $N_i$, partitions $(P_{i1}, \dots, P_{im})$ of $N_i$, and linear orders $\leq_i, \leq'_i$ of $N_i$ such that $(N_0, \leq_0, P_{01}, \dots, P_{0m}) = (N, \leq, P_1, \dots, P_m)$ and $(N_\ell, \leq'_\ell, P_{\ell 1}, \dots, P_{\ell m}) = (N_+(t'), \leq', P'_1, \dots, P'_m)$ and $(N_{i-1}, \leq'_{i-1}, P_{(i-1)1}, \dots, P_{(i-1)m}) \equiv_{q_1}^{\text{MSO}} (N_i, \leq_i, P_{i1}, \dots, P_{im})$.

We let $A^\ell := A'_{t'}$ and $t_\ell := t'$, and for $0 \leq i < \ell$ 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 $(N_i, P_{i1}, \dots, P_{im}) \equiv_{q_1}^{\text{MSO}} (N_+(t'), P'_1, \dots, P'_m)$.

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 $\leq_i, \leq'_i$ on $A^i$ that extend the corresponding orders on $N_i$ and coincide with the linear order induced by $\leq'$ on the copies of the sub-otxxs $A'_{u'}$ that we used to build $A^i$.

Then for $0 \leq i < \ell$, all $j \in [m]$, and all $u \in N_i$, if $u \in P_{ij}$ then $(A^i_u, \leq_i)$ and $(A^i_u, \leq'_i)$ are copies of $(A'_{u'}, \leq')$ for some $u' \in P'_j$, and hence $\mathrm{tp}^{\mathrm{MSO}}_q(A^i_u, \leq_i) = \mathrm{tp}^{\mathrm{MSO}}_q(A^i_u, \leq'_i) = \mathrm{tp}^{\mathrm{MSO}}_q(A'_{u'}, \leq') = \theta_j$. Since $(N_{i-1}, \leq'_{i-1}, P_{(i-1)1}, \dots, P_{(i-1)m}) \equiv^{\mathrm{MSO}}_{q_1} (N_i, \leq_i, P_{i1}, \dots, P_{im})$, it follows from Lemma 3.5 that we have $\mathrm{tp}^{\mathrm{MSO}}_q(A^{i-1}, \leq'_{i-1}) = \mathrm{tp}^{\mathrm{MSO}}_q(A^i, \leq_i)$ for all $i$. Moreover, as we have $(N_0, \leq_0) \models \mathsf{a\text{-}type}_\theta(P_{01}, \dots, P_{0m})$, again by Lemma 3.5 we have $\mathrm{tp}^{\mathrm{MSO}}_q(A^0, \leq_0) = \theta$.

This implies that $\theta$ is realizable and that $\theta \equiv_{co} \theta'$, or equivalently, $\theta' \in \langle\theta\rangle_{co}$. As this holds for all $\theta' \in \Theta_\theta$, we have $\Theta_\theta \subseteq \langle\theta\rangle_{co}$. We have $\theta \in \Theta_\theta$ because $(N_0, \leq_0) \models \mathsf{a\text{-}type}_\theta(P_{01}, \dots, P_{0m})$ implies $(N_0, \leq_0, P_{01}, \dots, P_{0m}) \models \chi^2_\theta$, and this implies $(N_0) \models \chi^3_\theta(P_{01}, \dots, P_{0m})$. ⌟

**Claim 2.** *Let $A$ be an otxx, $t \in V(T^A)$ an a-node, $\leq$ a compatible linear order of $A$, and $(P_1, \dots, P_m)$ the type partition of $N_+(t)$. Then the set $\Theta_t$ of all $\theta \in \Theta$ with $(N_+(t)) \models \chi^3_\theta(P_1, \dots, P_m)$ is compatible at $t$.*

*Proof.* Let $\theta_t := \mathrm{tp}^{\mathrm{MSO}}_q(A_t, \leq)$. Then for all $\theta \in \Theta_t$ we have $\theta_t \in \Theta_\theta$ and thus, by Claim 1, $\theta_t \in \langle\theta\rangle_{co}$. 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 $\Theta_t$ 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, $t \in V(T^A)$ an a-node, and $(P_1, \dots, P_m)$ a compatible cover of $N_+(t)$. Let $\Theta^t$ be the set of all $\theta \in \Theta$ such that $(N_+(t)) \models \mathsf{oi\text{-}a\text{-}type}_\theta(P_1, \dots, P_m)$. We need to prove that $\Theta^t$ is compatible at $t$.

For every $u \in N_+(t)$, let $\Theta^u := \{\theta_i \mid i \in [m] \text{ such that } u \in P_i\}$. As the cover $(P_1, \dots, P_m)$ is compatible, for all $u$ the set $\Theta^u$ is compatible at $u$. In particular, there is a $\theta_u \in \Theta^u$ and a compatible linear order $\leq_u$ of $A_u$ such that $\theta_u = \mathrm{tp}^{\mathrm{MSO}}_q(A_u, \leq_u)$ and $\Theta^u \subseteq \langle\theta_u\rangle_{co}$. Let $\leq^1$ be a compatible linear order of $A_t$ such that for all $u \in N_+(t)$, the restriction of $\leq^1$ to $U(A_u)$ 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 $(Q_1, \dots, Q_m)$ is the type partition of $N_+(t)$ in $(A_t, \leq^1)$, 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 $\leq^2$ of $A$ such that $\mathrm{tp}^{\mathrm{MSO}}_q(A_t, \leq^2) = \theta_t$ 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 $(N_+(t)) \models \mathsf{oi\text{-}a\text{-}type}_{\theta_t}(P_1, \dots, P_m)$. 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 we have $\theta \in \Theta_t(Q'_1, \dots, Q'_m)$. 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 $\leq^3$ of $A$ such that $\mathrm{tp}^{\mathrm{MSO}}_q(A_t, \leq^3) = \theta'_t$ and $\Theta_t(Q'_1, \dots, Q'_m) \subseteq \langle\theta'_t\rangle_{co}$.

It remains to prove that $\theta_t \equiv_{co} \theta'_t$, because then $\theta \in \Theta_t(Q'_1, \dots, Q'_m) \subseteq \langle\theta'_t\rangle_{co} = \langle\theta_t\rangle_{co}$. For each $u \in N_+(t)$, let $\theta_u := \mathrm{tp}^{\mathrm{MSO}}_q(A_u, \leq^2)$ and $\theta'_u := \mathrm{tp}^{\mathrm{MSO}}_q(A_u, \leq^3)$. 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 $\Theta^u$ 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 $\theta_t \equiv_{co} \theta$). □

### 3.4 Proofs of the lifting theorems

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

First of all, we turn $A$ into a structure $A^{\star\star}$ that is isomorphic to an otx of $A$. Using the theorem's precondition, this is possible by a CMSO-transduction that produces otxs with bounded adhesion. 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 $\varphi$ to a formula $\varphi^{\star\star}$, such that for each $A \in \mathcal{C}$ we have $A \models \varphi$ if, and only if, $A^{\star\star} \models \varphi^{\star\star}$. In particular, $\varphi^{\star\star}$ is still an order-invariant MSO-formula.

In order to test whether $A^{\star\star} \models \varphi^{\star\star}$ holds, we view $\varphi^{\star\star}$ as an MSO$[\tau^{\star\star} \cup \{\leq\}]$-formula and test whether $(A^{\star\star}, \leq) \models \varphi^{\star\star}$ holds for some total order $\leq$ over $U(A^{\star\star})$ compatible with $A^{\star\star}$. Using the terminology developed in Section 3.3, we ask whether $\varphi^{\star\star}$ is equivalent to a formula from a realizable type $\theta$ of $A^{\star\star}$. Due to the order-invariance of $\varphi^{\star\star}$, this is equivalent to asking whether each realizable type $\theta$ contains a formula equivalent to $\varphi^{\star\star}$. In order to have access to a realizable type of $A^{\star\star}$, we define a compatible set of types $\Theta'_r$ for the root $r$ by using a CMSO-formula that implements the following three parts: (1) It existentially guesses a cover $(P_1, \dots, P_m)$ of all nodes of the tree decomposition that *induces* the set of types $\Theta'_t := \{\theta_i \mid i \in [m] \text{ with } t \in P_i\}$ at each node $t$ of the tree decomposition. (2) It tests whether the induced set of types for each leaf is compatible. This is possible since leaves are always b-nodes and the substructures induced by their bags contain total orderings. (3) It compares the induced set of types of each inner node $t$ with the set of types that we get by applying Lemmas 3.6 (in the case of a b-node) or 3.7 (in the case of an a-nodes) to the cover $(P_1 \cap N_+(t), \dots, P_m \cap N_+(t))$ of its children $N_+(t)$.

Finally, we test whether $\varphi^{\star\star}$ is equivalent to a formula from a type $\theta \in \Theta'_r$. Overall, this results in a CMSO-formula $\psi^{\star\star}$ that is equivalent to $\varphi^{\star\star}$ on $A^{\star\star}$. Since $\varphi^{\star\star}$ on $A^{\star\star}$ is constructed to be equivalent to $\varphi$ on $A$ and CMSO-transductions preserve CMSO-definability, we know that there exists a CMSO-formula $\psi$ on $\tau$ that is equivalent to $\varphi$ 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 this class of structures [1]. The resulting proof of Theorem 3.2 produces an MSO-formula instead of a CMSO-formula. ☐

# 4 Defining Decompositions

During the course of the present section, we use MSO-transductions to extend graphs with tree decompositions for them. The first transduction (developed in Section 4.1) is used to prove Theorems 5.6 and 5.7, which apply to graphs of bounded tree width. The second transduction (reviewed in Section 4.2) is used to prove Theorems 5.10 and 5.11, which apply to graphs that exclude $K_{3,\ell}$ for some $\ell \in \mathbb{N}$ as a minor. The present section's results work with graphs instead of general structures. Thus, we set $\tau = \{E\}$ throughout the section where $E$ is the (binary) edge relation symbol.

The structures defined by the transductions are over the vocabulary $\tau^+ := \tau \cup \{V_S, V_T, E_T, R_\beta\}$ where $V_S$ and $V_T$ are unary, and $E_T$ and $R_\beta$ are binary. A *tree extension* (*tx* for short) of a graph $G = (V, E)$ is a $\tau^+$-structure $G^+$ that extends $G$ by a tree decomposition $(T, \beta)$ of $G$. Tree decompositions are encoded as part of txs just like they are encoded as part of otxs in Section 3, but without including a partial order. The below transductions turn graphs of a certain kind into tree extensions of a certain kind. In order to state the results concisely, we use the following terminology: whenever we talk about the bags and separators of a tree extension $G^+$, we refer to the bags and separators, respectively, of the tree decomposition $(T, \beta)$ encoded by $G^+$. For a class $\mathcal{C}$ of graphs and a class $\mathcal{D}$ of tree extensions, we say that an $\textsc{mso}[\tau, \tau^+]$-transduction $\Lambda$ *defines tree extensions from $\mathcal{D}$ for graphs from $\mathcal{C}$* if the following holds for every $G \in \mathcal{C}$: we have $\emptyset \subsetneq \Lambda[G] \subseteq \mathcal{D}$ and every $G^+ \in \Lambda[G]$ is isomorphic to a tree extension of $G$.

### 4.1 Defining Tree Decompositions into Graphs without Clique Separators

A *clique separator* in a graph $G$ is a set $S \subseteq V(G)$, such that $G[S]$ is a clique (that means, there is an edge in $G$ between every pair of vertices from $S$) and there are two vertices $v, w \in V(G) \setminus S$ that are disconnected in $G \setminus S$. In this case, $S$ *separates* $v$ and $w$. An *atom* is a graph without clique separators; in particular, atoms are connected graphs. We prove the following lemma.

**Lemma 4.1.** *Let $k \in \mathbb{N}$. There is an $\textsc{mso}[\tau, \tau^+]$-transduction $\Lambda_{\mathrm{tw}\leq k}$ that defines tree extensions for graphs of tree width at most k where (1) the bags induce subgraphs that are atoms, and (2) the separators of the tree decompositions are cliques.*

Our proof uses the graph-theoretic ideas behind a logspace algorithm [13] for constructing tree decomposition of the kind described by Lemma 4.1 and shows how to define the construction using an $\textsc{mso}$-transduction. The mentioned algorithm first constructs decompositions along small clique separators of the graph and, then, refines the decompositions by also taking larger clique separators into account. Since graphs of tree width at most $k$ only contain cliques of size at most $k+1$, applying $k+1$ refinement steps turns a given graph of tree width at most $k$ into a tree decomposition that proves the lemma.

Formally, constructing tree decompositions via clique separators of a growing size involves working with a refined notion of atoms. For $c \in \mathbb{N}$, a *$c$-clique separator* is a clique separator of size at most $c$ and a *$c$-atom* is a graph that does not contain clique separators of size at most $c$. Like atoms, $c$-atoms are connected by definition.

For a graph $G$ and a constant $c \in \mathbb{N}$, we build a graph $T_c$ where $V(T_c)$ consists of all maximal subgraphs of $G$ that are $c$-atoms, which are called *atom nodes*, and all $c$-clique separators, which are called *separator nodes*. In addition, to each $t \in V(T_c)$ we assign a bag $\beta_c(t) \subseteq V(G)$ as follows: if $t$ is an atom node, then $\beta_c(t)$ is the vertex set of the corresponding atom, and if $t$ is a separator node, then $\beta_c(t)$ is the corresponding separator. An edge is inserted between every atom node $t$ and separator node $u$ with $\beta_c(u) \subseteq \beta_c(t)$. While $T_c$ is not a tree in general, [13] proved that, if $G$ is a $(c-1)$-atom for $c \geq 1$, then $(T_c, \beta_c)$ is a tree decomposition for $G$.

**Fact 4.2.** *Let $c \geq 1$ and $G$ be a $(c-1)$-atom. Then $(T_c, \beta_c)$ is a tree decomposition for $G$. Moreover,*
1. *atom nodes are only connected to separator nodes and vice versa, and*
2. *all leaves are atom nodes.*

The previous fact provides us with a single step in the decomposition refinement procedure outlined above. We apply it in order to move from tree decompositions whose bags induce $(c-1)$-atoms to tree decompositions whose bags induce $c$-atoms. This is similar to the approach of [13], which is based on the following construction.

Let $G$ be a graph and $D_{(c-1)} = (T_{(c-1)}, \beta_{(c-1)})$ a tree decomposition of $G$, such that for each node $t \in V(T_{(c-1)})$ the bag $\beta(t)$ induces a maximal subgraphs of $G$ that is a $(c-1)$-atom or it induces a $(c-1)$-separator. Moreover, the tree decomposition satisfies the two properties stated in Fact 4.2: neighbors of atom nodes are only separator nodes and vice versa, and all leaves are atom nodes. We modify the tree decomposition into a decomposition $D_c$, such that it still satisfies the same properties, except that the constant $c-1$ is replaced by $c$. For each atom node $V(t)$, we consider the tree decomposition $D_c^t = (T_c^t, \beta_c^t)$ of the $(c-1)$-atom $G[\beta(t)])$ that we get from applying Fact 4.2. We replace $t$ with $D_c^t$ inside $D_{(c-1)}$ as follows: if $t$ is the root of $T_c^t$, we just replace it with $D_c^t$. If $t$ is not the root, it has a unique parent separator node $u$ and, in turn, $u$ has a unique parent atom node $v$. We replace $t$ with $D_c^t$ and connect $u$ to the root of $D_c^t$, which is constructed as an atom node whose bag contains all of $\beta(u)$. Similarly, $v$ is replaced with $D_c^v$ and the edge between $v$ and $u$ is redirected such that there is an edge to $u$ from the highest atom node in $D_c^v$ (with respect to the root of $D_c^v$) that contains all of $\beta(u)$, which is unique. The following fact follows from [13]. The arguments about the shape of the decomposition directly follow from the construction.

**Fact 4.3.** *Let $G$, $D_{(c-1)}$, and the constructed $D_c$ be defined as in the previous paragraph. Then $D_c$ is a tree decomposition for $G$. Moreover, in $D_c$,*

1. *atom nodes are only connected to separator nodes and vice versa, and*
2. *all leaves are atom nodes.*

The final proof of Lemma 4.1 shows how the construction of Fact 4.3 can be done by an MSO-transduction. It also needs to turn a given graph, which can possibly be disconnected, into a tree decomposition whose bags induce the connected components of the graph. Since this is a special case that is not covered by the above constructions, we first prove it separately. In the context of MSO-definable tree decompositions, we use the concept of tree extensions. In order to do that, we use the following convention: when we say that the bags of a tree decomposition (or tree extension) are $c$-atoms, we mean that the subgraphs induced by the bags are $c$-atoms. We frequently use the fact that there is an MSO-formula for each of the following properties of vertex subsets $V' \subseteq V$ of a given graph $G$: $V'$ is a clique separator, $V'$ is a $c$-clique separator for some fixed, but arbitrary, $c \in \mathbb{N}$, $G[V']$ is an atom, $G[V']$ is a $c$-atom for some fixed, but arbitrary, $c \in \mathbb{N}$.

**Lemma 4.4.** *There is an* $\mathrm{MSO}[\tau, \tau^+]$*-transduction* $\Lambda_{\mathrm{comp}}$ *that defines for every graph $G$ a tree extension whose tree decomposition*

1. *has a single node with an empty bag (representing the empty separator), and*
2. *for each component of $G$ exactly one node whose bag equals the vertex set of it.*

*Proof.* The main idea is to guess, via parameters, a set of vertices of the graph whose copies in the tree extension represent the atoms and separators in the decomposition; the term *represent* hints to the fact that we are able to define the vertex set of the corresponding atom or separator in an MSO-definable way from the atom node or separator node, respectively. The transduction $\Lambda_{\mathrm{comp}}$ has three parameters $\mathrm{ROOT}_0$, $\mathrm{ATOM}_0$, and $\mathrm{CLIQUE}_0$ and three levels: Level 1 contains copies of the vertices of the original graph $G$, level 2 contains the atom nodes of the decomposition, and level 3 contains the separator nodes of the decomposition.

First of all, the formula $\lambda_{\mathrm{VALID}}$ tests whether the parameters are chosen in a way that allows the other formulas to define the tree extension from them. It ensures the following properties: $\mathrm{ROOT}_0$ contains exactly one vertex that we call $v_r$ in the following, $\mathrm{ROOT}_0 \cup \mathrm{ATOM}_0$ contain exactly one vertex from each connected component of $G$, and $\mathrm{CLIQUE}_0$ contains exactly one vertex that we call $v_c$ with $v_c \in \mathrm{ATOM}_0$. Thus, $v_c$ is used to both represent an atom and to represent the unique separator in the construction.

For each $v \in \mathrm{ROOT}_0 \cup \mathrm{ATOM}_0$, the transduction defines $\beta(v,2)$ to be the vertex set of the connected component in which $v$ lies. Moreover, we set $\beta(v_c,3) := \emptyset$. We create an edge between $(v_c,3)$ and each $(v,2)$ for $v \in \mathrm{ATOM}_0$. Moreover, edges are oriented away from the root $v_r$. □

We are now ready to prove Lemma 4.1. The remaining difficulty for the proof lies in defining the construction of Fact 4.3 in an MSO-definable way, which involves defining the construction of Fact 4.2 simultaneously for all atom nodes.

*Proof of Lemma 4.1.* We first turn a given graph into a tree decomposition whose bags are the graph's connected components using $\Lambda_{\text{comp}}$ from Lemma 4.4. Next, we refine this decomposition $k+1$ times using MSO-transductions that implement the construction from Fact 4.3. Finally, the lemma follows since MSO-transductions are closed under composition.

Let $c \geq 1$ and $G^+$ be a tree extension with a tree decomposition $(T, \beta) = (T_{(c-1)}, \beta_{(c-1)})$ as described above. In order to MSO-define the construction of Fact 4.3, we use an MSO$[\tau^+, \tau^+]$-transduction $\Lambda_c$, which transforms tree extensions into tree extensions. Similar to the transduction of the proof of Lemma 4.4, it has three parameters, but this time they are called ROOT$_c$, ATOM$_c$, and CLIQUE$_c$. Moreover, it has three levels: to level 1 we copy the vertex set of the underlying graph and decomposition nodes whose bags are not refined, level 2 contains newly constructed atom nodes, and level 3 contains newly constructed separator nodes. The parameters have to satisfy certain properties similar to the ones in the proof of Lemma 4.4, but they are more involved due to the following reasons. First, we need to make sure that all atoms can be refined simultaneously. Second, we need to make sure that each new atom node represents a unique atom. In the proof of Lemma 4.4 the connected components, which are 0-atoms, are disjoint and, thus, it was possible to choose a vertex from each component. In the case of $c$-atoms for $c \geq 1$, a vertex can be part of multiple atoms. In order to work around this problem, we utilize the tree-like partial order that is given by the decomposition with respect to the chosen root.

We start with the existing tree extension $G^+$ and consider where it needs to be modified. Since $\Lambda_c[G^+]$ will be a refinement of $G^+$ where new separator nodes are added, but existing separator nodes do not change, all of the separator nodes present in $G^+$ can be copied to level 1 directly without modification. On the other hand, the atom nodes in $G^+$ are refined if they contain a $c$-clique separator, so altogether the formula $\lambda^1_{V_T}(t)$ is satisfied only for some $t \in V(T)$: either if $t$ is a separator node, that means, where $\beta(t)$ induces a clique of size up to $c$; or otherwise if the size of $\beta(t)$ is larger than $c$ and there is no $c$-clique separator. This effectively removes exactly those atom nodes which have a $c$-clique separator and which we thus need to decompose further. We define $\lambda^1_{R_\beta}$, such that $R_\beta(t) = \beta(t)$ because we do not want the bags of these copied nodes to change, and similarly, the edges between any pair of copied nodes $s,t$ remain the same, so we define $\lambda^{1,1}_{E_T}(s,t)$ to be satisfied precisely if $(s,t) \in E(T)$.

As a reminder, the indices of the formulas in a transduction specify a level for each of its free variables – so as an example for a binary relation like $E_T$, the formula $\lambda^{2,3}_{E_T}(v,w)$ being satisfied for two concrete vertices $a = v$ and $b = w$ would mean that the vertex $(a,2)$ (the copy of $a$ on level 2) is connected to the vertex $(b,3)$ (the copy of $b$ on level 3) in the tree $T$ defined by the transduction. The transduction then constructs the relation $E_T$ by taking the union over all satisfying assignments of $\lambda^{i,j}_{E_T}(v,w)$ for all pairs of levels $i,j$.

Next, we define the new atom and separator nodes, as well as their connectivity to the forest $D$ resulting from the described removal of atom nodes and their incident edges from $T$. Let $t \in V(T)$ be an atom node that is deleted and set $A_t := G[\beta(t)]$, which contains at least one $c$-clique separator. We define a partial tree decomposition $D_t$ of $A_t$ into $c$-atoms, and then show how $D_t$ is reinserted into the forest $D$ in place of the deleted node $t$. We keep in mind that $A_t$ is a $(c-1)$-atom and, thus, free of any clique separators up to size $c-1$. Like in the construction of Fact 4.3, the root atom node of each decomposition $D_t$ is chosen so that it contains the $C = \beta(s)$ where $s$ is the parent separator node of $t$ in $T$. If $t$ is itself the root of $T$ and thus has no parent, then consider $C = \emptyset$ in the following.

**Parameters of the transduction and their validity properties.** We describe the properties of the parameters verified by $\lambda_{\text{VALID}}$. They are used to single out a unique vertex of $G^+$ for each $c$-atom and each $c$-clique separator, as well as a unique $c$-atom assigned to the root of each partial tree decomposition $D_t$ with the property describe above.

The parameter $\text{ROOT}_c$ contains exactly one vertex for each $c$-atom that includes a $c$-clique separator. We aim to find some $r \in A_t \cap \text{ROOT}_c$ as the unique root vertex of a $c$-atom $A_t$. Since $r$ is supposed to represent the root node $(r,2)$ of $T_t$ that is later connected to the parent separator node $s$ of $t$ when reinserting the partial tree decomposition $D_t$ into the forest $D$, the root atom has to contain the clique $C$ in its own bag. There are potentially multiple atoms that satisfy this property. If we consider the tree decomposition from Fact 4.2 on the subgraph $A_t$, then the set of $c$-atoms containing the clique $C$ form a subtree (due to the cover and connectedness condition of any tree decomposition). The leaves of this subtree are $c$-atoms which contain at least one vertex $r$ that is not present in any of the other $c$-atoms from $A_t$ that include all of $C$. Note that this either immediately implies that either $r \notin C$, or there is just a single candidate $c$-atom, in which case we may freely pick an $r \notin C$. This suffices as a unique identifier of the root $c$-atom of $A_t$, because then any other $c$-atom containing $r$ cannot contain all of $C$. An MSO-formula can ensure that for each $(c-1)$-atom $A_t$ that is decomposed further, $\text{ROOT}_c$ contains a single root vertex $r$ from the described candidates for this $A_t$. Since $r \notin C$, the respective $r$ can only appear in bags in the subtree of $T$ below $t$. So if $r$ appeared again in the root of a different $(c-1)$-atom that gets decomposed further, it would necessarily be in the bag of the separator node just above that $(c-1)$-atom, a contradiction. So there is a one-to-one correspondence between $(c-1)$-atoms $A_t$ that get decomposed further and the vertices $r \in \text{ROOT}_c$. This shows $\text{ROOT}_c$ has the desired properties for all $A_t$ simultaneously on all of $G$.

For the other $c$-atom representatives we utilize the fact that the $c$-cliques between any two $c$-atoms in $A_t$ can be linearly ordered. To see this, remember the construction of the tree decomposition from Fact 4.2. In particular, for any vertex $v \in A_t$ outside of the root $c$-atom, we can define the $c$-clique separator *$S$ closest to $v$ compared to the root atom* in the sense that $S$ separates a vertex of the root atom of $A_t$ from $v$, but no other $c$-clique $S'$ separates a vertex of the root atom from both $v$ and a vertex of $S$. We define an MSO-formula $\mathsf{closest\text{-}clique\text{-}separator}^c(v,S)$, which is satisfied exactly for vertices $v$ and $c$-cliques $S$ that satisfy this property. Note that this formula works globally on all of $G^+$, because the root vertex $r$ of each $A_t$ an be retrieved from the parameter $\text{ROOT}_c$. So for each $c$-atom $A$ within $A_t$, we define the (nonempty) set $Z_A$ of vertices of this atom which are not in its closest $c$-clique separator. For different $c$-atoms, these sets are distinct – since an overlap would mean that this vertex would appear in the $c$-clique separator between them, which is then a closer clique separator for one of the $c$-atoms, a contradiction. Via the parameter $\text{ATOM}_c$, we guess a single vertex of $Z_A$ for each $c$-atom. An MSO-formula can verify that conversely, no two vertices of $\text{ATOM}_c$ are in the same set $Z_A$. This establishes the one-to-one correspondence of every $a \in \text{ATOM}_c$ to the sets $Z_A$ and thus, the non-root $c$-atoms $A$ in all of $G$. Remember that the root atoms are already covered above by $\text{ROOT}_c$.

To define representatives for the separator nodes of $T_t$, we make the following observation: in the tree decomposition, each separator node will have at least one atom node as its child. Consequently, we use the representative of a child atom nodes also as the representative of its closest $c$-clique separator towards the root vertex $r$. This overlap explains why we use separate levels for atom and separator nodes. We use the parameter $\text{CLIQUE}_c$ to guess these representatives, and have to only verify $\text{CLIQUE}_c \subseteq \text{ATOM}_c$, that no two vertices in $\text{CLIQUE}_c$ have the same closest $c$-clique separator, and that for each $c$-clique separator $S$ some vertex $v \in \text{CLIQUE}_c$ exists that has $S$ as its closest separator. This guarantees the one-to-one correspondence of $c$-clique separators and vertices in $\text{CLIQUE}_c$ not just for $A_t$, but for all of $G$.

**Defining the construction of Fact 4.2.** We follow the construction of the decompositions from Fact 4.2. We define the formula $\lambda^2_{V_T}(v)$ such that it is satisfied exactly for the vertices $v \in \text{ROOT}_c \cup \text{ATOM}_c$, and $\lambda^3_{V_T}(v)$ such that it is defined exactly for the vertices $v \in \text{CLIQUE}_c$. The properties of these parameters as discussed above can be defined in MSO.

We now know that for a separator node $(v,3)$ created in this way, the clique separator it represents is the closest $c$-clique separator $S$ towards the unique root $r \in \text{ROOT}_c \cap A_t$, which we can extract using the formula MSO- formula $\mathsf{closest\text{-}clique\text{-}separator}^c(v,S)$ and thus set $R_\beta(v,3) = S$ by defining the formula $\lambda^{3,1}(v,x)$ such that it is satisfied exactly for $x \in S$. Conversely, for an atom node $(v,2)$ created this way, we extract the $c$-atom $A$ it represents by finding the closest $c$-clique separator $S$ towards $r$. The atom $A$ is then the set of vertices which either have $S$ as its closest $c$-clique separator, which means they are in the set $Z_A$ defined above, or which are itself in $S$. We can then define $R_\beta(v,3) = A$ similarly to above. This sets up the bags of the separator and atom nodes to be exactly the set of vertices of the clique separator (and respectively, atom) which they represent.

Finally, we define the edges between nodes in $T_t$. Remember that the construction from Fact 4.2 connects an atom and a separator node if the bag of the separator node is completely contained in the bag of the atom node. We only have to define this as a directed tree decomposition rooted in the atom node $(r,2)$ of $A_t$ for the unique root representative $r \in A_t \cap \mathrm{ROOT}_c$.

We use the formula $\lambda_{E_T}^{3,2}$ to express that there is an edge from a separator node $(u,3)$ to an atom node $(v,2)$ precisely if the vertices $u$ and $v$ have the same closest $c$-clique separator $S$, which is unique. Then by the above, the bag of $(u,3)$ is precisely this separator $S$ and this $S$ is completely contained in the bag of $(v,2)$, so the desired property is satisfied. Similarly, we use the formula $\lambda_{E_T}^{2,3}$ to express that there is an edge from an atom node $(v,2)$ to a separator node $(u,3)$ precisely if the bag of $(u,3)$ is completely contained in the bag of $(v,2)$, but they do not have the same closest $c$-clique separator $S$. Thus, we constructed a tree decomposition $D_t$ as described in Fact 4.2.

**Defining the construction of Fact 4.3.** We now move from the view of the single $(c-1)$-atom $A_t$ and its tree decomposition $T_t$ to the global view on all of $G^+$. If we stopped defining the rest of the transduction here, the decomposition graph would now be the forest $D$ together with all partial tree decompositions $D_t$ for removed $(c-1)$-atom nodes $t$. It remains to define how this forest is merged back together into a single tree decomposition.

Let $t$ be a deleted $(c-1)$-atom node and $T_t$ the newly constructed tree of the partial decomposition $D_t$ into $c$-atoms on the bag $A_t := G[\beta(t)]$. Further let $s$ be the parent of $t$ in $T$, which is a separator node. We use the formula $\lambda_{E_T}^{1,2}$ to define the edges from $s$ to the root of $T_t$ and thereby reattach the partial tree decompositions at the appropriate position. So $\lambda_{E_T}^{1,2}(s,u)$ is satisfied if $s$ is a separator node with a deleted child node, $u \in \mathrm{ROOT}_c$, and $\beta(s)$ is the closest $c$-clique separator of $u$ since this means precisely that the node $(u,2)$ has the root atom of $D_t$ as its bag.

For the formulas $\lambda_{E_T}^{2,1}$, finding the correct point of attachment is a bit more involved. If $t$ had no child nodes, there is nothing to reattach. Otherwise we have to consider all former child nodes $s_1,\dots,s_n$ of the deleted node $t$, each of which is a separator node according to Fact 4.2. Each of their bags is a clique, and we would thus receive a valid tree decomposition if we connected each $s_j$ to an atom node $t_j$ of $T_t$ such that $\beta(s_j) \subseteq \beta(t_j)$ for all $j \in [n]$. Following the construction of Fact 4.3, to find a unique connection point, we take a closer look at the potential choices of the compatible atom nodes for a node $s_j$: due to the connectedness property, the set of nodes in $T_t$ which contain all of $\beta(s_j)$ is connected. This means that there is a unique atom node $t_j^*$ in this tree 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 in $T_t$ does not include $\beta(s_j)$. We can thus define $\lambda_{E_T}^{2,1}(t,s)$ to be satisfied precisely if $s = s_j$ and $t = t_j^*$ hold, which is MSO-definable. This concludes the reintegration of $T_t$ and finishes the description of the MSO-transduction that implements the construction of Fact 4.3. □

## 4.2 Defining Tree Decompositions into 3-Connected Components

A graph $G$ is *k-connected* if $|G| > k$ and $G$ has no separator $S \subseteq V(G)$ of size $|S| < k$. Courcelle [8] showed that one can use MSO-transductions to define tree decompositions into 3-connected components. We formulate this result with respect to the notion of tree extensions as Fact 4.5.

**Fact 4.5.** *There is an MSO-transduction $\Lambda_{\text{3-comp}}$ that defines tree extensions whose torsos (1) are 3-connected, cycles, a single edge, or a single vertex, and (2) separators have size at most 2 for all graphs.*

The torsos of the tree decomposition produced by Fact 4.5 always induce topological subgraphs; a *topological subgraph* $G'$ of a graph $G$ arises by taking a subgraph of $G$ and replacing some paths with edges. Later we use this insight since whenever a graph $G$ does not contain a certain graph $H$ as a minor, then this also holds for each of its topological subgraphs. In our application $H$ equals $K_{3,\ell}$ for some $\ell \in \mathbb{N}$.

# 5 Defining Orderings

In the previous section, we have seen 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 $\ell$. 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]. In order to state it formally, we introduce some terminology. Let $\tau$ be a vocabulary that does not contain the binary relation symbol $\leq$. We say that an MSO$[\tau, \tau \cup \{\leq\}]$-transduction $\Lambda$ *defines orderings* on a class $\mathcal{C}$ of $\tau$-structures if the following holds for every $A \in \mathcal{C}$: $\Lambda(A) \neq \emptyset$ and every $B \in \Lambda(A)$ is an expansion of $A$ with a binary relation $\leq^B$ that is a linear order of $U(B)$. 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 $G \setminus S$ is bounded by $f(|S|)$. The below fact refers to GSO-logic on graphs; it is defined by taking MSO-logic on graphs and extend it with the ability to quantify over subsets of a graph's edges [16].

**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 total orderings on $\mathcal{C}$.*

Since GSO-logic collapses to MSO-logic on every class of graphs that exclude a fixed minor [9] (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 fact has the following corollary.

**Corollary 5.2.** *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 an* MSO*-transduction $\Lambda_{\text{ORDER-SEP}}$ that defines total orderings on $\mathcal{C}$.*

## 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 bounded tree width and are atoms, 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.

Given a graph $G$, its *improved version* $G'$ is the graph with vertex set $V(G') := V(G)$ and $(v,w) \in E(G')$ holds for every two distinct vertices $v,w \in V(G')$ if, and only if, $(v,w) \in E(G)$ 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 [20]. 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 $k = \mathrm{tw}(G)$ than the original graph. The MSO-transduction of the below proposition is based on defining a constant number, $k+1$, of disjoint paths between pairs of vertices of the graph. This can be done by using $k+1$ set variables where each set colors the vertices of a single path that does not share vertices with other paths.

**Proposition 5.3.** *Let $k \in \mathbb{N}$. There is an* MSO*-transduction $\Lambda_{\text{IMPROVE}}$ 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 5.4.** *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 $G \in \mathcal{C}$ and $k := \mathrm{tw}(G)$. Let $S \subseteq V(G)$, and let $G_1, \ldots, G_n$ be the components of $G \setminus S$. We shall prove that $n \leq \binom{|S|}{2} \cdot k + 1$ holds.

Without loss of generality we assume that $n \geq 2$. For every $i \in [n]$, let $S_i$ be the set of neighbors of $G_i$ in $S$. As $G$ is an atom, $S_i$ is not a clique in $G$. Thus there are $u, v \in S_i$ such that $\{u, v\} \notin E(G)$. Since $G$ is improved, we have $u, v \in S_i$ for at most $k$ indices $i \in [n]$. As there are $\binom{|S|}{2}$ pairs $\{u, v\} \subseteq S$, this implies $n \leq \binom{|S|}{2} k$ and, thus, the above inequality holds. □

We get the following from combining Lemma 5.4 with Fact 5.1.

**Corollary 5.5.** *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 5.6.** *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 $\Lambda_{\text{IMPROVE}}$ from Proposition 5.3. Next, we apply the transduction $\Lambda$ of Lemma 4.1 to $G'$, which defines a tree extension $G^+$. 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 5.5 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 separator 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 5.7.** *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 5.6, 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 5.8.** *Let $\mathcal{C}$ be a class 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 $G \setminus S$. 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 $\ell$ 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 hence bounded separability. □

**Corollary 5.9.** *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 5.9, we can prove the following.

**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-MSO $=$ CMSO *on $\mathcal{C}$.*

*Proof.* The proof is similar to the proof of Theorem 5.6, except that we need to use different transductions 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 4.5. For constructing the bag orderings, we follow the arguments from Theorem 5.6, but apply Corollary 5.9 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 5.11.** *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 5.7. We take the proof of Theorem 5.10, but use the lifting theorem for $<$-inv-FO instead of the lifting theorem for $<$-inv-MSO. $\square$

# 6 Conclusions

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.


# Acknowledgements

We thank Pascal Schweitzer for the idea of Lemma 5.4.


# 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] M. Bojańczyk and M. Pilipczuk. Definability equals recognizability for graphs of bounded tree width. *CoRR*, abs/1605.03045, 2016. URL: http://arxiv.org/abs/1605.03045.

[4] M. Bojańczyk and M. Pilipczuk. Definability equals recognizability for graphs of bounded tree width. In *Proceedings of LICS 2016*. IEEE Computer Society, 2016. to appear.

[5] 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`.

[6] 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`.

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

[8] 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`.

[9] 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.

[10] 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.

[11] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer Verlag, 1995.

[12] K. Eickmeyer, M. Elberfeld, and F. Harwath. Expressivity and succinctness of order-invariant logics on depth-bounded structures. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS 2014), Part I*, LNCS, pages 256–266, 2014. doi:10.1007/978-3-662-44522-8_22.

[13] M. Elberfeld and P. Schweitzer. Canonizing Graphs of Bounded Tree Width in Logspace. In *Proceedings of the 33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, volume 47 of *LIPIcs*, pages 32:1–32:14. Schloss Dagstuhl LZI, 2016. doi:http://dx.doi.org/10.4230/LIPIcs.STACS.2016.32.

[14] S. Feferman and R. Vaught. The first order properties of algebraic systems. *Fundamenta Mathematicae*, 47(1):57–103, 1959.

[15] 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.

[16] E. Grädel, C. Hirsch, and M. Otto. Back and forth between guarded and modal logics. *ACM Trans. Comput. Logic*, 3:418–463, July 2002. doi:10.1145/507382.507388.

[17] Y. Gurevich. Toward logic tailored for computational complexity. In M. M. Richter, E. Börger, W. Oberschelp, B. Schinzel, and W. Thomas, editors, *Computation and Proof Theory*, volume 1104 of *Lecture Notes in Mathematics*, pages 175–216. Springer Verlag, 1984. doi:10.1007/BFb0099486.

[18] D. Lapoire. Recognizability equals monadic second-order definability for sets of graphs of bounded tree-width. In M. Morvan, C. Meinel, and D. Krob, editors, *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science (STACS 1998)*, volume 1373 of *LNCS*, pages 618–628. Springer Verlag, 1998.

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

[20] 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.

[21] J. Makowsky. Algorithmic uses of the Feferman–Vaught theorem. *Annals of Pure and Applied Logic*, 126(1–3):159–213, 2004. doi:10.1016/j.apal.2003.11.002.

[22] G. Ringel. Das Geschlecht des vollständigen paaren Graphen. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 28(3):139–150, 1965. doi:10.1007/BF02993245.

[23] G. Ringel. Der vollständige paare Graph auf nichtorientierbaren Flächen. *Journal für die reine und angewandte Mathematik*, 220:88–93, 1965. URL: http://eudml.org/doc/150703.

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

[25] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages: Volume 3 Beyond Words*, pages 389–455. Springer Berlin Heidelberg, 1997. `doi:10.1007/978-3-642-59126-6_7`.