

## ON THE MONADIC SECOND-ORDER TRANSDUCTION HIERARCHY

ACHIM BLUMENSATH<sup>a</sup> AND BRUNO COURCELLE<sup>b</sup>

<sup>a</sup> TU Darmstadt, Germany  
*e-mail address:* blumensath@mathematik.tu-darmstadt.de

<sup>b</sup> Institut Universitaire de France and Bordeaux University, LaBRI, France  
*e-mail address:* courcell@labri.fr

---

**ABSTRACT.** We compare classes of finite relational structures via monadic second-order transductions. More precisely, we study the preorder where we set  $C \sqsubseteq K$  if, and only if, there exists a transduction  $\tau$  such that  $C \subseteq \tau(K)$ . 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 all trees of height  $n$ , for each  $n \in \mathbb{N}$ , of all paths, of all trees, and of all grids.

### 1. INTRODUCTION

*Monadic second-order logic* (MSO) is one of the most expressive logics for which the theories of many interesting classes of structures are still decidable. In particular, the infinite binary tree and many linear orders have a decidable MSO-theory [Rab69, She75] and the same holds for many classes of (finite or infinite) structures with bounded *tree-width* [BCL07, RS86]. Furthermore, for every fixed MSO-sentence  $\varphi$  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  $\varphi$  [Bod96, FG06]. Examples of monadic second-order expressible graph properties are  $k$ -colourability, various types of connectivity, and planarity (via Kuratowski's well-known characterisation by forbidden configurations).

A variant of monadic second-order logic called *guarded second-order logic* (GSO) allows quantification not only over sets of elements but also over sets of edges (i.e., tuples from the relations) [GHO02]. 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 [See91, Cou95]. The proof of this result uses a deep theorem of graph minor theory by Robertson and Seymour:

---

*1998 ACM Subject Classification:* G.2.2, F.4.1.

*Key words and phrases:* Monadic Second-Order Logic, Guarded Second-Order Logic, Transductions, Hypergraphs.

<sup>b</sup> Supported by the GRAAL project of the 'Agence Nationale pour la Recherche'.

a set of graphs has bounded tree-width if and only if it excludes some planar graph as a minor [RS86].

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, using a detour via incidence structures, also for guarded second-order logic [BCL07, Cou91, Cou95, Cou03].

In the present article we classify classes of finite structures according to their ‘combinatorial complexity’. (Note that we do not consider decidability issues.) We will consider two ways to measure the complexity of such classes. On the one hand, we can use their tree-width and its variants. On the other hand, we can compare them via transductions. 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 briefly mentioned in Section 9.

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 (see Definition 3.4):

- (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 disjoint copies of  $\mathfrak{A}$ , augmented with appropriate relations between the copies;
- (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  $\tau$  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  $\mathfrak{B} = \tau(\mathfrak{A}, P_1, \dots, P_n)$  we can consider the tuple  $P_1, \dots, P_n$  as an *encoding* of  $\mathfrak{B}$  in  $\mathfrak{A}$ . The transduction  $\tau$  is the corresponding *decoding function*.

Each transduction  $\tau$  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  $\tau$  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  $\tau$ ). 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. Since, at the moment, a complete description of this hierarchy seems to be out of reach, we concentrate on a variant where we replace monadic second-order logic by *guarded second-order logic*. In this case, 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  $\text{edg}$  is a binary relation containing all pairs of adjacent vertices. But we also can use the *incidence representation* of  $\mathfrak{G}$ . 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  $\mathfrak{G}$ , and  $\text{in}$  is the incidence relation between vertices and edges. In a similar way, we can associate with every relational structure  $\mathfrak{A}$  its incidence structure  $\mathfrak{A}_{\text{in}}$  (see Definition 2.1) where the domain also contains elements for all tuples in some relation of  $\mathfrak{A}$ .

It is shown in [GHO02] that every GSO-formula  $\varphi$  talking about some structure  $\mathfrak{A}$  can be translated into an MSO-formula talking about the incidence structure  $\mathfrak{A}_{\text{in}}$ , and vice versa. Hence, we can use incidence structures to obtain an analogue  $\sqsubseteq_{\text{in}}$  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  $\sqsubseteq_{\text{in}}$  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  $n$ , for each  $n \in \mathbb{N}$ ;
- 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 complexity measures on structures that are compatible with MSO-transductions transforming incidence structures.

The upper levels of the hierarchy can be determined easily using techniques from graph minor theory developed by Robertson and Seymour, such as the notions of a minor and a tree decomposition. In particular, we employ two results characterising bounded tree-width and bounded path-width in terms of excluded minors [RS83, RS86].

For the lower levels, which consist of classes of bounded path-width, the characterisation is more complicated and requires new results relating tree decompositions and monadic second-order logic.

In Sections 2 and 3 we give basic definitions. Section 4 collects some known results from graph minor theory. We also introduce a new variant of tree-width and prove some of its basic properties. In Section 5 we expound the connections between tree-width and monadic second-order transductions. In Section 6 we introduce the transduction hierarchy and we state our main theorem. Its proof is contained in Sections 7 and 8. In the first one, we prove that the hierarchy is strict while, in the second one, we show that it covers every class. The final Section 9 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  $\mathcal{P}(X)$  for the power set of a set  $X$ . We denote tuples  $\bar{a}$  with a bar. The components of  $\bar{a}$  will be  $a_0, \dots, a_{n-1}$  where the length  $n$  will usually be implicit. We sometimes identify a tuple  $\bar{a}$  with the set of its components. For instance, we write  $c \in \bar{a}$  to express that  $c = a_i$ , for some  $i$ .

In this article all graphs, trees, and relational structures are finite. We will not repeat this finiteness assumption. A relational structure  $\mathfrak{A}$  is of the form  $\langle A, R_0^{\mathfrak{A}}, \dots, R_{m-1}^{\mathfrak{A}} \rangle$  with domain  $A$  and relations  $R_i^{\mathfrak{A}}$ . The *signature* of such a structure is the set  $\Sigma = \{R_0, \dots, R_{m-1}\}$  of relation symbols. In some proofs we will also use signatures with constant symbols denoting elements of the domain. We write  $\text{ar}(R)$  for the arity of a relation  $R$ . For a signature  $\Sigma$ , we denote by  $\text{STR}[\Sigma]$  the class of all  $\Sigma$ -structures. We write  $\mathfrak{A} \oplus \mathfrak{B}$  for the *disjoint union* of the structures  $\mathfrak{A}$  and  $\mathfrak{B}$ .

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  $A$  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 2.2.** 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 2.3.** Let  $k \in \mathbb{N}$ . A structure  $\mathfrak{A} = \langle A, \bar{R} \rangle$  is *k-sparse*<sup>1</sup> 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 2.4.** *Every incidence structure is 1-sparse.*

Let us fix our notation regarding trees and graphs.

**Definition 2.5.** A *directed graph* is a pair  $\langle V, \text{edge} \rangle$  where  $V$  is the set of *vertices* and  $\text{edge} \subseteq V \times V$  is the *edge relation*. Thus, graphs are by definition simple (without parallel edges). An *undirected graph* is a graph where the edge relation  $\text{edge}$  is symmetric. When speaking of a *graph* we will always mean an undirected one.

We regard a *coloured graph* as a relational structure  $\langle V, E_0, \dots, E_k, P_0, \dots, P_m \rangle$  with binary relations  $E_i$  and unary relations  $P_i$  that encode the colours of, respectively, the edges and the vertices. We allow graphs whose edges and vertices have several colours.

<sup>1</sup>[Cou03] introduced two notions of sparsity for hypergraphs: *k-sparse hypergraphs* and *uniformly k-sparse hypergraphs*. What we call *k-sparse* is a slight modification of the uniform version of [Cou03].

Trees play a major role in this article. Intuitively, a tree is a directed graph  $\mathfrak{T}$  with a unique vertex  $r$  of indegree 0, called the *root* of  $\mathfrak{T}$ , such that every vertex is reachable from  $r$  by a unique directed path. The actual definition we will use is slightly more concrete. It is based on the usual encoding of the vertices of a tree by finite sequences describing the path from the root to the given vertex. In fact, we introduce two notions of a tree: *order-trees* and *successor-trees*. The latter use the usual edge relation, while the former are equipped with the tree-order instead.

**Definition 2.6.** Let  $D$  be a set.

(a) For an ordinal  $\alpha$ , we denote by  $D^{<\alpha}$  the set of all sequences of elements of  $D$  of length less than  $\alpha$ . The *prefix relation* on  $D^{<\omega}$  is defined by

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

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

(b) A finite prefix closed subset  $T \subseteq D^{<\omega}$  is called a *tree domain*. Following our intuition that a vertex is represented by the path leading to it, we call the empty sequence  $\langle \rangle$  the *root* of  $T$  and the maximal elements of  $T$  its *leaves*. The domain of the *complete  $m$ -ary tree of height  $n$*  is  $m^{<n}$ . Hence,  $m^{<0}$  is the empty tree,  $m^{<1}$  the one consisting only of the root, and  $m^{<2}$  consists of a root and  $m$  leaves.

Given a tree domain  $T$  we can define the *successor relation*  $\text{edg}$  on  $T$  by setting

$$\langle x, y \rangle \in \text{edg} \quad : \text{iff} \quad y = xd \text{ for some } d \in D.$$

In this case we call  $y$  a *successor* of  $x$  and  $x$  the *predecessor* of  $y$ . A structure of the form  $\langle T, \text{edg} \rangle$  (and every structure isomorphic to it) is called a *successor-tree*.

Sometimes it is convenient to replace the successor relation  $\text{edg}$  by the tree order  $\preceq$ . Structures of the form  $\langle T, \preceq \rangle$  are called *order-trees*. A *coloured tree* is the expansion of a (order- or successor-) tree by unary predicates  $\bar{P}$ . (We do not require these predicates to be pairwise disjoint. Hence, every vertex may have none, one, or several colours.) We write  $\text{TREE}_m$  for the class of all order-trees  $\langle T, \preceq, P_0, \dots, P_{m-1} \rangle$  with  $m$  colours. The set of *leaves* of a tree  $\mathfrak{T}$  is denoted by  $\text{Lf}(\mathfrak{T})$ .

(c) Let  $\mathfrak{T} = \langle T, \preceq \rangle$  be an order-tree. The *level* of an element  $v \in T$  is the number of vertices  $u \in T$  with  $u \prec v$ . We denote it by  $|v|$ . The *height* of  $\mathfrak{T}$  is the least ordinal  $\alpha$  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 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  $v \preceq u$ , i.e., all vertices below  $v$ .

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

**Definition 2.7.** 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

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

### 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 [GHO02] for details). Hence, guarded second-order logic over a given structure  $\mathfrak{A}$  is equivalent to monadic second-order logic over its incidence structure  $\mathfrak{A}_{\text{in}}$ .

**Lemma 3.1** ([GHO02]).

(a) For every GSO-sentence  $\varphi$ , we can effectively construct an MSO-sentence  $\psi$  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  $\varphi$ , we can effectively construct a GSO-sentence  $\psi$  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 the treatment of 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  $|X| \equiv k \pmod{m}$  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 9 we will give a partial characterisation of the hierarchy for CMSO-transductions of graphs (not of incidence graphs). In this case the availability of counting predicates does make a difference.

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. Similarly, when we use structures with constants we actually regard each constant as a singleton set.

**Definition 3.2.** (a) The *rank*  $\text{qr}(\varphi)$  of a formula  $\varphi$  is the nesting depth of quantifiers in  $\varphi$ . 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 3.3.** We use the term ‘rank’ instead of the more natural ‘quantifier rank’ since in Section 9 below we will consider CMSO where the notion of rank has to be adapted 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 compositions of these.

**Definition 3.4.** (a) Let  $k \geq 2$  be a natural number. The operation  $\text{copy}_k$  maps a structure  $\mathfrak{A}$  to the expansion

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

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 of  $\mathfrak{A} \oplus \cdots \oplus \mathfrak{A}$  by the pair  $\langle a, i \rangle$ , we define

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

For  $k = 1$ , we set  $\text{copy}_1(\mathfrak{A}) := \mathfrak{A}$ .

(b) For  $m \in \mathbb{N}$ , we define the operation  $\text{exp}_m$  that maps a structure  $\mathfrak{A}$  to all possible expansions by  $m$  unary predicates  $Q_0, \dots, Q_{m-1} \subseteq A$ . Note that this operation is many-valued and that  $\text{exp}_0$  is just the identity.

(c) A *basic MSO-transduction* is a partial operation  $\tau$  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  $\tau$ . Given a structure  $\mathfrak{A}$  that satisfies the formula  $\chi$  the operation  $\tau$  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*  $\tau$  is a (many-valued) operation on relational structures of the form  $\tau_0 \circ \text{copy}_k \circ \text{exp}_m$  where  $\tau_0$  is a basic MSO-transduction. When the value of  $k$  does not matter, we will simply speak of a *transduction*.

Note that, due to  $\text{exp}_m$ , a structure can be mapped to several structures by a transduction. Consequently, we consider  $\tau(\mathfrak{A})$  as the *set* of possible values  $(\tau_0 \circ \text{copy}_k)(\mathfrak{A}, \bar{P})$  where  $\bar{P}$  ranges over all  $m$ -tuples of subsets of  $A$ .

For a class  $\mathcal{C}$ , we set

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

**Remark 3.5.** (a) The expansion by  $m$  unary predicates corresponds, in the terminology of [Cou95, Cou03], to using  $m$  *parameters*. We will use this terminology, for instance, in the proof of Theorem 5.5.

(b) Note that every basic MSO-transduction is a 1-copying MSO-transduction without parameters.

**Example 3.6.** (a) Let  $\Sigma$  be a signature and let  $r$  be the maximal arity of a relation in  $\Sigma$ . 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  $k = r(r-1)/2$ , for  $r \geq 2$ , and  $k = 1$ , for  $r \leq 1$ .

(b) For every fixed number  $n \in \mathbb{N}$ , we describe a transduction  $\tau$  transforming a path  $\mathfrak{P}$  of length  $l$  into the class of all trees of height  $n$  with  $l+1$  vertices.

We can encode a tree  $T$  of height  $n$  with  $m$  vertices as a finite word  $w$  of length  $m$  over the alphabet  $[n]$  as follows. Let  $v_0 <_{\text{lex}} \dots <_{\text{lex}} v_{m-1}$  be the enumeration of the vertices of  $T$  in lexicographic order, and let  $l_i$  be the level of  $v_i$ . We encode  $T$  by the word  $w := l_0 \dots l_{m-1}$ . A transduction can recover  $T$  from  $w$  as follows. Each position in  $w$  corresponds to a vertex. The predecessor of the  $i$ -th vertex  $v$  is the maximal vertex to the left of  $v$  whose label is less than  $l_i$ . Clearly, this predecessor relation is definable in monadic second-order logic.

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

**Lemma 3.7.** *Let  $\tau$  be a transduction. For every MSO-sentence  $\varphi$ , there exists an MSO-sentence  $\varphi^\tau$  such that, for all structures  $\mathfrak{A}$ ,*

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

*Furthermore, if  $\tau$  is quantifier-free, then the rank of  $\varphi^\tau$  is no larger than that of  $\varphi$ .*

**Corollary 3.8.** *For every quantifier-free transduction  $\tau$  and every  $m \in \mathbb{N}$ , there exists a function  $f_\tau$  on monadic theories of rank  $m$  such that*

$$\text{MTh}_m(\tau(\mathfrak{A})) = f_\tau(\text{MTh}_m(\mathfrak{A})), \quad \text{for all structures } \mathfrak{A}.$$

**Lemma 3.9** ([Cou91]). *For all transductions  $\sigma, \tau$  there exists a transduction  $\varrho$  such that  $\varrho = \sigma \circ \tau$ .*

As a further example note that we can use transductions to translate between order-trees and successor-trees.

**Lemma 3.10.** (a) *There exists a transduction  $\tau$  mapping an order-tree to the corresponding successor-tree.*

(b) *There exists a transduction  $\sigma$  mapping a successor-tree to the corresponding order-tree.*

Similarly there are transductions translating between a structure and its incidence structure.

**Lemma 3.11.** *For every signature  $\Sigma$ , there exists a transduction  $\tau$  such that  $\tau(\mathfrak{A}_{\text{in}}) = \mathfrak{A}$ , for all  $\mathfrak{A} \in \text{STR}[\Sigma]$ .*

The converse statement is a much deeper result and requires the structure in question to be  $k$ -sparse for some fixed  $k$ .

**Theorem 3.12** ([Cou03, Blu10]). *For every signature  $\Sigma$  and all numbers  $k \in \mathbb{N}$ , there exists an MSO-transduction  $\tau$  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 3.7 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 disjoint union operation can frequently be used for this purpose (for a proof of the following theorem see, e.g., Theorem 7.11 of [Lib04], or [Cou87]).

**Theorem 3.13.** *Let  $\Sigma$  and  $\Gamma$  be relational signatures with constants. For every  $m \in \mathbb{N}$ , there exists a (computable) binary operation  $\oplus_m$  on monadic theories of rank  $m$  such that*

$$\text{MTh}_m(\mathfrak{A} \oplus \mathfrak{B}) = \text{MTh}_m(\mathfrak{A}) \oplus_m \text{MTh}_m(\mathfrak{B}),$$

*for all  $\Sigma$ -structures  $\mathfrak{A}$  and  $\Gamma$ -structures  $\mathfrak{B}$ .*



Below we will mainly make use of the following corollary.

**Lemma 3.14.** *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  $\mathfrak{S}$ . Let  $\bar{c}$  be a tuple of vertices of  $\mathfrak{T}$  with  $v \not\preceq c_i$ , for all  $i$ . If  $\bar{a}$  are vertices of  $\mathfrak{T}_v$  and  $\bar{b}$  are vertices of  $\mathfrak{S}$  such that*

$$\text{MTh}_m(\mathfrak{T}_v, \bar{a}) = \text{MTh}_m(\mathfrak{S}, \bar{b})$$

*then it follows that*

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

**Proof.** Let  $\mathfrak{C}$  be the tree obtained from  $\mathfrak{T}$  by replacing the subtree  $\mathfrak{T}_v$  by a single vertex  $w$ . We define the following auxiliary predicates:

$$P := \{w\}, \quad Q_0 := \{x \in C \mid x \prec w\}, \quad Q_1 := T_v, \quad \text{and} \quad Q'_1 := S.$$

We construct a quantifier-free transduction  $\tau$  such that

$$\begin{aligned} \tau(\langle \mathfrak{C}, P, Q_0, \bar{c} \rangle \oplus \langle \mathfrak{T}_v, Q_1, \bar{a} \rangle) &= \langle \mathfrak{T}, \bar{a}\bar{c} \rangle, \\ \tau(\langle \mathfrak{C}, P, Q_0, \bar{c} \rangle \oplus \langle \mathfrak{S}, Q'_1, \bar{b} \rangle) &= \langle \mathfrak{T}', \bar{b}\bar{c} \rangle. \end{aligned}$$

If  $f_\tau$  is the function from Corollary 3.8 and  $\oplus_m$  the operation from Theorem 3.13, it follows that

$$\begin{aligned} \text{MTh}_m(\mathfrak{T}', \bar{b}\bar{c}) &= f_\tau(\text{MTh}_m(\mathfrak{C}, P, Q_0, \bar{c}) \oplus_m \text{MTh}_m(\mathfrak{S}, Q'_1, \bar{b})) \\ &= f_\tau(\text{MTh}_m(\mathfrak{C}, P, Q_0, \bar{c}) \oplus_m \text{MTh}_m(\mathfrak{T}_v, Q_1, \bar{a})) = \text{MTh}_m(\mathfrak{T}, \bar{a}\bar{c}), \end{aligned}$$

as desired.

Hence, it remains to define  $\tau$ . Let  $\{\preceq, P, Q_0, \bar{d}\}$  be the signature of  $\langle \mathfrak{C}, P, Q_0, \bar{c} \rangle$ , and  $\{\preceq, Q_1, \bar{e}\}$  the signature of  $\langle \mathfrak{T}_v, Q_1, \bar{a} \rangle$  and  $\langle \mathfrak{S}, Q'_1, \bar{b} \rangle$ . For  $\tau$  we can use the basic MSO-transduction consisting of the following formulae:

$$\begin{aligned} \chi &:= \text{true}, \\ \delta(x) &:= \neg Px, \\ \varphi_{\preceq}(x, y) &:= x \preceq y \vee (Q_0x \wedge Q_1y). \end{aligned}$$

□

#### 4. MINORS AND TREE DECOMPOSITIONS

Some properties of the transduction hierarchy, which we will introduce in Section 6 below, can be deduced from results about graph minors.

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

$$\mathfrak{G}/E := \langle W, \text{edg}_0 \rangle,$$

where  $W := V/E^*$  is the set of equivalence classes and  $\text{edg}_0$  contains an edge between classes  $[x]$  and  $[y]$  if and only if  $[x] \neq [y]$  and there are representatives  $u \in [x]$  and  $v \in [y]$  with  $\langle u, v \rangle \in \text{edg}$ .

(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  $\mathcal{C}$ .

One central tool in graph minor theory is the notion of a tree decomposition and the related complexity measures called tree-width and path-width. These notions extend in a natural way to relational structures.

**Definition 4.2.** 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 a rooted 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 the sets  $U_v$  the *components* of the decomposition, and  $T$  is its *underlying tree*.

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

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

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

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

(e) For a class  $\mathcal{C}$  of structures, we define  $\text{tw}(\mathcal{C})$  as the supremum of  $\text{tw}(\mathfrak{A})$ , for  $\mathfrak{A} \in \mathcal{C}$ , and similarly for  $\text{pw}(\mathcal{C})$  and  $\text{tw}_n(\mathcal{C})$ .

**Remark 4.3.** (a) The  $n$ -depth tree-width of a graph  $\mathfrak{G}$  is related to its *tree-depth*  $\text{td}(\mathfrak{G})$  as introduced by Nešetřil and Ossona de Mendez [NdM06a, NdM06b]. The tree-depth of a graph  $\mathfrak{G}$  is the least number  $n$  such that some orientation of  $\mathfrak{G}$  is a subgraph of some order-tree of height  $n$ . For a graph  $\mathfrak{G}$ , it follows that

- $\text{td}(\mathfrak{G}) \leq n$  implies  $\text{tw}_n(\mathfrak{G}) < n$ ;
- $\text{tw}_n(\mathfrak{G}) < k$  implies  $\text{td}(\mathfrak{G}) \leq nk$ .

These facts are easy to establish. We will not need them in the following.

(b) There are some simple relations between  $n$ -depth tree-width, path-width, and tree-width. For every graph  $\mathfrak{G}$ , we have

$$\text{tw}(\mathfrak{G}) \leq \text{tw}_{n+1}(\mathfrak{G}) \leq \text{tw}_n(\mathfrak{G}), \quad \text{for every } n \in \mathbb{N},$$

$$\text{and} \quad \text{tw}(\mathfrak{G}) = \text{tw}_n(\mathfrak{G}), \quad \text{for all sufficiently large } n \in \mathbb{N}.$$

Furthermore,

$$\text{pw}(\mathfrak{G}) < n(\text{tw}_n(\mathfrak{G}) + 1), \quad \text{for every } n \in \mathbb{N}.$$

(Let us sketch the proof of the last inequality: let  $(U_v)_{v \in T}$  be a tree decomposition of  $\mathfrak{G}$  of height  $n$  and width  $\text{tw}_n(\mathfrak{G})$ . As the components of a path decomposition of  $\mathfrak{G}$  we take all sets of the form  $U_{v_0} \cup \dots \cup U_{v_k}$ , where  $v_0 \dots v_k$  is a path from the root  $v_0$  to some leaf  $v_k$  of  $T$ .)

The next lemma shows that most questions regarding tree decompositions of a structure can be reduced to the corresponding questions about its Gaifman graph. 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  $U_v$  containing the whole clique. This implies that, for every clique  $C$  in  $\text{Gf}(\mathfrak{A})$  induced by some tuple  $\bar{c} \in R_i$ , there is some vertex  $v \in T$  with  $C \subseteq U_v$ . Hence, every tuple  $\bar{c} \in R_i$  is contained in some component  $U_v$ .  $\square$

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

**Theorem 4.5** (Excluded Tree Theorem [RS83]). *For each tree  $\mathfrak{T}$ , there exists a number  $k \in \mathbb{N}$  such that*

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

**Theorem 4.6** (Excluded Grid Theorem [RS86]). *For each planar graph  $\mathfrak{E}$ , there exists a number  $k \in \mathbb{N}$  such that*

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

**Corollary 4.7.** (a) *A class of graphs has bounded path-width if and only if it excludes some tree as a minor.*

(b) *A class of graphs has bounded tree-width if and only if it excludes some planar graph as a minor.*

We also need a variant of these theorems for  $n$ -depth tree-width. The next lemma contains the main technical argument.

**Lemma 4.8.** *Suppose that  $\mathfrak{G}$  is a graph that does not contain a path of length  $l$ . Then  $\mathfrak{G}$  has a tree decomposition of height at most  $l$  and width at most  $l - 1$ .*

**Proof.** Let  $\langle T, \preceq \rangle$  be a depth-first spanning order-tree of  $\mathfrak{G}$ , i.e., a spanning tree such that, for every edge  $(u, v)$  of  $\mathfrak{G}$  we have  $u \preceq v$  or  $v \preceq u$  (for details see, e.g., [Die06] where such spanning trees are called *normal*). We define a tree decomposition  $(U_v)_{v \in T}$  of  $\mathfrak{G}$  by setting

$$U_v := \{u \in T \mid u \preceq v\}.$$

Since  $T$  is depth-first, it follows that every edge  $(u, v)$  of  $\mathfrak{G}$  is contained in some component  $U_w$  where  $w$  is the maximum of  $u$  and  $v$ .

The height of the tree  $T$  can be at most  $l$  since  $\mathfrak{G}$  contains no path of length  $l$ . Furthermore, we have  $|U_v| = |v| + 1 \leq l$ . Hence, the width of the tree decomposition is at most  $l - 1$ .  $\square$

**Theorem 4.9** (Excluded Path Theorem). *For each path  $\mathfrak{P}$ , there exist numbers  $n, k \in \mathbb{N}$  such that*

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

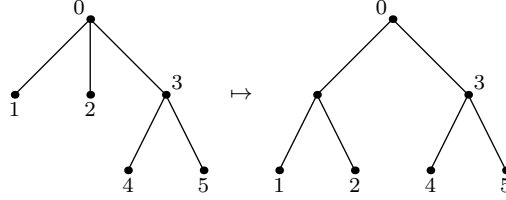
**Proof.** Suppose that  $\mathfrak{P} \notin \text{Min}(\mathfrak{G})$  and let  $l$  be the length of  $\mathfrak{P}$ . Then the preceding lemma implies that  $\text{twd}_l(\mathfrak{G}) < l$ .  $\square$

**Corollary 4.10.** (a) *A class of graphs has bounded  $n$ -depth tree-width, for some  $n$ , if and only if it excludes some path as a minor (equivalently, as a subgraph).*

(b) *A class of graphs has bounded tree-depth if and only if it excludes some path as a minor (equivalently, as a subgraph).*

We can also compute a bound on the  $n$ -depth tree-width in terms of the  $(n + 1)$ -depth tree-width. It will be needed in the proof of Theorem 8.2 below.

We say that the tree  $\langle S, \preceq \rangle$  can be *embedded* into a tree  $\langle T, \preceq \rangle$  if there exists an order-preserving injective mapping  $\langle S, \preceq \rangle \rightarrow \langle T, \preceq \rangle$ , i.e., if  $\langle S, \preceq \rangle$ , regarded as relational structure, is isomorphic to an induced substructure of  $\langle T, \preceq \rangle$ . For instance, we have an embedding



as indicated by the labels. If  $S$  can be embedded in  $T$  then  $S$  is isomorphic to a minor of  $T$ , when we consider  $S$  and  $T$  as graphs.

**Definition 4.11.** Let  $D = (U_v)_{v \in T}$  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

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

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

**Lemma 4.12.** *Let  $\mathfrak{G}$  be a graph and let  $D := (U_v)_{v \in T}$  be a tree decomposition of  $\mathfrak{G}$  of width  $k$  and height at most  $n + 1$ . If  $m \in \mathbb{N}$  is some number such that the tree  $m^{<n+1}$  cannot be embedded into  $T$ , then  $\text{twd}_n(\mathfrak{G}) < m(k + 1)$ .*

**Proof.** We construct a tree decomposition  $D'$  of height at most  $n$  and width at most  $m(k + 1) - 1$  as follows. Let  $P \subseteq T$  be the minimal (w.r.t.  $\subseteq$ ) set of vertices that contains

- every leaf of  $T$  at 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 (i) every vertex of  $T$  has less than  $m$   $F$ -successors; (ii) every path of  $T$  from the root to some leaf on level  $n$  contains at least one edge from  $F$ ; and (iii) no such path contains two consecutive edges from  $F$ .

The decomposition  $D' := D/F$  obtained by contracting all 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  $n$ . □

**Corollary 4.13.** *Let  $\mathcal{C}$  be a class such that  $k := \text{twd}_{n+1}(\mathcal{C}) < \infty$  and let  $m \in \mathbb{N}$ . If every structure  $\mathfrak{A} \in \mathcal{C}$  has a tree decomposition  $(U_v)_{v \in T}$  of width  $k$  and height at most  $n + 1$  such that the tree  $m^{<n+1}$  cannot be embedded into  $T$ , then  $\text{twd}_n(\mathcal{C}) < m(k + 1) < \infty$ .*

We conclude this section with a lemma that will be useful when constructing transductions  $\tau_n$  that transform a structure into their tree decompositions of height  $n$ . Our construction works for all tree decompositions that are *strict* in the following sense.

**Definition 4.14.** Let  $(U_v)_{v \in T}$  be a tree decomposition of a structure  $\mathfrak{A}$ .

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

$$\mu(a) := \min \{ v \in T \mid a \in U_v \}.$$

Note that  $\mu(a)$  is well-defined since, by the definition of a tree decomposition, there is at least one  $v \in T$  such that  $a \in U_v$ .

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

$$U_{\uparrow v} := \bigcup_{u \succeq v} U_u \setminus \bigcup_{u \prec v} U_u.$$

(c) The tree decomposition  $(U_v)_v$  is *strict* if, for every  $v \in T$ ,

- $U_v \cap \mu(A) \neq \emptyset$  (equivalently,  $U_v \setminus U_u \neq \emptyset$ , where  $u$  is the predecessor of  $v$ ) and
- if  $v$  is not the root of  $T$ , then the subgraph of  $\text{Gf}(\mathfrak{A})$  induced by the set  $U_{\uparrow v}$  is connected.

We conclude this section by a result implying that, for our purposes, it will be sufficient to consider only strict tree decompositions.

**Lemma 4.15.** *Let  $\mathfrak{G}$  be a graph. For every tree decomposition  $(U_v)_{v \in T}$  of  $\mathfrak{G}$ , there exists a strict tree decomposition  $(U'_v)_{v \in T'}$  of  $\mathfrak{G}$  whose width and height are at most those of  $(U_v)_{v \in T}$ .*

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

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

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

The limit  $(U_v^\omega)_{v \in T_\omega}$  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  $(u, v)$  of  $T_\omega$  such that  $U_v \cap \mu(V) = \emptyset$ . (Note that this implies  $U_v \subseteq U_u$ .) We construct the tree decomposition  $(U'_v)_{v \in T'}$  by contracting all edges in  $F$ . The details and the remaining verifications are left to the reader.  $\square$

## 5. TREE DECOMPOSITIONS AND TRANSDUCTIONS

In this section we relate the material presented in the preceding one to monadic second-order transductions. Let us start by showing that there is a transduction computing the minors of a graph.

**Lemma 5.1** ([Cou95]). *There exists a transduction  $\tau$  such that  $\tau(\mathfrak{G}_{\text{in}}) = \text{Min}(\mathfrak{G})$ , 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 four sets: the set of vertices we delete, the set of edges we delete, the set of edges we contract, and a set of vertices containing one representative of each contracted subgraph of  $\mathfrak{G}$  (these vertices serve as vertices of the resulting graph  $\mathfrak{H}$ ). With the help of these parameters we can define  $\mathfrak{H}$  inside of  $\mathfrak{G}_{\text{in}}$  by MSO-formulae.  $\square$

There is a close relationship between tree decompositions and transductions.

**Lemma 5.2.** *For every signature  $\Sigma$  and every number  $k \in \mathbb{N}$ , there exists a transduction  $\tau_k : \text{TREE}_0 \rightarrow \text{STR}_{\text{in}}[\Sigma]$  that maps an order-tree  $T$  to the class of all incidence structures  $\mathfrak{A}_{\text{in}}$  such that the corresponding  $\Sigma$ -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  $\Sigma$  and  $k$ .

Let  $\mathfrak{C}_0, \dots, \mathfrak{C}_{m-1}$  be an enumeration of all  $\Sigma$ -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  $T$  the 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  $\lambda$  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 an  $n$ -copying MSO-transduction where  $n$  is the maximal size of the structures  $(\mathfrak{C}_i)_{\text{in}}$ ,  $i < m$ .  $\square$

We have just seen that we can map a class of trees to a class of structures with these trees as tree decompositions. Conversely, if we only consider strict tree decompositions, we can define a transduction mapping a class of structures to the corresponding class of trees. Recall the function  $\mu : A \rightarrow T$  from Definition 4.14. that assigns to an element  $a \in A$  the minimal index  $v \in T$  such that  $a \in U_v$ .

**Proposition 5.3.** *For each number  $n \in \mathbb{N}$ , there exists an MSO-formula  $\varphi_n(x, y; \bar{Z})$  such that, for every strict tree decomposition  $D = (U_v)_{v \in T}$  of a graph  $\mathfrak{G}$  of height at most  $n$ , there are sets  $L_0, \dots, L_{n-1} \subseteq V$  such that*

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

**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. In particular,  $L_0 = U_{\emptyset}$  is the root component of the tree decomposition. 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

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

and  $a \sim b$  iff  $a \preceq b$  and  $b \preceq a$ , or if  $a, b \in L_0$ .

Clearly, the relation  $\preceq$  is MSO-definable with the help of the parameters  $\bar{L}$ . We claim that, for  $a, b \notin L_0$ , we have

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

( $\Leftarrow$ ) 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  $a \preceq b$ .

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

**Theorem 5.4.** *For each constant  $n \in \mathbb{N}$ , there exists a transduction  $\tau_n$  mapping a graph  $\mathfrak{G}$  to the class of all (underlying trees of) strict tree decompositions of  $\mathfrak{G}$  of height at most  $n$ .*

**Proof.** Let  $D = (U_v)_{v \in T}$  be a strict tree decomposition of  $\mathfrak{G}$ , and let  $\varphi_n(x, y; \bar{L})$  be the formula from Proposition 5.3 with parameters  $L_0, \dots, L_{n-1} \subseteq V$ . We can define the tree  $T$  underlying  $D$  as follows:

- Its root is any element of  $L_0 = U_{\emptyset}$ .
- For the other vertices of  $T$ , we choose one vertex in each  $\sim$ -class different from  $L_0$ . Note that  $\sim$  is definable with the help of  $\varphi_n$ .
- The ordering of  $T$  is defined by  $\varphi_n$ .

Hence, we obtain a transduction with parameters  $\bar{L}$  that transforms a graph into a ‘candidate’ tree decomposition. Via a backwards translation we can write down a formula stating that the candidate given by the parameters  $\bar{L}$  corresponds to an actual strict tree decomposition. We omit the details which are standard for this type of construction.  $\square$

In Lemma 5.2 we have seen how to obtain classes of bounded tree-width from classes of trees. Conversely, it is the case that every class obtained from a class of trees via a transduction has a bounded tree-width.

**Theorem 5.5.** *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  $m$ -coloured 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  $\mathfrak{T}$ .*

**Remark 5.6.** (a) Courcelle and Engelfriet [CE95] have shown that an incidence structure  $\mathfrak{A}_{\text{in}}$  obtained via a transduction  $\tau$  from an  $m$ -coloured tree  $\mathfrak{T}$  has bounded tree-width. Theorem 5.5 strengthens this result by proving that, if  $\mathfrak{A}_{\text{in}}$  is the image of a tree  $\mathfrak{T}$ , then we can use the same tree  $\mathfrak{T}$  as the tree underlying a tree decomposition of the given width.

(b) Lapoire has announced in [Lap98] a result somewhat related to Theorem 5.4. He claims that, for every  $k \in \mathbb{N}$ , there exists a transduction that transforms a given graph  $\mathfrak{G}$  of tree-width at most  $k$  to a coloured tree (like in the proof of Lemma 5.2) that encodes some tree decomposition of  $\mathfrak{G}$  of width at most  $k$ . Our result is less ambitious in the sense that

we only consider tree decompositions of a fixed height. This enables us to give a precise description of which tree decompositions (the strict ones) our transduction returns. Note that one can show that, for  $k \geq 2$ , there is no such transduction that would return *all* tree decompositions of  $\mathfrak{G}$  of width at most  $k$ .

We split the proof into several lemmas. 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 5.7.** Let  $\mathfrak{A}_{\text{in}} = \langle A \cup E, \bar{P}, \text{in}_0, \dots \rangle$  be an incidence structure.

(a) A *partition refinement* of  $\mathfrak{A}_{\text{in}}$  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.
- For the root  $\langle \rangle$ , we have  $W_{\langle \rangle} = A \cup E$
- For every internal vertex (i.e., non-leaf)  $u \in T$  with successors  $v_0, \dots, v_{n-1}$ , the sets  $W_{v_0}, \dots, W_{v_{n-1}}$  form a partition of  $W_u$ .
- $|W_u| = 1$ , for every leaf  $u \in T$ .
- $x \approx_v y$  and  $u \preceq v$  implies  $x \approx_u y$ .
- If  $u$  is an internal vertex of  $T$ ,  $v, w$  successors of  $u$ , not necessarily distinct, and  $x \in W_v, y \in W_w$  elements, then  $x \approx_u y$  implies either

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

$$(x, e) \in \text{in}_i \Leftrightarrow (y, e) \in \text{in}_i$$

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

$$(a, x) \in \text{in}_i \Leftrightarrow (a, y) \in \text{in}_i.$$

Note that it follows that, for every element  $x \in A \cup E$ , there is some leaf  $u \in T$  such that  $W_u = \{x\}$ .

(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  $\mathfrak{A}_{\text{in}}$  is the minimal width of a partition refinement of  $\mathfrak{A}_{\text{in}}$ .

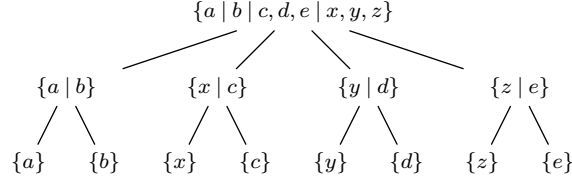
The notion of a partition refinement and of partition-width are adaptations of definitions from [Blu06, Blu03]. Up to a factor of 2, the partition-width of an incidence structure and its clique-width coincide.

**Example 5.8.** 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  $\mathfrak{A}_{\text{in}} = \langle A \cup E, P_R, \text{in}_0, \text{in}_1, \text{in}_2 \rangle$  with  $E = \{x, y, z\}$ . We obtain a partition refinement





where we have indicated the partition into  $\approx_v$ -classes by vertical bars. This partition refinement has width 4.

**Lemma 5.9.** *For every partition refinement  $\Pi = (W_v, \approx_v)_{v \in T}$  of an incidence structure  $\mathfrak{A}_{\text{in}} = \langle A \cup E, \bar{P}, \text{in}_0, \dots, \text{in}_{r-1} \rangle$ , there exists a tree decomposition  $D = (U_v)_{v \in T}$  of  $\mathfrak{A}$  with the same underlying tree  $T$  such that*

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

**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  $W_{l(x)} = \{x\}$ . We claim that the desired tree decomposition  $(U_u)_{u \in T}$  of  $\mathfrak{A}$  is given by

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

where

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

Note that the connectedness condition holds since  $(v, e) \in \text{in}_i$  implies that  $v$  belongs to precisely those components  $U_u$  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  $u = l(\bar{c})$ , for some  $\bar{c} \in E$ , then  $U_u = C_u$  consists of the components of  $\bar{c}$ . Hence,  $|U_u| = |\bar{c}| \leq r$ . 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)  $|C_u| \leq r \cdot |W_u / \approx_u|$ .

Since  $B_u, D_u \subseteq W_u$  it then 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  $i$  such that  $(x, e) \in \text{in}_i$  and  $u \not\preceq l(e)$ . We have  $(y, e) \in \text{in}_i$ , for every  $y \in W_u$  such that  $y \approx_u x$ . 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  $i$  such that  $(x, e) \in \text{in}_i$  and  $u = l(x) \sqcap l(e)$ . Let  $v$  be the successor of  $u$  such that  $v \preceq l(e)$ . We have  $(y, e) \in \text{in}_i$ , for all  $y \in W_u \setminus W_v$  such that  $y \approx_u x$ . 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  $j$  such that  $(y, f) \in \text{in}_j$  and  $l(y) \sqcap l(f) \preceq u$ . 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  $e \in E$  such that  $(x, e) \in \text{in}_i$  and  $u \preceq l(e)$ . Set

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

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

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

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

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

*Then there exists a number  $n \in \mathbb{N}$  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  $n$ .*

**Proof.** Let  $\langle \chi, \delta(x), (\varphi_{P_R}(x))_R, (\varphi_{\text{in}_i}(x, y))_{i < r} \rangle$  be the definition scheme of  $\tau$ , and 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

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

$$\text{and } x \approx_u y \quad : \text{iff} \quad \text{MTh}_h(\mathfrak{T}_v, x) = \text{MTh}_h(\mathfrak{T}_w, y),$$

where  $v, w$  are the 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 that it only depends on  $h$  and not on the input tree  $\mathfrak{T}$ .

It remains to show that  $\Pi$  is actually a partition refinement. First, let us prove that  $x \approx_v y$  and  $u \preceq v$  implies  $x \approx_u y$ . It is sufficient to consider the case that  $u$  is the predecessor of  $v$ . Then the general case follows by induction. Hence, suppose that  $v$  is a successor of  $u$ , that  $w, w'$  are successors of  $v$ , and that  $x, y$  are leaves with  $w \preceq x$  and  $w' \preceq y$  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 3.14, 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 successors of  $u$  and suppose that  $x, y$  are leaves with  $v \preceq x$  and  $w \preceq y$  such that  $x \approx_u y$ . We distinguish two cases.

Suppose that  $x, y \in A$  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  $z \in A \setminus (W_v \cup W_w)$  be an element. 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  $(z, y) \in \text{in}_i$ .  $\square$

**Proof of Theorem 5.5.** (1) First, suppose that  $\tau : \text{TREE}_m \rightarrow \text{STR}_{\text{in}}[\Sigma]$  is a basic MSO-transduction such that, for every  $m$ -coloured order-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 5.10 that there is a number  $w \in \mathbb{N}$  such that, for every tree  $\mathfrak{T}$ , we can find a partition refinement of  $\mathfrak{A}_{\text{in}} \in \tau(\mathfrak{T})$  with underlying tree  $\mathfrak{T}$  whose width is at most  $w$ . By Lemma 5.9 it follows that  $\mathfrak{A}$  has a tree decomposition  $(U_v)_{v \in T}$  with underlying tree  $\mathfrak{T}$  and whose width is less than  $k := w(r + 3)$ .

(2) If  $\tau$  is a basic MSO-transduction such that

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

then we can argue similarly. Let  $\tau'$  be the MSO-transduction 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  $\tau'$  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  $\tau$  is a non-copying MSO-transduction as in (2) but with  $p$  parameters. We can regard  $\tau$  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 structure  $\tau(\mathfrak{T}, \bar{P})$  has a tree decomposition of the required form.

(4) Finally, consider the general case. Suppose that  $\tau$  is  $l$ -copying. Given  $\mathfrak{T}$  let  $\mathfrak{T}^+$  be the tree obtained from  $\mathfrak{T}$  by adding  $l$  new successors to every vertex of  $\mathfrak{T}$ . Formally, suppose that  $T \subseteq D^{<\omega}$ , for some finite 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 parameters  $\bar{S}$  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  $H^+ = (U_v^+)_{v \in T^+}$  of  $\tau^+(\mathfrak{T}^+) = \tau(\mathfrak{T})$ . Let  $H = (U_v)_{v \in T}$  be the tree decomposition obtained from  $H^+$  by contracting every edge leading to a leaf in  $T^+ \setminus T$ . Then we have

$$\text{wd}(H) + 1 \leq (l + 1)(\text{wd}(H^+) + 1)$$

and  $\text{wd}(H^+) \leq w$ , for some  $w \in \mathbb{N}$  independent of  $\mathfrak{T}$ . □

## 6. THE TRANSDUCTION HIERARCHY

The focus of our investigation lies on the following preorder on classes of structures which compares their ‘encoding powers’ 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 6.4.

**Definition 6.1.** Let  $\mathcal{C}, \mathcal{K} \subseteq \text{STR}$ . We define the following relations.

- (a)  $\mathcal{C} \sqsubseteq \mathcal{K}$  if there exists a transduction  $\tau$  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)  $\mathcal{C} \triangleleft \mathcal{K}$  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  $\sqsubseteq_{\text{in}}$ ,  $\equiv_{\text{in}}$ , and  $\triangleleft_{\text{in}}$  are defined analogously to  $\sqsubseteq$ ,  $\equiv$ ,  $\triangleleft$  by replacing  $\sqsubseteq$  everywhere by  $\sqsubseteq_{\text{in}}$ .

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

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

**Lemma 6.2.**  $\sqsubseteq_{\text{in}}$  is a preorder on  $\mathcal{P}(\text{STR})$ .

**Definition 6.3.** 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)  $\mathcal{T}_{\text{bin}}$  is the class of all binary trees.
- (c)  $\mathcal{T}_\omega$  is the class of all trees.
- (d)  $\mathcal{P}$  is the class of all 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 6.4.** *We have the following hierarchy:*

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

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

**Remark 6.5.** There is a lot of flexibility in the choice of representatives for the various levels. For instance, we could replace  $\mathcal{T}_n$  by the class of *all* trees of height at most  $n$ ,  $\mathcal{T}_\omega$  by  $\mathcal{T}_{\text{bin}}$  or  $\{2^{<n} \mid n \in \mathbb{N}\}$ , and  $\mathcal{G}$  by the class of square grids.

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 6.6.** *We have*

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

**Proof.** In the example before Lemma 3.7, we have constructed transductions  $\tau_n$  such that  $\mathcal{T}_n \subseteq \tau_n(\mathcal{P})$ . Hence,  $\mathcal{T}_n \sqsubseteq_{\text{in}} \mathcal{P}$ . The remaining assertions follow from the observation that, by Lemma 5.1,  $\mathcal{C} \subseteq \text{Min}(\mathcal{K})$  implies  $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{K}$ .  $\square$

Let us collect some easy properties of the hierarchy. Our first result states that  $\mathcal{G}$  is a representative of the top level of the transduction hierarchy.

**Lemma 6.7.**  $\text{STR}[\Sigma] \sqsubseteq_{\text{in}} \mathcal{G}$

**Proof.** Recall that the  $m \times n$  grid is the undirected 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

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

and

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

It is easy to write down a formula checking that the parameters  $P_m$  and  $Q_m$  are correctly chosen (see, e.g., [Cou97]).

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

**Lemma 6.8.**  $\mathcal{T}_\omega \equiv_{\text{in}} \mathcal{T}_{\text{bin}}$ .

**Proof.** For one direction, note that  $\mathcal{T}_{\text{bin}} \subseteq \mathcal{T}_\omega$  implies  $\mathcal{T}_{\text{bin}} \sqsubseteq_{\text{in}} \mathcal{T}_\omega$ . Conversely, each finite tree can be obtained as minor of a binary tree. Therefore, we have  $\mathcal{T}_\omega \subseteq \text{Min}(\mathcal{T}_{\text{bin}}) \sqsubseteq_{\text{in}} \mathcal{T}_{\text{bin}}$ .  $\square$

**Lemma 6.9.** *We have  $\mathcal{C} \equiv_{\text{in}} \mathcal{T}_1$  if and only if  $\mathcal{C}$  is finite and contains at least one nonempty structure.*

As indicated in the example before Lemma 3.7, there exists a transduction mapping an incidence structure  $\mathfrak{A}_{\text{in}}$  to the incidence structure  $\text{Gf}(\mathfrak{A})_{\text{in}}$  of the Gaifman graph of  $\mathfrak{A}$ .

**Lemma 6.10.** *For every class  $\mathcal{C}$  of structures, we have  $\text{Gf}(\mathcal{C}) \sqsubseteq_{\text{in}} \mathcal{C}$ .*

The next result is just a restatement of Lemma 5.1 in our current terminology.

**Lemma 6.11.** *For every class  $\mathcal{C}$  of graphs we have  $\text{Min}(\mathcal{C}) \equiv_{\text{in}} \mathcal{C}$ .*

## 7. STRICTNESS OF THE HIERARCHY

In this section we prove that the hierarchy is strict. Using the results of Section 5 we can characterise each level of the hierarchy in terms of tree-width and its variants.

**Theorem 7.1.** *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 5.2 and  $(\Rightarrow)$  follows from Theorem 5.5.  $\square$

**Corollary 7.2.** *Let  $\mathcal{C}$  be a class of  $\Sigma$ -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.** (a) Suppose that  $\text{pwd}(\mathcal{C}) = \infty$ . Then Theorem 4.5 implies that  $\mathcal{T}_\omega \subseteq \text{Min}(\text{Gf}(\mathcal{C}))$ . Hence, the claim follows from Lemmas 6.10 and 6.11.

(b) Suppose that  $\text{twd}(\mathcal{C}) = \infty$ . Then Theorem 4.6 implies that  $\mathcal{G} \subseteq \text{Min}(\text{Gf}(\mathcal{C}))$ . As in (a), the claim follows from Lemmas 6.10 and 6.11.  $\square$

**Corollary 7.3.** *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*  $\mathcal{G} \sqsubseteq_{\text{in}} \mathcal{C}$ .

In particular, it follows that the upper part of the hierarchy is strict:

**Corollary 7.4.**  $\mathcal{P} \triangleleft_{\text{in}} \mathcal{T}_\omega \triangleleft_{\text{in}} \mathcal{G}$

**Proof.** Both assertions follow from Theorem 7.1 and Corollary 7.3.

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, by Theorem 7.1 (a), that  $\mathcal{T}_\omega \not\sqsubseteq_{\text{in}} \mathcal{P}$ . Hence,  $\mathcal{P} \sqsubset_{\text{in}} \mathcal{T}_\omega$ . Finally, if  $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_\omega$  then  $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{P}$ , by Corollary 7.3 (a). Consequently, we have  $\mathcal{P} \triangleleft_{\text{in}} \mathcal{T}_\omega$ .

Similarly, the fact that  $\mathcal{T}_\omega \sqsubseteq_{\text{in}} \mathcal{G}$  follows from Lemma 6.7. Since  $\text{twd}(\mathcal{G}) = \infty$ , Theorem 7.1 (b) implies that  $\mathcal{T}_\omega \sqsubset_{\text{in}} \mathcal{G}$ . Finally, we obtain  $\mathcal{T}_\omega \triangleleft_{\text{in}} \mathcal{G}$  by Corollary 7.3 (b).  $\square$

Let us turn to the lower part. We start with two technical lemmas.

**Definition 7.5.** Let  $\mathfrak{T} = \langle T, \leq \rangle$  be an order-tree. Vertices  $v_0, \dots, v_{m-1}$  of  $T$  are *horizontally related* via a vertex  $w$  if all  $v_i$  are at the same level of the tree and  $v_i \sqcap v_k = w$ , for all  $0 \leq i < k < m$ .

**Lemma 7.6.** *Let  $\mathfrak{T}$  be a coloured order-tree of height  $n$ , and suppose that  $\tau$  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  $v_0, \dots, v_{m-1}$  of  $\mathfrak{T}$  that are horizontally related via  $w$  and fix some number  $l < k$ . Let  $x_i$  be the successor of  $w$  with  $x_i \preceq v_i$ . 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 at least  $m - 1$  elements of the set  $\{\langle v_0, l \rangle, \dots, \langle v_{m-1}, l \rangle\}$  (these are elements of the domain of  $\tau(\mathfrak{T})$ ) are horizontally related in  $\tau(\mathfrak{T})$ .*

**Proof.** Let  $\varphi_{ss'}(x, y)$  be the formula defining the successor relation in  $\tau(\mathfrak{T})$  between vertices of the form  $\langle x, s \rangle$  and  $\langle y, s' \rangle$ . By assumption the rank of  $\varphi_{ss'}(x, y)$  is at most  $r$ .

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 the formula

$$\begin{aligned} \psi_{s_0 \dots s_{h-1}}(v) := \exists y_0 \cdots \exists y_{h-1} \Bigg[ \bigwedge_{i < h-1} \varphi_{s_i s_{i+1}}(y_i, y_{i+1}) \wedge \varphi_{s_{h-1} l}(y_{h-1}, v) \\ \wedge \neg \exists z \bigvee_{s < k} \varphi_{ss_0}(z, y_0) \Bigg] \end{aligned}$$

expresses that there exists a path of the form  $\langle y_0, s_0 \rangle, \dots, \langle y_{h-1}, s_{h-1} \rangle, \langle v, l \rangle$  from the root  $\langle y_0, s_0 \rangle$  of  $\tau(\mathfrak{T})$  to the vertex  $\langle v, l \rangle$ . By assumption on  $v_i$  and Lemma 3.14, 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  $u_0 \sqcap v_0 \preceq w$  in  $\mathfrak{T}$ . By  $(*)$  and Lemma 3.14, 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  $u_0 \sqcap v_i \preceq w$ . Note that there can be at most one index  $i$  that does not satisfy this condition since, if we had  $u_0 \sqcap v_i \succeq w$  and  $u_0 \sqcap v_j \succeq w$ , for  $i \neq j$ , then we would have  $v_i \sqcap v_j \succ w$  and  $v_0, \dots, v_{m-1}$  would not be horizontally related via  $w$ . It follows that

$$\mathfrak{T} \models \varphi_{s_0 l}(u_0, v_0) \quad \text{implies} \quad \mathfrak{T} \models \varphi_{s_0 l}(u_0, v_i), \quad \text{for all } i \text{ as above.}$$

Hence,  $\langle u_0, s_0 \rangle$  is the common predecessor of all the  $\langle v_i, l \rangle$ , except for possibly one of them. (For an index  $i$  with  $u_0 \sqcap v_i \succeq w$  our composition argument does not work since in that case  $(*)$  does not imply that the theories of  $(\mathfrak{T}, u_0, v_0)$  and  $(\mathfrak{T}, u_0, v_i)$  coincide.)

It remains to consider the case that  $w \prec u_0 \sqcap v_0$ . Setting

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

we have

$$\mathfrak{T}_{x_0} \models \exists z[|z| = |u_0| \wedge \eta_{u_0}(z) \wedge \varphi_{s_0 l}(z, v_0)].$$

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

$$\mathfrak{T}_{x_i} \models \exists z[|z| = |u_0| \wedge \eta_{u_0}(z) \wedge \varphi_{s_0 l}(z, v_i)], \quad \text{for all } i < m.$$

Consequently, we have  $|u_i| = |u_0|$ , for all  $i$ , and  $u_0, \dots, u_{m-1}$  are horizontally related via  $w$ . Since the vertices  $\langle u_0, s_0 \rangle, \dots, \langle u_{m-1}, s_0 \rangle$  are on level  $h-1$  in  $\tau(\mathfrak{T})$ , we can apply the induction hypothesis and it follows that all but at most one of them are horizontally related via some vertex  $w'$ . Therefore, the same holds for their successors  $\langle v_0, l \rangle, \dots, \langle v_{m-1}, l \rangle$ .  $\square$

**Definition 7.7.** We denote by  $B(n, k, c)$  the number of functions  $m^{<n} \rightarrow \mathcal{P}([c])$  with  $m \leq k$ . Intuitively, each such function corresponds to a vertex-colouring of the tree  $m^{<n}$  with  $c$  colours.

**Lemma 7.8.** For  $n \geq 1$  and  $k \geq 2$ , we have

$$2^{ck^{n-1}} \leq B(n, k, c) \leq k2^{2ck^{n-1}}.$$

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

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

Since  $|[m]^{<n}| = \sum_{i < n} m^i = m^{n-1} + \sum_{i < n-1} m^i$  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

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

□

**Theorem 7.9.**  $\mathcal{T}_n \sqsubset_{\text{in}} \mathcal{T}_{n+1}$

**Proof.** For a contradiction, suppose that there exists a transduction  $\tau$  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  $\mathcal{T}_{n+1}^{\text{col}} := \exp_1(\mathcal{T}_{n+1})$  be the class of all coloured successor-trees with one colour whose underlying tree is in  $\mathcal{T}_{n+1}$ . Since the successor-trees in  $\mathcal{T}_n$  are 1-sparse we can construct an MSO-transduction  $\sigma_0$  such that  $(\mathcal{T}_n)_{\text{in}} \subseteq \sigma_0(\mathcal{T}_n)$ . Since  $\mathcal{T}_n^{\text{ord}} \equiv \mathcal{T}_n$  we can combine  $\tau$  and  $\sigma_0$  to a transduction  $\sigma$  such that  $\mathcal{T}_{n+1}^{\text{col}} \subseteq \sigma(\mathcal{T}_n^{\text{ord}})$ . By Lemma 7.6, 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  $\sigma$  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. □



## 8. COMPLETENESS OF THE HIERARCHY

We have shown that the hierarchy presented in Theorem 6.4 is strict. To conclude the proof of the theorem it therefore remains to show that there are no additional classes. We have already seen in Corollary 7.4 that  $\mathcal{T}_\omega$  and  $\mathcal{G}$  are the only classes above  $\mathcal{P}$ . Next we shall prove that  $\mathcal{T}_n \triangleleft \mathcal{T}_{n+1}$ .

**Lemma 8.1.** *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 underlying a 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 4.15, 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  $k$  and with an underlying tree  $T$  into which we can embed the tree  $m^{<n+1}$ . According to Proposition 5.3 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 8.2.** *Let  $\mathcal{C}$  be a class of structures. Then  $\mathcal{T}_{n+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$  implies  $\text{twd}_n(\mathcal{C}) < \infty$ .*

**Proof.** Suppose that  $\mathcal{T}_{n+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$ . Then  $\mathcal{P} \not\sqsubseteq_{\text{in}} \mathcal{C}$  since  $\mathcal{T}_{n+1} \sqsubseteq_{\text{in}} \mathcal{P}$ . By Lemmas 6.10 and 6.11, this implies 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 4.9, it follows that there are numbers  $k, l \in \mathbb{N}$  such that  $\text{twd}_l(\mathcal{C}) < k$ .

By induction on  $l$ , we prove that  $\text{twd}_l(\mathcal{C}) < \infty$  implies  $\text{twd}_n(\mathcal{C}) < \infty$ . For  $l \leq n$ , there is nothing to do. For  $l > n$ , we have  $\mathcal{T}_{n+1} \sqsubseteq_{\text{in}} \mathcal{T}_l$ , which implies that  $\mathcal{T}_l \not\sqsubseteq_{\text{in}} \mathcal{C}$ . Consequently, it follows by Lemma 8.1 and Corollary 4.13 that  $\text{twd}_{l-1}(\mathcal{C}) < \infty$ . By induction hypothesis, the result follows.  $\square$

By Lemma 5.2 we obtain the following results.

**Corollary 8.3.** *If  $\mathcal{T}_{n+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$  then  $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_n$ .*

**Corollary 8.4.**  *$\mathcal{T}_n \triangleleft_{\text{in}} \mathcal{T}_{n+1}$ .*

To conclude the proof of Theorem 6.4 it remains to show that there are no classes between the lower part of the hierarchy and its upper part.

**Lemma 8.5.** *Let  $\mathcal{C}$  be a class of structures. If  $\mathcal{T}_n \sqsubseteq_{\text{in}} \mathcal{C}$ , for all  $n \in \mathbb{N}$ , then  $\mathcal{P} \sqsubseteq_{\text{in}} \mathcal{C}$ .*

**Proof.** We show the contraposition. Suppose that  $\mathcal{P} \not\sqsubseteq_{\text{in}} \mathcal{C}$ . We have to show that there is some  $n$  such that  $\mathcal{T}_n \not\sqsubseteq_{\text{in}} \mathcal{C}$ . As in the proof of Theorem 8.2 it follows that there are numbers  $k, l \in \mathbb{N}$  such that  $\text{twd}_l(\mathcal{C}) < k$ . Hence, we can use Lemma 5.2 to obtain a transduction  $\tau$  witnessing that  $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_l$ . By Theorem 7.9 we have  $\mathcal{T}_{l+1} \not\sqsubseteq_{\text{in}} \mathcal{T}_l$ . It follows that  $\mathcal{T}_{l+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$ , as desired.  $\square$

**Corollary 8.6.** *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 8.5, there is some  $n \in \mathbb{N}$  such that  $\mathcal{T}_{n+1} \not\sqsubseteq_{\text{in}} \mathcal{C}$ . Hence, Corollary 8.3 implies that  $\mathcal{C} \sqsubseteq_{\text{in}} \mathcal{T}_n$ .  $\square$

Together, Corollaries 7.4, 8.4, and 8.6 (and the fact that every class  $\mathcal{C}$  satisfies  $\emptyset \sqsubseteq_{\text{in}} \mathcal{C} \sqsubseteq_{\text{in}} \mathcal{G}$ ) show that every class of  $\Sigma$ -structures is  $\equiv_{\text{in}}$ -equivalent to some of the classes in Theorem 6.4. This completes the proof of this theorem.

## 9. 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 classes of 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. (For an example of nonlinearity, note that the class of all countable trees and the class of all finite grids are incomparable.)

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 7.9 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 3.13 also goes through for CMSO.

For logics much weaker than MSO, on the other hand, it seems unrealistic to hope for a complete description of the corresponding transduction hierarchy. For instance, a related hierarchy for first-order logic was investigated by Mycielski, Pudlák, and Stern in [MPS90]. The results they obtain indicate that the structure of the resulting hierarchy is very complicated.

Finally, let us address question (i). When using transductions between 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 9.1.** *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  $\varrho$  such that  $\mathcal{C} = \varrho(\mathcal{C}_{\text{in}})$ . Since  $\mathcal{S}$  is  $k$ -sparse we can also find a transduction  $\sigma$  such that  $\mathcal{S}_{\text{in}} = \sigma(\mathcal{S})$ . Consequently,

$$\mathcal{C}_{\text{in}} \subseteq \tau(\mathcal{S}_{\text{in}}) \quad \text{implies} \quad \mathcal{C} \subseteq (\varrho \circ \tau \circ \sigma)(\mathcal{S}),$$

and

$$\mathcal{S} \subseteq \tau(\mathcal{C}) \quad \text{implies} \quad \mathcal{S}_{\text{in}} \subseteq (\sigma \circ \tau \circ \varrho)(\mathcal{C}_{\text{in}}).$$

□

**Theorem 9.2.** *We have the following hierarchy:*

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

**Proof.** Note that all classes in Theorem 9.2 are 2-sparse. For 2-sparse classes  $\mathcal{C}$  and  $\mathcal{K}$ , Lemma 9.1 implies that

$$\mathcal{C} \sqsubseteq \mathcal{K} \quad \text{iff} \quad \mathcal{C} \sqsubseteq_{\text{in}} \mathcal{K}.$$

Consequently, the result follows from Theorem 6.4.

□

**Open Problem 9.3.** Is there any class  $\mathcal{C} \subseteq \text{STR}[\Sigma]$  which is not  $\equiv$ -equivalent to some class in the above hierarchy?

**Remark 9.4.** If we only consider classes of graphs and if we use CMSO-transductions instead of MSO-transductions, then the following result can be used as replacement of Theorem 4.6:

**Theorem 9.5** ([CO07]). *Let  $\mathcal{C}$  be a class of graphs with unbounded clique-width. There exists a CMSO-transduction  $\tau$  with  $\mathcal{G} \subseteq \tau(\mathcal{C})$ .*

This eliminates some possibilities for intermediate classes of graphs in the hierarchy of Theorem 9.2, but to complete the picture we still need analogues of Proposition 5.3 and of Theorems 4.5 and 4.9. Furthermore, the techniques of [CO07] are specific to graphs (or, more generally, to relational structures where all relations are binary). Even with the results of [CO07] one cannot exclude the existence of a class  $\mathcal{C}$  of arbitrary relational structures strictly between  $\mathcal{T}_\omega$  and  $\mathcal{G}$  in the CMSO-transduction hierarchy.

Let us make a final comment about relational structures. An incidence structure  $\mathfrak{A}_{\text{in}}$  can be seen as a bipartite labelled directed graph (see the remark after Definition 2.1). Furthermore, it is 1-sparse. Hence, our results use tools from graph theory, in particular those of [RS83, RS86, CO07]. However, there is currently no encoding of relational structures as labelled graphs that could help to solve question (i) above.

## REFERENCES

- [BCL07] A. Blumensath, T. Colcombet, and C. Löding. Logical Theories and Compatible Operations. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata: History and Perspectives*, pages 73–106. Amsterdam University Press, 2007.
- [Blu03] A. Blumensath. *Structures of Bounded Partition Width*. Ph.D. Thesis, RWTH Aachen, Aachen, 2003.
- [Blu06] A. Blumensath. A Model Theoretic Characterisation of Clique-Width. *Annals of Pure and Applied Logic*, 142:321–350, 2006.
- [Blu10] A. Blumensath. Guarded Second-Order Logic, Spanning Trees, and Network Flows. *Logical Methods in Computer Science*, 6, 2010.
- [Bod96] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal of Computing*, 25:1305–1317, 1996.
- [CE95] B. Courcelle and J. Engelfriet. A Logical Characterization of the Sets of Hypergraphs Defined by Hyperedge Replacement Grammars. *Math. System Theory*, 28:515–552, 1995.
- [CO07] B. Courcelle and S.-I. Oum. Vertex-Minors, Monadic Second-Order Logic, and a Conjecture by Seese. *Journal Combinatorial Theory B*, 97:91–126, 2007.
- [Cou87] B. Courcelle. An axiomatic definition of context-free rewriting and its application to NLC graph grammars. *Theoretical Computer Science*, 55:141–181, 1987.
- [Cou91] B. Courcelle. The monadic second-order logic of graphs V: On closing the gap between definability and recognizability. *Theoretical Computer Science*, 80:153–202, 1991.
- [Cou95] B. Courcelle. The monadic second-order logic of graphs VIII: Orientations. *Annals of Pure and Applied Logic*, 72:103–143, 1995.
- [Cou97] B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In [Roz97], pages 313–400. 1997.
- [Cou03] B. Courcelle. The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications. *Theoretical Computer Science*, 299:1–36, 2003.
- [Die06] R. Diestel. *Graph Theory*. Springer, 3rd edition, 2006.
- [FG06] J. Flum and M. Grohe. *Parametrized Complexity Theory*. Springer Verlag, 2006.
- [GHO02] E. Grädel, C. Hirsch, and M. Otto. Back and Forth Between Guarded and Modal Logics. *ACM Transactions on Computational Logics*, pages 418–463, 2002.

- [Lap98] 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*, pages 618–628, 1998.
- [Lib04] L. Libkin. *Elements of Finite Model Theory*. Springer Verlag, 2004.
- [MPS90] J. Mycielski, P. Pudlák, and A. S. Stern. *A lattice of chapters of mathematics (interpretations between theorems)*. Mem. Amer. Math. Soc. 426. AMS, 1990.
- [NdM06a] J. Nešetřil and P. Ossona de Mendez. Linear time low tree-width partitions and algorithmic consequences. In *Proc. 38th Annual ACM Symposium on Theory of Computing, STOC*, pages 391–400, 2006.
- [NdM06b] J. Nešetřil and P. Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics*, 27:1022–1041, 2006.
- [Rab69] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.
- [Roz97] G. Rozenberg, editor. *Handbook of graph grammars and computing by graph transformations*, volume 1: Foundations. World Scientific, 1997.
- [RS83] N. Robertson and P. D. Seymour. Graph Minors. I. Excluding a Forest. *Journal of Combinatorial Theory B*, 35:39–61, 1983.
- [RS86] N. Robertson and P. D. Seymour. Graph Minors. V. Excluding a Planar Graph. *Journal of Combinatorial Theory B*, 41:92–114, 1986.
- [See91] D. Seese. The structure of the models of decidable monadic theories of graphs. *Annals of Pure and Applied Logic*, 53:169–195, 1991.
- [She75] S. Shelah. The Monadic Second Order Theory of Order. *Annals of Mathematics*, 102:379–419, 1975.