



HAL
open science

On the monadic second-order transduction hierarchy

Achim Blumensath, Bruno Courcelle

► **To cite this version:**

Achim Blumensath, Bruno Courcelle. On the monadic second-order transduction hierarchy. 2008. hal-00287223v1

HAL Id: hal-00287223

<https://hal.science/hal-00287223v1>

Preprint submitted on 11 Jun 2008 (v1), last revised 7 Apr 2010 (v3)

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

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

ON THE MONADIC SECOND-ORDER TRANSDUCTION HIERARCHY

Achim Blumensath and Bruno Courcelle*

TU Darmstadt, Mathematik
Schloßgartenstraße 7
64289 Darmstadt, Germany
blumensath@mathematik.tu-darmstadt.de

Institut Universitaire de France and Bordeaux University, LaBRI
33405 Talence, France
courcell@labri.fr

We compare classes of finite relational structures via monadic second-order transductions. More precisely, we study the preorder

$$\mathcal{C} \sqsubseteq \mathcal{K} \quad : \text{iff} \quad \mathcal{C} \subseteq \tau(\mathcal{K}) \quad \text{for some transduction } \tau.$$

If we only consider classes of *incidence structures* we can completely describe the resulting hierarchy. It is linear of order type $\omega + 3$. Each level can be characterised in terms of a suitable variant of tree-width. Canonical representatives of the various levels are: the class of (i) all trees of height n , for $n \in \mathbb{N}$; (ii) all paths; (iii) all trees; and (iv) all grids.

1 INTRODUCTION

Monadic second-order logic (MSO) is one of the most expressive logics where quite a few interesting theories are still decidable. In particular, the infinite binary tree

*Supported by the GRAAL project of the 'Agence Nationale pour la Recherche'.

has a decidable MSO-theory [15, 19] and the same holds for many classes of (finite or infinite) structures with bounded *tree-width* [4, 17]. Furthermore, for every fixed MSO-sentence φ and every class \mathcal{C} of finite structures with bounded tree-width, there is a linear-time algorithm that checks whether a given structure from \mathcal{C} satisfies φ [5, 10]. Examples of the expressive power of monadic second-order logic include graph properties like k -colourability, various types of connectivity, and planarity (via Kuratowski’s well-known characterisation by forbidden configurations).

We also consider a variant of monadic second-order logic called *guarded second-order logic* (GSO) which allows quantification not only over sets of elements but also over sets of edges (i.e., tuples from the relations) [11]. The above mentioned linear-time algorithms can be adapted to this logic. There are tight links between guarded second-order logic and tree-width: every class of (finite or infinite) relational structures with a decidable GSO-theory has bounded tree-width. This gives a sort of converse to the above mentioned decidability results [18, 7]. The proof of this result uses a deep theorem of graph minor theory by Robertson and Seymour: a set of graphs has bounded tree-width if and only if it excludes some planar graph as a minor [17].

To compare the MSO-theories or GSO-theories of two classes of structures we can use *monadic second-order transductions*, a certain kind of interpretations suitable both, for monadic second-order logic and, with a technical trick, also for guarded second-order logic [4, 6, 7, 8].

In the present article we classify classes of finite structures according to their ‘combinatorial complexity’. (Note that we do not consider decidability issues.) We have already mentioned two ways to define the complexity of such classes. We can compare them via transductions, or we can see which variants of tree-width are bounded for the class. As it turns out these two approaches are equivalent and they give rise to the same hierarchy. This indicates the robustness of our definitions and their intrinsic interest. Other possible hierarchies, based on different logics, will be considered in Section 7.

Let us give more details. An MSO-transduction is a transformation of relational structures specified by monadic second-order formulae. As graphs can be represented by relational structures, we can use MSO-transductions as transformations between graphs. An MSO-transduction is a generalisation of the following kind of operations:

- (i) the definition of a relational structure “inside” another one (in model theory this is called an *interpretation*);
- (ii) the replacement of a structure \mathfrak{A} by the union of a fixed number of dis-

joint copies of \mathfrak{A} , augmented with appropriate relations between the copies (see Definition 3.3 (a));

(iii) the expansion of a given structure \mathfrak{A} by a fixed number of unary predicates, called *parameters*. Usually, these predicates are arbitrary subsets of the domain, but we also may have a formula imposing restrictions on them.

Because of the possibility to use parameters, a transduction τ is a many-valued map in general. (We may also think of it as non-deterministic.) Each relational structure \mathfrak{A} has several images $\tau(\mathfrak{A}, P_1, \dots, P_n)$ depending on the choice of the parameters $P_1, \dots, P_n \subseteq A$. If $\mathfrak{B} = \tau(\mathfrak{A}, \bar{P})$ we can consider \bar{P} as an *encoding* of \mathfrak{B} in \mathfrak{A} . The transduction τ is the corresponding *decoding function*.

Each transduction extends in a canonical way to a transformation between classes of structures. If \mathcal{C} and \mathcal{K} are classes of relational structures with $\mathcal{C} \subseteq \tau(\mathcal{K})$, we can think of τ as a way of encoding the structures in \mathcal{C} by elements of \mathcal{K} . For instance, every finite graph can be encoded in a sufficiently large finite square grid (by a fixed transduction τ). Every finite tree of height at most n (for fixed n) can be encoded in a sufficiently long finite path. But it is not the case that all finite trees can be encoded by paths (by a single transduction).

The purpose of this article is to classify classes of finite relational structures according to their *encoding powers*. We will compare classes \mathcal{C} and \mathcal{K} of structures by the following preorder:

$$\mathcal{C} \sqsubseteq \mathcal{K} \quad : \text{iff} \quad \mathcal{C} \subseteq \tau(\mathcal{K}) \quad \text{for some MSO-transduction } \tau.$$

We attack the problem of determining the structure of this preorder. Unfortunately, we only obtain a partial result since we are still missing some tools needed to give a complete characterisation.

In the case of *guarded second-order logic* we are more fortunate. Here, the corresponding hierarchy can be described completely. To obtain a corresponding notion of transduction we cannot simply change the definition of an MSO-transduction to use GSO-formulae since the resulting notion of transduction would not yield a reduction between GSO-theories, and it even would not be closed under composition. Instead, we will take a detour by combining ordinary MSO-transductions with a well-known translation between GSO and MSO.

This translation is based on *incidence structures*. Let us first describe this notion for undirected graphs where it is very natural. There are two canonical ways to encode a graph \mathfrak{G} by a relational structure. We can use its *adjacency representation* which is a structure $\langle V, \text{edg} \rangle$ where the domain V consists of all vertices of \mathfrak{G} and edg is a binary relation containing all pairs of adjacent vertices. But we also

can use the *incidence representation* of \mathcal{G} . This is the structure $\langle V \cup E, \text{in} \rangle$ where the domain $V \cup E$ contains both, the vertices and the edges of \mathcal{G} , and in is the incidence relation between vertices and edges. In a similar way, we can associate with every relational structure \mathfrak{A} its incidence structure \mathfrak{A}_{in} (see Definition 2.1) where the domain also contains elements for all tuples in some relation of \mathfrak{A} .

It is shown in [11] that every GSO-formula φ talking about some structure \mathfrak{A} can be translated into an MSO-formula talking about the incidence structure \mathfrak{A}_{in} , and vice versa. Hence, we can use incidence structures to obtain an analogue \sqsubseteq_{in} of the preorder \sqsubseteq suitable for guarded second-order logic. We set

$$\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{K} \quad \text{: iff} \quad \mathcal{C}_{\text{in}} \subseteq \tau(\mathcal{K}_{\text{in}}) \quad \text{for some MSO-transduction } \tau,$$

where $\mathcal{C}_{\text{in}} := \{ \mathfrak{A}_{\text{in}} \mid \mathfrak{A} \in \mathcal{C} \}$. The main result of the present article is a complete characterisation of the resulting hierarchy for classes of finite structures. We show that the preorder \sqsubseteq_{in} is linear of order type $\omega + 3$. It turns out that every class of finite structures is equivalent to one of the following classes, listed in increasing order of generality:

- ◆ trees of height at most $n \in \mathbb{N}$;
- ◆ paths;
- ◆ arbitrary trees (equivalently, binary trees);
- ◆ (square) grids.

Each of these levels can be characterised in terms of tree decompositions. Hence, we also obtain a corresponding hierarchy of all complexity measures on structures that are compatible with MSO-transductions.

The upper levels of the hierarchy can be determined easily using the machinery of minors and tree decompositions developed by Robertson and Seymour. In particular, we employ two results characterising bounded tree-width and bounded path-width in terms of excluded minors [16, 17].

For the lower levels consisting of classes of bounded path-width, the characterisation is more complicated requiring the development of new results relating tree decomposition and monadic second-order logic.

In Sections 2 and 3 we give basic definitions. Section 4 collects some known results from graph minor theory. In particular, we present the connection between tree-width and monadic second-order transductions. We also prove some new results in this direction. In Section 5 we introduce the transduction hierarchy and we describe the structure of its upper part. The lower part is studied in Section 6. The final Section 7 contains some extension of our results to other logics and some open problems in this direction.

2 PRELIMINARIES

Let us fix our notation. We set $[n] := \{0, \dots, n-1\}$ and we write $\wp(X)$ for the power set of a set X . In this article we only consider finite relational structures $\mathfrak{A} = \langle A, R_0^{\mathfrak{A}}, \dots, R_{m-1}^{\mathfrak{A}} \rangle$. The *signature* of such a structure is the set $\Sigma = \{R_0, \dots, R_{m-1}\}$ of relation symbols. We write $\text{ar}(R)$ for the arity of a relation R . For a signature Σ , we denote by $\text{STR}[\Sigma]$ the class of all finite Σ -structures.

We mainly consider *incidence structures*. These are representations of structures \mathfrak{A} where we have added new elements to the domain, one for each tuple in the relations of \mathfrak{A} .

Definition 2.1. Let $\mathfrak{A} = \langle A, R_0^{\mathfrak{A}}, \dots, R_{m-1}^{\mathfrak{A}} \rangle$ be a structure and let r be the maximal arity of a relation R_i . The *incidence structure* of \mathfrak{A} is the structure

$$\mathfrak{A}_{\text{in}} := \langle A \cup E, P_{R_0}, \dots, P_{R_{m-1}}, \text{in}_0, \dots, \text{in}_{r-1} \rangle,$$

where we extend the domain by

$$E := R_0^{\mathfrak{A}} \cup \dots \cup R_{m-1}^{\mathfrak{A}},$$

and the relations are

$$\begin{aligned} P_{R_i} &:= \{ \bar{c} \in E \mid \bar{c} \in R_i^{\mathfrak{A}} \}, \\ \text{in}_i &:= \{ (a, \bar{c}) \in A \times E \mid |\bar{c}| > i \text{ and } a = c_i \}. \end{aligned}$$

The class of all incidence structures is $\text{STR}_{\text{in}}[\Sigma] := \{ \mathfrak{A}_{\text{in}} \mid \mathfrak{A} \in \text{STR}[\Sigma] \}$.

Remark. Note that incidence structures are binary (i.e., their relations have arity at most 2). Hence, they can be regarded as bipartite labelled directed graphs.

One important property of incidence structures is the fact that they are *sparse*, i.e., their relations contain few tuples.

Definition 2.2. Let $k \in \mathbb{N}$. A structure $\mathfrak{A} = \langle A, \bar{R} \rangle$ is *k-sparse*¹ if, for every subset $X \subseteq A$ and all relations R_i , we have

$$|R_i \cap X^{\text{ar}(R_i)}| \leq k \cdot |X|.$$

Lemma 2.3. *If \mathfrak{A} is a structure with relations of arity at most r then \mathfrak{A}_{in} is r -sparse.*

¹In [8] the term *uniformly k-sparse* was used.

Sometimes it is possible to reduce results about relational structures to statements about undirected graphs. One way to do so consists in replacing the structure by its *Gaifman graph*.

Definition 2.4. The *Gaifman graph* of a structure $\mathfrak{A} = \langle A, \bar{R} \rangle$ is the undirected graph

$$\text{Gf}(\mathfrak{A}) := \langle A, \text{edg} \rangle,$$

with the same domain A and with the edge relation

$$\text{edg} := \{ (u, v) \mid u \neq v \text{ and there is some } \bar{c} \in R_i^{\mathfrak{A}} \text{ with } u, v \in \bar{c} \}.$$

Let us fix our notation regarding trees.

Definition 2.5. Let D be a set.

(a) We denote by $D^{<\omega}$ the set of all finite sequences of elements of D . The *prefix relation* on $D^{<\omega}$ is defined by

$$x \leq y \quad \text{iff} \quad y = xz, \quad \text{for some } z \in D^{<\omega}.$$

The infimum of x and y with respect to \leq , i.e., their longest common prefix, is denoted by $x \sqcap y$.

(b) A prefix closed subset $T \subseteq D^{<\omega}$ is called a *tree domain*. The domain of the *complete m -ary tree of height n* is $m^{<n}$. We distinguish two ways to represent a tree as a relational structure.

An *order tree* is a structure isomorphic to a partial order of the form $\langle T, \leq \rangle$ where T is a tree domain. If we replace the order \leq by the corresponding immediate successor relation we obtain a *successor tree* $\langle T, E \rangle$. A *coloured tree* is the expansion of a (order or successor) tree by unary predicates \bar{P} . (Hence, every vertex may have none, one, or several colours.) We write TREE_m for the class of all order trees $\langle T, \leq, P_0, \dots, P_{m-1} \rangle$ with m colours. The set of *leaves* of a tree \mathfrak{T} is denoted by $\text{Lf}(\mathfrak{T})$.

(c) Let $\mathfrak{T} = \langle T, \leq \rangle$ be an order tree. The *level* of an element $v \in T$ is the number of its predecessors. We denote it by $|v|$. The *height* of \mathfrak{T} is the least ordinal α greater than the level of every element of T . Hence, the empty tree has height 0 and the tree with a single vertex has height 1. The *out-degree* of \mathfrak{T} is the maximal number of immediate successors of a vertex of \mathfrak{T} . For successor trees we define these notions analogously.

(d) Let \mathfrak{T} be a tree and v a vertex of \mathfrak{T} . The *subtree* of \mathfrak{T} *rooted* at v is the subtree \mathfrak{T}_v consisting of all vertices u with $v \leq u$.

3 MONADIC SECOND-ORDER LOGIC AND TRANSDUCTIONS

Monadic second-order logic (MSO) is the extension of first-order logic by set variables and quantifiers over such variables. An important variant of MSO is guarded second-order logic (GSO) where one can quantify not only over sets of elements but also over sets of tuples from the relations (see [11] for details). Hence, guarded second-order logic over a given structure \mathfrak{A} is equivalent to monadic second-order logic over its incidence structure \mathfrak{A}_{in} .

Lemma 3.1 (Grädel, Hirsch, Otto [11]).

(a) *For every GSO-sentence φ , there exists an MSO-sentence ψ such that*

$$\mathfrak{A} \models \varphi \quad \text{iff} \quad \mathfrak{A}_{\text{in}} \models \psi, \quad \text{for all structures } \mathfrak{A}.$$

(b) *For every MSO-sentence φ , there exists a GSO-sentence ψ such that*

$$\mathfrak{A}_{\text{in}} \models \varphi \quad \text{iff} \quad \mathfrak{A} \models \psi, \quad \text{for all structures } \mathfrak{A}.$$

Throughout the article we will consistently work with incidence structures, thereby avoiding to treat with guarded second-order logic. In particular, all formulae are tacitly assumed to be MSO-formulae.

Besides MSO and GSO we also consider their *counting extensions* CMSO and CGSO. These add predicates of the form $|X| \cong k \pmod{m}$ to, respectively, MSO and GSO, where X is a set variable and k, m are numbers. All of our results for GSO go through also for CGSO, i.e., for CMSO-transductions between incidence structures. In Section 7 we will give a partial characterisation of the hierarchy for CMSO. Here, some results rely on the availability of counting predicates. Our results for MSO are strictly weaker.

To state the composition theorem below it is of advantage to work with a variant of MSO without first-order variables. This variant has atomic formulae of the form $X \subseteq Y$ and $R\bar{Z}$, for set variables X, Y, Z_0, Z_1, \dots , where a formula of the form $R\bar{Z}$ states that there are elements $a_i \in Z_i$ such that the tuple \bar{a} is in R . Note that every general monadic second-order formula with first-order variables can be brought into this restricted form by replacing all first-order variables by set variables and adding the condition that these sets are singletons.

Whenever we speak of MSO we will have this version in mind. In particular, the following definition of the rank of a formula is based on this variant. When writing down concrete formulae, on the other hand, we will allow the use of

first-order variables to improve readability. We regard every such formula as an abbreviation of a formula of the restricted form.

Definition 3.2. (a) The *rank* $\text{qr}(\varphi)$ of a formula φ is the nesting depth of quantifiers in φ . Formulae of rank 0 are called *quantifier-free*.

(b) The *monadic theory of rank m* of a structure \mathfrak{A} is

$$\text{MTh}_m(\mathfrak{A}) := \{ \varphi \in \text{MSO} \mid \mathfrak{A} \models \varphi, \text{qr}(\varphi) \leq m \}.$$

For a tuple \bar{a} of elements of \mathfrak{A} , we also consider the monadic theory $\text{MTh}_m(\mathfrak{A}, \bar{a})$ of the expansion $\langle \mathfrak{A}, \bar{a} \rangle$.

Remark. We use the term ‘rank’ instead of the more natural ‘quantifier rank’ since in Section 7 below we will consider CMSO where the notion of rank has to be adapted in order for our results to go through.

In order to compare the monadic theories of two classes of structures we employ MSO-transductions. To simplify the definition we introduce three simple operations and we obtain MSO-transductions as a combination of these.

Definition 3.3. (a) Let $k \in \mathbb{N}$ be a number. The operation copy_k maps a structure \mathfrak{A} to the expansion

$$\text{copy}_k(\mathfrak{A}) := \langle \mathfrak{A} \oplus \dots \oplus \mathfrak{A}, \sim, P_0, \dots, P_{k-1} \rangle$$

of the disjoint union of k copies of \mathfrak{A} by the following relations. Denoting the copy of an element $a \in A$ in the i -th component by the pair $\langle a, i \rangle$, we define

$$P_i := \{ \langle a, i \rangle \mid a \in A \} \quad \text{and} \quad \langle a, i \rangle \sim \langle b, j \rangle \text{ : iff } a = b.$$

(b) For $m \in \mathbb{M}$, we define the operation exp_m that maps a structure \mathfrak{A} to the class of all expansions by m unary predicates $Q_0, \dots, Q_{m-1} \subseteq A$. Note that this operation is many-valued.

(c) A *basic MSO-transduction* is a partial operation τ on relational structures described by a list

$$\langle \chi, \delta(x), \varphi_0(\bar{x}), \dots, \varphi_{s-1}(\bar{x}) \rangle$$

of MSO-formulae called the *definition scheme* of τ . Given a structure \mathfrak{A} that satisfies the formula χ the operation τ produces the structure

$$\tau(\mathfrak{A}) := \langle D, R_0, \dots, R_{s-1} \rangle$$

where

$$D := \{ a \in A \mid \mathfrak{A} \models \delta(a) \} \quad \text{and} \quad R_i := \{ \bar{a} \in D^{\text{ar}(R_i)} \mid \mathfrak{A} \models \varphi_i(\bar{a}) \}.$$

If $\mathfrak{A} \not\models \chi$ then $\tau(\mathfrak{A})$ remains undefined.

(d) A *k-copying MSO-transduction* τ is a (many-valued) operation on relational structures of the form $\tau_o \circ \text{copy}_k \circ \text{exp}_m$ where τ_o is a basic MSO-transduction.

Note that, due to exp_m , a structure can be mapped to several structures by a transduction. Consequently, we consider $\tau(\mathfrak{A})$ as the *set* of possible values $(\tau_o \circ \text{copy}_k)(\mathfrak{A}, \bar{P})$ where \bar{P} ranges over all m -tuples of subsets of A .

For classes \mathcal{C} , we set

$$\tau(\mathcal{C}) := \bigcup \{ \tau(\mathfrak{A}) \mid \mathfrak{A} \in \mathcal{C} \}.$$

Remark. (a) Every basic MSO-transduction is a k -copying MSO-transduction since we can choose $\text{copy}_1 = \text{id}$ and $\text{exp}_0 = \text{id}$.

(b) The expansion by m unary predicates corresponds, in the terminology of [7, 8], to using m *parameters*. We will use this terminology, for instance, in the proof of Proposition 4.6.

Example. The operation mapping an incidence structure $\mathfrak{A}_{\text{in}} \in \text{STR}_{\text{in}}[\Sigma]$ to the structure $\text{Gf}(\mathfrak{A})_{\text{in}}$ is a k -copying MSO-transduction where $k = r(r-1)/2$ and r is the maximal arity of a relation in Σ .

The two most important properties of MSO-transductions are summarised in the following lemmas.

Lemma 3.4. *Let τ be an MSO-transduction. For every MSO-sentence φ , there exists an MSO-sentence φ^τ such that, for all structures \mathfrak{A} ,*

$$\mathfrak{A} \models \varphi^\tau \quad \text{iff} \quad \mathfrak{B} \models \varphi \quad \text{for some } \mathfrak{B} \in \tau(\mathfrak{A}).$$

Lemma 3.5 (Courcelle [6]). *For all MSO-transductions σ, τ there exists an MSO-transduction ρ with $\rho = \sigma \circ \tau$.*

As a first example of transductions note that we can use them to translate between order trees and successor trees.

Lemma 3.6. (a) *There exists a transduction τ mapping an order tree to the corresponding successor tree.*

(b) *There exists a transduction σ mapping a successor tree to the corresponding order tree.*

A similar, but deeper, result states that there are transductions translating between a structure and its incidence structure.

Lemma 3.7. *For every signature Σ , there exists a transduction τ such that $\tau(\mathfrak{A}_{\text{in}}) = \mathfrak{A}$, for all $\mathfrak{A} \in \text{STR}[\Sigma]$.*

Theorem 3.8 (Courcelle [8, 1]). *For every signature Σ and all numbers $k \in \mathbb{N}$, there exist an MSO-transduction τ such that $\tau(\mathfrak{A}) = \mathfrak{A}_{\text{in}}$, for all k -sparse structures $\mathfrak{A} \in \text{STR}[\Sigma]$.*

We have seen in Lemma 3.4 that transductions relate the monadic theories of two structures. We also need techniques to relate the monadic theory of a structure to those of its substructures. The following generalisation of the disjoint union operation can frequently be used for this purpose (see [4] for an overview; for the special case of linear orders, a readable proof can be found in [20]).

Definition 3.9. Let $\mathfrak{J} = \langle I, \bar{S} \rangle$ be a structure and $(\mathfrak{A}^{(i)})_{i \in I}$ a family of structures $\mathfrak{A}^{(i)} = \langle A^{(i)}, \bar{R}^{(i)} \rangle$ indexed by elements i of \mathfrak{J} .

The *generalised sum* of $(\mathfrak{A}^{(i)})_{i \in I}$ is the structure

$$\sum_{i \in \mathfrak{J}} \mathfrak{A}^{(i)} := \langle U, \approx, \bar{R}', \bar{S}' \rangle$$

with domain $U := \{ \langle a, i \rangle \mid i \in I, a \in A^{(i)} \}$ and relations

$$\begin{aligned} \langle a, i \rangle \approx \langle b, j \rangle & \quad \text{iff} \quad i = j, \\ R'_l & := \{ (\langle a_0, i \rangle, \dots, \langle a_{n-1}, i \rangle) \mid i \in I \text{ and } \bar{a} \in R_l^{(i)} \}, \\ S'_l & := \{ (\langle a_0, i_0 \rangle, \dots, \langle a_{n-1}, i_{n-1} \rangle) \mid \bar{i} \in S_l \}. \end{aligned}$$

Theorem 3.10 (Shelah [19]). *For every MSO-sentence φ , we can construct a finite sequence of MSO-sentences $\chi_0, \dots, \chi_{s-1}$ and an MSO-sentence ψ such that*

$$\sum_{i \in \mathfrak{J}} \mathfrak{A}^{(i)} \models \varphi \quad \text{iff} \quad \langle \mathfrak{J}, \llbracket \chi_0 \rrbracket, \dots, \llbracket \chi_{s-1} \rrbracket \rangle \models \psi,$$

where $\llbracket \chi \rrbracket := \{ i \in I \mid \mathfrak{A}^{(i)} \models \chi \}$. Furthermore, we can choose the formulae χ_i to be of the same rank as φ .

This theorem is usually called the *Composition Theorem*. Below we will mainly make use of the following corollary.

Lemma 3.11. *Let \mathfrak{T} be an order tree and $v \in T$ a vertex. Suppose that \mathfrak{T}' is the order tree obtained from \mathfrak{T} by replacing the subtree \mathfrak{T}_v by some tree \mathfrak{G} . Let \vec{c} be a tuples of vertices of \mathfrak{T} with $v \notin c_i$, for all i . If \vec{a} are vertices of \mathfrak{T}_v and \vec{b} are vertices of \mathfrak{G} such that*

$$\text{MTh}_h(\mathfrak{T}_v, \vec{a}) = \text{MTh}_h(\mathfrak{G}, \vec{b})$$

then it follows that

$$\text{MTh}_h(\mathfrak{T}, \vec{a}\vec{c}) = \text{MTh}_h(\mathfrak{T}', \vec{b}\vec{c}).$$

Proof. Let \mathfrak{J} be the tree obtained from \mathfrak{T} by replacing the subtree \mathfrak{T}_v by a single vertex w . We can obtain \mathfrak{T} by a quantifier-free basic MSO-transduction from the generalised sum $\sum_{i \in \mathfrak{J}} \mathfrak{A}_i$ where $\mathfrak{A}_w := \mathfrak{T}_v$ and \mathfrak{A}_i is a singleton, for every $i \neq w$. In the same way, we obtain \mathfrak{T}' from the sum $\sum_{i \in \mathfrak{J}} \mathfrak{A}'_i$ where $\mathfrak{A}'_w := \mathfrak{G}$ and $\mathfrak{A}'_i := \mathfrak{A}_i$, for $i \neq w$. Let \vec{d}_i be the subtuple of $\vec{a}\vec{c}$ whose components belong to \mathfrak{A}_i and let \vec{d}'_i be the corresponding subtuple of $\vec{b}\vec{c}$. Since

$$\text{MTh}_h(\mathfrak{A}_i, \vec{d}_i) = \text{MTh}_h(\mathfrak{A}'_i, \vec{d}'_i), \quad \text{for all } i \in I,$$

it follows by Theorem 3.10 that

$$\text{MTh}_h(\mathfrak{T}, \vec{a}\vec{c}) = \text{MTh}_h(\mathfrak{T}', \vec{b}\vec{c}). \quad \square$$

4 MINORS, TREE DECOMPOSITIONS, AND TRANSDUCTIONS

Many properties of the transduction hierarchy we will introduce in the next section can be deduced from results in graph minor theory.

Definition 4.1. (a) Let $\mathfrak{G} = \langle V, E \rangle$ be an undirected graph and $F \subseteq E$ a set of edges. We denote by F^* the reflexive and transitive closure of F . The graph \mathfrak{G}/F is obtained by contracting all edges in F . Formally, we have

$$\mathfrak{G}/F := \langle \pi[V], \pi[E \setminus F^*] \rangle,$$

where $\pi : V \rightarrow V/F^*$ is the projection corresponding to the equivalence relation F^* .

(b) A *minor* of a graph \mathfrak{G} is a graph that can be obtained from \mathfrak{G} by first deleting some vertices and edges and then contracting some of the remaining edges. For a class \mathcal{C} of graphs, we denote by $\text{Min}(\mathcal{C})$ the class of all minors of graphs in \mathcal{C} .

Lemma 4.2 (Courcelle [7]). *There exists a transduction τ with $\tau(\mathfrak{G}_{\text{in}}) = \text{Min}(\mathfrak{G})$, for every graph \mathfrak{G} .*

Proof. A minor \mathfrak{H} of \mathfrak{G} is obtained by deleting vertices, deleting edges, and contracting edges. Hence, we can encode \mathfrak{H} by three sets: the set of vertices we delete, the set of edges we delete, and the set of edges we contract. With the help of these parameters we can define \mathfrak{H} inside of \mathfrak{G}_{in} by MSO-formulae. \square

Besides the notion of a minor we consider tree decompositions of structures and the corresponding notions of tree-width and path-width.

Definition 4.3. Let $\mathfrak{A} = \langle A, \bar{R} \rangle$ be a structure.

(a) A *tree decomposition* of \mathfrak{A} is a family $D = (U_v)_{v \in T}$ of (possibly empty) subsets $U_v \subseteq A$ indexed by an order tree T such that

- for every element $a \in A$, the set $\{v \in T \mid a \in U_v\}$ is nonempty and connected in T ;
- for every tuple $\bar{c} \in R_i$, there is some index $v \in T$ with $\bar{c} \subseteq U_v$.

We call T the *underlying tree* of the decomposition.

The *width* of a tree decomposition $D = (U_v)_{v \in T}$ is the number

$$\text{wd } D := \sup_{v \in T} (|U_v| - 1).$$

(The summand -1 has mainly aesthetic reasons.)

(b) The *tree-width* $\text{twd}(\mathfrak{A})$ of \mathfrak{A} is the minimal width of a tree decomposition of \mathfrak{A} .

(c) The *path-width* $\text{pwd}(\mathfrak{A})$ of \mathfrak{A} is the minimal width of a tree decomposition of \mathfrak{A} where the underlying tree is a path.

(d) The *n -depth tree-width* $\text{twd}_n(\mathfrak{A})$ of \mathfrak{A} is the minimal width of a tree decomposition of \mathfrak{A} whose underlying tree has height at most n .

Remark. The n -depth tree-width is related to the *tree-depth* $\text{td}(\mathfrak{G})$ of a graph \mathfrak{G} introduced by Nešetřil and Ossona de Mendez [13, 14]. The tree-depth of a connected graph \mathfrak{G} is the least number n such that \mathfrak{G} is a subgraph of some order tree of height n (ignoring edge directions). It follows that

- (a) $\text{td}(\mathfrak{G}) \leq n$ implies $\text{twd}_n(\mathfrak{G}) < n$.
- (b) $\text{twd}_n(\mathfrak{G}) < k$ implies $\text{td}(\mathfrak{G}) \leq nk$.

The next lemma shows that most questions regarding the tree decompositions of structures can be reduced to the corresponding questions for their Gaifman graphs. For many of the following results it is therefore sufficient to consider graphs.

Lemma 4.4. *Let \mathfrak{A} be a structure. A family $(U_v)_{v \in T}$ is a tree decomposition of \mathfrak{A} if and only if it is a tree decomposition of $\text{Gf}(\mathfrak{A})$.*

Proof. (\Rightarrow) is immediate. (\Leftarrow) follows from the fact that every tree decomposition of a clique has one component covering the whole clique. This implies that, for every clique C in $\text{Gf}(\mathfrak{A})$, there is some vertex $v \in T$ with $C \subseteq U_v$. \square

There is a close relationship between tree decompositions and transductions.

Lemma 4.5. *For every signature Σ and every number $k \in \mathbb{N}$, there exists a transduction $\tau_k : \text{TREE}_o \rightarrow \text{STR}_{\text{in}}[\Sigma]$ that maps an order tree T to the class of all incidence structures \mathfrak{A}_{in} such that the corresponding Σ -structure \mathfrak{A} has a tree decomposition of width at most k with underlying tree T .*

Proof. Suppose that \mathfrak{A} is a structure which has a tree decomposition $(U_v)_{v \in T}$ of width k . We prove that \mathfrak{A} can be defined from a colouring of T where the number of colours depends only on Σ and k .

Let $\mathfrak{C}_0, \dots, \mathfrak{C}_{m-1}$ be an enumeration of all Σ -structures whose domain is a subset of $[k+1]$. For each $v \in T$, let \mathfrak{U}_v be the substructure of \mathfrak{A} induced by U_v . It follows that, for every $v \in T$, we can find some index $\lambda(v)$ such that $\mathfrak{U}_v \cong \mathfrak{C}_{\lambda(v)}$. Let $\pi_v : \mathfrak{U}_v \rightarrow \mathfrak{C}_{\lambda(v)}$ be the corresponding isomorphism.

Furthermore, we associate with each edge (u, v) of T a binary relation

$$R(u, v) := \{ (\pi_u(a), \pi_v(a)) \mid a \in U_u \cap U_v \} \subseteq [k+1] \times [k+1].$$

We can recover \mathfrak{A} from T with the help of the vertex colouring λ and the edge colouring R . We form the disjoint union of all structures $(\mathfrak{C}_{\lambda(v)})_{\text{in}}$, for $v \in T$, and we identify two elements $i \in C_{\lambda(u)}$ and $j \in C_{\lambda(v)}$ if (u, v) is an edge of T such that $(i, j) \in R(u, v)$. This can be performed by a n -copying MSO-transduction where n is the maximal size of the structures $(\mathfrak{C}_i)_{\text{in}}$, $i < m$. \square

The converse is more involved. We defer the proof of the following proposition to the end of this section.

Proposition 4.6. *For every transduction $\tau : \text{TREE}_m \rightarrow \text{STR}_{\text{in}}[\Sigma]$, there exists a number $k \in \mathbb{N}$ such that, for each tree \mathfrak{T} with image $\mathfrak{A}_{\text{in}} \in \tau(\mathfrak{T})$, the structure \mathfrak{A} has a tree decomposition of width at most k where the underlying tree is \mathfrak{T} .*

In order to separate the higher classes of the hierarchy, we employ two deep results of Robertson and Seymour about excluded minors.

Theorem 4.7 (Excluded Tree Theorem [16]). *For each tree \mathfrak{T} , there exists a number $k \in \mathbb{N}$ such that*

$$\mathfrak{T} \notin \text{Min}(\mathfrak{G}) \text{ implies } \text{pwd}(\mathfrak{G}) < k, \quad \text{for every graph } \mathfrak{G}.$$

Theorem 4.8 (Excluded Grid Theorem [17]). *For each planar graph \mathfrak{E} , there exists a number $k \in \mathbb{N}$ such that*

$$\mathfrak{E} \notin \text{Min}(\mathfrak{G}) \text{ implies } \text{twd}(\mathfrak{G}) < k, \quad \text{for every graph } \mathfrak{G}.$$

For the separation of the lower degrees of the hierarchy we will need a result stating that the class of trees underlying a tree decomposition of a given structure can be obtained from the structure with the help of a transduction. In general, it is impossible to obtain *all* such trees via a transduction, but we can get *some* of them. There is a general result of Lapoire [12] to this effect. We will only need the special case of tree decompositions of bounded height where the proof is much simpler. As convenience for the reader we give a complete proof for this special case.

Definition 4.9. Let $(U_\nu)_{\nu \in T}$ be a tree decomposition of a structure \mathfrak{A} .

(a) We define a function $\mu : A \rightarrow T$ by

$$\mu(a) := \min \{ \nu \in T \mid a \in U_\nu \}.$$

(b) For $\nu \in T$, we set

$$U_{\uparrow\nu} := \bigcup_{u \geq \nu} U_u \setminus \bigcup_{u < \nu} U_u.$$

(c) The tree decomposition $(U_\nu)_{\nu \in T}$ is *strict* if, for every $\nu \in T$,

- ♦ $U_\nu \cap \mu(A) \neq \emptyset$, and
- ♦ if ν is not the root of T then $U_{\uparrow\nu}$ is connected.

Definition 4.10. Let $D = (U_\nu)_{\nu \in T}$ be a tree decomposition and let F be a set of edges of (the successor tree corresponding to) T . The tree decomposition D/F obtained by *contracting* the edges in F is

$$D/F := (U'_{[\nu]})_{[\nu] \in T/F},$$

where $U'_{[\nu]} := \bigcup_{u \in [\nu]} U_u$.

Lemma 4.11. *Let \mathfrak{G} be a graph. For every tree decomposition $(U_\nu)_{\nu \in T}$ of \mathfrak{G} , there exists a strict tree decomposition $(U'_\nu)_{\nu \in T'}$ of \mathfrak{G} whose width and height are at most those of $(U_\nu)_{\nu \in T}$.*

Proof. By induction on $n \in \mathbb{N}$, we construct a sequence $(U_\nu^n)_{\nu \in T_n}$ of tree decompositions such that, $U_{\uparrow \nu}^n$ is connected, for every $\nu \in T_n$ with $0 < |\nu| \leq n$. (Recall that $|\nu|$ denotes the level of ν .) Furthermore, the restriction of T_n to the set of vertices of level at most n coincides with the corresponding restriction of T_{n+1} , and we have $U_\nu^{n+1} = U_\nu^n$, for all $\nu \in T_{n+1}$ with $|\nu| \leq n$. It follows that the sequence has a limit $(U_\nu^\omega)_{\nu \in T_\omega}$ where

$$T_\omega := \bigcup_{n \in \mathbb{N}} \{ \nu \in T_n \mid |\nu| \leq n \} \quad \text{and} \quad U_\nu^\omega := U_\nu^{|\nu|}.$$

We start the construction with $T_0 := T$ and $U_\nu^0 := U_\nu$. Suppose that we have already defined $(U_\nu^n)_{\nu \in T_n}$. For every vertex $\nu \in T_n$ of level $|\nu| = n+1$ we modify the tree decomposition as follows. Let C_0, \dots, C_{m-1} be an enumeration of the connected components of $U_{\uparrow \nu}^n$. We replace in T_n the subtree rooted at ν by m copies S_0, \dots, S_{m-1} of the subtree, all attached to the predecessor of ν . For $u \in S_i$ we define $U_u^{n+1} := U_u^n \cap C_i$. We can do these modifications for all vertices of level $n+1$ simultaneously. Let $(U_\nu^{n+1})_{\nu \in T_{n+1}}$ be the resulting tree decomposition.

The limit $(U_\nu^\omega)_{\nu \in T_\omega}$ of this sequence satisfies the connectedness requirement of a strict tree decomposition. To also satisfy the other condition we proceed as follows. Let F be the set of all edges (u, ν) of T_ω such that $U_\nu \cap \mu(V) = \emptyset$. (Note that this implies $U_\nu \subseteq U_u$.) We construct the tree decomposition $(U'_\nu)_{\nu \in T'}$ by contracting all edges in F . \square

Theorem 4.12. *For each constant $n \in \mathbb{N}$, there exists an MSO-formula $\varphi_n(x, y; \vec{Z})$ such that, for every strict tree decomposition $D = (U_\nu)_{\nu \in T}$ of a graph \mathfrak{G} of height at most n , there are sets $L_0, \dots, L_{n-1} \subseteq V$ such that*

$$\mathfrak{G} \models \varphi_n(a, b; \vec{L}) \quad \text{iff} \quad \mu(a) \leq \mu(b).$$

Proof. Given D we use the sets

$$L_i := \{ a \in V \mid |\mu(a)| = i \}$$

of all elements that first appear at level i of the tree. For $k < n$, let $\mathfrak{G}_{\geq k}$ be the subgraph of \mathfrak{G} induced by $L_k \cup \dots \cup L_{n-1}$. For $a \in L_i$ and $b \in L_j$ we define

$$a \leq b \quad \text{iff} \quad i \leq j \text{ and } a, b \text{ belong to the same connected component of } \mathfrak{G}_{\geq i},$$

and $a \sim b$ iff $a \leq b$ and $b \leq a$.

Clearly, the relation \leq is MSO-definable with the help of the parameters \bar{L} . We claim that

$$a \leq b \quad \text{iff} \quad \mu(a) \leq \mu(b).$$

(\Leftarrow) Suppose that $\mu(a) \leq \mu(b)$. Then $b \in U_{\uparrow\mu(a)}$. Furthermore, $U_{\uparrow\mu(a)}$ is connected since D is strict. Hence, $U_{\uparrow\mu(a)}$ is a connected component of $\mathfrak{G}_{\geq i}$ containing both a and b . Since $|\mu(a)| \leq |\mu(b)|$ it follows that $a \leq b$.

(\Rightarrow) Suppose that $a \leq b$. Then there exists an undirected path π in $\mathfrak{G}_{\geq |\mu(a)|}$ connecting a and b . Since $U_{\uparrow u} \cap U_{\uparrow v} = \emptyset$, for all $u \neq v$ with $|u| = |v|$, it follows that π is contained in some $U_{\uparrow v}$ with $|v| = |\mu(a)|$. Since a is a vertex of π we must have $v = \mu(a)$. Furthermore, $b \in U_{\uparrow\mu(a)}$ since b is also a vertex of π . This implies that $\mu(a) \leq \mu(b)$. \square

Corollary 4.13. *For each constant $n \in \mathbb{N}$, there exists a transduction τ_n mapping a graph \mathfrak{G} to the class of all (underlying trees of) strict tree decompositions of \mathfrak{G} of height at most n .*

We also need a variant of Theorems 4.7 and 4.8 for n -depth tree-width. The next lemma contains the main technical argument.

Lemma 4.14. *Suppose that \mathfrak{G} is a graph that does not contain a path of length l and let $D = (U_v)_{v \in T}$ be a strict tree decomposition of \mathfrak{G} of width k . Then the height of T is at most l^{k+1} .*

Proof. We will prove the following claim. Let $v_o < \dots < v_n$ be a path in T of length $n + 1$ and let $C \subseteq V$ be a set of vertices such that $C \subseteq U_{v_i}$, for all i . Then we claim that $n < l^{k+1-|C|}$.

Clearly, the statement of the lemma follows from this claim by setting $C := \emptyset$. We prove the claim by induction on $|C|$, starting with large sets C . If $|C| = k + 1$ then we have $U_{v_i} = C$, for all $i \leq n$. As D is strict this implies that $n = o < l^o$. For the induction step, suppose that $|C| \leq k$. For every $i < n$, we select an element $a_i \in U_{v_{i+1}} \setminus U_{v_i}$. Since D is strict there exists a path $\pi = b_o \dots b_{m-1}$ in \mathfrak{G} connecting a_o and a_{n-1} such that $\pi \subseteq V \setminus U_o$. In particular, we have $\pi \cap C = \emptyset$. By assumption on \mathfrak{G} we have $m < l$. Since every component U_{v_i} contains at least one of the elements b_j it follows that there is some index j such that at least $n/m > n/l$ of the U_{v_i} contain the vertex b_j . Suppose that $b_j \in U_{v_i}$, for $s \leq i \leq s + n/l$. Then $C \cup \{b_j\}$ is contained in all these U_{v_i} and it follows by induction hypothesis that $n/l < l^{k-|C|}$. Hence, $n < l^{k+1-|C|}$. \square

Theorem 4.15 (Excluded Path Theorem). *For each path \mathfrak{P} , there exist numbers $n, k \in \mathbb{N}$ such that*

$$\mathfrak{P} \notin \text{Min}(\mathfrak{G}) \quad \text{implies} \quad \text{twd}_n(\mathfrak{G}) < k, \quad \text{for every graph } \mathfrak{G}.$$

Proof. By Theorem 4.7, there is a number k such that $\mathfrak{P} \notin \text{Min}(\mathfrak{G})$ implies $\text{pwd}(\mathfrak{G}) < k$. Hence, \mathfrak{G} has a tree decomposition D of width smaller than k . We can use Lemma 4.11 to find a strict tree decomposition D' of \mathfrak{G} of the same width. According to the preceding lemma D' has height at most $n := l^{k+1}$ where l is the length of \mathfrak{P} . Consequently, $\text{twd}_n \mathfrak{G} < k$. \square

It remains to prove Proposition 4.6. As a technical tool we introduce a second kind of hierarchical decompositions of structures and a corresponding notion of width. To simplify the definition we will only consider incidence structures.

Definition 4.16. Let $\mathfrak{A}_{\text{in}} = \langle A \cup E, \bar{P}, \text{in}_o, \dots \rangle$ be an incidence structure.

(a) A *partition refinement* of \mathfrak{A}_{in} is a family $\Pi = (W_v, \approx_v)_{v \in T}$ of pairs consisting of a subset $W_v \subseteq A \cup E$ and an equivalence relation \approx_v on W_v with the following properties:

- ◆ The index set T is a tree.
- ◆ $W_{\langle \rangle} = A \cup E$
- ◆ For every internal vertex $u \in T$ with immediate successors v_o, \dots, v_{n-1} , the sets $W_{v_o}, \dots, W_{v_{n-1}}$ form a partition of W_u .
- ◆ $|W_u| = 1$, for every leaf $u \in T$.
- ◆ $x \approx_v y$ and $u \leq v$ implies $x \approx_u y$.
- ◆ If u is an internal vertex of T , v, w immediate successors of u , not necessarily distinct, and $x \in W_v, y \in W_w$ elements, then $x \approx_u y$ implies either

$$x, y \in A \text{ and } (x, e) \in \text{in}_i \Leftrightarrow (y, e) \in \text{in}_i \text{ for every } e \in E \setminus (W_v \cup W_w)$$

$$\text{or } x, y \in E \text{ and } (a, x) \in \text{in}_i \Leftrightarrow (a, y) \in \text{in}_i \text{ for every } a \in A \setminus (W_v \cup W_w).$$

(b) The *width* of a partition refinement $\Pi = (W_v, \approx_v)_{v \in T}$ is the maximum number of equivalence classes realised in some component W_v :

$$\text{wd } \Pi := \max_{v \in T} |W_v / \approx_v|.$$

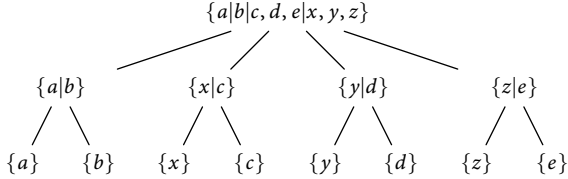
The *partition-width* of the structure \mathfrak{A}_{in} is the minimal width of a partition refinement of \mathfrak{A}_{in} .

The notion of a partition refinement and of partition-width is an adaptation of definitions from [3, 2]. Up to a factor of 2, the partition-width of an incidence structure and its clique-width coincide.

Example. Let $\mathfrak{A} = (A, R)$ be a structure with domain $A = \{a, b, c, d, e\}$ and a ternary relation

$$R = \{ \underbrace{(a, b, c)}_x, \underbrace{(a, b, d)}_y, \underbrace{(a, b, e)}_z \}.$$

Its incidence structure is $\mathfrak{A}_\epsilon = (A \cup E, P_R, \text{in}_o, \text{in}_1, \text{in}_2)$ with $E = \{x, y, z\}$. We obtain a partition refinement



where we have indicated the partition into \approx_v -classes by vertical bars. Note that this partition refinement has width 4.

Lemma 4.17. *For every partition refinement $\Pi = (W_v, \approx_v)_{v \in T}$ of an incidence structure $\mathfrak{A}_{\text{in}} = \langle A \cup E, \tilde{P}, \text{in}_o, \dots, \text{in}_{r-1} \rangle$, there exists a tree decomposition $D = (U_u)_{u \in T}$ of \mathfrak{A}_{in} with the same underlying tree T such that*

$$\text{wd } D \leq (r + 3) \cdot \text{wd } \Pi.$$

Proof. Let $l : A \cup E \rightarrow \text{Lf}(T)$ be the function assigning to every $x \in A \cup E$ the unique leaf $l(x)$ of T such that $U_{l(x)} = \{x\}$. We claim that the desired tree decomposition $(U_u)_{u \in T}$ of \mathfrak{H} is given by

$$U_u := B_u \cup C_u \cup D_u$$

where

$$\begin{aligned}
B_u &:= \{ v \in A \mid u \leq l(v) \text{ and } (v, e) \in \text{in}_i \text{ for some } i < r \text{ and } e \in E \text{ with} \\
&\quad u \not\leq l(e) \}, \\
C_u &:= \{ v \in A \mid u \not\leq l(v) \text{ and } (v, e) \in \text{in}_i \text{ for some } i < r \text{ and } e \in E \text{ with} \\
&\quad u \leq l(e) \}, \\
D_u &:= \{ v \in A \mid (v, e) \in \text{in}_i \text{ for some } i < r \text{ and } e \in E \text{ with} \\
&\quad l(v) \sqcap l(e) = u \}.
\end{aligned}$$

Note that the connectedness condition holds since $(v, e) \in \text{in}_i$ implies that v belongs to every component U_u such that u lies on the path from $l(v)$ to $l(e)$.

It remains to prove that $|U_u| \leq (r+3) \cdot \text{wd } \Pi$. If $u = l(\tilde{c})$, for some $\tilde{c} \in E$, then $U_u = C_u$ consists the components of \tilde{c} . Hence, $|U_u| = |\tilde{c}| \leq r$. Therefore, we may assume that $u \notin l[E]$. Let

$$[x]_u := \{ y \in W_u \mid y \approx_u x \}$$

denote the \approx_u -class of x . We prove the following bounds.

- (1) $|[x]_u| = 1$, for all $x \in B_u$.
- (2) $|[x]_u \cap U_u| \leq 2$, for all $x \in D_u$.
- (3) $|C_u| \leq r \cdot |W_u / \approx_u|$.

Then it follows that $|U_u| = |B_u \cup C_u \cup D_u| \leq (r+3) \cdot |W_u / \approx_u|$.

(1) Let $x \in B_u$. There is some tuple $e \in E$ and some index i with $(x, e) \in \text{in}_i$ and $u \not\leq l(e)$. We have $(y, e) \in \text{in}_i$, for every $y \in W_u$ with $y \approx_u x$. Since x is the only such element it follows that $[x]_u = \{x\}$.

(2) Let $x \in D_u$. There is some tuple $e \in E$ and some i with $(x, e) \in \text{in}_i$ and $u = l(x) \sqcap l(e)$. Let v be the immediate successor of u with $v \leq l(e)$. We have $(y, e) \in \text{in}_i$, for all $y \in W_u \setminus W_v$ with $y \approx_u x$. Hence, $[x]_u \setminus W_v = \{x\}$.

Suppose that there is some element $y \in [x]_u \cap W_v \cap U_u$. By definition of U_u there is some tuple $f \in E$ and some j with $(y, f) \in \text{in}_j$ and $l(y) \sqcap l(f) \leq u$. As above it follows that $[x]_u \cap W_v = \{y\}$. Consequently, we have $|[x]_u \cap U_u| \leq 2$.

(3) Let $x \in C_u$ and consider some tuple $e \in E$ with $(x, e) \in \text{in}_i$ and with $u \leq l(e)$. Set

$$I_u(e) := \{ z \in A \mid (z, e) \in \text{in}_i \text{ for some } i \text{ and } l(z) \not\leq u \}.$$

For $e, f \in E \cap W_u$, it follows that

$$e \approx_u f \quad \text{implies} \quad I_u(e) = I_u(f).$$

Furthermore, we obviously have $|I_u(e)| \leq |e| \leq r$. It follows that C_u contains at most $r \cdot |W_u/\approx_u|$ vertices. \square

Lemma 4.18. *Let $\tau : \text{TREE}_m \rightarrow \text{STR}_{\text{in}}[\Sigma]$ be a basic MSO-transduction such that, for every tree \mathfrak{T} with image $\mathfrak{A}_{\text{in}} \in \tau(\mathfrak{T})$, we have*

$$A \cup E = \text{Lf}(\mathfrak{T}) \quad \text{and} \quad A \cap E = \emptyset.$$

Then there exists a finite constant w such that, for every order tree \mathfrak{T} , we can find a partition refinement $(W_v, \approx_v)_{v \in T}$ of $\tau(\mathfrak{T})$ of width at most w .

Proof. Let $(\chi, \delta(x), (\varphi_{P_R}(x))_R, (\varphi_{\text{in}_i}(x, y))_i)$ be the definition scheme of τ . Let h be the maximal rank of these formulae.

Given \mathfrak{T} we define the desired partition refinement $\Pi = (W_u, \approx_u)_{u \in T}$ by setting

$$W_u := \{x \in \text{Lf}(T) \mid u \leq x\},$$

and $x \approx_u y$:iff $\text{MTh}_h(\mathfrak{T}_v, x) = \text{MTh}_h(\mathfrak{T}_w, y)$,
where v, w are the immediate successors of u with
 $x \in W_v$ and $y \in W_w$.

(If u is a leaf of T then $W_u = \{x\}$ and we take the equality relation for \approx_u .) Note that the index of \approx_v is finite and it only depends on h and not on the input tree \mathfrak{T} .

It remains to show that Π is really a partition refinement. First, let us prove that $x \approx_v y$ and $u \leq v$ implies $x \approx_u y$. It is sufficient to consider the case that u is the immediate predecessor of v . Then the general case follows by induction. Hence, suppose that v is the immediate successor of u , that w, w' are immediate successors of v , and that x, y are leafs with $w \leq x$ and $w' \leq y$ such that $x \approx_v y$. Then we have

$$\text{MTh}_h(\mathfrak{T}_w, x) = \text{MTh}_h(\mathfrak{T}_{w'}, y),$$

which, by Lemma 3.11, implies that

$$\text{MTh}_h(\mathfrak{T}_v, x) = \text{MTh}_h(\mathfrak{T}_v, y).$$

Consequently, we have $x \approx_u y$.

We also have to show that the incidence relation is invariant under \approx_u . Let v, w be immediate successors of u and suppose that x, y are leaves with $v \leq x$ and $w \leq y$ such that $x \approx_u y$. We have to distinguish two cases.

Suppose that $x, y \in V$ and let $e \in E \setminus (W_v \cup W_w)$ be an edge. Since

$$\text{MTh}_h(\mathfrak{T}_v, x) = \text{MTh}_h(\mathfrak{T}_w, y),$$

it follows that

$$\mathfrak{T} \models \varphi_{\text{in}_i}(x, e) \quad \text{iff} \quad \mathfrak{T} \models \varphi_{\text{in}_i}(y, e).$$

Hence, $(x, e) \in \text{in}_i$ iff $(y, e) \in \text{in}_i$.

Now, suppose that $x, y \in E$ and let $z \in V \setminus (W_v \cup W_w)$ be a vertex. Since

$$\text{MTh}_h(\mathfrak{T}_v, x) = \text{MTh}_h(\mathfrak{T}_w, y),$$

it follows that

$$\mathfrak{T} \models \varphi_{\text{in}_i}(z, x) \quad \text{iff} \quad \mathfrak{T} \models \varphi_{\text{in}_i}(z, y).$$

Hence, $(z, x) \in \text{in}_i$ iff $(z, y) \in \text{in}_i$. □

Proof of Proposition 4.6. (1) First, let us consider the case that the transduction $\tau : \text{TREE}_m \rightarrow \text{STR}_{\text{in}}[\Sigma]$ is a basic MSO-transduction such that, for every tree $\mathfrak{T} \in \text{dom}(\tau)$ with image $\mathfrak{A}_{\text{in}} \in \tau(\mathfrak{T})$, we have

$$A \cup E = \text{Lf}(\mathfrak{T}) \quad \text{and} \quad A \cap E = \emptyset.$$

It follows by Lemma 4.18 that there is a constant $w \in \mathbb{N}$ such that, for every tree \mathfrak{T} , we can find a partition refinement of $\tau(\mathfrak{T})$ with underlying tree \mathfrak{T} whose width is at most w . By Lemma 4.17 it follows that $\tau(\mathfrak{T})$ has a tree decomposition $(U_v)_{v \in T}$ with underlying tree \mathfrak{T} and whose width is bounded by $k := w(r + 3)$.

(2) If τ is a basic MSO-transduction such that

$$A \cup E \subseteq \text{Lf}(\mathfrak{T}) \quad \text{and} \quad A \cap E = \emptyset, \quad \text{for } \mathfrak{A}_{\text{in}} \in \tau(\mathfrak{T}) \text{ with } \mathfrak{T} \in \text{dom}(\tau),$$

then we can argue similarly. Let τ' be the MSO-interpretation mapping \mathfrak{T} to the structure obtained from $\tau(\mathfrak{T})$ by adding one isolated element for every leaf of \mathfrak{T} that does not correspond to an element of $\tau(\mathfrak{T})$. Then τ' is of the form considered in (1) and we obtain a tree decomposition $(U_v)_{v \in T}$ of $\tau'(\mathfrak{T})$. Deleting from every

component U_v all elements not in $\tau(\mathfrak{T})$ we obtain the desired tree decomposition of $\tau(\mathfrak{T})$.

(3) Suppose that τ is a non-copying MSO-transduction as in (2) but with p parameters. We can regard τ as a basic MSO-transduction $\text{TREE}_{m+p} \rightarrow \text{STR}_{\text{in}}[\Sigma]$. By (2) it follows that, for every value of the parameters \bar{P} , the hypergraph $\tau(\mathfrak{T}, \bar{P})$ has a tree decomposition of the required form.

(4) Finally, consider the general case. Suppose that τ is l -copying. Given \mathfrak{T} let \mathfrak{T}^+ be the tree obtained from \mathfrak{T} by adding l new immediate successors to every vertex of \mathfrak{T} . Formally, suppose that $T \subseteq D^{<\omega}$, for some set D . W.l.o.g. we may assume that $D \cap [l] = \emptyset$. We define the domain $T^+ \subseteq (D \cup [l])^{<\omega}$ of \mathfrak{T}^+ by

$$T^+ := T \cup T \times [l].$$

Furthermore, we add new colour predicates

$$S_i := T \times \{i\}, \quad \text{for } i \in [l].$$

Note that every element of $\tau(\mathfrak{T})$ is of the form $\langle v, i \rangle$ where $i \in [l]$ and $v \in T$. Hence, each such element corresponds to a leaf $vi \in T \times [l] \subseteq T^+$. Using the colours \tilde{S} we can construct a basic MSO-transduction $\tau^+ : \text{TREE}_{m+l} \rightarrow \text{STR}_{\text{in}}[\Sigma]$ satisfying the conditions of (3) such that $\tau^+(\mathfrak{T}^+) = \tau(\mathfrak{T})$. By (3), we obtain a tree decomposition $D^+ = (U_v^+)_{v \in T^+}$ of $\tau^+(\mathfrak{T}^+) = \tau(\mathfrak{T})$. Let $D = (U_v)_{v \in T}$ be the tree decomposition obtained from D^+ by contracting every edge leading to a leaf in $T^+ \setminus T$. Then

$$\text{wd } D + 1 \leq (l + 1)(\text{wd } D^+ + 1). \quad \square$$

5 THE TRANSDUCTION HIERARCHY

The focus of our investigation lies on the following preorder on classes of structures which compares their ‘encoding power’ with respect to MSO-transductions. Our main result is a complete description of the hierarchy induced by this preorder. It will be given in Theorem 5.4.

Definition 5.1. Let $\mathcal{C}, \mathcal{K} \subseteq \text{STR}$. We define the following relations.

- (a) $\mathcal{C} \sqsubseteq \mathcal{K}$ if there exists a transduction τ such that $\mathcal{C} \subseteq \tau(\mathcal{K})$.
- (b) $\mathcal{C} \sqsubset \mathcal{K}$ if $\mathcal{C} \sqsubseteq \mathcal{K}$ and $\mathcal{K} \not\sqsubseteq \mathcal{C}$.

- (c) $\mathcal{C} \equiv \mathcal{K}$ if $\mathcal{C} \sqsubseteq \mathcal{K}$ and $\mathcal{K} \sqsubseteq \mathcal{C}$.
- (d) $\mathcal{C} \sqsubseteq \mathcal{K}$ if $\mathcal{C} \sqsubset \mathcal{K}$ and there is no class \mathcal{D} with $\mathcal{C} \sqsubset \mathcal{D} \sqsubset \mathcal{K}$.
- (e) $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{K}$ if $\mathcal{C}_{\text{in}} \sqsubseteq \mathcal{K}_{\text{in}}$.
- (f) The relations \sqsubset_{in} , \equiv_{in} , and \sqsubseteq_{in} are defined analogously to \sqsubset , \equiv , \sqsubseteq by replacing \sqsubseteq everywhere by \sqsubseteq_{in} .

The *transduction hierarchy* is the hierarchy of classes $\mathcal{C} \subseteq \text{STR}$ induced by the relation \sqsubseteq_{in} .

As transductions are closed under composition it follows that the relation \sqsubseteq_{in} is a preorder, i.e., it is reflexive and transitive.

Lemma 5.2. \sqsubseteq_{in} is a preorder on $\wp(\text{STR})$.

Definition 5.3. We consider the following subclasses of $\text{STR}[\{\text{edg}\}]$. (All trees below are considered to be successor trees.)

- (a) $\mathcal{T}_n := \{ m^{<n} \mid m \in \mathbb{N} \}$ is the set of all complete m -ary trees of height n .
- (b) $\mathcal{T}_{\text{bin}} := \{ 2^{<n} \mid n \in \mathbb{N} \}$ is the set of all binary trees.
- (c) \mathcal{T}_ω is the set of all trees.
- (d) \mathcal{P} is the class of all finite paths.
- (e) \mathcal{G} is the class of all rectangular grids.

The following description of the transduction hierarchy is the main result of the present paper.

Theorem 5.4. *We have the following hierarchy:*

$$\emptyset \sqsubseteq_{\text{in}} \mathcal{T}_0 \sqsubseteq_{\text{in}} \mathcal{T}_1 \sqsubseteq_{\text{in}} \cdots \sqsubseteq_{\text{in}} \mathcal{T}_n \sqsubseteq_{\text{in}} \cdots \sqsubset_{\text{in}} \mathcal{P} \sqsubseteq_{\text{in}} \mathcal{T}_\omega \equiv_{\text{in}} \mathcal{T}_{\text{bin}} \sqsubseteq_{\text{in}} \mathcal{G}$$

For every signature Σ , every class $\mathcal{C} \subseteq \text{STR}[\Sigma]$ is \equiv_{in} -equivalent to some class in this hierarchy.

It is straightforward to show that the above classes form an increasing chain. The hard part is to prove that the chain is strictly increasing and that there are no further classes.

Lemma 5.5. *We have*

$$\emptyset \sqsubseteq_{\text{in}} \mathcal{T}_0 \sqsubseteq_{\text{in}} \mathcal{T}_1 \sqsubseteq_{\text{in}} \cdots \sqsubseteq_{\text{in}} \mathcal{T}_n \sqsubseteq_{\text{in}} \cdots \sqsubseteq_{\text{in}} \mathcal{P} \sqsubseteq_{\text{in}} \mathcal{T}_\omega \sqsubseteq_{\text{in}} \mathcal{G}.$$

Proof. Note that $\mathcal{C} \subseteq \text{Min}(\mathcal{K})$ implies $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{K}$, by Lemma 4.2. All but one of the above claims follow from this observation. It only remains to prove that $\mathcal{T}_n \sqsubseteq_{\text{in}} \mathcal{P}$. We can encode each tree T of height n as a finite word w over the alphabet $[n]$ as follows. Let $\nu_0 <_{\text{lex}} \cdots <_{\text{lex}} \nu_{m-1}$ be an enumeration of T in lexicographic order and let l_i be the level of ν_i . We encode T by the word $w := l_0 \dots l_{m-1}$. Note that we can recover T from w as follows. Each position in w corresponds to a vertex. The predecessor of the i -th vertex ν is the maximal vertex to the left of ν whose label is less than l_i . Clearly, this predecessor relation is definable in monadic second-order logic. \square

Let us collect some easy properties of the hierarchy. Our first result states that \mathcal{G} belongs to the greatest degree in the transduction hierarchy.

Lemma 5.6. $\text{STR}[\Sigma] \sqsubseteq_{\text{in}} \mathcal{G}$

Proof. The $m \times n$ grid is the graph $\mathfrak{G} = \langle V, \text{edg} \rangle$ with vertices $V = [m] \times [n]$ and edge relation

$$\text{edg} = \{ (\langle i, k \rangle, \langle j, l \rangle) \mid |i - j| + |k - l| = 1 \}.$$

Before encoding arbitrary structures in such grids we describe a transduction mapping \mathfrak{G} to its directed variant $\langle V, E_0, E_1 \rangle$ where

$$E_0 := \{ (\langle i, k \rangle, \langle i + 1, k \rangle) \mid i < m - 1, k < n \},$$

and $E_1 := \{ (\langle i, k \rangle, \langle i, k + 1 \rangle) \mid i < m, k < n - 1 \}.$

This can be done with the help of the parameters $P_0, P_1, P_2, Q_0, Q_1, Q_2 \subseteq V$ where

$$P_m := \{ \langle i, k \rangle \mid i \equiv m \pmod{3} \},$$

and $Q_m := \{ \langle i, k \rangle \mid k \equiv m \pmod{3} \}.$

Then

$$E_0 = \{ (u, v) \mid u \in P_i \text{ and } v \in P_j \text{ for some } i \equiv j - 1 \pmod{3} \},$$

and $E_1 = \{ (u, v) \mid u \in Q_i \text{ and } v \in Q_j \text{ for some } i \equiv j - 1 \pmod{3} \}.$

To show that $\text{STR}[\Sigma] \sqsubseteq_{\text{in}} \mathcal{G}$, suppose that $\mathfrak{A} \in \text{STR}[\Sigma]$ is a structure with $\mathfrak{A}_{\text{in}} = \langle A \cup E, (P_R)_R, \text{in}_0, \dots, \text{in}_{r-1} \rangle$. Fix enumerations a_0, \dots, a_{m-1} of A and e_0, \dots, e_{n-1} of E . By the above remarks, it is sufficient to encode \mathfrak{A}_{in} in the directed $m \times n$ grid. Consider the following subsets of $[m] \times [n]$:

$$\begin{aligned} A' &:= [m] \times \{0\}, & P'_R &:= \{ \langle 0, k \rangle \mid e_k \in P_R \}, \\ E' &:= \{0\} \times [n], & I'_l &:= \{ \langle i, k \rangle \mid (a_i, e_k) \in \text{in}_l \}. \end{aligned}$$

Then \mathfrak{A}_{in} can be recovered from \mathfrak{G} by an MSO-transduction using these sets as parameters. \square

Lemma 5.7. $\mathcal{T}_\omega \sqsubseteq_{\text{in}} \mathcal{T}_{\text{bin}}$.

Proof. $\mathcal{T}_{\text{bin}} \subseteq \mathcal{T}_\omega$ implies $\mathcal{T}_{\text{bin}} \sqsubseteq_{\text{in}} \mathcal{T}_\omega$. Conversely, each finite tree can be obtained as minor of a binary tree. Hence, $\mathcal{T}_\omega \subseteq \text{Min}(\mathcal{T}_{\text{bin}}) \sqsubseteq_{\text{in}} \mathcal{T}_{\text{bin}}$. \square

Lemma 5.8. We have $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_1$ if and only if \mathcal{C} is finite and contains at least one nonempty structure.

As indicated in the example on page 9 there exist a transduction mapping an incidence structure \mathfrak{A}_{in} to the incidence structure $\text{Gf}(\mathfrak{A})_{\text{in}}$ of the Gaifman graph.

Lemma 5.9. For every class \mathcal{C} of structures, we have $\text{Gf}(\mathcal{C}) \sqsubseteq_{\text{in}} \mathcal{C}$.

From the results of Section 4 we can deduce the following facts.

Lemma 5.10. For every class \mathcal{C} of graphs we have $\text{Min}(\mathcal{C}) \sqsubseteq_{\text{in}} \mathcal{C}$.

Proof. This follows directly from Lemma 4.2. \square

Theorem 5.11. Let $\mathcal{C} \subseteq \text{STR}[\Sigma]$.

- (a) $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{P}$ iff $\text{pwd}(\mathcal{C}) < \infty$.
- (b) $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_\omega$ iff $\text{twd}(\mathcal{C}) < \infty$.
- (c) $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_n$ iff $\text{twd}_n(\mathcal{C}) < \infty$.

Proof. In each case (\Leftarrow) follows from Lemma 4.5 and (\Rightarrow) follows from Proposition 4.6. \square

Corollary 5.12. Let \mathcal{C} be a class of structures.

(a) $\text{pwd}(\mathcal{C}) = \infty$ implies $\mathcal{T}_\omega \sqsubseteq_{\text{in}} \mathcal{C}$.

(b) $\text{twd}(\mathcal{C}) = \infty$ implies $\mathcal{G} \sqsubseteq_{\text{in}} \mathcal{C}$.

Proof. For (a), Theorem 4.7 implies that $\mathcal{T}_\omega \subseteq \text{Min}(\text{Gf}(\mathcal{C}))$, while, for (b), Theorem 4.8 implies that $\mathcal{G} \subseteq \text{Min}(\text{Gf}(\mathcal{C}))$. Consequently, both claims follow from Lemma 5.10. \square

Corollary 5.13. *Let $\mathcal{C} \subseteq \text{STR}[\Sigma]$.*

(a) $\mathcal{C} \not\sqsubseteq_{\text{in}} \mathcal{P}$ implies $\mathcal{T}_\omega \sqsubseteq_{\text{in}} \mathcal{C}$.

(b) $\mathcal{C} \not\sqsubseteq_{\text{in}} \mathcal{T}_\omega$ implies $\mathcal{G} \sqsubseteq_{\text{in}} \mathcal{C}$.

Corollary 5.14. $\mathcal{P} \sqsubseteq_{\text{in}} \mathcal{T}_\omega \sqsubseteq_{\text{in}} \mathcal{G}$

Proof. Both inclusions follow from Theorem 5.11 and Corollary 5.13.

For the first one, note that we have $\mathcal{P} \sqsubseteq_{\text{in}} \mathcal{T}_\omega$ since $\mathcal{P} \subseteq \text{Min}(\mathcal{T}_\omega)$. Conversely, $\text{pwd}(\mathcal{T}_\omega) = \infty$ implies $\mathcal{T}_\omega \not\sqsubseteq_{\text{in}} \mathcal{P}$. Hence, $\mathcal{P} \subset_{\text{in}} \mathcal{T}_\omega$. Finally, if $\mathcal{C} \subset_{\text{in}} \mathcal{T}_\omega$ then $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{P}$, by Corollary 5.13 (a). Consequently, we have $\mathcal{P} \sqsubseteq_{\text{in}} \mathcal{T}_\omega$.

Similarly, the fact that $\mathcal{T}_\omega \sqsubseteq_{\text{in}} \mathcal{G}$ follows from Lemma 5.6. Since $\text{twd}(\mathcal{G}) = \infty$ Lemma 5.11 implies that $\mathcal{T}_\omega \subset_{\text{in}} \mathcal{G}$. Finally, we obtain $\mathcal{T}_\omega \sqsubseteq_{\text{in}} \mathcal{G}$ by Corollary 5.13 (b). \square

6 THE LOWER PART OF THE HIERARCHY

In this section we finish the proof of Theorem 5.4 by showing that $\mathcal{T}_n \sqsubseteq_{\text{in}} \mathcal{T}_{n+1}$, for all n . First, we prove that $\mathcal{T}_n \not\sqsubseteq_{\text{in}} \mathcal{T}_{n+1}$.

Definition 6.1. Let $\mathfrak{T} = \langle T, \leq \rangle$ be an order tree. Vertices v_0, \dots, v_{m-1} are *horizontally related via w* if all v_i are at the same level of the tree and $v_i \sqcap v_k = w$, for all $i \neq k$.

Lemma 6.2. *Let \mathfrak{T} be a coloured order tree of height n , and suppose that τ is a parameterless k -copying MSO-transduction of rank r such that $\tau(\mathfrak{T})$ is a successor tree of height at most $n + 1$.*

Consider vertices v_0, \dots, v_{m-1} that are horizontally related via w in \mathfrak{T} and fix some number $l < k$. Let x_i be the successor of w with $x_i \leq v_i$. If, for all $i, j < m$, we have

$$\text{MTh}_{r+2n+1}(\mathfrak{T}_{x_i}, v_i) = \text{MTh}_{r+2n+1}(\mathfrak{T}_{x_j}, v_j),$$

then we can remove one vertex from the set $\{\langle v_0, l \rangle, \dots, \langle v_{m-1}, l \rangle\}$ in such a way that the remaining ones are horizontally related in $\tau(\mathfrak{T})$.

Proof. Let $\varphi_{ij}(x, y)$ be the formula defining the successor relation in $\tau(\mathfrak{T})$ between vertices of the form $\langle x, i \rangle$ and $\langle y, j \rangle$.

First, note that a vertex $\langle v, l \rangle$ is on level h in $\tau(\mathfrak{T})$ if and only if there are indices $s_0, \dots, s_{h-1} < k$ such that

$$\mathfrak{T} \models \psi_{s_0 \dots s_{h-1}}(v)$$

where

$$\begin{aligned} \psi_{s_0 \dots s_{h-1}}(v) := \exists x_0 \dots \exists x_{h-1} \left[\bigwedge_{i < h-1} \varphi_{s_i s_{i+1}}(x_i, x_{i+1}) \wedge \varphi_{s_{h-1} l}(x_{h-1}, v) \right. \\ \left. \wedge \neg \exists y \bigvee_{s < k} \varphi_{ss_0}(y, x_0) \right]. \end{aligned}$$

By assumption on v_i and Lemma 3.11, we have

$$\text{MTh}_{r+2n+1}(\mathfrak{T}, v_i) = \text{MTh}_{r+2n+1}(\mathfrak{T}, v_j), \quad \text{for all } i, j.$$

Since the rank of $\psi_{s_0 \dots s_{h-1}}$ is $h + r + 1 \leq r + 2n + 1$ it follows that

$$\mathfrak{T} \models \psi_{s_0 \dots s_{h-1}}(v_i) \quad \text{iff} \quad \mathfrak{T} \models \psi_{s_0 \dots s_{h-1}}(v_j).$$

Hence, all vertices $\langle v_0, l \rangle, \dots, \langle v_{m-1}, l \rangle$ are on the same level h in $\tau(\mathfrak{T})$. We prove by induction on h that

$$(*) \quad \text{MTh}_{r+n+h+1}(\mathfrak{T}_{x_i}, v_i) = \text{MTh}_{r+n+h+1}(\mathfrak{T}_{x_j}, v_j)$$

implies that all but at most one of $\langle v_0, l \rangle, \dots, \langle v_{m-1}, l \rangle$ are horizontally related.

Let $\langle u_i, s_i \rangle$ be the predecessor of $\langle v_i, l \rangle$ in $\tau(\mathfrak{T})$, that is,

$$\mathfrak{T} \models \varphi_{s_i l}(u_i, v_i).$$

We distinguish two cases.

First suppose that $u_0 \sqcap v_0 \leq w$. By $(*)$ and Lemma 3.11, we have

$$\text{MTh}_{r+n+h+1}(\mathfrak{T}, u_0, v_0) = \text{MTh}_{r+n+h+1}(\mathfrak{T}, u_0, v_i),$$

for all i such that $u_o \sqcap v_i \leq w$. Hence,

$$\mathfrak{T} \models \varphi_{s_o l}(u_o, v_o) \quad \text{implies} \quad \mathfrak{T} \models \varphi_{s_o l}(u_o, v_i), \quad \text{for all such } i,$$

and $\langle u_o, s_o \rangle$ is the common predecessor of all the $\langle v_i, l \rangle$, except for possibly one of them. (There might be an index i with $u_o \sqcap v_i \leq w$. For such an index our composition argument does not work since in that case $(*)$ does not imply that the theories of (\mathfrak{T}, u_o, v_o) and (\mathfrak{T}, u_o, v_i) coincide.)

It remains to consider the case that $w < u_o \sqcap v_o$. Setting

$$\eta_{u_i} := \bigwedge \text{MTh}_{r+n+h-1+1}(\mathfrak{T}_{x_i}, u_i)$$

we have

$$\mathfrak{T}_{x_o} \models \exists z[|z| = |u_o| \wedge \eta_{u_o}(z) \wedge \varphi_{s_o l}(z, v_o)].$$

Since the rank of this formula is $r + n + h + 1$ it follows that

$$\mathfrak{T}_{x_i} \models \exists z[|z| = |u_o| \wedge \eta_{u_o}(z) \wedge \varphi_{s_o l}(z, v_i)], \quad \text{for all } i < m.$$

Consequently, we have $|u_i| = |u_o|$, for all i , and u_o, \dots, u_{m-1} are horizontally related via w . By induction hypothesis it follows that all but at most one of the vertices $\langle u_o, s_o \rangle, \dots, \langle u_{m-1}, s_o \rangle$ are horizontally related via some vertex w' . Therefore, the same holds for their successors $\langle v_o, l \rangle, \dots, \langle v_{m-1}, l \rangle$. \square

Definition 6.3. We denote by $B(n, k, c)$ the number of all vertex-coloured trees with height at most n , out-degree at most k , and c colours. More precisely, we define $B(n, k, c)$ as the number of all functions of the form $\lambda : m^{<n} \rightarrow \wp([c])$ with $m \leq k$.

Lemma 6.4. For $n \geq 1$ and $k \geq 2$, we have

$$2^{c k^{n-1}} \leq B(n, k, c) \leq k 2^{2c k^{n-1}}.$$

Proof. For $m \geq 2$, we have

$$m^n \leq m^n + \sum_{i < n} m^i = m^n + \frac{m^n - 1}{m - 1} \leq 2m^n.$$

Since $|[m]^{<n}| = \sum_{i < n} m^i$ it follows that

$$m^{n-1} \leq |[m]^{<n}| \leq 2m^{n-1}.$$

Therefore, we can bound

$$B(n, k, c) = 2^{cn} + \sum_{m=2}^k 2^{c|[m]^{<n}|}$$

from above by

$$B(n, k, c) \leq 2^{cn} + \sum_{m=2}^k 2^{c2m^{n-1}} \leq k2^{2ck^{n-1}}$$

and from below by

$$B(n, k, c) \geq 2^{cn} + \sum_{m=2}^k 2^{cm^{n-1}} \geq 2^{ck^{n-1}}. \quad \square$$

Theorem 6.5. $\mathcal{T}_n \sqsubset_{\text{in}} \mathcal{T}_{n+1}$

Proof. For a contradiction, suppose that there exists an MSO-transduction τ such that $(\mathcal{T}_{n+1})_{\text{in}} \subseteq \tau((\mathcal{T}_n)_{\text{in}})$. Let $\mathcal{T}_n^{\text{ord}}$ be the class of all order trees corresponding to successor trees in \mathcal{T}_n , and let $\mathcal{T}_{n+1}^{\text{col}} := \text{exp}_1(\mathcal{T}_{n+1})$ be the class of all coloured successor trees with one colour whose underlying tree is in \mathcal{T}_{n+1} . As the successor trees in \mathcal{T}_n are sparse we can construct an MSO-transduction σ_{\circ} such that $(\mathcal{T}_n)_{\text{in}} \subseteq \sigma_{\circ}(\mathcal{T}_n)$. Since $\mathcal{T}_n^{\text{ord}} \equiv \mathcal{T}_n$ we can combine τ and σ_{\circ} to a transduction σ such that $\mathcal{T}_{n+1}^{\text{col}} \subseteq \sigma(\mathcal{T}_n^{\text{ord}})$. By Lemma 6.2, it follows that there is some constant d such that every tree $\mathfrak{T} \in \sigma(\mathcal{T}_n^{\text{ord}})$ with out-degree at most k is of the form $\sigma(\mathfrak{T}')$ where $\mathfrak{T}' \in \mathcal{T}_n^{\text{ord}}$ has out-degree at most dk . (The out-degree of an order tree is the out-degree of the corresponding successor tree.) Suppose that σ uses c parameters. There are

$$B(n, dk, c) \leq dk2^{2c(dk)^{n-1}}$$

colourings of trees in $\mathcal{T}_n^{\text{ord}}$ with out-degree at most dk . On the other hand, there are

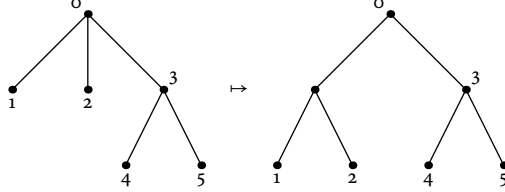
$$B(n+1, k, 1) \geq 2^{k^n}$$

trees in $\mathcal{T}_{n+1}^{\text{col}}$ with out-degree at most k . For large k it follows that

$$B(n, dk, c) \leq dk2^{2cd^{n-1}k^{n-1}} < 2^{k^n} = B(n+1, k, 1).$$

Consequently, there is some tree in $\mathcal{T}_{n+1}^{\text{col}}$ that is not the image of a tree in $\mathcal{T}_n^{\text{ord}}$. A contradiction. \square

To conclude the proof of Theorem 5.4 it remains to show that every degree in the lower part of the hierarchy is equivalent to one of the classes \mathcal{T}_n . We say that the tree $m^{<n+1}$ can be *embedded* into a tree T if there exists an embedding of order trees $\langle m^{<n+1}, \leq \rangle \rightarrow \langle T, \leq \rangle$, i.e., if $\langle m^{<n+1}, \leq \rangle$, regarded as relational structure, is isomorphic to an induced substructure of $\langle T, \leq \rangle$. For instance, we have an embedding



Lemma 6.6. *Let \mathcal{C} be a class with $k := \text{twd}_{n+1}(\mathcal{C}) < \infty$. If there is some number $m \in \mathbb{N}$ such that every structure $\mathfrak{A} \in \mathcal{C}$ has a tree decomposition $D(\mathfrak{A}) := (U_v)_{v \in T}$ of width k and height at most $n + 1$ such that the tree $m^{<n+1}$ cannot be embedded into T , then $\text{twd}_n(\mathcal{C}) < m(k + 1) < \infty$.*

Proof. For each $\mathfrak{A} \in \mathcal{C}$ we construct a tree decomposition $D'(\mathfrak{A})$ of height at most n and width at most mk as follows.

Suppose that $D(\mathfrak{A}) = (U_v)_{v \in T}$ and let $P \subseteq T$ be the minimal (w.r.t. \subseteq) set of vertices that contains

- ◆ every leaf of T with level n and
- ◆ every vertex that has at least m successors in P .

Since $m^{<n+1}$ cannot be embedded into T it follows that P does not contain the root of T . Let F be the set of all edges of T linking a vertex in $T \setminus P$ to a vertex in P . By definition of P it follows that every vertex of T has less than m F -successors and every path of T from the root to some leaf on level n contains exactly one edge from F .

The decomposition $D'(\mathfrak{A}) := D(\mathfrak{A})/F$ obtained by contraction the edges in F has width at most

$$k + 1 + (m - 1)(k + 1) - 1 < m(k + 1).$$

Furthermore, the height of the underlying tree is at most n . □

Lemma 6.7. *Let \mathcal{C} be a class of structures. If, for every number $m \in \mathbb{N}$, there exists a structure $\mathfrak{A} \in \mathcal{C}$ such that we can embed $m^{<n+1}$ into every tree decomposition $(U_v)_{v \in T}$ of \mathfrak{A} of width k , then $\mathcal{T}_{n+1} \sqsubseteq_{\text{in}} \mathcal{C}$.*

Proof. By Lemma 4.11, it follows that, for every $m \in \mathbb{N}$, there is a structure in \mathcal{C} with a strict tree decomposition of width at most k into which we can embed the tree $m^{<n+1}$. According to Theorem 4.12 there is a transduction mapping \mathcal{C} to the class of trees underlying these strict tree decompositions. Hence, there exists a class $\mathcal{K} \sqsubseteq_{\text{in}} \mathcal{C}$ of successor trees containing, for every $m \in \mathbb{N}$, some tree into which we can embed $m^{<n+1}$. Hence, $\mathcal{T}_{n+1} \subseteq \text{Min}(\mathcal{K}) \sqsubseteq_{\text{in}} \mathcal{K} \sqsubseteq_{\text{in}} \mathcal{C}$. \square

Theorem 6.8. *If $\mathcal{T}_{n+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$ then $\text{twd}_n(\mathcal{C}) < \infty$.*

Proof. Note that $\mathcal{G} \not\sqsubseteq_{\text{in}} \mathcal{C}$ since $\mathcal{T}_{n+1} \sqsubseteq_{\text{in}} \mathcal{G}$. Hence, $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_\omega$ and $k := \text{twd}(\mathcal{C}) < \infty$. Similarly, $\mathcal{P} \not\sqsubseteq_{\text{in}} \mathcal{C}$ since $\mathcal{T}_{n+1} \sqsubseteq_{\text{in}} \mathcal{P}$. Hence, there exists a constant l such that no structure $\mathfrak{A} \in \mathcal{C}$ has a tree decomposition of width k and height l . This implies that $\text{twd}_l(\mathcal{C}) = \text{twd}(\mathcal{C}) = k$.

If $l \leq n$ then we are done. Otherwise, we have $\mathcal{T}_l \not\sqsubseteq_{\text{in}} \mathcal{C}$. Hence, Lemma 6.6 and Lemma 6.7 together imply that $\text{twd}_{l-1}(\mathcal{C}) < \infty$. Repeating this argument it follows that $\text{twd}_i(\mathcal{C}) < \infty$, for all $n \leq i \leq l$. \square

Corollary 6.9. *If $\mathcal{T}_{n+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$ then $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_n$.*

Corollary 6.10. $\mathcal{T}_n \sqsubseteq_{\text{in}} \mathcal{T}_{n+1}$.

To conclude the proof of Theorem 5.4 it remains to show that there are no classes between the lower part of the hierarchy and its upper part.

Lemma 6.11. *If $\mathcal{T}_n \sqsubseteq_{\text{in}} \mathcal{C}$, for all $n \in \mathbb{N}$, then $\mathcal{P} \sqsubseteq_{\text{in}} \mathcal{C}$.*

Proof. Suppose that $\mathcal{P} \not\sqsubseteq_{\text{in}} \mathcal{C}$. By Lemma 5.10, it follows that $\mathcal{P} \not\subseteq \text{Min}(\text{Gf}(\mathcal{C}))$. Therefore, we can find a path that is not in $\text{Min}(\text{Gf}(\mathcal{C}))$. By Theorem 4.15, it follows that $\text{twd}_n \mathcal{C} < k$, for some $n, k \in \mathbb{N}$. Hence, we can use Lemma 4.5 to obtain an MSO-transduction τ witnessing that $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_n$. Consequently, we have $\mathcal{T}_{n+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$, as desired. \square

Corollary 6.12. *If $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{P}$ then there is some $n \in \mathbb{N}$ such that $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_n$.*

Proof. By Lemma 6.11, there is some $n \in \mathbb{N}$ such that $\mathcal{T}_{n+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$. Hence, Corollary 6.9 implies that $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_n$. \square

7 PROSPECTS AND CONCLUSION

Above we have obtained a complete description of the transduction hierarchy for classes of finite incidence structures. The most surprising result is that the hierarchy is linear. At this point there are at least three natural directions in which to proceed.

- (i) We can study the hierarchy for classes of structures, instead of their incidence structures.
- (ii) We can consider the hierarchy for classes of infinite structures.
- (iii) We can replace MSO by a different logic.

An answer to (ii) seems within reach, at least if we restrict our attention to countable structures. Although the resulting hierarchy is no longer linear we can adapt most of our techniques to this setting.

Concerning question (iii), let us remark that all results above go through if we use CMSO instead of MSO. We only need the right definition of rank for CMSO. In the proof of Theorem 6.5 we needed the fact that there are only finitely many theories of bounded rank. We can ensure this for CMSO by defining the rank as the least number n such that

- ◆ the nesting depth of quantifiers is at most n and
- ◆ in every cardinality predicate $|X| \equiv k \pmod{m}$ we have $m \leq n$.

One can check that, with this definition of rank, the proof of Theorem 3.10 also goes through for CMSO.

For logics much weaker than MSO, on the other hand, it seems to be unrealistic to hope for a complete description of the corresponding transduction hierarchy.

Finally, let us address question (i). When using transductions between the original structures instead of their incidence structures we can transfer some of the above results to the corresponding hierarchy. But we presently have no complete description since we miss some of the corresponding excluded minor results.

Lemma 7.1. *Let $\mathcal{C}, \mathcal{S} \subseteq \text{STR}$ and suppose that \mathcal{S} is k -sparse.*

- (a) $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{S}$ implies $\mathcal{C} \sqsubseteq \mathcal{S}$.
- (b) $\mathcal{S} \sqsubseteq \mathcal{C}$ implies $\mathcal{S} \sqsubseteq_{\text{in}} \mathcal{C}$.

Proof. There is a transduction ρ such that $\mathcal{C} = \rho[\mathcal{C}_{\text{in}}]$. Since \mathcal{S} is sparse we can also find a transduction σ such that $\mathcal{S}_{\text{in}} = \sigma[\mathcal{S}]$. Consequently,

$$\begin{aligned} \mathcal{C}_{\text{in}} \subseteq \tau[\mathcal{S}_{\text{in}}] \quad &\text{implies} \quad \mathcal{C} \subseteq (\rho \circ \tau \circ \sigma)[\mathcal{S}], \\ \text{and} \quad \mathcal{S} \subseteq \tau[\mathcal{C}] \quad &\text{implies} \quad \mathcal{S}_{\text{in}} \subseteq (\sigma \circ \tau \circ \rho)[\mathcal{C}_{\text{in}}]. \end{aligned} \quad \square$$

Theorem 7.2. *We have the following hierarchy:*

$$\emptyset \sqsubset \mathcal{T}_0 \sqsubset \mathcal{T}_1 \sqsubset \dots \sqsubset \mathcal{T}_n \dots \sqsubset \mathcal{P} \sqsubset \mathcal{T}_\omega \sqsubset \mathcal{G} \equiv \text{STR}[\Sigma]$$

Proof. Note that all classes in the above hierarchy are sparse. For sparse classes \mathcal{C} and \mathcal{K} , Lemma 7.1 implies that

$$\mathcal{C} \subseteq \mathcal{K} \quad \text{iff} \quad \mathcal{C} \subseteq_{\text{in}} \mathcal{K}.$$

Consequently, the claim follows from Theorem 5.4. \square

Open Problem. *Is there any class $\mathcal{C} \subseteq \text{STR}[\Sigma]$ which is not \equiv -equivalent to some class in the above hierarchy?*

Remark. If we only consider classes of graphs and we use CMSO-transductions instead of MSO-transductions then we can replace Theorem 4.8 by the following result:

Theorem 7.3 (Courcelle, Oum [9]). *Let \mathcal{C} be a class of graphs with $\text{cwd}\mathcal{C} = \infty$. There exists a CMSO-transduction τ with $\mathcal{G} \subseteq \tau[\mathcal{C}]$.*

This eliminates some possibilities for intermediate classes of graphs in the hierarchy, but we still need analogues of Theorems 4.7, 4.15, and 4.12 to complete the picture. Furthermore, the techniques of [9] are specific to graphs (or, more generally, to relational structures where all relations are binary). Even with these results one cannot exclude the existence of a class \mathcal{C} strictly between \mathcal{T}_ω and \mathcal{G} in the CMSO-transduction hierarchy.

Let us make a final comment about relational structures. An incidence structure \mathfrak{A}_{in} can be seen as a bipartite labelled directed graph (see the remark after Definition 2.1). Furthermore, it is k -sparse where k is the maximal arity of a relation of \mathfrak{A} . Hence, our results use tools from graph theory, in particular those of [16, 17, 9]. However, there is no encoding of relational structures as labelled graphs that could help to solve question (i) above.

REFERENCES

- [1] A. BLUMENSATH, *Guarded Second-Order Logic and Finite-to-One Maps*. submitted.
- [2] ———, *Structures of Bounded Partition Width*, Ph. D. Thesis, RWTH Aachen, Aachen, 2003.
- [3] ———, *A Model Theoretic Characterisation of Clique-Width*, *Annals of Pure and Applied Logic*, 142 (2006), pp. 321–350.
- [4] A. BLUMENSATH, T. COLCOMBET, AND C. LÖDING, *Logical Theories and Compatible Operations*, in *Logic and Automata: History and Perspectives*, J. Flum, E. Grädel, and T. Wilke, eds., Amsterdam University Press, 2007, pp. 73–106.
- [5] H. L. BODLAENDER, *A linear time algorithm for finding tree-decompositions of small treewidth*, *SIAM Journal of Computing*, 25 (1996), pp. 1305–1317.
- [6] B. COURCELLE, *The monadic second-order logic of graphs V: On closing the gap between definability and recognizability*, *Theoretical Computer Science*, 80 (1991), pp. 153–202.
- [7] ———, *The monadic second-order logic of graphs VIII: Orientations*, *Annals of Pure and Applied Logic*, 72 (1995), pp. 103–143.
- [8] ———, *The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications*, *Theoretical Computer Science*, 299 (2003), pp. 1–36.
- [9] B. COURCELLE AND S.-I. OUM, *Vertex-Minors, Monadic Second-Order Logic, and a Conjecture by Seese*, *Journal Combinatorial Theory B*, 97 (2007), pp. 91–126.
- [10] J. FLUM AND M. GROHE, *Parametrized Complexity Theory*, Springer Verlag, 2006.
- [11] E. GRÄDEL, C. HIRSCH, AND M. OTTO, *Back and Forth Between Guarded and Modal Logics*, *ACM Transactions on Computational Logics*, (2002), pp. 418–463.
- [12] D. LAPOIRE, *Recognizability Equals Monadic Second-Order Definability for Sets of Graphs of Bounded Tree-Width*, in *Proc. 15th Annual Symp. on Theoretical Aspects of Computer Science*, STACS, LNCS, 1373, 1998, pp. 618–628.
- [13] J. NEŠETŘIL AND P. O. DE MENDEZ, *Linear time low tree-width partitions and algorithmic consequences*, in *Proc. 38th Annual ACM Symposium on Theory of Computing*, STOC, 2006, pp. 391–400.
- [14] ———, *Tree-depth, subgraph coloring and homomorphism bounds*, *European Journal of Combinatorics*, 27 (2006), pp. 1022–1041.
- [15] M. O. RABIN, *Decidability of second-order theories and automata on infinite trees*, *Trans. Amer. Math. Soc.*, 141 (1969), pp. 1–35.
- [16] N. ROBERTSON AND P. D. SEYMOUR, *Graph Minors. I. Excluding a Forest*, *Journal of Combinatorial Theory B*, 35 (1983), pp. 39–61.

- [17] ———, *Graph Minors. V. Excluding a Planar Graph*, *Journal of Combinatorial Theory B*, 41 (1986), pp. 92–114.
- [18] D. SEESE, *The structure of the models of decidable monadic theories of graphs*, *Annals of Pure and Applied Logic*, 53 (1991), pp. 169–195.
- [19] S. SHELAH, *The Monadic Second Order Theory of Order*, *Annals of Mathematics*, 102 (1975), pp. 379–419.
- [20] W. THOMAS, *Ehrenfeucht Games, the Composition Method, and the Monadic Theory of Ordinal Words*, *LNCS*, 1261 (1997), pp. 118–143.