



**HAL**  
open science

# Expressiveness Results for an Inductive Logic of Separated Relations

Radu Iosif, Florian Zuleger

► **To cite this version:**

Radu Iosif, Florian Zuleger. Expressiveness Results for an Inductive Logic of Separated Relations. 2022. hal-03826324

**HAL Id: hal-03826324**

**<https://hal.science/hal-03826324>**

Preprint submitted on 24 Oct 2022

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

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

# Expressiveness Results for an Inductive Logic of Separated Relations

Radu Iosif<sup>1</sup> and Florian Zuleger<sup>2</sup>

<sup>1</sup> Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, 38000, France

<sup>2</sup> Institute of Logic and Computation, Technische Universität Wien, Austria

**Abstract.** In this paper we study a Separation Logic of Relations (SLR) and compare its expressiveness to (Monadic) Second Order Logic [(M)SO]. SLR is based on the well-known Symbolic Heap fragment of Separation Logic, whose formulae are composed of points-to assertions, inductively defined predicates, with the separating conjunction as the only logical connective. SLR generalizes the Symbolic Heap fragment by supporting general relational atoms, instead of only points-to assertions. In this paper, we restrict ourselves to finite relational structures, and hence only consider Weak (M)SO, where quantification ranges over finite sets. Our main results are that SLR and MSO are incomparable on structures of unbounded treewidth, while SLR can be embedded in SO in general. Furthermore, MSO becomes a strict subset of SLR, when the treewidth of the models is bounded by a parameter and all vertices attached to some hyperedge belong to the interpretation of a fixed unary relation symbol. We also discuss the problem of identifying a fragment of SLR that is equivalent to MSO over models of bounded treewidth.

## 1 Introduction

The theory of finite (hyper-)graphs is important for many areas of computing, such as e.g., static program analysis [36], database theory and knowledge representation [1]. Graphs are naturally encoded as logical structures over finite signatures of relation symbols. Graph properties are traditionally expressed in classical second-order logics, where quantifiers range over either individuals or relations. The most prominent logics in this class are the Monadic Second Order Logics, where quantification is over vertices only ( $MSO_1$ ), or both vertices and edges ( $MSO_2$ ), and sets thereof [22]. These logics are quite powerful, since properties as e.g., Hamiltonicity, 3-colorability, planarity, etc. can be expressed.

The  $MSO_{1,2}$  logics have been instrumental in establishing equivalences between *definability* and *recognizability*, which are cornerstones of formal language theory. For words [13] and ranked trees [26] definability and recognizability coincide, whereas the situation is more complex for graphs: definable sets are always recognizable, but there are recognizable sets of graphs that are not definable [19]. The equivalence between definability and recognizability for graphs can be recovered by considering only those sets of graphs that are *inductively defined* by (i.e., are components of the least solutions of) a finite set of recursive equations with set variables, written using an algebra of hyperedge-replacement operations<sup>3</sup> [20,6]. Moreover, these are exactly the sets of

<sup>3</sup> A hyperedge replacement substitutes a graph for a hyperedge in another graph.

graphs in which the *treewidth* (a positive integer that indicates how close the graph is to a tree) is topped by a fixed constant (see e.g., [22, Theorem 2.83]).

Another way of describing graph properties is to extend first-order logic with a *spatial connective* that decomposes graphs [16,34,46]. Intuitively, a formula  $\phi * \psi$  denotes graphs that can be decomposed into subgraphs satisfying  $\phi$  and  $\psi$ , respectively, whose sets of edges are disjoint and the only common vertices are denoted by variables or constant symbols from the signature.

It is known that first-order logic can only express local properties of graphs [32], whereas the interplay of spatial and boolean connectives allows to describe certain non-local properties, such as reachability between vertices [23,10]. A general way of specifying non-local properties of finite graphs is adding a *least fixpoint* operator [44] or *inductive definitions* [2]. A most prominent example is the *Separation Logic* (SL) of Ishtiaq, O’Hearn [34] and Reynolds [46], where inductive definitions are used to define the shapes of datastructures in the heap memory of a program. For instance, the rules

$$(1) \text{ls}(x,y) \leftarrow x = y \quad (2) \text{ls}(x,y) \leftarrow \exists z . x \mapsto z * \text{ls}(z,y)$$

define finite singly-linked list segments, that are either (1) empty with equal endpoints, or (2) consist of a single cell  $x$  *separated from* the rest of the list segment  $\text{ls}(z,y)$ . The graphs used in the semantics of SL are described as finite partial functions from vertices to sequences of vertices of a fixed length  $k$ , called *k-heaps* ( $k = 1$  in the example above).

In general, most recursive datastructures (singly- and doubly-linked lists, trees, etc.) can be defined using only existentially quantified separated (spatial) conjunctions of atoms, that are either equalities  $x = y$ , disequalities  $x \neq y$  and points-to propositions  $x \mapsto (y_1, \dots, y_k)$ . This simple subset of SL is referred to as the *Symbolic Heap* fragment. The decision problems of model checking [12], satisfiability [11], robustness properties [35] and entailment<sup>4</sup> [18,37,29,30,43] for this fragment have been extensively studied, yielding a detailed map of decidability and complexity results.

Extending SL beyond heaps to general relational signatures has been first considered in the context of *role logic* [38], to describe properties of relational databases and type systems of object-oriented languages. Another, quite recent, application has been writing Hoare-style proofs of correctness of dynamically reconfigurable distributed systems [4,7,9]. In this logic, called *Separation Logic of Relations* (SLR), the separating conjunction is understood as splitting the interpretation of each relation symbol from the signature into disjoint parts. For instance,  $R(x_1, \dots, x_n) * R(y_1, \dots, y_n)$  means that  $R$  is interpreted as a relation consisting of two distinct tuples i.e., that values of  $x_i$  and  $y_i$  differ for at least some  $1 \leq i \leq n$ . Note the interpretation of a relation atom  $R(x_1, \dots, x_n)$  as a singleton relations consisting of the tuple of values of  $x_1, \dots, x_n$ , respectively, that matches the denotation of a SL points-to atom  $x \mapsto (y_1, \dots, y_k)$  as a singleton heap.

**Relation between SLR and SL** In general, symbolic heaps with inductive definitions are at least as expressive for SLR as for SL<sup>5</sup>. For instance, the previous definition of a list segment can written in a relational signature having at least a unary relation  $D$  and a binary relation  $H$ , as follows:

$$\text{rel\_ls}(x,y) \leftarrow x = y \quad \text{rel\_ls}(x,y) \leftarrow \exists z . D(x) * H(x,z) * \text{rel\_ls}(z,y)$$

<sup>4</sup> Since symbolic heaps do not have negation, entailment cannot be reduced to satisfiability.

<sup>5</sup> Except for the functionality of the heap, which cannot be expressed without negations.

$\llbracket \mathcal{L}_1 \rrbracket \subseteq \llbracket \mathcal{L}_2 \rrbracket$	$\llbracket \mathcal{L}_1 \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \mathcal{L}_2 \rrbracket$	SLR		MSO		SO	
SLR		✓	?	× (§4)	× (§4)	✓ (§6)	✓ (§8)
MSO		× (§5)	✓ (§7)	✓	✓ (§8)	✓	✓ (§8)
SO		× (§5)	?	× (§8)	× (§8)	✓	✓ (§8)

**Table 1.** A comparison of SLR, MSO and SO in terms of expressiveness, where ✓ means that the inclusion holds, × means it does not and ? denotes an open problem, with reference to the sections where (non-trivial) proofs are given.

Note that the  $D(x)$  atoms joined by separating conjunction ensure that all the vertices are pairwise disjoint, except for the last one denoted by  $y$ . In contrast, the rules:

$$\text{fold\_ls}(x, y) \leftarrow x = y \quad \text{fold\_ls}(x, y) \leftarrow \exists z . H(x, z) * \text{fold\_ls}(z, y)$$

describes “foldable” lists in which any two vertices can be fused into one. Moreover, SLR can describe graphs of unbounded degree<sup>6</sup>, such as stars with a central vertex and outgoing binary edges  $E$  to frontier vertices, all belonging to a set  $N$ :

$$\text{star}(x) \leftarrow N(x) * \text{node}(x) \quad \text{node}(x) \leftarrow x = x \quad \text{node}(x) \leftarrow \exists y . E(x, y) * N(y) * \text{node}(x)$$

Such definitions are typically beyond the reach of standard SL symbolic heaps, because  $k$ -heaps can only represent graphs of (out-)degree bounded by  $k$ .

**Our contributions** We compare the expressiveness of SLR with (monadic) second-order logic (M)SO. Strictly speaking, by MSO we understand the fragment of  $\text{MSO}_1$  [22], without cardinality constraints<sup>7</sup> (we consider the extension of our results to cardinality constraints an interesting question left for future work). In this paper, we are only interested in finite relational structures, and hence only consider Weak (M)SO, where quantification is restricted to *finite* sets of tuples of vertices (in  $\text{MSO}_1$  quantification is over sets of vertices). In the remainder, (M)SO will always refer to Weak (M)SO.

Table 1 summarizes our results. For a set  $\mathcal{L}$  of logical formulæ written using a finite set  $\Sigma$  of relation and constant symbols, we denote by  $\llbracket \mathcal{L} \rrbracket$  the set of sets of graphs that are models of a formula  $\phi$  from  $\mathcal{L}$ . For a unary relation symbol  $\mathcal{D}$  not in  $\Sigma$ , considered fixed in the rest of the paper, we say that a graph is *guarded* if all vertices attached to some hyperedge belong to the set denoted by  $\mathcal{D}$ . Then  $\llbracket \mathcal{L} \rrbracket_{\mathcal{D}}^k$  denotes the set of sets of guarded models of treewidth at most  $k$  of a formula from  $\mathcal{L}$ , where the signature of  $\mathcal{L}$  is extended with  $\mathcal{D}$ . With this notation,  $\llbracket \mathcal{L}_1 \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \mathcal{L}_2 \rrbracket$  means that the logic  $\mathcal{L}_2$  is at least as expressive as  $\mathcal{L}_1$ , when only guarded models of treewidth at most  $k$  are considered. Note that  $\llbracket \mathcal{L} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \mathcal{L} \rrbracket$  is not a trivial statement, in general, because it asserts the existence of a formula of  $\mathcal{L}$  that defines the set of guarded graphs of treewidth at most  $k$ .

The most interesting results of our comparison are that:

1. SLR and MSO are incomparable on unguarded graphs of unbounded treewidth i.e., there are formulæ in each of the logics that do not have an equivalent in the other,
2. SO is at least as expressive as SLR, when considering unguarded graphs of unbounded treewidth,
3. SLR is strictly more expressive than MSO, when considering guarded graphs of bounded treewidth; this result is probably the most interesting, as it gives a sense of the expressive power of SLR, with emphasis on the treewidth parameter.

<sup>6</sup> The degree of a graph is the maximum number of hyperedges involving a vertex.

<sup>7</sup> The cardinality of a set variable equals  $p$  modulo  $q$ , where  $0 \leq p < q$  are constants.

Note that, when considering SLR-definable sets of bounded treewidth, we systematically assume these graphs to be guarded. We state as an open problem and conjecture that every infinite SLR-definable set of graphs is necessarily guarded, in a hope that the guardedness condition can actually be lifted. Similar conditions have been used to e.g., obtain decidability of entailments between SL symbolic heaps [33,37] and of invariance for assertions written in a fragment of SLR for describing distributed networks [9]. Moreover, the problem of checking if a given set of inductive definitions defines a guarded set of graphs is decidable for these logics [35,8].

A further natural question asks for a fragment of SLR with the same expressive power as MSO, over graphs of bounded treewidth. This is also motivated by the need for a general fragment of SLR with a decidable entailment<sup>8</sup> problem, that is instrumental in designing automated program analyses. Unfortunately, such a definition is challenging because the MSO-definability of the sets defined by SLR is an undecidable problem, whereas treewidth boundedness of such sets remains an open problem.

For space reasons, the proofs of technical results are given in Appendix A.

**Related work** The problem of comparing the expressiveness of SL [46] with that of classical logics received a fair amount of attention. For instance, a first proof of undecidability of the satisfiability problem for first-order SL [15] is based on a reduction to Trakhtenbrot’s undecidability result for first-order logic on finite models [27]. This proof relies on 2-heaps to encode arbitrary binary relations as  $R(x,y) \stackrel{\text{def}}{=} \exists z . z \mapsto (x,y) * \text{true}$ . A more refined proof for 1-heaps was given by Brochenin, Demri and Lozes [10]. In particular, they show that SO has the same expressivity as SL over 1-heaps. A related line of work, pioneered by Lozes [40], is the translation of quantifier-free SL formulae into boolean combinations of *core formulae*, belonging to a small set of very simple patterns. This enables a straightforward translation of the quantifier-free fragment of SL into first-order logic, over unrestricted signatures with both relation and function symbols, that was subsequently extended to two quantified variables [24] and restricted quantifier prefixes [28]. Moreover, a translation of quantifier-free SL into first-order logic, based on the small model property of the former, has been described in [14].

The above references focus exclusively on fragments of SL without inductive definitions, but with arbitrary combinations of boolean (conjunction, negation) and spatial (separating conjunction, magic wand) connectives. A non-trivial attempt of generalizing the technique of translating SL into boolean combinations of core formulae to *reachability* and *list segment* predicates is given in [25]. Moreover, an extensive comparison between the expressiveness of various models of separation i.e., spatial, as in SL, and contextual (subtree-like), as in Ambient Logic [17], can be found in [42].

An early combination of spatial connective for graph decomposition with (least fixpoint) recursion is Graph Logic (GL) [16], whose expressiveness is compared to that of  $\text{MSO}_2$  [23]. For reasons related to its applications, GL quantifies over the vertices and edge labels of a graph, unlike  $\text{MSO}_2$  that quantifies over vertices, edges and sets thereof. Another fairly subtle difference is that GL can describe graphs with multiple edges that involve the same vertices and same label, whereas the models of  $\text{MSO}_2$  are simple graphs. Without recursion, GL can be translated into  $\text{MSO}_2$  and it has been showed that

<sup>8</sup> The entailment problem  $\phi \models_{\Delta} \psi$  asks if every model of  $\phi$  is also a model of  $\psi$ , when the inductive predicates in both  $\phi$  and  $\psi$  are interpreted by a given set of definitions  $\Delta$ .

MSO<sub>2</sub> is strictly more expressive than GL without edge label quantification [5]. Little is known for GL with recursion, besides that it can express PSPACE-complete model checking problems [23], whereas model checking is PSPACE-complete for MSO [47].

The separating conjunction used in SLR has been first introduced in role logic [38], a logic designed to reason about properties of record fields in object-oriented programs. This logic uses separating conjunction in combination with boolean connectives and first-order quantifier (ranging over vertices) and has no recursive constructs (least fix-points or inductive definitions). A bothways translation between role logic and SO has been described in [39]. These translations rely on boolean connectives and first-order quantifiers, instead of least fixpoint recursion, which is the case in our work.

## 2 Definitions

For a set  $A$ , we denote by  $\text{pow}(A)$  its powerset and  $A^1 \stackrel{\text{def}}{=} A$ ,  $A^{i+1} \stackrel{\text{def}}{=} A^i \times A$ , for all  $i \geq 1$ , where  $\times$  is the Cartesian product, and  $A^+ \stackrel{\text{def}}{=} \bigcup_{i \geq 1} A^i$ . The cardinality of a finite set  $A$  is denoted by  $\|A\|$ . Given integers  $i$  and  $j$ , we write  $[i, j]$  for the set  $\{i, i+1, \dots, j\}$ , assumed to be empty if  $i > j$ . For a function  $f : A \rightarrow B$  and a subset  $S \subseteq A$ , we denote by  $\text{dom}(f) = A$  the domain of  $f$  and by  $f|_S$  the restriction of  $f$  to  $S$ . A function  $f$  is said to be *effectively computable* if there exists a Turing machine  $\mathcal{M}$ , such that, for any  $a \in \text{dom}(f)$ ,  $\mathcal{M}$  computes  $f(a)$  in a finite number of steps.

**Signatures and Structures** Let  $\Sigma = \{R_1, \dots, R_N, c_1, \dots, c_M\}$  be a finite *signature*, where  $R_i$  are relation symbols of arity  $\#R_i \geq 1$  and  $c_j$  are constant symbols i.e., function symbols of arity zero. Additionally, we assume the existence of a unary relation symbol  $\mathcal{D}$ , not in  $\Sigma$ . Unless stated otherwise, we consider  $\Sigma$  and  $\mathcal{D}$  to be fixed in the following.

A *structure* is a pair  $(U, \sigma)$ , where  $U$  is a non-empty set, called *universe*, and  $\sigma : \Sigma \rightarrow U \cup \text{pow}(U^+)$  is a function that maps each relation symbol  $R$  to a relation  $\sigma(R) \subseteq U^{\#R}$  and each constant  $c$  to an element  $\sigma(c)$  of the universe. We denote by  $\text{Str}(\Sigma)$  (resp.  $\text{Str}(\Sigma, \mathcal{D})$ ) the set of structures over the signature  $\Sigma$  (resp.  $\Sigma \cup \{\mathcal{D}\}$ ). To give the statements of the results in this paper, we introduce the following notions:

$$\begin{aligned} \text{Rel}(\sigma) &\stackrel{\text{def}}{=} \{u_k \mid \langle u_1, \dots, u_{\#R_i} \rangle \in \sigma(R_i), i \in [1, N], k \in [1, \#R_i]\} \\ \text{Dom}(\sigma) &\stackrel{\text{def}}{=} \text{Rel}(\sigma) \cup \{\sigma(c_1), \dots, \sigma(c_M)\} \end{aligned}$$

The structure  $(U, \sigma)$  is *locally finite* if  $\text{Dom}(\sigma)$  is finite (i.e., the interpretation of each relation symbol from the signature is finite) and *finite* if  $U$  is finite. In this paper, we consider only locally finite structures but allow the universe  $U$  to be infinite. Another important class are the *guarded* structures, defined below:

**Definition 1.** A structure  $(U, \sigma) \in \text{Str}(\Sigma, \mathcal{D})$  is *guarded* if and only if  $\text{Rel}(\sigma|_{\Sigma}) \subseteq \sigma(\mathcal{D})$ .

The *spatial composition* (composition for short) of structures is defined below:

**Definition 2.** Two structures  $(U_1, \sigma_1)$  and  $(U_2, \sigma_2)$  are said to be:

1. *compatible* if and only if  $U_1 = U_2$  and  $\sigma_1(c_j) = \sigma_2(c_j)$ , for all  $j \in [1, M]$ , and
2. *disjoint* if and only if  $\sigma_1(R_i) \cap \sigma_2(R_i) = \emptyset$ , for all  $i \in [1, N]$ .

The composition of two compatible and disjoint structures is the structure  $(U_1, \sigma_1 \bullet \sigma_2)$ , where  $(\sigma_1 \bullet \sigma_2)(R_i) = \sigma_1(R_i) \cup \sigma_2(R_i)$  and  $(\sigma_1 \bullet \sigma_2)(c_j) = \sigma_1(c_j) = \sigma_2(c_j)$ , for all  $i \in [1, N]$  and  $j \in [1, M]$ , undefined for structures that are not compatible or not disjoint.

In the following, we identify structures that differ by a renaming of elements from the universe and consider only classes of structures that are closed under isomorphism:

**Definition 3.** Two structures  $(U_1, \sigma_1)$  and  $(U_2, \sigma_2)$  are isomorphic, denoted  $(U_1, \sigma_1) \simeq (U_2, \sigma_2)$  if and only if there exists a bijection  $h : U_1 \rightarrow U_2$ , such that:

1. for each relation symbol  $R \in \Sigma$  and each tuple  $\langle u_1, \dots, u_{\#R} \rangle \in U_1^{\#R}$ , we have:

$$\langle u_1, \dots, u_{\#R} \rangle \in \sigma_1(R) \iff \langle h(u_1), \dots, h(u_{\#R}) \rangle \in \sigma_2(R)$$

2. for each constant symbol  $c \in \Sigma$ , we have  $h(\sigma_1(c)) = \sigma_2(c)$ .

**Graphs and Treewidth** A (finite) graph is a pair  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , such that  $\mathcal{V}$  is a (finite) set of vertices and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is a set of edges. All graphs considered in this paper are directed i.e.,  $\mathcal{E}$  is not a symmetric relation. Graphs are naturally encoded as structures:

**Definition 4.** Each graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is encoded by the structure  $(\mathcal{V}, \sigma_{\mathcal{G}})$  over the signature  $\Gamma \stackrel{\text{def}}{=} \{\mathcal{V}, \mathcal{E}\}$ , where  $\#\mathcal{V} = 1$  and  $\#\mathcal{E} = 2$ , such that  $\sigma_{\mathcal{G}}(\mathcal{V}) = \mathcal{V}$  and  $\sigma_{\mathcal{G}}(\mathcal{E}) = \mathcal{E}$ .

A path in  $\mathcal{G}$  is a sequence of pairwise distinct vertices  $v_1, \dots, v_n$ , such that  $(v_i, v_{i+1}) \in \mathcal{E}$ , for all  $i \in [1, n-1]$ . We say that  $v_1, \dots, v_n$  is an undirected path if  $\{(v_i, v_{i+1}), (v_{i+1}, v_i)\} \cap \mathcal{E} \neq \emptyset$  instead, for all  $i \in [1, n-1]$ . A set of vertices  $V \subseteq \mathcal{V}$  is connected in  $\mathcal{G}$  if and only if there is an undirected path in  $\mathcal{G}$  between any two vertices in  $V$ . A graph  $\mathcal{G}$  is connected if and only if  $\mathcal{V}$  is connected in  $\mathcal{G}$ . A clique is a graph such that each two distinct nodes are the endpoints of an edge, the direction of which is not important. We denote by  $\mathcal{K}_n$  the set of cliques with  $n$  vertices.

Given a set  $\Lambda$  of labels, a  $\Lambda$ -labeled tree is a tuple  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$ , where  $(\mathcal{N}, \mathcal{F})$  is a graph,  $r \in \mathcal{N}$  is a designated vertex called the root, such that there exists a path in  $(\mathcal{N}, \mathcal{F})$  from  $r$  to any other vertex  $v \in \mathcal{N} \setminus \{r\}$ ,  $r$  has no incoming edge  $(p, r) \in \mathcal{F}$  and no vertex  $n$  has two incoming edges  $(m, n), (p, n) \in \mathcal{F}$ , for  $m \neq p$ . The mapping  $\lambda : \mathcal{N} \rightarrow \Lambda$  associates each vertex of the tree a label from  $\Lambda$ .

**Definition 5.** A tree decomposition of a structure  $(U, \sigma)$  over the signature  $\Sigma$  is a  $\text{pow}(U)$ -labeled tree  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$ , such that the following hold:

1. for each relation symbol  $R \in \Sigma$  and each tuple  $\langle u_1, \dots, u_{\#R} \rangle \in \sigma(R)$  there exists  $n \in \mathcal{N}$ , such that  $\{u_1, \dots, u_{\#R}\} \subseteq \lambda(n)$ , and
2. for each  $u \in \text{Dom}(\sigma)$ , the set  $\{n \in \mathcal{N} \mid u \in \lambda(n)\} \neq \emptyset$  is connected in  $(\mathcal{N}, \mathcal{F})$ .

The width of the tree decomposition is  $\text{tw}(\mathcal{T}) \stackrel{\text{def}}{=} \max_{n \in \mathcal{N}} \|\lambda(n)\| - 1$ . The treewidth of the structure  $\sigma$  is  $\text{tw}(\sigma) \stackrel{\text{def}}{=} \min\{\text{tw}(\mathcal{T}) \mid \mathcal{T} \text{ is a tree decomposition of } \sigma\}$ .

It is clear that two isomorphic structures have the same treewidth. In the following, we consider set of structures that are closed under isomorphism. Such a set is treewidth-bounded if and only if the set of corresponding treewidths is finite and treewidth-unbounded otherwise. A set is strictly treewidth-unbounded if and only if it is treewidth-unbounded and any infinite subset is treewidth-unbounded. The following result can be found in several textbooks (see e.g., [22]) and is restated here for self-containment:

**Proposition 1.** The set  $\{\mathcal{K}_n \mid n \in \mathbb{N}\}$  is strictly treewidth-unbounded.

### 3 Logics

We introduce two logics for reasoning about structures over a (fixed) relational signature  $\Sigma = \{R_1, \dots, R_N, c_1, \dots, c_M\}$ . The first such logic, called the *Separation Logic of Relations* (SLR), uses a set of *first-order variables*  $\mathbb{V}^{(1)} = \{x, y, \dots\}$  and a set of *predicates*  $\mathbb{A} = \{A, B, \dots\}$  of arities  $\#A, \#B$ , etc. We use the symbols  $\xi, \chi \in \mathbb{V}^{(1)} \cup \{c_1, \dots, c_M\}$  to denote *terms* i.e., either first-order variables or constants. The formulæ of SLR are defined by the following syntax:

$$\phi := \xi = \chi \mid \xi \neq \chi \mid R(\xi_1, \dots, \xi_{\#R}) \mid A(\xi_1, \dots, \xi_{\#A}) \mid \phi * \psi \mid \exists x . \phi$$

The formulæ  $\xi = \chi$  and  $\xi \neq \chi$  are called *equalities* and *disequalities*,  $R(\xi_1, \dots, \xi_{\#R})$  and  $A(\xi_1, \dots, \xi_{\#A})$  are called *relation* and *predicate atoms*, respectively. By *atom* we mean any of the above atomic formulæ. A formula with no occurrences of predicate atoms (resp. existential quantifiers) is called *predicate-free* (resp. *quantifier-free*). A variable is *free* if it does not occur within the scope of an existential quantifier and let  $\text{fv}(\phi)$  be the set of free variables of  $\phi$ . A *sentence* is a formula with no free variables. A *substitution*  $\phi[x_1/\xi_1 \dots x_n/\xi_n]$  replaces simultaneously every occurrence of the free variable  $x_i$  by the term  $\xi_i$  in  $\phi$ , for all  $i \in [1, n]$ .

The predicates from  $\mathbb{A}$  are interpreted as sets of structures, defined inductively by a finite set of rules. Before defining the semantics of SLR formulæ, we introduce the rules that assign meaning to predicates:

**Definition 6.** A set of inductive definitions (SID)  $\Delta$  is a finite set of rules of the form  $A(x_1, \dots, x_{\#A}) \leftarrow \phi$ , where  $x_1, \dots, x_{\#A}$  are pairwise distinct variables, called *parameters*, such that  $\text{fv}(\phi) \subseteq \{x_1, \dots, x_{\#A}\}$ . An atom  $\alpha$  occurs in a rule  $A(x_1, \dots, x_{\#A}) \leftarrow \phi$  if and only if it occurs in  $\phi$ .

Note that having distinct parameters in a rule is without loss of generality, as e.g., a rule  $A(x_1, x_1) \leftarrow \phi$  can be equivalently written as  $A(x_1, x_2) \leftarrow x_1 = x_2 * \phi$ . As a convention, we shall always use the names  $x_1, \dots, x_{\#A}$  for the parameters of a rule that defines  $A$ .

The semantics of SLR formulæ is given by the satisfaction relation  $(U, \sigma) \models_{\Delta}^v \phi$  between structures and formulæ. This relation is parameterized by a *store*  $v : \mathbb{V}^{(1)} \rightarrow U$  mapping the free variables of a formula into elements of the universe and an SID  $\Delta$ . We write  $v[x \leftarrow u]$  for the store that maps  $x$  into  $u$  and agrees with  $v$  on all variables other than  $x$ . For a term  $\xi$ , we denote by  $(\sigma, v)(\xi)$  the value  $\sigma(\xi)$  if  $\xi$  is a constant, or  $v(\xi)$  if  $\xi$  is a first-order variable. The satisfaction relation is the least relation that satisfies the following conditions:

$$\begin{aligned} (U, \sigma) \models_{\Delta}^v \xi \bowtie \chi &\iff \sigma(R) = \emptyset, \text{ for all } R \in \Sigma \text{ and} \\ &\quad (\sigma, v)(\xi) \bowtie (\sigma, v)(\chi), \text{ for all } \bowtie \in \{=, \neq\} \\ (U, \sigma) \models_{\Delta}^v R(\xi_1, \dots, \xi_{\#R}) &\iff \sigma(R) = \{ \langle (\sigma, v)(\xi_1), \dots, (\sigma, v)(\xi_{\#R}) \rangle \} \\ &\quad \text{and } \sigma(R') = \emptyset, \text{ for all } R' \in \Sigma \setminus \{R\} \\ (U, \sigma) \models_{\Delta}^v A(\xi_1, \dots, \xi_{\#A}) &\iff (U, \sigma) \models_{\Delta}^v \phi[x_1/\xi_1, \dots, x_{\#A}/\xi_{\#A}], \text{ for some rule} \\ &\quad A(x_1, \dots, x_{\#A}) \leftarrow \phi \text{ from } \Delta \\ (U, \sigma) \models_{\Delta}^v \phi_1 * \phi_2 &\iff \text{there exist structures } (U, \sigma_1) \text{ and } (U, \sigma_2), \text{ such that} \\ &\quad \sigma = \sigma_1 \bullet \sigma_2 \text{ and } (U, \sigma_i) \models_{\Delta}^v \phi_i, \text{ for all } i = 1, 2 \\ (U, \sigma) \models_{\Delta}^v \exists x . \phi &\iff (U, \sigma) \models_{\Delta}^{v[x \leftarrow u]} \phi, \text{ for some } u \in U \end{aligned}$$

Equivalently, the satisfaction relation can be defined by considering all *finite* unfoldings of the above rules, until no predicate atom is left. Note that this means that every structure  $(U, \sigma)$ , such that  $(U, \sigma) \models_{\Delta}^v \phi$ , for some formula  $\phi$ , is locally finite, since it is defined by a least fixpoint iteration over the rules in  $\Delta$ .

If  $\phi$  is a sentence, the satisfaction relation does not depend on the store, in which case  $(U, \sigma)$  is a  $\Delta$ -model of  $\phi$ , written  $(U, \sigma) \models_{\Delta} \phi$ . We define  $\llbracket \phi | \Delta \rrbracket \stackrel{\text{def}}{=} \{(U, \sigma) \mid (U, \sigma) \models_{\Delta} \phi\}$  the set of  $\Delta$ -models of  $\phi$ . By  $\llbracket \phi | \Delta \rrbracket_{\mathfrak{D}}^k$  we denote the set of guarded structures of treewidth at most  $k$  from  $\llbracket \phi | \Delta \rrbracket$ . We write  $\llbracket \text{SLR} \rrbracket \stackrel{\text{def}}{=} \{\llbracket \phi | \Delta \rrbracket \mid \phi \text{ is a SLR formula, } \Delta \text{ is a SID}\}$  and  $\llbracket \text{SLR} \rrbracket_{\mathfrak{D}}^k \stackrel{\text{def}}{=} \{\llbracket \phi | \Delta \rrbracket_{\mathfrak{D}}^k \mid \phi \text{ is a formula of SLR, } \Delta \text{ is a SID}\}$ . A set is *SLR-defined* if it is of the form  $\llbracket \phi | \Delta \rrbracket$ , for a formula  $\phi$  and a SID  $\Delta$ . By the result below, every set of SLR-defined structures is a union of equivalence classes of isomorphism (Def. 3):

**Proposition 2.** *Given structures  $(U, \sigma) \simeq (U', \sigma')$ , for any sentence  $\phi$  of SLR and any SID  $\Delta$ , we have  $(U, \sigma) \models_{\Delta} \phi \iff (U', \sigma') \models_{\Delta} \phi$ .*

The other logic considered in this paper is *Weak Second Order Logic* (SO) defined using a set of *second-order variables*  $\mathbb{V}^{(2)} = \{X, Y, \dots\}$ , in addition to first-order variables. As usual, we denote by  $\#X$  the arity of a second-order variable  $X$ . Terms and atoms are defined in the same way as in SLR. The formulae of SO are defined by the following syntax:

$$\psi := \xi = \chi \mid R(\xi_1, \dots, \xi_{\#R}) \mid X(\xi_1, \dots, \xi_{\#X}) \mid \neg\psi \mid \psi \wedge \psi \mid \exists x . \psi \mid \exists X . \psi$$

As usual, we write  $\xi \neq \chi \stackrel{\text{def}}{=} \neg\xi = \chi$ ,  $\psi_1 \vee \psi_2 \stackrel{\text{def}}{=} \neg(\neg\psi_1 \wedge \neg\psi_2)$ ,  $\psi_1 \rightarrow \psi_2 \stackrel{\text{def}}{=} \neg\psi_1 \vee \psi_2$ ,  $\forall x . \psi \stackrel{\text{def}}{=} \neg\exists x . \neg\psi$  and  $\forall X . \psi \stackrel{\text{def}}{=} \neg\exists X . \neg\psi$ . The Weak Monadic Second Order Logic (MSO) is the fragment of SO restricted to second-order variables of arity one only.

The semantics of SO is given by a relation  $(U, \sigma) \models^v \psi$ , where the store  $v : \mathbb{V}^{(1)} \cup \mathbb{V}^{(2)} \rightarrow U \cup \text{pow}(U^+)$  maps each first-order variable  $x \in \mathbb{V}^{(1)}$  to an element of the universe  $v(x) \in U$  and each second-order variable  $X \in \mathbb{V}^{(2)}$  to a finite relation  $v(X) \subseteq U^{\#X}$ . The satisfaction relation of SO is defined inductively on the structure of formulae:

$$\begin{aligned} (U, \sigma) \models^v \xi = \chi &\iff (\sigma, v)(\xi) = (\sigma, v)(\chi) \\ (U, \sigma) \models^v R(\xi_1, \dots, \xi_{\#R}) &\iff \langle (\sigma, v)(\xi_1), \dots, (\sigma, v)(\xi_{\#R}) \rangle \in \sigma(R) \\ (U, \sigma) \models^v X(\xi_1, \dots, \xi_{\#X}) &\iff \langle (\sigma, v)(\xi_1), \dots, (\sigma, v)(\xi_{\#X}) \rangle \in v(X) \\ (U, \sigma) \models^v \exists X . \psi &\iff (U, \sigma) \models^{v[X \leftarrow V]} \psi, \text{ for some finite set } V \subseteq U^{\#X} \end{aligned}$$

The semantics of negation, conjunction and first-order quantification is standard and omitted for brevity. Note the difference between (dis-)equalities and relation atoms in SLR and SO; in the former, (dis-)equality (relation) atoms hold in an empty (singleton) structure, whereas no cardinality upper bounds occur in SO, in these particular cases.

If  $\phi$  is a sentence, we write  $(U, \sigma) \models \phi$  instead of  $(U, \sigma) \models^v \phi$  and define  $\llbracket \phi \rrbracket \stackrel{\text{def}}{=} \{(U, \sigma) \mid (U, \sigma) \models \phi\}$  and  $\llbracket \phi \rrbracket_{\mathfrak{D}}^k$  for the restriction of  $\llbracket \phi \rrbracket$  to guarded structures of treewidth at most  $k$ . We write  $\llbracket (\text{M})\text{SO} \rrbracket \stackrel{\text{def}}{=} \{\llbracket \phi \rrbracket \mid \phi \text{ is a (M)SO formula}\}$  and  $\llbracket (\text{M})\text{SO} \rrbracket_{\mathfrak{D}}^k \stackrel{\text{def}}{=} \{\llbracket \phi \rrbracket_{\mathfrak{D}}^k \mid \phi \text{ is a (M)SO formula}\}$ . A set is *(M)SO-defined* if it is of the form  $\llbracket \phi \rrbracket$ , for a formula  $\phi$  of (M)SO. It is well-known (see e.g., [27]) that SO sentences cannot distinguish between isomorphic structures (Def. 3).

The aim of this paper is comparing the expressive powers of SLR, MSO and SO, with respect to the properties that can be defined in these logics. To ease the upcoming technical developments, we consider only locally finite structures over a fixed infinite universe  $\mathbb{U}$ . This is because, in general, MSO and SO can express statements related to an upper bound on the cardinality of the universe (i.e., there are most  $n$  elements in the universe), whereas SLR can only express lower bounds. To avoid such technical discrepancies, from now on we consider that the universe is fixed and infinite, in close analogy with SL, where the set of memory locations is typically infinite<sup>9</sup>. Further, from now on, the notation is simplified by writing  $\sigma$  instead of  $(\mathbb{U}, \sigma)$ .

In this paper, we are concerned with the problems  $[[\mathcal{L}_1]] \subseteq [[\mathcal{L}_2]]$  and  $[[\mathcal{L}_1]]_{\mathfrak{D}}^k \subseteq [[\mathcal{L}_2]]$ , where  $\mathcal{L}_1$  and  $\mathcal{L}_2$  are any of the logics SLR, MSO and SO, respectively. In particular, for  $[[\mathcal{L}_1]]_{\mathfrak{D}}^k \subseteq [[\mathcal{L}_2]]$ , we implicitly assume that  $\mathcal{L}_1$  and  $\mathcal{L}_2$  are sets of formulæ over the relational signature  $\Sigma \cup \{\mathfrak{D}\}$ . Table 1 summarizes our results, with references to the sections in the paper where the proofs (for the non-trivial ones) can be found, and the remaining open problems.

#### 4 $[[\text{SLR}]]_{\mathfrak{D}}^k \not\subseteq [[\text{MSO}]]$

We exhibit an SLR-definable set of structures of treewidth one that is not MSO-definable, implying also that  $[[\text{SLR}]] \not\subseteq [[\text{MSO}]]$ . The idea is to encode words over a binary alphabet  $\mathcal{A} = \{a, b\}$  by lists whose vertices are labeled with symbols from  $\mathcal{A}$ . A list is connected acyclic graph, in which each vertex has at most one incoming and at most one outgoing edge. A word  $w = a_1 \dots a_n$  is encoded by the guarded structure  $\sigma_w \in \text{Str}(\Sigma, \mathfrak{D})$ , where  $\Sigma = \{E, P_a, P_b, b, e\}$ ,  $\sigma_w(\mathfrak{D}) = \{v_1, \dots, v_n\}$ ,  $\sigma_w(E) = \{(v_i, v_{i+1}) \mid i \in [1, n-1]\}$ ,  $\sigma_w(P_\alpha) = \{v_i \mid i \in [1, n], a_i = \alpha\}$ , for all  $\alpha \in \mathcal{A}$ ,  $\sigma_w(b) = v_1$  and  $\sigma_w(e) = v_n$ . Note that  $\text{tw}(\sigma_w) = 1$ , for any word  $w \in \mathcal{A}^*$ .

A *context-free grammar*  $G = (N, \mathcal{X}, \Delta)$  consists of a finite set  $N$  of *nonterminals*, a start symbol  $\mathcal{X} \in N$  and a finite set  $\Delta$  of *productions* of the form  $\mathcal{Y} \rightarrow w$ , where  $\mathcal{Y} \in N$  and  $w \in (N \cup \mathcal{A})^*$ . Given finite strings  $u, v \in (N \cup \mathcal{A})^*$ , the relation  $u \triangleright v$  replaces a nonterminal  $\mathcal{Y}$  of  $u$  by the right-hand side  $w$  of a production  $\mathcal{Y} \rightarrow w$  and  $\triangleright^*$  denotes the reflexive and transitive closure of  $\triangleright$ . The *language* of  $G$  is the set  $\mathcal{L}(G) \stackrel{\text{def}}{=} \{w \in \mathcal{A}^* \mid \mathcal{X} \triangleright^* w\}$ . A language  $L$  is *context-free* if and only if there exists a context-free grammar  $G$ , such that  $L = \mathcal{L}(G)$ .

A language  $L$  is *recognizable* if and only if the equivalence relation  $u \simeq_L v$ , defined as  $uw \in L \iff vw \in L$ , for all  $w \in \mathcal{A}^*$ , has finitely many equivalence classes. A language is recognizable if and only if it is definable by an MSO formula over the signature  $\{p_a, p_b, s\}$ , where  $p_\alpha$  is the set of positions from  $[1, n]$  labeled by  $\alpha \in \mathcal{A}$  and  $s(x, y) \iff y = (x \bmod n) + 1$  [13]. Every recognizable language is context-free but not viceversa e.g.,  $\{a^n b^n \mid n \in \mathbb{N}\}$  is context-free but not recognizable.

**Proposition 3.** *Given a context-free grammar  $G = (N, \mathcal{X}, \Delta)$ , there exists an SID  $\Delta_G$  and a binary predicate symbol  $A_{\mathcal{X}} \in \text{def}(\Delta_G)$ , such that  $w \in \mathcal{L}(G) \iff \sigma_w \models_{\Delta_G} A_{\mathcal{X}}(b, e)$ , for all  $w \in \mathcal{A}^*$ .*

<sup>9</sup> An exception is the finite version of SL with universal quantification and magic wand, but without inductive definitions from [28].

Let  $L$  be a non-recognizable context-free language and suppose that there exists an MSO formula  $\phi_L$ , over the signature  $\{E, P_a, P_b, b, e, \mathcal{D}\}$ , that defines the set  $\mathcal{S}_L \stackrel{\text{def}}{=} \{\sigma_w \mid w \in L\}$ . Then there exists also an MSO formula  $\psi_L$  over the signature  $\{p_a, p_b, s\}$ , interpreted over finite words, that defines  $L$ . Note that  $\psi_L$  can be obtained directly from  $\phi_L$ , by replacing each atom  $E(\xi, \chi)$  by  $s(\xi) = \chi$  and each atom  $P_\alpha(\xi)$  by  $p_\alpha(\xi)$ . Then the language  $L$  is recognizable, which contradicts with the choice of  $L$ . Moreover, we have  $\{\text{tw}(\sigma) \mid \sigma \in \mathcal{S}_L\} = \{1\}$ , proving that  $\llbracket \text{SLR} \rrbracket_{\mathcal{D}}^k \not\subseteq \llbracket \text{MSO} \rrbracket$ , for any given  $k \geq 1$ .

## 5 $\llbracket \text{MSO} \rrbracket \not\subseteq \llbracket \text{SLR} \rrbracket$

We prove that the MSO-definable set of cliques  $\{\mathcal{K}_n \mid n \in \mathbb{N}\}$  is not SLR-definable. Because MSO is a syntactic fragment of SO, this also implies  $\llbracket \text{SO} \rrbracket \not\subseteq \llbracket \text{SLR} \rrbracket$ . First, observe that the set  $\{\mathcal{K}_n \mid n \in \mathbb{N}\} \subseteq \text{Str}(\{V, E\})$  is defined by the formula:

$$\forall x \forall y . V(x) \wedge V(y) \wedge x \neq y \rightarrow E(x, y) \vee E(y, x)$$

Note that the quantifiers range over finite (singleton) sets, hence the above is a formula of Weak MSO. As shown in Proposition 1, the set  $\{\mathcal{K}_n \mid n \in \mathbb{N}\}$  is strictly treewidth-unbounded. It is then sufficient to prove that SLR cannot define strictly treewidth-unbounded sets. More precisely, for each SLR sentence  $\phi$  and SID  $\Delta$ , we prove the existence of an integer  $W \geq 1$ , such that, for each structure  $\sigma \in \llbracket \phi \mid \Delta \rrbracket$  there exists another structure  $\bar{\sigma} \in \llbracket \phi \mid \Delta \rrbracket$ , with  $\text{tw}(\bar{\sigma}) \leq W$ . Then each infinite set  $\llbracket \phi \mid \Delta \rrbracket$  has an infinite treewidth-bounded subset i.e., it is not strictly treewidth-unbounded (Proposition 4).

A first ingredient of the proof is that each SID can be transformed into an equivalent SID in which the variables that occur existentially quantified in the rules of  $\Delta$  are not constrained by equalities.

**Definition 7.** A rule  $A(x_1, \dots, x_{\#A}) \leftarrow \exists y_1 \dots \exists y_n . \psi$ , where  $\psi$  is a quantifier-free formula, is normalized if and only if no equality atom  $x = y$  occurs in  $\psi$ , for distinct variables  $x, y \in \{x_1, \dots, x_{\#A}\} \cup \{y_1, \dots, y_n\}$ . An SID is normalized if and only if it consists of normalized rules.

**Lemma 1.** Given an SID  $\Delta$ , one can build a normalized SID  $\Delta'$  such that  $\text{def}(\Delta) \subseteq \text{def}(\Delta')$  and for each structure  $\sigma$  and each predicate atom  $A(\xi_1, \dots, \xi_{\#A})$ , we have  $\sigma \models_{\Delta} \exists \xi_{i_1} \dots \exists \xi_{i_n} . A(\xi_1, \dots, \xi_{\#A}) \iff \sigma \models_{\Delta'} \exists \xi_{i_1} \dots \exists \xi_{i_n} . A(\xi_1, \dots, \xi_{\#A})$ , where  $\{\xi_{i_1}, \dots, \xi_{i_n}\} = \{\xi_1, \dots, \xi_{\#A}\} \cap \mathbb{V}^{(1)}$ .

A consequence of the above result is that, in the absence of equality constraints in a normalized SID, each existentially quantified variable instantiated by the inductive definition of the satisfaction relation can be assigned a distinct element of the universe. For instance, the  $\text{fold\_ls}(x, y)$  formula (§1) defines an infinite set of graphs whose edges are given by the interpretation of a relation symbol  $H$ , such that there exists an Eulerian path visiting all edges exactly once, and all vertices possibly more than once. Since there are no equality constraints (except for the case of empty graphs), each model of  $\text{fold\_ls}(x, y)$  can be expanded into an acyclic list from  $x$  to  $y$ , that never visits the same

vertex twice, except at the endpoints. This latter graph has treewidth two, if the vertices denoted by  $x$  and  $y$  coincide, and one otherwise.

Formally, we write  $\sigma \models_{\Delta}^{\nu \downarrow \mathcal{U}} \phi$  if and only if the satisfaction relation  $\sigma \models_{\Delta}^{\nu} \phi$  can be established by considering stores that are injective over the interpretation of existentially quantified variables and map these variables into elements from an infinite set  $\mathcal{U} \subseteq \mathbb{U}$ . More precisely, the inductive definition of  $\models_{\Delta}^{\nu \downarrow \mathcal{U}}$  is the same as the definition of  $\models_{\Delta}$  (§3), except for the following cases:

$$\begin{aligned} \sigma \models_{\Delta}^{\nu \downarrow \mathcal{U}} \phi_1 * \phi_2 &\iff \text{there exist structures } \sigma_1 \bullet \sigma_2 = \sigma \text{ and infinite sets } \mathcal{U}_1 \text{ and } \mathcal{U}_2, \\ &\quad \mathcal{U}_1 \cap \mathcal{U}_2 = \emptyset, \mathcal{U}_1 \cup \mathcal{U}_2 \subseteq \mathcal{U}, \text{ such that } \sigma_i \models_{\Delta}^{\nu \downarrow \mathcal{U}_i} \phi_i, i = 1, 2 \\ \sigma \models_{\Delta}^{\nu \downarrow \mathcal{U}} \exists x . \phi &\iff \sigma \models_{\Delta}^{\nu[x \leftarrow u] \downarrow \mathcal{U}} \phi, \text{ for some } u \in \mathcal{U} \setminus \nu(\text{fv}(\phi) \setminus \{x\}) \end{aligned}$$

**Lemma 2.** *Given a normalized SID  $\Delta$ , a predicate atom  $A(\xi_1, \dots, \xi_{\#A})$  and an infinite set  $\mathcal{U} \subseteq \mathbb{U}$ , for each structure  $\sigma$  and a store  $\nu$ , such that  $\sigma \models_{\Delta}^{\nu} A(\xi_1, \dots, \xi_{\#A})$ , there exists a structure  $\sigma'$ , such that  $\sigma' \models_{\Delta}^{\nu \downarrow \mathcal{U}} A(\xi_1, \dots, \xi_{\#A})$  and  $\|\text{Dom}(\sigma)\| \leq \|\text{Dom}(\sigma')\|$ .*

We show that the models defined on injective stores have bounded treewidth:

**Lemma 3.** *Given a normalized SID  $\Delta$  and a predicate atom  $A(\xi_1, \dots, \xi_{\#A})$ , we have  $\text{tw}(\sigma) \leq W$ , for each structure  $\sigma$  and store  $\nu$ , such that  $\sigma \models_{\Delta}^{\nu \downarrow \mathcal{U}} A(\xi_1, \dots, \xi_{\#A})$ , for some infinite set  $\mathcal{U} \subseteq \mathbb{U}$ , where  $W \geq 1$  is a constant depending only on  $\Delta$ .*

The proof of  $\llbracket \text{MSO} \rrbracket \not\subseteq \llbracket \text{SLR} \rrbracket$  relies on the following result:

**Proposition 4.** *Given a SLR sentence  $\phi$  and an SID  $\Delta$ , the set  $\llbracket \phi | \Delta \rrbracket$  is either finite or it has an infinite subset of bounded treewidth.*

## 6 $\llbracket \text{SLR} \rrbracket \subseteq \llbracket \text{SO} \rrbracket$

Since SLR and MSO define incomparable treewidth-unbounded sets of structures, it is natural to ask for a logic that subsumes them both. In this section, we prove that SO is such a logic. Since MSO is a syntactic subset of SO, we have  $\llbracket \text{MSO} \rrbracket \subseteq \llbracket \text{SO} \rrbracket$  trivially.

In the rest of this section, we show that  $\llbracket \text{SLR} \rrbracket \subseteq \llbracket \text{SO} \rrbracket$  using the fact that each model of a predicate atom in SLR is built according to a *finite unfolding tree* indicating the partial order in which the rules of the SID are used in the inductive definition of the satisfaction relation<sup>10</sup>. More precisely, the model can be decomposed into pairwise disjoint substructures, each being the model of the quantifier- and predicate-free subformula of a rule in the SID, such that there is a one-to-one mapping between the nodes of the tree and the decomposition of the model. We use second-order variables, interpreted as finite relations, to define the unfolding tree and the mapping between the nodes of the unfolding tree and the tuples in the interpretation of the relation symbols from the model. These second-order variables are existentially quantified and the resulting SO formula describes the model alone, without the unfolding tree that witnesses its decomposition according to the rules of the SID.

<sup>10</sup> Unfolding trees are for SIDs what derivation trees are for context-free grammars.

In the following, let  $\Sigma \stackrel{\text{def}}{=} \{R_1, \dots, R_N, c_1, \dots, c_M\}$  be the signature of SLR and let  $\Delta \stackrel{\text{def}}{=} \{r_1, \dots, r_R\}$  be a given SID. Without loss of generality, for each relation symbol  $R_i \in \Sigma$ , we assume that there is at most one occurrence of an atom  $R_i(y_1, \dots, y_{\#R_i})$  in each rule from  $\Delta$ . If this is not the case, we split the rule by introducing a new predicate symbol for each relation atom with relation symbol  $R_i$ , until the condition is satisfied.

We define unfolding trees formally. For a tree  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  and a vertex  $n \in \mathcal{N}$ , we denote by  $\mathcal{T}[n]$  the subtree of  $\mathcal{T}$  whose root is  $n$ . For a quantifier- and predicate-free formula  $\phi$ , we denote by  $\phi^n$  the formula in which every relation atom  $R(x_1, \dots, x_{\#R})$  is annotated as  $R^n(x_1, \dots, x_{\#R})$ . Atoms  $R^n(x_1, \dots, x_{\#R})$  (and consequently formulas  $\phi^n$ ) have the same semantics as atoms  $R(x_1, \dots, x_{\#R})$  (resp. formulas  $\phi$ ); these annotations serve purely as explanatory devices in our construction (Proposition 5) that keep track of the node of the unfolding tree where a relation atom was introduced.

**Definition 8.** An unfolding tree for a predicate atom  $A(\xi_1, \dots, \xi_{\#A})$  is a  $\Delta$ -labeled tree  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$ , such that  $\lambda(r) \in \text{def}_\Delta(A)$  and, for each vertex  $n \in \mathcal{N}$ , if  $B_1(z_{1,1}, \dots, z_{1,\#B_1}), \dots, B_h(z_{h,1}, \dots, z_{h,\#B_h})$  are the predicate atoms that occur in  $\lambda(n)$ , then  $p_1, \dots, p_h$  are the children of  $n$  in  $\mathcal{T}$ , such that  $\lambda(p_\ell) \in \text{def}_\Delta(B_\ell)$ , for all  $\ell \in [1, h]$ . An unfolding tree  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  for a predicate atom  $A(\xi_1, \dots, \xi_{\#A})$  gives rise to a predicate-free formula defined inductively as follows:

$$\Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A})) \stackrel{\text{def}}{=} \left( \exists y_1 \dots \exists y_m \cdot \psi^r * \bigstar_{\ell=1}^h \Theta(\mathcal{T}[p_\ell], B_\ell(z_{\ell,1}, \dots, z_{\ell,\#B_\ell})) \right) [x_1/\xi_1, \dots, x_{\#A}/\xi_{\#A}],$$

where  $\lambda(r) = A(x_1, \dots, x_{\#A}) \leftarrow \exists y_1 \dots \exists y_m \cdot \psi^r * \bigstar_{\ell=1}^h B_\ell(z_{\ell,1}, \dots, z_{\ell,\#B_\ell})$ , for a quantifier- and predicate-free formula  $\psi$ ,  $p_1, \dots, p_h$  are the children of the root  $r$  in  $\mathcal{T}$ , corresponding to the predicate atoms  $B_1(z_{1,1}, \dots, z_{1,\#B_1}), \dots, B_h(z_{h,1}, \dots, z_{h,\#B_h})$ , respectively.

The unfolding trees of a predicate atom describe the set of models of that predicate atom. The following lemma is standard and we include it for self-containment reasons:

**Lemma 4.** For any structure  $\sigma$  and store  $v$ ,  $\sigma \models_\Delta^v A(\xi_1, \dots, \xi_{\#A})$  iff there exists an unfolding tree  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  for  $A(\xi_1, \dots, \xi_{\#A})$ , such that  $\sigma \models^v \Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A}))$ .

Without loss of generality, we build a SO formula that defines the models of a relation atom  $A(\xi_1, \dots, \xi_{\#A})$ . This is because, for any formula  $\phi$  with  $\text{fv}(\phi) = \{x_1, \dots, x_n\}$ , one can add a new rule  $A_\phi(x_1, \dots, x_n) \leftarrow \phi$ , for a predicate symbol  $A_\phi$  not occurring in  $\Delta$ . Let  $P$  be the maximum number of occurrences of predicate atoms in a rule from  $\Delta$ . We use second-order variables  $Y_1, \dots, Y_P$  of arity 2, for the edges of the tree and  $X_0, \dots, X_R$  of arity 1, for the labels of the nodes in the tree i.e., the rules of  $\Delta$ . Then the SO formula  $\mathfrak{T}(x, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P)$  is defined as the conjunction of SO formulae that describe the following facts:

1. the root  $x$  belongs to  $X_i$ , for some  $r_i \in \text{def}_\Delta(A)$ ,
2. the sets  $X_1, \dots, X_R$  are pairwise disjoint,
3. each vertex in  $X_1 \cup \dots \cup X_R$  is reachable from  $x$  by a path with edges  $Y_1, \dots, Y_P$ ,
4. each vertex in  $X_1 \cup \dots \cup X_R$ , except for  $x$ , has exactly one incoming edge,
5.  $x$  has no incoming edge,

6. each vertex from  $X_i$  has exactly  $h$  outgoing edges  $Y_1, \dots, Y_h$ , each to a vertex from  $X_{j_\ell}$ , respectively, such that  $r_{j_\ell} \in \text{def}_\Delta(B_\ell)$ , for all  $\ell \in [1, h]$ , where  $B_1(z_{1,1}, \dots, z_{1, \#B_1}), \dots, B_h(z_{h,1}, \dots, z_{h, \#B_h})$  are the predicate atoms that occur in  $r_i$ .

We build a SO formula  $\mathfrak{F}(\xi_1, \dots, \xi_{\#A}, x, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P, \{\{Z_{k,\ell}\}_{\ell=1}^{\#R_k}\}_{k=1}^N)$  expressing the relationship between the unfolding tree  $\mathcal{T}$  and the relations  $\sigma(R_i)$ . We recall that, for every node  $n$  of  $\mathcal{T}$  and every relation atom  $R$ , there is at most one annotated relation atom  $R_k^n(\xi_1, \dots, \xi_{\#R_i})$  in  $\Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A}))$ . The formula  $\mathfrak{F}$  now uses second-order variables  $Z_{k,\ell}$ , of arity two that encode (partial) functions mapping a tree vertex  $n$  to the value of  $\xi_\ell$  for the (unique) annotated atom  $R_k^n(\xi_1, \dots, \xi_{\#R_i})$  (in case such an atom exists). The formula  $\mathfrak{F}$  is the conjunction of following SO-definable facts:

- (i) each second-order variable  $Z_{k,\ell}$  denotes a functional binary relation,  
(ii) for each tree node labeled by a rule  $r_i$  and each atom  $R_k(\xi_1, \dots, \xi_{\#R_k})$  occurring at that node, the interpretation of  $R_k$  contains a tuple, whose elements are related to the node via  $Z_{k,1}, \dots, Z_{k, \#R_k}$ :

$$\bigwedge_{i \in [1, R]} \bigwedge_{R_k \text{ occurs in } r_i} \forall y \cdot X_i(y) \rightarrow \exists z_1 \dots \exists z_{\#R_k} \cdot R_k(z_1, \dots, z_{\#R_k}) \wedge \bigwedge_{\ell \in [1, \#R_k]} Z_{k,\ell}(y, z_\ell)$$

- (iii) for any (not necessarily distinct) rules  $r_i$  and  $r_j$  such that an atom with relation symbol  $R_k$  occurs in both, the tuples (from the interpretation of  $R_k$ ) are distinct:

$$\bigwedge_{i, j \in [1, R]} \bigwedge_{R_k \text{ occurs in } r_i, r_j} \forall y \forall y' \forall z_1 \forall z'_1 \dots \forall z_{\#R_k} \forall z'_{\#R_k} \cdot \\ (X_i(y) \wedge X_j(y') \wedge \bigwedge_{\ell \in [1, \#R_k]} (Z_{k,\ell}(y, z_\ell) \wedge Z_{k,\ell}(y', z'_\ell))) \rightarrow \bigvee_{\ell \in [1, \#R_k]} z_\ell \neq z'_\ell$$

- (iv) each tuple from the interpretation of  $R_k$  must have been introduced by a relation atom with relation symbol  $R_k$  that occurs in a rule  $r_i$ :

$$\bigwedge_{k \in [1, N]} \forall z_1 \dots \forall z_{\#R_k} \cdot R_k(z_1, \dots, z_{\#R_k}) \rightarrow \exists y \cdot \bigvee_{R_k \text{ occurs in } r_i} (X_i(y) \wedge \bigwedge_{\ell \in [1, \#R_k]} Z_{k,\ell}(y, z_\ell))$$

- (v) two terms  $\xi_m$  and  $\chi_n$  that occur in two relation atoms  $R_k(\xi_1, \dots, \xi_{\#R_k})$  and  $R_\ell(\chi_1, \dots, \chi_{\#R_\ell})$  within rules  $r_i$  and  $r_j$ , respectively, and are constrained to be equal (i.e., via equalities and parameter passing), must be equated:

$$\bigwedge_{k, \ell \in [1, N]} \bigwedge_{R_k \text{ occurs in } r_i} \bigwedge_{R_\ell \text{ occurs in } r_j} \bigwedge_{m \in [1, \#R_k]} \bigwedge_{n \in [1, \#R_\ell]} \forall y \forall y' \forall z' \forall z'' \cdot \\ (X_i(y) \wedge X_j(y') \wedge \text{isEq}_{k,\ell,m,n}(y, y', \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P) \wedge Z_{k,m}(y, z) \wedge Z_{\ell,n}(y', z')) \rightarrow z = z''$$

The formula  $\text{isEq}_{k,\ell,m,n}(y, y', \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P)$  asserts that there is a path in the unfolding tree between the store values (i.e., vertices of the tree) of  $y$  and  $y'$ , such that the  $m$ -th and  $n$ -th variables of the relation atoms  $R_k(z_1, \dots, z_{\#R_k})$  and  $R_\ell(z'_1, \dots, z'_{\#R_\ell})$  are bound to the same value.

- (vi) a disequality  $\xi \neq \chi$  that occurs in a rule  $r_i$  is propagated throughout the tree to each pair of variables that occur within two relation atoms  $R_k(\xi_1, \dots, \xi_{\#R_k})$  and  $R_\ell(\chi_1, \dots, \chi_{\#R_\ell})$  in rules  $r_{j_k}$  and  $r_{j_\ell}$ , respectively, such that  $\xi$  is bound  $\xi_r$  and  $\chi$  to  $\chi_s$  by equality atoms and parameter passing:

$$\bigwedge_{\xi \neq \chi \text{ occurs in } r_i} \bigwedge_{k, \ell \in [1, N]} \bigwedge_{R_k \text{ occurs in } r_{j_k}} \bigwedge_{R_\ell \text{ occurs in } r_{j_\ell}} \bigwedge_{r \in [1, \#R_k]} \bigwedge_{s \in [1, \#R_\ell]} \forall y \forall y' \forall y'' \forall z' \forall z'' \cdot \\ (X_i(y) \wedge X_{j_k}(y') \wedge X_{j_\ell}(y'') \wedge Z_{k,r}(y', z') \wedge Z_{\ell,s}(y'', z'') \wedge \\ \text{varEq}_{\xi,k,r}(y, y', \{X_i\}_{i=1}^N, \{Y_j\}_{j=1}^M) \wedge \text{varEq}_{\chi,\ell,s}(y, y'', \{X_i\}_{i=1}^N)) \rightarrow z' \neq z''$$

The formula  $varEq_{\xi,k,r}(x,y, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P)$  states that the variable  $\xi$  occurring in the label of the unfolding tree vertex  $x$  is bound to the variable  $z_r$  that occurs in a relation atom  $R_k(z_1, \dots, z_{\#R_k})$  in the label of the vertex  $y$ .

- (vii) each term in  $A(\xi_1, \dots, \xi_{\#A})$  that is bound to a variable from a relation atom  $R_k(z_1, \dots, z_{\#R_k})$  in the unfolding, must be equated to that variable:

$$\bigwedge_{r_i \in \text{def}_\Delta(A)} \bigwedge_{j \in [1, \#A]} \forall y \forall z. (X_i(x) \wedge varEq_{\xi_j, k, r}(x, y, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P) \wedge Z_{k,r}(y, z)) \rightarrow \xi_j = z$$

The formulæ  $isEq_{k,\ell,m,n}(x,y, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P)$  and  $varEq_{\xi,k,r}(x,y, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P)$  above are definable in MSO, using standard tree automata construction techniques, similar to the definition of MSO formulæ that track parameters in an unfolding tree for SL, with edges definable by MSO formulæ over the signature of SL [33]. To avoid clutter, we defer such definitions to a long version of this paper.

Summing up, the SO formula defining the models of the predicate atom  $A(\xi_1, \dots, \xi_{\#A})$  with respect to the SID  $\Delta$  is the following:

$$\mathfrak{A}_\Delta^A(\xi_1, \dots, \xi_{\#A}) \stackrel{\text{def}}{=} \exists x \exists \{X_i\}_{i=1}^R \exists \{Y_j\}_{j=1}^P \exists \{Z_{1,\ell}\}_{\ell=1}^{\#R_1} \dots \exists \{Z_{K,\ell}\}_{\ell=1}^{\#R_K} . \\ \mathfrak{T}(x, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P) \wedge \mathfrak{F}(\xi_1, \dots, \xi_{\#A}, x, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P, \{\{Z_{k,\ell}\}_{\ell=1}^{\#R_k}\}_{k=1}^N)$$

The correctness of the above construction is proved in the following proposition, that also shows  $\llbracket \text{SLR} \rrbracket \subseteq \llbracket \text{SO} \rrbracket$ :

**Proposition 5.** *Given an SID  $\Delta$  and a predicate atom  $A(\xi_1, \dots, \xi_{\#A})$ , for each structure  $\sigma$  and store  $\nu$ , we have  $\sigma \models_\Delta^\nu A(\xi_1, \dots, \xi_{\#A}) \iff \sigma \Vdash^\nu \mathfrak{A}_\Delta^A(\xi_1, \dots, \xi_{\#A})$ .*

## 7 $\llbracket \text{MSO} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \text{SLR} \rrbracket$

We prove that, for any MSO sentence  $\phi$  and any integer  $k \geq 1$ , there exists an SID  $\Delta(k, \phi)$  that defines a predicate  $A_{k,\phi}$  of arity zero, such that  $\llbracket A_{k,\phi}(\cdot) \mid \Delta(k, \phi) \rrbracket = \llbracket \phi \rrbracket_{\mathcal{D}}^k$  i.e., the set of guarded models of  $\phi$  of treewidth at most  $k$  corresponds to the set of structures SLR-defined by the predicate atom  $A_{k,\phi}(\cdot)$  interpreted in the SID  $\Delta(k, \phi)$ . Our proof leverages from a result of Courcelle [21], stating that the set of models of bounded treewidth of a given MSO sentence can be described by a set of recursive equations, written using an algebra of operations on structures. This result follows up in a long-standing line of work (known as Feferman-Vaught theorems [41]) that reduces the evaluation of an MSO sentence on the result of an algebraic operation to the evaluation of several related sentences in the arguments of the respective operation.

### 7.1 Courcelle's Theorem

In order to explain our construction (given in §7.2), we recall first a result of Courcelle. Because this result applies to general structures  $(U, \sigma)$ , we do not assume the usual  $U = \mathbb{U}$  in this section. Courcelle gives a characterization of the structures of bounded treewidth that satisfy a given MSO formula  $\phi$  by an effectively constructible set of recursive equations. This set of equations uses two operations on structures, namely *glue* and *fgcst<sub>j</sub>*, that are lifted to sets of structures, as usual. The result is developed in two

steps. The first step builds a generic set of equations, that characterizes all structures of bounded treewidth. This set of equations is then refined, in the second step, to describe only models of  $\phi$ .

**Operations on Structures** Let  $\Sigma_1$  and  $\Sigma_2$  be two (possibly overlapping) signatures. The *glueing* operation  $glue : Str(\Sigma_1) \times Str(\Sigma_2) \rightarrow Str(\Sigma_1 \cup \Sigma_2)$  is the union of structures with *disjoint universes*, followed by fusion of the elements denoted by constants. Formally, given  $S_i = (U_i, \sigma_i)$ , for  $i = 1, 2$ , such that  $U_1 \cap U_2 = \emptyset$ , let  $\sim$  be the least equivalence relation on  $U_1 \cup U_2$  such that  $\sigma_1(c) \sim \sigma_2(c)$ , for all  $c \in \Sigma \cap \Sigma'$ . Let  $[u]$  be the equivalence class of  $u \in U_1 \cup U_2$  with respect to  $\sim$ . Then  $glue(S_1, S_2) \stackrel{\text{def}}{=} (U, \sigma)$ , where  $U \stackrel{\text{def}}{=} \{[u] \mid u \in U_1 \cup U_2\}$  and  $\sigma$  is defined as follows:

$$\sigma(R) \stackrel{\text{def}}{=} \begin{cases} \{\langle [u_1], \dots, [u_{\#R}] \rangle \mid \langle u_1, \dots, u_{\#R} \rangle \in \sigma_1(R)\}, & \text{if } R \in \Sigma_1 \setminus \Sigma_2 \\ \{\langle [u_1], \dots, [u_{\#R}] \rangle \mid \langle u_1, \dots, u_{\#R} \rangle \in \sigma_2(R)\}, & \text{if } R \in \Sigma_2 \setminus \Sigma_1 \\ \{\langle [u_1], \dots, [u_{\#R}] \rangle \mid \langle u_1, \dots, u_{\#R} \rangle \in \sigma_1(R) \cup \sigma_2(R)\}, & \text{if } R \in \Sigma_1 \cap \Sigma_2 \end{cases}$$

$$\sigma(c) \stackrel{\text{def}}{=} \begin{cases} [\sigma_1(c)], & \text{if } c \in \Sigma_1 \\ [\sigma_2(c)], & \text{if } c \in \Sigma_2 \setminus \Sigma_1 \end{cases}$$

Since we match isomorphic structures, the nature of the elements of  $U$  (i.e., equivalence classes) is not important. The *forget* operation  $fgcst_j : Str(\Sigma) \rightarrow Str(\Sigma \setminus \{c_j\})$  simply drops the constant  $c_j$  from the domain of its argument.

**Structures of Bounded Treewidth** Let  $k \geq 1$  be an integer,  $\Sigma = \{R_1, \dots, R_N, c_1, \dots, c_M\}$  be a signature and  $\Pi = \{c_{M+1}, \dots, c_{M+k+1}\}$  be a set of constants disjoint from  $\Sigma$ , called *ports*. We consider variables  $Y_i$ , for all subsets  $\Pi_i \subseteq \Pi$ , denoting sets of structures over the signature  $\Sigma \cup \Pi_i$ . The equation system now consists of recursive equations of the form  $Y_0 \supseteq f(Y_1, \dots, Y_n)$ , where each  $f$  is either *glue*,  $fgcst_{M+j}$ , for any  $j \in [1, k+1]$ , or a singleton relation of type  $R_i$ , of type  $R_i$ , consisting of a tuple with at most  $k+1$  distinct elements, for any  $i \in [1, N]$ . We denote this set of equations by  $Tw(k)$ . The structures of treewidth at most  $k$  correspond to a component of the least solution of  $Tw(k)$ , in the domain of tuples of sets ordered by pointwise inclusion, see e.g., [22, Theorem 2.83].

**Models of MSO Formulae** We recall that the *quantifier rank*  $qr(\phi)$  of an MSO formula  $\phi$  is the maximal depth of nested quantifiers i.e.,  $qr(\phi) \stackrel{\text{def}}{=} 0$  if  $\phi$  is an atom,  $qr(\neg\phi_1) \stackrel{\text{def}}{=} qr(\phi_1)$ ,  $qr(\phi_1 \wedge \phi_2) \stackrel{\text{def}}{=} \max(qr(\phi_1), qr(\phi_2))$  and  $qr(\exists x . \phi_1) = qr(\exists X . \phi_1) \stackrel{\text{def}}{=} qr(\phi_1) + 1$ . We denote by  $\mathbb{F}_{\text{MSO}}^r$  the set of MSO sentences of quantifier rank at most  $r$ . This set is finite, up to logical equivalence. For a structure  $S = (U, \sigma)$ , we define its *r-type* as  $type^r(S) \stackrel{\text{def}}{=} \{\phi \in \mathbb{F}_{\text{MSO}}^r \mid S \models \phi\}$ . We assume the sentences in  $type^r(S)$  to use the signature over which  $S$  is defined; this signature will be clear from the context in the following.

**Definition 9.** An operation  $f : Str(\Sigma_1) \times \dots \times Str(\Sigma_n) \rightarrow Str(\Sigma_{n+1})$  is (effectively) MSO-compatible<sup>11</sup> iff  $type^r(f(S_1, \dots, S_n))$  depends only on (and can be effectively computed from)  $type^r(S_1), \dots, type^r(S_n)$  by an abstract operation  $f^\# : (\text{pow}(\mathbb{F}_{\text{MSO}}^r))^n \rightarrow \text{pow}(\mathbb{F}_{\text{MSO}}^r)$ .

Courcelle establishes that glueing and forgetting of constants are effectively MSO-compatible, with effectively computable abstract operations  $glue^\#$  and  $fgcst_{M+i}^\#$ , for

<sup>11</sup> Also referred to as smooth operations in [41].

$i \in [1, k+1]$ , see e.g., [21, Lemmas 3.2 and 3.3]. Then one can build from  $Tw(k)$  a set of recursive equations  $Tw^\sharp(k)$  of the form  $Y_0^{\tau_0} = f(Y_1^{\tau_1}, \dots, Y_n^{\tau_n})$ , where  $Y_0 = f(Y_1, \dots, Y_n)$  is an equation from  $Tw(k)$  and  $\tau_0, \dots, \tau_n$  are  $qr(\phi)$ -types such that  $\tau_0 = f^\sharp(\tau_1, \dots, \tau_n)$ . Intuitively, each annotated variable  $Y^\tau$  denotes the set of structures whose  $qr(\phi)$ -type is  $\tau$ , from the  $Y$ -component of the least solution of  $Tw(k)$ . The set of models of  $\phi$  of treewidth at most  $k$  is the union of the  $Y^\tau$ -components of the least solution of  $Tw^\sharp(k)$ , such that  $\phi \in \tau$ , see e.g., [21, Theorem 3.6].

## 7.2 Encoding Types in SLR

We explain the proof for  $\llbracket MSO \rrbracket_{\mathfrak{D}}^k \subseteq \llbracket SLR \rrbracket$ . Instead of using the set of recursive equations  $Tw(k)$  from the previous subsection, we give an SID  $\Delta(k)$  that characterizes the guarded structures of bounded treewidth (Figure 1a). The idea is to use the separating conjunction for simulating the glueing operation. However, the separating conjunction is interpreted as composition of compatible structures, so that the universes cannot be disjoint, as required by the glueing operation. Note that this condition is stronger than the disjointness of structures i.e., the disjointness of the interpretations of each relation symbol from the signature (Def. 2).

This is also the place where we make use of the assumption that we only consider guarded structures i.e., we use the unary relation symbol  $\mathfrak{D}$  to enforce disjointness, by means of composition. Intuitively, in the SID  $\Delta(k)$ ,  $\mathfrak{D}$  “collects” the values assigned to the existentially quantified variables created by rule (2) and the top-level rule (4) during the unraveling. In particular, the relation symbol  $\mathfrak{D}$  ensures that (i) the variables of a predicate atom are mapped to pairwise distinct values and (ii) the composition of two guarded structures is the same as glueing them.

To alleviate the presentation, the SID  $\Delta(k)$  defines only structures  $(\mathbb{U}, \sigma) \in Str(\Sigma, \mathfrak{D})$  with at least  $k+1$  distinct elements in  $\sigma(\mathfrak{D})$  (rule 4) and  $\sigma(R) \neq \emptyset$  for at least one relation symbol  $R \in \Sigma$  (rule 3). The cases of structures such that  $\|\sigma(\mathfrak{D})\| \leq k$  or  $\bigcup_{R \in \Sigma} \sigma(R) = \emptyset$  can be dealt with easily, by adding more rules to  $\Delta(k)$ . We show below that  $\Delta(k)$  defines all structures of  $k$ -bounded treewidth (except for the aforementioned corner cases):

**Lemma 5.** *For any guarded structure  $(\mathbb{U}, \sigma) \in Str(\Sigma, \mathfrak{D})$ , such that  $\|\sigma(\mathfrak{D})\| \geq k+1$  and  $\sigma(R) \neq \emptyset$ , for at least some  $R \in \Sigma$ , we have  $tw(\sigma) \leq k$  if and only if  $(\mathbb{U}, \sigma) \models_{\Delta(k)} A_k()$ .*

The second step of our construction is the annotation of the rules in  $\Delta(k)$  with  $qr(\phi)$ -types, in order to obtain an SID  $\Delta(k, \phi)$  describing the models of an MSO sentence  $\phi$ , of treewidth at most  $k$ . For this, we consider the set of ports  $\Pi = \{c_{M+1}, \dots, c_{M+k+1}\}$  disjoint from  $\Sigma$ , which we use to encode the values of the variables  $x_1, \dots, x_{k+1}$ :

**Definition 10.** *Let  $\Sigma = \{R_1, \dots, R_N, c_1, \dots, c_M\}$  be a signature,  $\Pi = \{c_{M+1}, \dots, c_{M+k+1}\}$  be a set of constants not in  $\Sigma$ , and  $\mathbf{v}$  be a store mapping  $x_1, \dots, x_{k+1}$  to elements of  $\mathbb{U}$ . For a structure  $(\mathbb{U}, \sigma) \in Str(\Sigma, \mathfrak{D})$  with  $\sigma(\mathfrak{D}) \cap \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\} = \emptyset$ , we define a structure  $encode(\sigma, \mathbf{v}) \in Str(\Sigma \cup \Pi, \mathfrak{D})$ , with universe  $\mathbb{U}$ , that agrees with  $\sigma$  over  $\Sigma$  and maps  $c_{M+i}$  to  $\mathbf{v}(x_i)$ , for all  $i \in [1, k+1]$ , such that  $encode(\sigma, \mathbf{v})(\mathfrak{D}) = \sigma(\mathfrak{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ .*

The correctness of our construction will rely on the fact that the composition acts like glueing, for structures with universe  $\mathbb{U}$ , whose sets of elements involved in some relation may only overlap at the interpretation of the ports from  $\Pi$ :

$$A(x_1, \dots, x_{k+1}) \leftarrow A(x_1, \dots, x_{k+1}) * A(x_1, \dots, x_{k+1}) \quad (1)$$

$$A(x_1, \dots, x_{k+1}) \leftarrow \exists y . \mathcal{D}(y) * A(x_1, \dots, x_{k+1})[x_i/y], \text{ for all } i \in [1, k+1] \quad (2)$$

$$A(x_1, \dots, x_{k+1}) \leftarrow R(y_1, \dots, y_{\#R}), \text{ for all } R \in \Sigma \text{ and } y_1, \dots, y_{\#R} \in \{x_1, \dots, x_{k+1}\} \quad (3)$$

$$A_k() \leftarrow \exists x_1 \dots \exists x_{k+1} . \mathcal{D}(x_1) * \dots * \mathcal{D}(x_{k+1}) * A(x_1, \dots, x_{k+1}) \quad (4)$$

(a)

$$A^\tau(x_1, \dots, x_{k+1}) \leftarrow A^{\tau_1}(x_1, \dots, x_{k+1}) * A^{\tau_2}(x_1, \dots, x_{k+1}), \text{ where } \tau = \text{glue}^\sharp(\tau_1, \tau_2) \quad (5)$$

$$A^\tau(x_1, \dots, x_{k+1}) \leftarrow \exists y . \mathcal{D}(y) * A^{\tau_1}(x_1, \dots, x_{k+1})[x_i/y], \text{ for all } i \in [1, k+1], \quad (6)$$

where  $\tau = \text{glue}^\sharp(\text{fgcst}_{M+i}^\sharp(\tau_1), \rho_i)$  for the type  $\rho_i$  of some structure  $S \in \text{Str}(\{c_{M+i}\}, \mathcal{D})$  with singleton universe and  $S \Vdash \mathcal{D}(c_{M+i})$ ,

$$A^\tau(x_1, \dots, x_{k+1}) \leftarrow R(y_1, \dots, y_{\#R}), \text{ for some } y_1, \dots, y_{\#R} \in \{x_1, \dots, x_{k+1}\}, \quad (7)$$

where  $\tau = \text{type}^{\text{qr}(\phi)}(S)$ ,  $S \in \text{Str}(\Sigma \cup \{c_{M+1}, \dots, c_{M+k+1}\}, \mathcal{D})$  and  $S \Vdash R(y_1, \dots, y_{\#R})[x_1/c_{M+1}, \dots, x_{k+1}/c_{M+k+1}] * * \ast_{1 \leq i < j \leq k+1} c_{M+i} \neq c_{M+j} * \ast_{1 \leq i \leq k+1} \mathcal{D}(c_{M+i})$

$$A_{k,\phi}() \leftarrow \exists x_1 \dots \exists x_{k+1} . \mathcal{D}(x_1) * \dots * \mathcal{D}(x_{k+1}) * A^\tau(x_1, \dots, x_{k+1}) \quad (8)$$

for all  $\tau$  such that  $\phi \in \tau$

(b)

**Fig. 1.** The SID  $\Delta(k)$  defining structures of treewidth at most  $k$  (a) and its annotation  $\Delta(k, \phi)$  defining the models of an MSO sentence  $\phi$ , of treewidth at most  $k$  (b)

**Lemma 6.** For an integer  $r \geq 0$ , a store  $\mathbf{v}$  and disjoint compatible structures  $(\mathbb{U}, \sigma_1), (\mathbb{U}, \sigma_2) \in \text{Str}(\Sigma \cup \Pi, \mathcal{D})$ , such that  $\text{Rel}(\sigma_1) \cap \text{Rel}(\sigma_2) \subseteq \{\sigma_1(c_{M+1}), \dots, \sigma_1(c_{M+k+1})\}$  and  $(\sigma_1(\mathcal{D}) \cup \sigma_2(\mathcal{D})) \cap \{\mathbf{v}(x_i) \mid i \in [1, m]\} = \emptyset$ , we have:

$$\text{type}^r(\text{encode}(\sigma_1 \bullet \sigma_2, \mathbf{v})) = \text{glue}^\sharp(\text{type}^r(\text{encode}(\sigma_1, \mathbf{v})), \text{type}^r(\text{encode}(\sigma_2, \mathbf{v})))$$

Let  $\Delta(k, \phi)$  be the set of rules from Figure 1 (b). The main property of this SID is stated and proved below:

**Proposition 6.** Given  $k \geq 1$  and an MSO sentence  $\phi$ , for any guarded structure  $(\mathbb{U}, \sigma) \in \text{Str}(\Sigma, \mathcal{D})$ , we have: (1)  $(\mathbb{U}, \sigma) \Vdash \phi$  and  $\text{tw}(\sigma) \leq k$  if and only if (2)  $(\mathbb{U}, \sigma) \models_{\Delta(k, \phi)} A_{k, \phi}()$ .

The above result shows that SLR can define the guarded models  $\sigma \in \text{Str}(\Sigma, \mathcal{D})$  of a given MSO formula whose treewidth is bounded by a given integer. We do not know, for the moment, if this result holds on unguarded structures as well.

The above construction of the SID  $\Delta(k, \phi)$  is effectively computable, except from rule 7, where one needs to determine the type of a structure  $S = (\mathbb{U}, \sigma)$  with infinite universe. However, we argue in the following that determining this type can be reduced to computing the type of a finite structure. This type can be determined by solving finitely many MSO model checking problems on finite structures, each of which being known to be PSPACE-complete [47].

Given an integer  $n \geq 0$  and a structure  $S = (\mathbb{U}, \sigma) \in \text{Str}(\Sigma)$ , we define the finite structure  $S^n = (U^n, \sigma)$ , where  $U^n \stackrel{\text{def}}{=} \text{Dom}(\sigma) \cup \{u_1, \dots, u_n\}$ , for pairwise distinct elements  $u_1, \dots, u_n \in \mathbb{U} \setminus \text{Dom}(\sigma)$ . Then, for any quantifier rank  $r$ , the structures  $S$  and  $S^{2^r}$  have the same  $r$ -type, as shown below:

**Lemma 7.** Given  $r \geq 0$  and  $S = (\mathbb{U}, \sigma) \in \text{Str}(\Sigma)$ , we have  $\text{type}^r(S) = \text{type}^r(S^{2^r})$ .

## 8 The Remaining Cases

We present the results from Table 1, not already covered by §4, §5, §6 and §7.

$\llbracket \text{SO} \rrbracket_{\mathcal{D}}^k \not\subseteq \llbracket \text{MSO} \rrbracket$  First,  $\llbracket \text{SO} \rrbracket_{\mathcal{D}}^k \not\subseteq \llbracket \text{MSO} \rrbracket$  follows from the fact that any non-recognizable context-free word language, corresponding to a non-MSO-definable set of structures can be defined in SLR (Proposition 3). Since  $\llbracket \text{SLR} \rrbracket \subseteq \llbracket \text{SO} \rrbracket$ , we obtain that  $\llbracket \text{SO} \rrbracket_{\mathcal{D}}^k \not\subseteq \llbracket \text{MSO} \rrbracket$ . Moreover,  $\llbracket \text{SO} \rrbracket \not\subseteq \llbracket \text{MSO} \rrbracket$  follows from the fact that our counterexample involves only structures of treewidth one (i.e., lists encoding words as in §4).

$\llbracket \text{SLR} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \text{SO} \rrbracket$  By applying the translation of SLR to SO from §6 to  $\Delta(k)$  (Fig. 1a) and to a given SID  $\Delta$  defining a predicate  $A$  of zero arity, respectively, and taking the conjunction of the results with the SO formula defining guarded structures<sup>12</sup>, we obtain an SO formula that defines the set  $\llbracket A(\cdot) | \Delta \rrbracket_{\mathcal{D}}^k$ , thus proving that  $\llbracket \text{SLR} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \text{SO} \rrbracket$ .

$\llbracket (\text{M})\text{SO} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket (\text{M})\text{SO} \rrbracket$  For each given  $k \geq 1$ , there exists an MSO formula  $\theta_k$  that defines the structures of treewidth at most  $k$  [22, Proposition 5.11]. This is a consequence of the Graph Minor Theorem proved by Robertson and Seymour [45], combined with the fact that bounded treewidth graphs are closed under taking minors and that the property of having a given finite minor is MSO-definable<sup>13</sup>. Then, for any given (M)SO formula  $\phi$ , the (M)SO formula  $\phi \wedge \theta_k$  defines the models of  $\phi$  of treewidth at most  $k$ .

**Open Problems** The following problems from Table 1 are currently open:  $\llbracket \text{SLR} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \text{SLR} \rrbracket$  and  $\llbracket \text{SO} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \text{SLR} \rrbracket$ , both conjectured to have a negative answer. In particular, the difficulty concerning  $\llbracket \text{SLR} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \text{SLR} \rrbracket$  is that, in order to ensure treewidth boundedness, it seems necessary to force the composition of structures to behave like glueing (see the definition of  $\Delta(k)$  in Fig. 1a), which is however difficult to ensure without the additional predicate symbol  $\mathcal{D}$ .

Since  $\llbracket \text{MSO} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \text{SLR} \rrbracket$  but  $\llbracket \text{MSO} \rrbracket \not\subseteq \llbracket \text{SLR} \rrbracket$ , it is natural to ask for the existence of a fragment of SLR that describes only MSO-definable families of structures of bounded treewidth. Unfortunately, since SLR can define context-free languages (Prop. 3), the MSO-definability of the set of models of a SLR formula is undecidable, as a consequence of the undecidability of the recognizability of context-free languages [31]. On the other hand, the treewidth-boundedness of the set of models of a SLR formula is an open problem, related to the open problem  $\llbracket \text{SLR} \rrbracket_{\mathcal{D}}^k \subseteq \llbracket \text{SLR} \rrbracket$  above.

## 9 Conclusions

We have compared the expressiveness of SLR, MSO and SO, in general and for models of bounded treewidth. Interestingly, we found that SLR and MSO are, in general, incomparable and subsumed by SO, whereas the models of bounded treewidth of MSO can be defined by SLR, modulo augmenting the signature with a unary relation symbol used to store the elements that occur in the original structure.

<sup>12</sup>  $\bigwedge_{R \in \Sigma} \forall x_1 \dots \forall x_{\#R} \cdot R(x_1, \dots, x_{\#R}) \rightarrow \bigwedge_{i \in [1, \#R]} \mathcal{D}(x_i)$ .

<sup>13</sup> The proof of Robertson and Seymour does not build  $\theta_k$ , see [3] for an effective proof.

## References

1. S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, 2000.
2. P. Aczel. An introduction to inductive definitions. In *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, pages 739–782. Elsevier, 1977.
3. I. Adler, M. Grohe, and S. Kreutzer. Computing excluded minors. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '08*, page 641–650, USA, 2008. Society for Industrial and Applied Mathematics.
4. E. Ahrens, M. Bozga, R. Iosif, and J. Katoen. Reasoning about distributed reconfigurable systems. In *Object-Oriented Programming, Systems, Languages and Applications (OOP-SLA)*, volume To appear, 2022.
5. T. Antonopoulos and A. Dawar. Separating graph logic from mso. In L. de Alfaro, editor, *Foundations of Software Science and Computational Structures*, pages 63–77, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
6. M. Bojańczyk and M. Pilipczuk. Definability equals recognizability for graphs of bounded treewidth. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*, page 407–416, New York, NY, USA, 2016. Association for Computing Machinery.
7. M. Bozga, L. Bueri, and R. Iosif. Decision problems in a logic for reasoning about reconfigurable distributed systems. In J. Blanchette, L. Kovács, and D. Pattinson, editors, *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, volume 13385 of *Lecture Notes in Computer Science*, pages 691–711. Springer, 2022.
8. M. Bozga, L. Bueri, and R. Iosif. Decision problems in a logic for reasoning about reconfigurable distributed systems. *CoRR*, abs/2202.09637, 2022.
9. M. Bozga, L. Bueri, and R. Iosif. On an invariance problem for parameterized concurrent systems. In B. Klin, S. Lasota, and A. Muscholl, editors, *33rd International Conference on Concurrency Theory, CONCUR 2022, September 12-16, 2022, Warsaw, Poland*, volume 243 of *LIPICs*, pages 24:1–24:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
10. R. Brochenin, S. Demri, and É. Lozes. On the almighty wand. *Inf. Comput.*, 211:106–137, 2012.
11. J. Brotherston, C. Fuhs, J. A. N. Pérez, and N. Gorogiannis. A decision procedure for satisfiability in separation logic with inductive predicates. In T. A. Henzinger and D. Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 25:1–25:10. ACM, 2014.
12. J. Brotherston, N. Gorogiannis, M. I. Kanovich, and R. Rowe. Model checking for symbolic-heap separation logic with inductive predicates. In R. Bodík and R. Majumdar, editors, *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pages 84–96. ACM, 2016.
13. J. Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6:66–92, 1960.
14. C. Calcagno, P. Gardner, and M. Hague. From separation logic to first-order logic. In *Foundations of Software Science and Computational Structures*, pages 395–409, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
15. C. Calcagno, H. Yang, and P. W. O’Hearn. Computability and complexity results for a spatial assertion language for data structures. In R. Hariharan, M. Mukund, and V. Vinay, editors,

- FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science, 21st Conference, Bangalore, India, December 13-15, 2001, Proceedings*, volume 2245 of *Lecture Notes in Computer Science*, pages 108–119. Springer, 2001.
16. L. Cardelli, P. Gardner, and G. Ghelli. A Spatial Logic for Querying Graphs. In P. Widmayer, F. T. Ruiz, R. M. Bueno, M. Hennessy, S. Eidenbenz, and R. Conejo, editors, *Proceedings of the 29<sup>th</sup> International Colloquium on Automata, Languages and Programming (ICALP'02)*, volume 2380 of *Lecture Notes in Computer Science*, pages 597–610. Springer, July 2002.
  17. L. Cardelli and A. D. Gordon. Anytime, anywhere: Modal logics for mobile ambients. In *Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '00, page 365–377, New York, NY, USA, 2000. Association for Computing Machinery.
  18. B. Cook, C. Haase, J. Ouaknine, M. J. Parkinson, and J. Worrell. Tractable reasoning in a fragment of separation logic. In J. Katoen and B. König, editors, *CONCUR 2011 - Concurrency Theory - 22nd International Conference, CONCUR 2011, Aachen, Germany, September 6-9, 2011. Proceedings*, volume 6901 of *Lecture Notes in Computer Science*, pages 235–249. Springer, 2011.
  19. B. Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.
  20. B. Courcelle. The monadic second-order logic of graphs v: on closing the gap between definability and recognizability. *Theoretical Computer Science*, 80(2):153–202, 1991.
  21. B. Courcelle. The monadic second-order logic of graphs VII: graphs as relational structures. *Theor. Comput. Sci.*, 101(1):3–33, 1992.
  22. B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2012.
  23. A. Dawar, P. Gardner, and G. Ghelli. Expressiveness and Complexity of Graph Logic. *Information and Computation*, 205(3):263–310, Feb. 2007.
  24. S. Demri and M. Deters. Expressive completeness of separation logic with two variables and no separating conjunction. *ACM Trans. Comput. Log.*, 17(2):12, 2016.
  25. S. Demri, É. Lozes, and A. Mansutti. The effects of adding reachability predicates in quantifier-free separation logic. *ACM Trans. Comput. Log.*, 22(2):14:1–14:56, 2021.
  26. J. Doner. Tree acceptors and some of their applications. *J. Comput. Syst. Sci.*, 4(5):406–451, oct 1970.
  27. H. Ebbinghaus and J. Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, 1995.
  28. M. Echenim, R. Iosif, and N. Peltier. The bernays-schönfinkel-ramsey class of separation logic with uninterpreted predicates. *ACM Trans. Comput. Log.*, 21(3):19:1–19:46, 2020.
  29. M. Echenim, R. Iosif, and N. Peltier. Decidable Entailments in Separation Logic with Inductive Definitions: Beyond Establishment. In C. Baier and J. Goubault-Larrecq, editors, *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*, volume 183 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
  30. M. Echenim, R. Iosif, and N. Peltier. Unifying decidable entailments in separation logic with inductive definitions. In A. Platzer and G. Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 183–199. Springer, 2021.
  31. S. Greibach. A note on undecidable properties of formal languages. *Math. Systems Theory*, 2:1–6, 1968.
  32. W. Hanf. Model-theoretic methods in the study of elementary logic. In *The Theory of Models, Studies in Logic and the Foundations of Mathematics*, pages 132–145. North-Holland, 2014.

33. R. Iosif, A. Rogalewicz, and J. Simáček. The tree width of separation logic with recursive definitions. In M. P. Bonacina, editor, *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, volume 7898 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 2013.
34. S. S. Ishtiaq and P. W. O’Hearn. BI as an assertion language for mutable data structures. In C. Hankin and D. Schmidt, editors, *Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, London, UK, January 17-19, 2001*, pages 14–26. ACM, 2001.
35. C. Jansen, J. Katelaan, C. Matheja, T. Noll, and F. Zuleger. Unified reasoning about robustness properties of symbolic-heap separation logic. In *European Symposium on Programming (ESOP)*, volume 10201 of *Lecture Notes in Computer Science*, pages 611–638. Springer, 2017.
36. N. D. Jones and S. S. Muchnick. A flexible approach to interprocedural data flow analysis and programs with recursive data structures. In *Proceedings of the 9th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL ’82, page 66–74, New York, NY, USA, 1982. Association for Computing Machinery.
37. J. Katelaan and F. Zuleger. Beyond symbolic heaps: Deciding separation logic with inductive definitions. In E. Albert and L. Kovács, editors, *LPAR 2020: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Alicante, Spain, May 22-27, 2020*, volume 73 of *EPiC Series in Computing*, pages 390–408. EasyChair, 2020.
38. V. Kuncak and M. Rinard. Generalized records and spatial conjunction in role logic. In *Static Analysis*, pages 361–376, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
39. V. Kuncak and M. C. Rinard. On spatial conjunction as second-order logic. *CoRR*, cs.LO/0410073, 2004.
40. É. Lozes. *Expressivité des logiques spatiales*. Thèse de doctorat, Laboratoire de l’Informatique du Parallélisme, ENS Lyon, France, Nov. 2004.
41. J. A. Makowsky. Algorithmic uses of the feferman-vaught theorem. *Ann. Pure Appl. Log.*, 126(1-3):159–213, 2004.
42. A. Mansutti. *Logiques de séparation : complexité, expressivité, calculs. (Reasoning with separation logics : complexity, expressive power, proof systems)*. PhD thesis, University of Paris-Saclay, France, 2020.
43. C. Matheja, J. Pagel, and F. Zuleger. A decision procedure for guarded separation logic. *ACM Trans. Comput. Logic*, apr 2022. just accepted, available online.
44. Y. N. Moschovakis. *Elementary Induction on Abstract Structures*, volume 77 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1974.
45. M. R. F. R. G. Downey. *Parameterized Complexity*. Springer New York, NY, 1999.
46. J. C. Reynolds. Separation logic: A logic for shared mutable data structures. In *17th IEEE Symposium on Logic in Computer Science (LICS 2002), 22-25 July 2002, Copenhagen, Denmark, Proceedings*, pages 55–74. IEEE Computer Society, 2002.
47. M. Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 137–146. ACM, 1982.

## A Proofs

**Proposition 1.** *The set  $\{\mathcal{K}_n \mid n \in \mathbb{N}\}$  is strictly treewidth-unbounded.*

*Proof (Proof of Proposition 1).* Let  $\mathcal{K}_n \stackrel{\text{def}}{=} (\mathcal{V}, \mathcal{E})$ , modulo isomorphism. We prove that  $\text{tw}(\mathcal{K}_n) = n - 1$ , for all  $n \geq 2$ , by showing that, in every tree decomposition  $(\mathcal{N}, \mathcal{F}, r, \lambda)$

of  $\mathcal{X}_v \stackrel{\text{def}}{=} (\mathcal{V}, \mathcal{E})$  there is a node  $p \in \mathcal{N}$ , such that  $\lambda(p) = \mathcal{V}$ . By induction on  $n \geq 2$ , the base case  $n = 2$  follows immediately from point (1) of Def. 5. For the inductive step  $n > 2$ , let  $u \in \mathcal{V}$  be a vertex and let  $p \in \mathcal{N}$  be the node of least depth, such that  $u \in \lambda(p)$ . By point (2) of Def. 5, this node is unique. Let  $\{v_1, \dots, v_{n-1}\} \stackrel{\text{def}}{=} \mathcal{V} \setminus \{u\}$ . By the inductive hypothesis, there exists a node  $m \in \mathcal{N}$ , such that  $\lambda(m) = \{v_1, \dots, v_{n-1}\}$ , because the restriction of  $(\mathcal{V}, \mathcal{E})$  to  $\{v_1, \dots, v_{n-1}\}$  is a clique and  $(\mathcal{N}, \mathcal{F}, r, \lambda)$  is also a tree decomposition of that clique. Since  $(u, v_1), \dots, (u, v_{n-1}) \in \mathcal{E}$ , by point (1) of Def. 5, there exists nodes  $q_1, \dots, q_{n-1} \in \mathcal{N}$ , such that  $u, v_i \in \lambda(q_i)$ , for all  $i \in [1, n-1]$ . By the choice of  $p$ , these nodes are all descendants of  $p$ , because  $u \in \lambda(q_i)$ , for all  $i \in [1, n-1]$  and  $p$  has the lowest depth among all such nodes. We distinguish two cases:

- $m$  is not a descendant of  $p$ , then let  $q$  be the deepest common ancestor of  $m$  and  $p$ . Since  $v_i \in \lambda(q_i) \cap \lambda(m)$ , we obtain  $v_i \in \lambda(q)$ , for all  $i \in [1, n-1]$ , by point (2) of Def. 5. This leads to  $u, v_1, \dots, v_{n-1} \in \lambda(p)$ , by the same argument.
- $m$  is a descendant of  $p$  and let  $q'_1, \dots, q'_{n-1}$  be the deepest common ancestors of  $m$  and  $q_1, \dots, q_{n-1}$ , respectively. Since  $v_i \in \lambda(q_i) \cap \lambda(m)$ , we obtain  $v_i \in \lambda(q'_i)$ , for all  $i \in [1, n-1]$ , by point (2) of Def. 5. Since  $q'_1, \dots, q'_{n-1}$  are all ancestors of  $m$ , they are linearly ordered by the ancestor relation and let  $q'_k$  be the deepest such node. Then we obtain  $u, v_1, \dots, v_{n-1} \in \lambda(q'_k)$ , by point (2) of Def. 5.

**Proposition 2.** *Given structures  $(U, \sigma) \simeq (U', \sigma')$ , for any sentence  $\phi$  of SLR and any SID  $\Delta$ , we have  $(U, \sigma) \models_{\Delta} \phi \iff (U', \sigma') \models_{\Delta} \phi$ .*

*Proof (Proof of Proposition 2).* By induction on the definition of the satisfaction relation  $\models_{\Delta}$ , we show that  $\sigma \models_{\Delta}^v \psi \iff \sigma' \models_{\Delta}^{h \circ v} \psi$ , for any store  $v$  and any bijection  $h : U \rightarrow U'$ , such that, for any relation symbol  $R \in \Sigma$ , we have  $\langle u_1, \dots, u_n \rangle \in \sigma(R) \iff \langle h(u_1), \dots, h(u_n) \rangle \in \sigma'(R)$  and, for any constant  $c \in \Sigma$ , we have  $h(\sigma(c)) = \sigma'(c)$ , as in Def. 3. We consider the following cases:

- $\phi = \xi \bowtie \chi$ , for  $\bowtie \in \{=, \neq\}$ : we prove only the case where  $\xi \in \mathbb{V}^{(1)}$  and  $\chi \in \{c_1, \dots, c_M\}$ , the other cases being similar. By the above point, for all  $\bowtie \in \{=, \neq\}$ , by the definition of  $h$ , we have:
  - $\sigma(R) = \emptyset \iff \sigma'(R) = \emptyset$ , for all  $R \in \Sigma$ , and
  - $v(\xi) \bowtie \sigma(\chi) \iff h(v(\xi)) \bowtie h(\sigma(\chi)) \iff (h \circ v)(\xi) \bowtie \sigma'(\chi)$ .
- $\phi = R(\xi_1, \dots, \xi_{\#R})$ :  $\langle (\sigma, v)(\xi_1), \dots, (\sigma, v)(\xi_{\#R}) \rangle \in \sigma(R) \iff \langle h((\sigma, v)(\xi_1)), \dots, h((\sigma, v)(\xi_{\#R})) \rangle \in \sigma'(R)$ . If  $\xi_1, \dots, \xi_{\#R} \in \mathbb{V}^{(1)}$ , the latter condition is  $\langle (h \circ v)(\xi_1), \dots, (h \circ v)(\xi_{\#R}) \rangle \in \sigma'(R)$ . Else, if  $\xi_1, \dots, \xi_{\#R} \in \{c_1, \dots, c_M\}$ , the condition is  $\langle h(\sigma(\xi_1)), \dots, h(\sigma(\xi_{\#R})) \rangle \in \sigma'(R)$ . The general case  $\xi_1, \dots, \xi_{\#R} \in \mathbb{V}^{(1)} \cup \{c_1, \dots, c_M\}$  is a combination of the above cases.
- $\phi = A(\xi_1, \dots, \xi_{\#A})$ : this case follows by the induction hypothesis.
- $\phi = \phi_1 * \phi_2$ :  $\sigma \models_{\Delta}^v \phi_1 * \phi_2$  if and only if there exists disjoint and compatible structures  $\sigma_1 \bullet \sigma_2 = \sigma$ , such that  $\sigma_i \models_{\Delta}^v \phi_i$ , for all  $i = 1, 2$ . We define the structures  $\sigma'_1$  and  $\sigma'_2$  as follows, for  $i = 1, 2$ :
  - $\sigma'_i(R) = \{ \langle h(u_1), \dots, h(u_{\#R}) \rangle \mid \langle u_1, \dots, u_{\#R} \rangle \in \sigma_i(R) \}$ , for all relation symbols  $R \in \Sigma$ ,
  - $\sigma'_i(c) = h(\sigma_i(c))$ , for all constant symbols  $c \in \Sigma$ .

Then  $\sigma'_1$  and  $\sigma'_2$  are disjoint and compatible and  $\sigma' = \sigma'_1 \bullet \sigma'_2$ . Moreover, by the inductive hypothesis, we have  $\sigma'_i \models_{\Delta}^v \phi_i$ , for all  $i = 1, 2$ , leading to  $\sigma' \models_{\Delta}^v \phi_1 * \phi_2$ .

- $\phi = \exists x . \psi$ : by the inductive hypothesis, we obtain  $\sigma \models_{\Delta}^{v[x \leftarrow u]} \psi \iff \sigma' \models_{\Delta}^{ho(v[x \leftarrow u])} \psi \iff \sigma' \models_{\Delta}^{(ho v)[x \leftarrow h(u)]} \psi \iff \sigma' \models_{\Delta}^{ho v} \exists x . \psi$ .

**Proposition 3.** *Given a context-free grammar  $G = (N, \mathcal{X}, \Delta)$ , there exists an SID  $\Delta_G$  and a binary predicate symbol  $A_{\mathcal{X}} \in \text{def}(\Delta_G)$ , such that  $w \in \mathcal{L}(G) \iff \sigma_w \models_{\Delta_G} A_{\mathcal{X}}(b, e)$ , for all  $w \in \mathcal{A}^*$ .*

*Proof (Proof of Proposition 3).* Assume w.l.o.g that the context-free grammar  $G$  does not produce the empty word and that it is in Greibach normal form i.e., contains only production rules of the form  $\mathcal{Y}_0 \rightarrow \alpha \mathcal{Y}_1 \dots \mathcal{Y}_i$ , where  $\mathcal{Y}_0, \dots, \mathcal{Y}_i \in N$ , for some  $i \geq 0$  and some  $\alpha \in \mathcal{A}$ . For each nonterminal  $\mathcal{Y}$ , we consider a binary relation symbol  $A_{\mathcal{Y}}(x_1, x_2)$  and for each production rule as above, we consider a rule:

$$A_{\mathcal{Y}_0}(x_1, x_2) \leftarrow \exists y_1 \dots \exists y_i . \mathcal{D}(x_1) * P_{\alpha}(x_1) * E(x_1, y_1) * A_{\mathcal{Y}_1}(y_1, y_2) * E(y_2, y_3) * \dots * E(y_{i-1}, y_i) * A_{\mathcal{Y}_i}(y_i, x_2)$$

if  $i \geq 1$  and  $A_{\mathcal{Y}_0}(x_1, x_2) \leftarrow \mathcal{D}(x_1) * P_{\alpha}(x_1) * x_1 = x_2$ , if  $i = 0$ . Let  $\Delta$  be the set of the rules above and let  $\mathcal{L}_{\mathcal{Y}}(G) \stackrel{\text{def}}{=} \{w \in \mathcal{A}^* \mid \mathcal{Y} \triangleright^* w\}$ , for all  $\mathcal{Y} \in N$ . Let  $w = a_1 \dots a_n \in \mathcal{A}^*$  be any word and  $\sigma_w \in \text{Str}(\{\mathcal{D}, E, P_{\alpha}, P_b, b, e\})$  be the structure encoding  $w$ . We prove that  $\mathcal{Y} \triangleright^* w \iff (\sigma_w, v) \models_{\Delta} A_{\mathcal{Y}}(x, y)$ , for any word  $w \in \mathcal{A}^*$  and any nonterminal  $\mathcal{Y} \in N$ , where  $v$  is any store such that  $v(x) = \sigma_w(b)$  and  $v(y) = \sigma_w(e)$ .

“ $\implies$ ” By induction on the length  $n \geq 1$  of the derivation of  $w$  from  $\mathcal{Y}$ . The base case  $n = 1$  corresponds to a production rule  $\mathcal{Y} \rightarrow \alpha$  in  $G$  that yields the rule  $A_{\mathcal{Y}}(x_1, x_2) \leftarrow \mathcal{D}(x_1) * P_{\alpha}(x_1) * x_1 = x_2$  in  $\Delta$  and  $\sigma_{\alpha} \models_{\Delta} A_{\mathcal{Y}}(b, e)$  follows. For the inductive step  $n > 1$ , we assume w.l.o.g. that the derivation is ordered such that each nonterminal is fully expanded before another nonterminal from the same rule (every derivation of a context-free grammar can be reordered in this way). Let  $\mathcal{Y}_0 \rightarrow \alpha \mathcal{Y}_1 \dots \mathcal{Y}_i$  be the first rule of the derivation and let  $\mathcal{Y}_j \triangleright^* w_j$  be the sub-derivations of  $\mathcal{Y}_1, \dots, \mathcal{Y}_i$ , respectively. Then  $w = \alpha w_1 \dots w_i$ . We can now choose structures  $\sigma_0 \bullet \sigma_1 \bullet \dots \bullet \sigma_i = \sigma_w$ , such that  $\sigma_0(\mathcal{D}) = \{v_0\}$ ,  $\sigma_0(P_{\alpha}) = \{v_0\}$ ,  $\sigma_0(E) = \{(v_j, v_{j+1}) \mid j \in [0, n-1]\}$ ,  $\sigma_0(b) = v_0$  and  $\sigma_0(e) = v_n$ , for some set of pairwise different universe elements  $v_0, \dots, v_n \in \mathbb{U}$ . We denote by  $\sigma'_j$ , for  $1 \leq j \leq i$ , the structure that is identical to  $\sigma_j$  except that we set  $\sigma'_j(b) = v_j$  and  $\sigma'_j(e) = v_{j+1}$ , where we set  $v_{n+1} = \sigma_w(e)$ . We recognize that  $\sigma'_j$  is isomorphic to  $\sigma_{w_j}$ . By the inductive hypothesis, we have  $\sigma_{w_j} \models_{\Delta} A_{\mathcal{Y}_j}(b, e)$  for all  $1 \leq j \leq i$ . By Prop. 2 we get that  $\sigma'_j \models_{\Delta} A_{\mathcal{Y}_j}(b, e)$ . Let  $v$  be a store with  $v(y_j) = v_j$  for all  $1 \leq j \leq i$ . We now recognize that:

$$(\sigma_0 \bullet \sigma_1 \bullet \dots \bullet \sigma_i, v) \models_{\Delta} \mathcal{D}(b) * P_{\alpha}(b) * E(b, y_1) * A_{\mathcal{Y}_1}(y_1, y_2) * E(y_2, y_3) * \dots * E(y_{i-1}, y_i) * A_{\mathcal{Y}_i}(y_i, e).$$

Hence,  $\sigma_w \models_{\Delta} A_{\mathcal{Y}}(b, e)$ , as required.

“ $\impliedby$ ” By induction on the length of the unfolding that defines the satisfaction relation. In the base case, we have  $\sigma_{\alpha} \models_{\Delta} \mathcal{D}(b) * P_{\alpha}(b) * b = e$ , leading to  $\mathcal{Y} \triangleright^* \alpha$ , for a rule  $\mathcal{Y} \rightarrow \alpha$  of  $G$ . For the inductive step, assume that:

$$\sigma_w \models_{\Delta}^v \mathcal{D}(b) * P_{\alpha}(b) * E(b, y_1) * A_{\mathcal{Y}_1}(y_1, y_2) * E(y_2, y_3) * \dots * E(y_{i-1}, y_i) * A_{\mathcal{Y}_i}(y_i, e)$$

for some store  $v$ , where  $A_{\mathcal{Y}_0}(x_1, x_2) \leftarrow \exists y_1 \dots \exists y_i . V(x_1) * P_{\alpha}(x_1) * E(x_1, y_1) * A_{\mathcal{Y}_1}(y_1, y_2) * E(y_2, y_3) * \dots * E(y_{i-1}, y_i) * A_{\mathcal{Y}_i}(y_i, x_2)$  is a rule of  $\Delta$ , for some  $i \geq 1$ . Then  $G$  has a rule  $\mathcal{Y} \rightarrow \alpha \mathcal{Y}_1 \dots \mathcal{Y}_i$ . Moreover, there exist structures  $\sigma_0 \bullet \sigma_1 \bullet \dots \bullet \sigma_i = \sigma_w$ , such that:

- $\sigma_0 \models^v \mathcal{D}(\mathbf{b}) * P_\alpha(\mathbf{b}) * E(\mathbf{b}, y_1) * E(y_2, y_3) * \dots * E(y_{i-1}, y_i)$ ,
- $(\sigma_j, \mathbf{v}) \models_\Delta A_{\mathcal{Y}_j}(y_j, y_{j+1})$ , for all  $j \in [1, i-1]$ , and
- $(\sigma_i, \mathbf{v}) \models_\Delta A_{\mathcal{Y}_i}(y_i, e)$ .

Let  $\sigma'_j$  be the structures that agree with  $\sigma_j$ , except that  $\sigma'_j(\mathbf{b}) = \mathbf{v}'(y_j)$ ,  $\sigma'_j(e) = \mathbf{v}(y_{j+1})$ , for all  $j \in [1, i-1]$ , and  $\sigma'_i(e) = \sigma_w(e)$ . It is easy to show that there exist words  $w_1, \dots, w_i$ , such that  $w = \alpha w_1 \dots w_i$  and  $\sigma_{w_j} \simeq \sigma'_j$ , for all  $j \in [1, i]$ . By Prop. 2, we obtain  $\sigma_{w_j} \models_\Delta A_{\mathcal{Y}_j}(\mathbf{b}, e)$ , hence  $\mathcal{Y}_j \triangleright^* w_j$ , by the inductive hypothesis, thus leading to  $\mathcal{Y} \triangleright^* w$ .

**Lemma 1.** *Given an SID  $\Delta$ , one can build a normalized SID  $\Delta'$  such that  $\text{def}(\Delta) \subseteq \text{def}(\Delta')$  and for each structure  $\sigma$  and each predicate atom  $A(\xi_1, \dots, \xi_{\#A})$ , we have  $\sigma \models_\Delta \exists \xi_{i_1} \dots \exists \xi_{i_n} . A(\xi_1, \dots, \xi_{\#A}) \iff \sigma \models_{\Delta'} \exists \xi_{i_1} \dots \exists \xi_{i_n} . A(\xi_1, \dots, \xi_{\#A})$ , where  $\{\xi_{i_1}, \dots, \xi_{i_n}\} = \{\xi_1, \dots, \xi_{\#A}\} \cap \mathbb{V}^{(1)}$ .*

*Proof (Proof of Lemma 1).* Let  $\Delta$  be an SID. For each predicate  $A \in \text{def}(\Delta)$  and each partition  $\{I_1, \dots, I_k\}$  of  $[1, \#A]$ , we consider a fresh predicate  $A_{I_1, \dots, I_k}$  of arity  $k \geq 1$ , not in  $\text{def}(\Delta)$ . Let  $\Delta'$  be the SID obtained from  $\Delta$  by introducing, for each rule:

$$A(x_1, \dots, x_{\#A}) \leftarrow \exists y_1 \dots \exists y_m . \phi * \bigstar_{\ell=1}^h B^\ell(z_{\ell,1}, \dots, z_{\ell, \#B^\ell}) \in \Delta \quad (9)$$

where  $\phi$  is a quantifier- and predicate-free formula and for each equivalence relation  $\approx$  on the set of variables  $\{x_1, \dots, x_{\#A}\} \cup \{y_1, \dots, y_m\}$  that is compatible with all equalities in  $\phi$  i.e.,  $x = y$  occurs in  $\phi$  only if  $x \approx y$ , the following rules:

$$A_{I_1, \dots, I_k}(x_1, \dots, x_k) \leftarrow (\exists y_{j_1} \dots \exists y_{j_n} . \psi * \bigstar_{\ell=1}^k B_{J_1^\ell, \dots, J_{s^\ell}^\ell}^\ell(z_{r_{\ell,1}}, \dots, z_{r_{\ell, s^\ell}})) \quad (10)$$

$$[x_{i_1}/x_1, \dots, x_{i_k}/x_k]$$

$$A(x_1, \dots, x_{\#A}) \leftarrow A_{I_1, \dots, I_k}(x_{i_1}, \dots, x_{i_k}) \quad (11)$$

where:

- $\approx$  induces the partitions  $\{I_1, \dots, I_k\}$  of  $[1, \#A]$  and  $\{J_1^\ell, \dots, J_{s^\ell}^\ell\}$  of  $[1, \#B^\ell]$ , for each  $\ell \in [1, k]$ ,
- $x_{i_j}$  and  $z_{r_{\ell, s^\ell}}$  are the first in their  $\approx$ -equivalence classes, respectively, in the total order  $x_1 < \dots < x_{\#A} < y_1 < \dots < y_m$ ,
- $\psi$  is obtained from  $\phi$  by replacing each variable  $x$ , such that  $x \approx x_{i_j}$  with  $x_{i_j}$ , respectively each  $z$ , such that  $z \approx z_{r_{\ell, s^\ell}}$  with  $z_{r_{\ell, s^\ell}}$ , and removing the trivial equalities of the form  $x = x$ ,
- the quantifier prefix  $\exists y_{j_1} \dots \exists y_{j_n}$  is the result of eliminating from  $\exists y_1 \dots \exists y_m$  the variables that do not occur in  $\text{fv}(\psi) \cup \bigcup_{\ell=1}^k \{z_{\ell, r_1}, \dots, z_{\ell, r_{s^\ell}}\}$ .

In particular, one can remove from  $\Delta'$  the rules containing unsatisfiable disequalities of the form  $x \neq x$ , obtained from the above transformations. We are left with proving the equivalence from the statement.

“ $\implies$ ” Assume that  $\sigma \models_\Delta A(\xi_1, \dots, \xi_{\#A})$  for a store  $\mathbf{v}$ , and let  $\approx$  be the equivalence relation over  $\{\xi_1, \dots, \xi_{\#A}\}$ , defined as  $\xi_i \approx \xi_j \iff (\sigma, \mathbf{v})(\xi_i) = (\sigma, \mathbf{v})(\xi_j)$ . We now prove by induction that  $\sigma \models_\Delta A(\xi_1, \dots, \xi_{\#A})$  implies  $\sigma \models_{\Delta'} A_{I_1, \dots, I_k}(\xi_{i_1}, \dots, \xi_{i_k})$ , where  $\{I_1, \dots, I_k\}$  are the partitions of  $[1, \#A]$  induced by  $\approx$  and the  $\xi_{i_j}$  are minimal representatives of  $I_j$  in some fixed total order. Since  $\sigma \models_\Delta A(\xi_1, \dots, \xi_{\#A})$ , there is a rule (9) in  $\Delta$ ,

a store  $v'$ , that agrees with  $v$  over  $\xi_1, \dots, \xi_{\#A}$  and structures  $\sigma_0 \bullet \dots \bullet \sigma_h = \sigma$ , such that  $\sigma_0 \models_{\Delta}^{v'} \phi \bar{s}$  and  $\sigma_i \models_{\Delta}^{v'} B^\ell(z_{\ell,1}, \dots, z_{\ell, \#B^\ell}) \bar{s}$ , for all  $\ell \in [1, h]$ , where  $\bar{s} \stackrel{\text{def}}{=} [x_1/\xi_1, \dots, x_n/\xi_n]$  is the substitution that replaces the formal parameters by terms. Let  $\approx'$  be the equivalence over  $x_1 < \dots < x_{\#A} < y_1 < \dots < y_m$  defined as  $x \approx y \iff v'(x) = v'(y)$ . By the inductive hypothesis, we have  $\sigma_i \models_{\Delta'}^{v'} B_{J_1, \dots, J_{s^\ell}}^\ell(z_{J_{\ell,1}}, \dots, z_{r_{\ell, s^\ell}}) \bar{s}$ , where  $z_{r_{\ell,1}}, \dots, z_{r_{\ell, s^\ell}}$  is the sequence of minimal representatives wrt  $\approx'$ . Since  $\approx' \supseteq \approx$ , there exists a rule of type (10) in  $\Delta'$  allowing to infer that  $\sigma \models_{\Delta'}^v A_{I_1, \dots, I_k}(\xi_{i_1}, \dots, \xi_{i_k})$ . We finally obtain  $\sigma \models_{\Delta'}^v A(\xi_1, \dots, \xi_{\#A})$ , by a rule of type (11).  
" $\Leftarrow$ " Assume that  $\sigma \models_{\Delta'}^v A(\xi_1, \dots, \xi_{\#A})$ . Then, by a rule of type (11) from  $\Delta'$ , we must have  $\sigma \models_{\Delta'}^v A_{I_1, \dots, I_k}(\xi_{i_1}, \dots, \xi_{i_k})$ . We now prove by induction that  $\sigma \models_{\Delta'}^v A_{I_1, \dots, I_k}(\xi_{i_1}, \dots, \xi_{i_k})$  implies that  $\sigma \models_{\Delta}^{v'} A(\xi_1, \dots, \xi_{\#A})$ , where  $v'$  is a store that maps each  $\xi_j \in \mathbb{V}^{(1)}$ , such that  $j \in I_j$ , into  $(\sigma, v)(\xi_{i_j})$ . Assume that  $\sigma \models_{\Delta'}^{v''} (\Psi * \ast_{\ell=1}^h B_{J_1, \dots, J_{s^\ell}}^\ell(z_{r_{\ell,1}}, \dots, z_{r_{\ell, s^\ell}})) \bar{s}$ , by a rule of type (10), where  $v''$  is a store that agrees with  $v$  over  $\xi_{i_1}, \dots, \xi_{i_k}$  and  $\bar{s} \stackrel{\text{def}}{=} [x_1/\xi_{i_1}, \dots, x_k/\xi_{i_k}]$  is a substitution. Then there exists structures  $\sigma_0 \bullet \dots \bullet \sigma_h = \sigma$ , such that  $\sigma_0 \models_{\Delta'}^{v''} \Psi \bar{s}$  and  $\sigma_\ell \models_{\Delta'}^{v''} B_{J_1, \dots, J_{s^\ell}}^\ell(z_{r_{\ell,1}}, \dots, z_{r_{\ell, s^\ell}}) \bar{s}$ , for all  $\ell \in [1, h]$ . By the definition of  $\Delta'$ , there exists a rule of type (9) in  $\Delta$  and a corresponding equivalence relation  $\approx$  over  $x_1 < \dots < x_{\#A} < y_1 < \dots < y_m$ , which induces the partitions  $\{I_1, \dots, I_k\}$  of  $[1, \#A]$  and  $\{J_1^\ell, \dots, J_{s^\ell}^\ell\}$  of  $[1, \#B^\ell]$ , for each  $\ell \in [1, k]$ , and  $x_{i_j}$  and  $z_{r_{\ell, s^\ell}}$  are the first in their  $\approx$ -equivalence classes. We can now choose a store  $v'''$ , such that:

- $(\sigma, v''')(\xi_j) = (\sigma, v'')(\xi_{i_j})$ , for all  $j \in [1, k]$ , and
- $(\sigma, v''')(z_{j\bar{s}}) = (\sigma, v'')(z_{r_{\ell, q}\bar{s}})$  if  $j \in J_q^\ell$ , for all  $\ell \in [1, h]$  and  $q \in [1, s^\ell]$ .

By the definition of  $\Psi$ , we have  $\sigma_0 \models_{\Delta'}^{v'''} \phi$ . By the inductive hypothesis, we obtain  $\sigma_\ell \models_{\Delta}^{v'''} B^\ell(z_{\ell,1}, \dots, z_{\ell, \#B^\ell})$ , for all  $\ell \in [1, h]$ . Hence  $\sigma \models_{\Delta}^{v'''} A(\xi_1, \dots, \xi_{\#A})$ , by a rule of type (9).

**Lemma 2.** *Given a normalized SID  $\Delta$ , a predicate atom  $A(\xi_1, \dots, \xi_{\#A})$  and an infinite set  $\mathcal{U} \subseteq \mathbb{U}$ , for each structure  $\sigma$  and a store  $v$ , such that  $\sigma \models_{\Delta}^v A(\xi_1, \dots, \xi_{\#A})$ , there exists a structure  $\sigma'$ , such that  $\sigma' \models_{\Delta}^{v \downarrow \mathcal{U}} A(\xi_1, \dots, \xi_{\#A})$  and  $\|\text{Dom}(\sigma)\| \leq \|\text{Dom}(\sigma')\|$ .*

*Proof (Proof of Lemma 2).* The structure  $\sigma'$  is built inductively on the definition of the satisfaction relation  $\sigma \models_{\Delta}^v A(x_1, \dots, x_{\#A})$ . Since no existentially quantified variable is constrained by equality during this derivation, one can use the definition of  $\models_{\Delta}^{v \downarrow \mathcal{U}}$  instead, thus ensuring that  $\sigma' \models_{\Delta}^{v \downarrow \mathcal{U}} A(x_1, \dots, x_{\#A})$ . Moreover, since the values of all existentially quantified variables are pairwise distinct in  $\sigma'$ , it follows that  $\|\text{Dom}(\sigma)\| \leq \|\text{Dom}(\sigma')\|$ .

**Lemma 3.** *Given a normalized SID  $\Delta$  and a predicate atom  $A(\xi_1, \dots, \xi_{\#A})$ , we have  $\text{tw}(\sigma) \leq W$ , for each structure  $\sigma$  and store  $v$ , such that  $\sigma \models_{\Delta}^{v \downarrow \mathcal{U}} A(\xi_1, \dots, \xi_{\#A})$ , for some infinite set  $\mathcal{U} \subseteq \mathbb{U}$ , where  $W \geq 1$  is a constant depending only on  $\Delta$ .*

*Proof (Proof of Lemma 3).* Let  $W$  be the maximal number of variables that occurs free or bound in the right-hand side of a rule from  $\Delta$ . Given an infinite set  $\mathcal{U} \subseteq \mathbb{U}$ , we build a tree decomposition  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  of  $\sigma$ , inductively on the definition of the satisfaction relation  $\sigma \models_{\Delta}^{v \downarrow \mathcal{U}} A(\xi_1, \dots, \xi_{\#A})$ , such that  $\bigcup_{n \in \mathcal{N}} \lambda(n) \subseteq \mathcal{U} \cup \{(\sigma, v)(\xi_1), \dots, (\sigma, v)(\xi_{\#A})\}$ .

Assume that  $\sigma \models_{\Delta}^{v \Downarrow \mathcal{U}} A(\xi_1, \dots, \xi_{\#A})$  is the consequence of a rule  $A(x_1, \dots, x_{\#A}) \leftarrow \exists y_1 \dots \exists y_m \cdot \Psi * \bigstar_{\ell=1}^k B_{\ell}(z_1^{\ell}, \dots, z_{\#B_{\ell}}^{\ell})$  from  $\Delta$ , where  $\Psi$  is a quantifier- and predicate-free formula, such that  $\sigma \models_{\Delta}^{v' \Downarrow \mathcal{U}} \exists y_1 \dots \exists y_m \cdot \Psi * \bigstar_{\ell=1}^k B_{\ell}(z_1^{\ell}, \dots, z_{\#B_{\ell}}^{\ell})$ , where  $v'$  is the store that maps  $x_i$  into  $v(\xi_i)$ , for all  $i \in [1, \#A]$ . Then there exists a store  $v''$  that agrees with  $v'$  over  $x_1, \dots, x_{\#A}$ , such that  $v''(y_1), \dots, v''(y_m) \in \mathcal{U}$  are pairwise distinct, and structures  $\sigma_0 \bullet \sigma_1 \bullet \dots \bullet \sigma_k = \sigma$ , such that  $\sigma_0 \models^{v''} \Psi$  and  $\sigma_{\ell} \models_{\Delta}^{v'' \Downarrow \mathcal{U}_{\ell}} B_{\ell}(z_1^{\ell}, \dots, z_{\#B_{\ell}}^{\ell})$ , for all  $\ell \in [1, k]$ , where  $\mathcal{U}_1, \dots, \mathcal{U}_k$  is a partition of  $\mathcal{U}$  into  $k$  infinite sets. Note that, because  $\mathcal{U}$  is infinite, such a partition always exists. By the inductive hypothesis, there exists a tree decomposition  $\mathcal{T}_{\ell} = (\mathcal{N}_{\ell}, \mathcal{F}_{\ell}, r_{\ell}, \lambda_{\ell})$  of  $\sigma_{\ell}$ , such that  $\text{tw}(\mathcal{T}_{\ell}) \leq W$  and  $\bigcup_{n \in \mathcal{N}_{\ell}} \lambda_{\ell}(n) \subseteq \mathcal{U}_{\ell} \cup \{v''(z_1^{\ell}), \dots, v''(z_{\#B_{\ell}}^{\ell})\}$ , for each  $\ell \in [1, k]$ .

We define the tree decomposition  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  such that  $\mathcal{T}_1, \dots, \mathcal{T}_k$  are the immediate subtrees of the root and  $\lambda(r) = \{v''(x_1), \dots, v''(x_{\#A})\} \cup \{v''(y_1), \dots, v''(y_m)\}$ . Then, for each relation atom  $R(z_1, \dots, z_{\#R})$  that occurs in  $\Psi$ , the set  $\{v''(z_1), \dots, v''(z_{\#R})\}$  is a subset of the label of the root, thus fulfilling point (1) of Def. 5. To check point (2) of Def. 5, let  $u \in \lambda_i(n_i) \cap \lambda_j(n_j)$ , where  $n_i \in \mathcal{N}_i$  and  $n_j \in \mathcal{N}_j$ , for some  $1 \leq i < j \leq k$ . Since  $\sigma_i \models_{\Delta}^{v'' \Downarrow \mathcal{U}_i} B_i(z_1^i, \dots, z_{\#B_i}^i)$ ,  $\sigma_j \models_{\Delta}^{v'' \Downarrow \mathcal{U}_j} B_j(z_1^j, \dots, z_{\#B_j}^j)$  and  $U_i \cap U_j = \emptyset$ , we obtain that  $u$  is not the image of an existentially quantified variable via  $v''$ , hence  $u = v''(z)$ , for some  $z \in \{z_1^i, \dots, z_{\#B_i}^i\} \cap \{z_1^j, \dots, z_{\#B_j}^j\} \subseteq \{\xi_1, \dots, \xi_{\#A}\} \cup \{y_1, \dots, y_m\}$ . Then  $u \in \lambda(r)$ , thus fulfilling point (2) of Def. 5. We have  $\text{tw}(\mathcal{T}) = \max\{\|\lambda(r)\|, \text{tw}(\mathcal{T}_1), \dots, \text{tw}(\mathcal{T}_k)\} \leq W$ , since  $\|\lambda(r)\| \leq \#A + m \leq W$ , by the definition of  $\lambda(r)$ , and  $\text{tw}(\mathcal{T}_i) \leq W$ , for all  $i \in [1, k]$ , by the inductive hypothesis.

**Proposition 4.** *Given a SLR sentence  $\phi$  and an SID  $\Delta$ , the set  $\llbracket \phi | \Delta \rrbracket$  is either finite or it has an infinite subset of bounded treewidth.*

*Proof (Proof of Proposition 4).* Given a sentence  $\phi$ , we introduce a fresh predicate  $A_{\phi}$  of arity zero and consider the rule  $A_{\phi}() \leftarrow \phi$ . Then, for each structure  $\sigma$ , we have  $\sigma \models_{\Delta} \phi$  if and only if  $\sigma \models_{\Delta_{\phi}} A_{\phi}()$ , where  $\Delta_{\phi} \stackrel{\text{def}}{=} \Delta \cup \{A_{\phi}() \leftarrow \phi\}$ . Let  $\Delta'_{\phi}$  be the normalized SID, such that  $\sigma \models_{\Delta'_{\phi}} A_{\phi}()$ , by Lemma 1. Given an infinite subset  $\mathcal{U}$  of  $\mathbb{U}$ , there exists a structure  $\sigma'$  and a store  $v$ , such that  $\sigma' \models_{\Delta'_{\phi}}^{v \Downarrow \mathcal{U}} A_{\phi}()$  and  $\|\text{Dom}(\sigma)\| \leq \|\text{Dom}(\sigma')\|$ , by Lemma 2. By Lemma 3, we also obtain  $\text{tw}(\sigma') \leq W$ , where  $W$  depends only of  $\Delta'_{\phi}$  and hence of  $\Delta$  and  $\phi$ . If  $A_{\phi}()$  has finitely many isomorphic models, there is nothing to prove. Otherwise, consider an infinite sequence  $\sigma_1, \sigma_2, \dots$  of models of  $A_{\phi}()$ , such that  $\|\text{Dom}(\sigma_i)\| < \|\text{Dom}(\sigma_{i+1})\|$ , for all  $i \geq 1$  (note that such a sequence always exists as otherwise there would only be finitely many isomorphic model). Then there exists a sequence of models  $\sigma'_1, \sigma'_2, \dots$  of  $A_{\phi}()$ , such that  $\|\text{Dom}(\sigma_i)\| \leq \|\text{Dom}(\sigma'_i)\|$  and  $\text{tw}(\sigma'_i) \leq W$ , for all  $i \geq 1$ . By going to a subsequence, if necessary, one can retrieve an infinite treewidth-bounded set of models of  $\phi$ .

**Lemma 4.** *For any structure  $\sigma$  and store  $v$ ,  $\sigma \models_{\Delta}^v A(\xi_1, \dots, \xi_{\#A})$  iff there exists an unfolding tree  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  for  $A(\xi_1, \dots, \xi_{\#A})$ , such that  $\sigma \models^v \Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A}))$ .*

*Proof (Proof of Lemma 4).* We omit the annotations from  $\phi^n$ , in this proof and simply write  $\phi$  because the annotations are not relevant for this proof. “ $\Rightarrow$ ” By induction on the

definition of the satisfaction relation  $\sigma \models_{\Delta}^v A(\xi_1, \dots, \xi_{\#A})$ . Assume the relation holds because:

$$\sigma \models_{\Delta}^{v'} \psi \bar{s} * \bigstar_{\ell=1}^h B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s}$$

for a rule  $r : A(x_1, \dots, x_{\#A}) \leftarrow \exists y_1 \dots \exists y_m \cdot \psi * \bigstar_{\ell=1}^h B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}})$  from  $\Delta$ , where  $\psi$  is a quantifier- and predicate-free formula,  $\bar{s} \stackrel{\text{def}}{=} [x_1/\xi_1, \dots, x_{\#A}/\xi_{\#A}]$  is a substitution and  $v'$  is a store that agrees with  $v$  over  $\{\xi_1, \dots, \xi_{\#A}\} \cap \mathbb{V}^{(1)}$ . Then there exist structures  $\sigma_0 \bullet \sigma_1 \bullet \dots \bullet \sigma_h = \sigma$ , such that  $\sigma_0 \models_{\Delta}^{v'} \psi \bar{s}$  and  $\sigma_{\ell} \models_{\Delta}^{v'} B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s}$ , for all  $\ell \in [1, h]$ . By the induction hypothesis, there exist unfolding trees  $\mathcal{T}_{\ell} = (\mathcal{N}_{\ell}, \mathcal{F}_{\ell}, r_{\ell}, \lambda_{\ell})$  for  $B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s}$ , such that  $(\sigma_{\ell}, v') \models_{\Delta} \Theta(\mathcal{T}_{\ell}, B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s})$ , for all  $\ell \in [1, h]$ . Then  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  is defined as  $\mathcal{N} \stackrel{\text{def}}{=} \{r\} \cup \bigcup_{\ell=1}^h \mathcal{N}_{\ell}$ ,  $\mathcal{F} \stackrel{\text{def}}{=} \{(r, r_{\ell}) \mid \ell \in [1, h]\} \cup \bigcup_{\ell=1}^h \mathcal{F}_{\ell}$  and  $\lambda = \{(r, r)\} \cup \bigcup_{\ell=1}^h \lambda_{\ell}$ , assuming w.l.o.g. that  $\mathcal{N}_{\ell} \cap \mathcal{N}_k = \emptyset$ , for all  $1 \leq \ell < k \leq h$  and  $r \notin \bigcup_{\ell=1}^h \mathcal{N}_{\ell}$ . The check that  $\sigma \models_{\Delta}^v \Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A}))$  is routine.

“ $\Leftarrow$ ” By induction on the structure of  $\mathcal{T}$ . Let  $p_1, \dots, p_h$  be the children of  $r$  in  $\mathcal{T}$ . By Def. 8, we have:

$$\sigma \models_{\Delta}^{v'} \psi \bar{s} * \bigstar_{\ell=1}^h \Theta(\mathcal{T}[p_{\ell}], B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s})$$

where  $\bar{s} \stackrel{\text{def}}{=} [x_1/\xi_1, \dots, x_{\#A}/\xi_{\#A}]$  is a substitution,  $v'$  is a store that agrees with  $v$  over  $\{\xi_1, \dots, \xi_{\#A}\} \cap \mathbb{V}^{(1)}$  and  $\lambda(r) = A(x_1, \dots, x_{\#A}) \leftarrow \exists y_1 \dots \exists y_m \cdot \psi * \bigstar_{\ell=1}^h B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}})$  is a rule from  $\Delta$ . Then there exist structures  $\sigma_0 \bullet \sigma_1 \bullet \dots \bullet \sigma_h = \sigma$ , such that  $\sigma_0 \models_{\Delta}^{v'} \psi \bar{s}$  and  $\sigma_{\ell} \models_{\Delta}^{v'} \Theta(\mathcal{T}[p_{\ell}], B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s})$ , for all  $\ell \in [1, h]$ . Since  $\mathcal{T}[p_{\ell}]$  is an unfolding tree for  $B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s}$ , by the inductive hypothesis, we obtain  $\sigma_{\ell} \models_{\Delta}^{v'} B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s}$ , for all  $\ell \in [1, h]$ . Then, we have  $\sigma \models_{\Delta}^{v'} \psi \bar{s} * \bigstar_{\ell=1}^h B_{\ell}(z_{\ell,1}, \dots, z_{\ell, \#B_{\ell}}) \bar{s}$ , leading to  $\sigma \models_{\Delta}^v A(\xi_1, \dots, \xi_{\#A})$ .

**Proposition 5.** *Given an SID  $\Delta$  and a predicate atom  $A(\xi_1, \dots, \xi_{\#A})$ , for each structure  $\sigma$  and store  $v$ , we have  $\sigma \models_{\Delta}^v A(\xi_1, \dots, \xi_{\#A}) \iff \sigma \Vdash^v \mathfrak{A}_{\Delta}^A(\xi_1, \dots, \xi_{\#A})$ .*

*Proof (Proof of Proposition 5).* “ $\Rightarrow$ ” By Lemma 4, there exists an unfolding tree  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  of  $A(\xi_1, \dots, \xi_{\#A})$ , such that  $\sigma \models_{\Delta}^v \Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A}))$ . Let  $\Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A})) = \exists y_1 \dots \exists y_K \cdot \Phi$ , where  $\Phi$  is a quantifier- and predicate-free formula. Note that, by Def. 8, no second-order variables occur in  $\Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A}))$ . Hence there exists a store  $v'$  that agrees with  $v$  over  $\xi_1, \dots, \xi_{\#A}$ , such that  $\sigma \models_{\Delta}^{v'} \Phi$ . We define another store  $v''$ , that agrees with  $v$  and  $v'$  over  $\xi_1, \dots, \xi_{\#A}$  such that, moreover, we have:

- $v''(x) = r$ ,
- $v''(X_i) = \{n \in \mathcal{N} \mid \lambda(n) = r_i\}$ , for all  $i \in [1, R]$ ,
- $v''(Y_j) = \{(n, m) \in \mathcal{N} \times \mathcal{N} \mid m \text{ is the } j\text{-th child of } n\}$ , for all  $j \in [1, P]$ ; we consider that the order between the children of a vertex in an unfolding tree is the syntactic order of their corresponding predicate atoms, in the sense of Def. 8,
- $v''(Z_{k,\ell}) = \{(n, (\sigma, v')(\xi_{\ell})) \mid n \in \mathcal{N}, R_k^{\ell}(\xi_1, \dots, \xi_{\#R_k}) \text{ occurs in } \Phi\}$ , for all  $k \in [1, N]$  and  $\ell \in [1, \#R_k]$ .

We have  $\sigma \Vdash^{v''} \mathfrak{A}(x, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P)$  because  $\mathcal{T}$  is an unfolding tree for  $A(\xi_1, \dots, \xi_{\#A})$ , by Def. 8. The proof of  $\sigma \Vdash^{v''} \mathfrak{F}(\xi_1, \dots, \xi_{\#A}, x, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P, \{\{Z_{k,\ell}\}_{\ell=1}^{\#R_k}\}_{k=1}^N)$  follows from  $(\sigma, v') \models \Phi$  and the definition of  $v''$ , by the points (i-vii) from the definition of  $\mathfrak{F}$ . We obtain  $\sigma \Vdash^v \mathfrak{A}_{\Delta}^A(\xi_1, \dots, \xi_{\#A})$  from the definition of  $\mathfrak{A}_{\Delta}^A$ .

” $\Leftarrow$ ” There exists a store  $\mathbf{v}'$  that agrees with  $\mathbf{v}$  over  $\xi_1, \dots, \xi_{\#A}$ , such that:

$$\sigma \models^{\mathbf{v}'} \mathfrak{A}(x, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P) \quad (12)$$

$$\sigma \models^{\mathbf{v}'} \mathfrak{F}(\xi_1, \dots, \xi_{\#A}, x, \{X_i\}_{i=1}^R, \{Y_j\}_{j=1}^P, \{\{Z_{k,\ell}\}_{\ell=1}^{\#R_k}\}_{k=1}^N) \quad (13)$$

By (12) we obtain an unfolding tree  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  for  $A(\xi_1, \dots, \xi_{\#A})$ , such that:

- $\mathcal{N} = \bigcup_{i=1}^R \mathbf{v}'(X_i)$ ,
- $\mathcal{F} = \bigcup_{j=1}^P \mathbf{v}'(Y_j)$ ,
- $\lambda(n) = r_i \iff n \in \mathbf{v}'(X_i)$ , for all  $n \in \mathcal{N}$  and  $i \in [1, R]$ .

Let  $\Theta(\mathcal{T}, A(\xi_1, \dots, \xi_{\#A})) = \exists y_1 \dots \exists y_K \cdot \Phi$ , where  $\Phi$  is a quantifier- and predicate-free formula (Def. 8). By Lemma 4, it is sufficient to show the existence of a store  $\mathbf{v}''$  that agrees with  $\mathbf{v}$  over  $\xi_1, \dots, \xi_{\#A}$ , such that  $\sigma \models^{\mathbf{v}''} \Phi$ . Let  $f_{k,\ell}$  denote the partial mapping defined by  $\mathbf{v}'(Z_{k,\ell})$ , for each  $k \in [1, N]$  and  $\ell \in [1, \#R]$ , by point (i) of the definition of  $\mathfrak{F}$ .

For each  $r \in [1, K]$ , we define  $\mathbf{v}''(y_r) \stackrel{\text{def}}{=} f_{k,\ell}(n)$  if  $y_r$  occurs in or is constrained to be equal to a term  $\xi_\ell$  that occurs in an annotated relation atom  $R_k^n(\xi_1, \dots, \xi_{\#R_k})$  from  $\Phi$ . Note that there can be at most one such relation atom in  $\Phi$ , because of the assumption that in each rule from  $\Delta$  at most one relation atom  $R_k(z_1, \dots, z_{\#R_k})$  occurs. Otherwise, if  $y_r$  is not constrained in  $\Phi$  to be equal to a term that occurs in a relation atom,  $\mathbf{v}''(y_r)$  is given an arbitrary fresh value. Because the equalities and disequalities from  $\Phi$  are taken care of by points (v) and (vi) from the definition of  $\mathfrak{F}$ , it remains to check the satisfaction of the relation atoms from  $\Phi$ . To this end, we define a decomposition  $\sigma = \bigotimes_{n \in \mathcal{N}, k \in [1, N]} \sigma_{n,k}$  such that  $\sigma_{n,k} \models^{\mathbf{v}''} R_k(\xi_1, \dots, \xi_{\#R_k})$ , for all relation atoms  $R_k^n(\xi_1, \dots, \xi_{\#R_k})$  from  $\Phi$ . Such a decomposition is possible due to points (ii-iv) from the definition of  $\mathfrak{F}$ .

**Definition 11.** A tree decomposition  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  of a structure  $(U, \sigma)$  is said to be reduced if and only if the following hold:

1. for each  $R \in \Sigma$  and each  $\langle u_1, \dots, u_{\#R} \rangle \in \sigma(R)$  there exists a leaf  $n \in \mathcal{N}$  such that  $\{u_1, \dots, u_{\#R}\} \subseteq \lambda(n)$ , called the witness of  $\langle u_1, \dots, u_{\#R} \rangle \in \sigma(R)$ ,
2. every leaf witnesses exactly one tuple  $\langle u_1, \dots, u_{\#R} \rangle \in \sigma(R)$ ,
3.  $\mathcal{T}$  is a binary tree i.e., a tree where each node has at most two children,
4. if  $n \in \mathcal{N}$  has two children  $m_1, m_2 \in \mathcal{N}$  then  $\lambda(n) = \lambda(m_1) = \lambda(m_2)$ ,
5. if  $n \in \mathcal{N}$  has one child  $m \in \mathcal{N}$  then either  $\lambda(n) = \lambda(m)$  and  $m$  witnesses a tuple  $\langle u_1, \dots, u_{\#R} \rangle \in \sigma(R)$ , or  $\|\lambda(n) \setminus \lambda(m)\| = \|\lambda(m) \setminus \lambda(n)\| = 1$ ,
6.  $\|\lambda(n)\| = k + 1$ , for all  $n \in \mathcal{N}$ .

**Lemma 8.** If a structure  $(U, \sigma)$  has a tree decomposition of width  $k$ , then it also has a reduced tree decomposition of width  $k$ .

*Proof (Proof of Lemma 8).* The properties (1-6) can be proven directly from Definition 5. The main tool is the introduction of intermediate nodes into the tree decomposition. We list some of the cases. For (1), if a node witnesses more than one tuple, then we introduce an intermediate node between itself and its parent that is labeled by the same set of vertices. The intermediate node then becomes the witness for the this tuple. This process can be iterated until every node witnesses at most one tuple. For (3), if a node  $n_0$  has children  $m_0, \dots, m_l$  we can introduce new nodes  $n_1, \dots, n_{l-1}$  with  $\lambda(m_0) = \lambda(m_i)$ ,

and create a new tree decomposition which agrees with the old tree decomposition except that we remove the edges from  $n_0$  to the children  $m_0, \dots, m_l$  and add edges from  $n_i$  to  $m_i$  and  $n_{i+1}$  for all  $i < l - 1$  and edges from  $n_{l-1}$  to  $m_{l-1}$  and  $m_l$ . For (6), note that we always consider structures over some infinite universe. Hence, we can always extend the labels of the tree-decomposition with some fresh vertices from the universe.

**Lemma 9.** *Let  $\sigma \in \text{Str}(\Sigma, \mathcal{D})$  be a structure and  $\mathbf{v}$  be a store, such that  $\sigma \models_{\Delta(k)}^{\mathbf{v}} A(x_1, \dots, x_{k+1})$  and  $\mathbf{v}(x_i) \notin \sigma(\mathcal{D})$ , for all  $1 \leq i \leq k + 1$ . Then, there exists a reduced tree decomposition  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  of  $\sigma$ , such that  $\text{tw}(\mathcal{T}) = k$ ,  $\lambda(r) = \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$  and  $\lambda(n) \subseteq \sigma(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ , for all  $n \in \mathcal{N}$ .*

*Proof (Proof of Lemma 9).* We prove the claim by induction on the number of rule applications. The claim clearly holds for the base case, by rule (3). We now consider the rule (1) i.e., we assume that  $\sigma \models_{\Delta(k)}^{\mathbf{v}} A(x_1, \dots, x_{k+1}) * A(x_1, \dots, x_{k+1})$ . Then, there exist structures  $\sigma_1$  and  $\sigma_2$ , such that  $(\sigma_i, \mathbf{v}) \models_{\Delta(k)} A(x_1, \dots, x_{k+1})$ , for  $i = 1, 2$  and  $\sigma_1 \bullet \sigma_2 = \sigma$ . We note that the latter implies that  $\sigma_1(\mathcal{D}) \cap \sigma_2(\mathcal{D}) = \emptyset$  ( $\dagger$ ). We now apply the inductive hypothesis and obtain reduced tree decompositions  $\mathcal{T}_i$  for  $\sigma_i$  whose respective roots are labelled by  $\{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$  and  $\lambda(n) \subseteq \sigma(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ , for all nodes  $n$  of  $\mathcal{T}_i$  and all  $i = 1, 2$ . We obtain a reduced tree decomposition for  $\sigma$  by composing  $\mathcal{T}_1$  and  $\mathcal{T}_2$  with a fresh root node labelled by  $\{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ . Note that  $\mathcal{T}$  is indeed a tree decomposition because the only elements that may appear in labels of both  $\mathcal{T}_1$  and  $\mathcal{T}_2$  must belong to  $\{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ , by ( $\dagger$ ). Clearly, we have  $\lambda(n) \subseteq \sigma_1(\mathcal{D}) \cup \sigma_2(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\} = \sigma(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ , for all nodes  $n$  of the resulting tree decomposition  $\mathcal{T}$ . We now consider the rule (2) i.e., we assume that  $\sigma \models_{\Delta(k)}^{\mathbf{v}} \exists y . \mathcal{D}(y) * A(x_1, \dots, x_{k+1})[x_i/y]$ . Then, there is an element  $u \notin \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$  such that  $\sigma' \models_{\Delta(k)}^{\mathbf{v}[x_i \leftarrow u]}$   $A(x_1, \dots, x_{k+1})$ , where the structure  $\sigma'$  agrees with  $\sigma$ , except that  $\sigma(\mathcal{D}) = \sigma'(\mathcal{D}) \setminus \{u\}$ . We now apply the inductive hypothesis and obtain a reduced tree decomposition  $\mathcal{T}_1$  for  $\sigma'$  whose root is labelled by  $(\{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\} \setminus \{\mathbf{v}(x_i)\}) \cup \{u\}$  and  $\lambda(n) \subseteq ((\sigma'(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}) \setminus \{\mathbf{v}(x_i)\}) \cup \{u\}$ , for all nodes  $n$  of  $\mathcal{T}_1$  ( $\ddagger$ ). We can now obtain a reduced tree decomposition  $\mathcal{T}$  for  $\sigma$  by composing  $\mathcal{T}_1$  with an additional root labelled by  $\{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ . Note that  $\mathcal{T}$  is indeed a tree decomposition because  $\mathbf{v}(x_i) \notin \lambda(n)$  for every node  $n$  of  $\mathcal{T}_1$ , because of ( $\ddagger$ ),  $u \neq \mathbf{v}(x_i)$  and the assumption that  $\mathbf{v}(x_i) \notin \sigma(\mathcal{D})$ . We have  $\lambda(n) \subseteq \sigma'(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\} \cup \{u\} = \sigma(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$  for all nodes  $n$  of the resulting tree decomposition  $\mathcal{T}$ .

**Lemma 10.** *Let  $\sigma \in \text{Str}(\Sigma, \mathcal{D})$  be a structure with  $\text{tw}(\sigma) \leq k$  witnessed by some reduced tree decomposition  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$ , with  $\lambda(r) = \{u_1, \dots, u_{k+1}\}$  and  $\sigma(\mathcal{D}) = \bigcup_{n \in \mathcal{N}} \lambda(n) \setminus \{u_1, \dots, u_{k+1}\}$ , and let  $\mathbf{v}$  be a store with  $\mathbf{v}(x_i) = u_i$ , for all  $i \in [1, k + 1]$ . Then,  $\sigma \models_{\Delta(k)}^{\mathbf{v}} A(x_1, \dots, x_{k+1})$ .*

*Proof (Proof of Lemma 10).* The proof goes by induction on the structure of  $\mathcal{T}$ . The claim clearly holds for the base case, where  $\mathcal{T}$  consists of a single leaf, by rule (3). Consider first the case where the root of  $\mathcal{T}$  has two children. The subtrees  $\mathcal{T}_1$  and  $\mathcal{T}_2$  rooted in the two children induce substructures  $\sigma_1$  and  $\sigma_2$  of  $\sigma$ , where  $\langle v_1, \dots, v_{\#\mathbb{R}} \rangle \in \sigma_i(\mathbb{R})$

iff  $\langle v_1, \dots, v_{\#R} \rangle$  is witnessed by some leaf of  $\mathcal{T}_i$ , for  $i = 1, 2$ . Because  $\mathcal{T}$  is a reduced tree decomposition, there is at most one leaf that witnesses a tuple  $\langle v_1, \dots, v_{\#R} \rangle \in \sigma_i(R)$ . Hence, we have  $\sigma = \sigma_1 \bullet \sigma_2$ . From the inductive hypothesis we obtain  $\sigma_i \models_{\Delta(k)}^v A(x_1, \dots, x_{k+1})$ , for  $i = 1, 2$ . Hence  $\sigma_1 \bullet \sigma_2 \models_{\Delta(k)}^v A(x_1, \dots, x_{k+1}) * A(x_1, \dots, x_{k+1})$ , thus  $\sigma \models_{\Delta(k)}^v A(x_1, \dots, x_{k+1})$ , by rule (1). Consider now the case where the root of  $\mathcal{T}$  has a single child which is not a leaf and consider the subtree  $\mathcal{T}_1$  rooted at this child. Then, there is an element  $u \notin \{u_1, \dots, u_{k+1}\}$ , such that the root of  $\mathcal{T}_1$  is labeled by  $\{u_1, \dots, u_{i-1}, u, u_{i+1}, \dots, u_{k+1}\}$ . Let  $\sigma'$  be the structure that agrees with  $\sigma$  except that we have  $\sigma'(\mathcal{D}) = \sigma(\mathcal{D}) \setminus \{u\}$ . By the inductive hypothesis, we obtain  $\sigma' \models_{\Delta(k)}^{v[x_i \leftarrow u]} A(x_1, \dots, x_{k+1})$ . Hence,  $\sigma \models_{\Delta(k)}^v \exists y . \mathcal{D}(y) * A(x_1, \dots, x_{k+1})[x_i/y]$ , thus  $\sigma \models_{\Delta(k)}^v A(x_1, \dots, x_{k+1})$ , by rule (2).

**Lemma 5.** *For any guarded structure  $(\mathbb{U}, \sigma) \in \text{Str}(\Sigma, \mathcal{D})$ , such that  $\|\sigma(\mathcal{D})\| \geq k+1$  and  $\sigma(R) \neq \emptyset$ , for at least some  $R \in \Sigma$ , we have  $\text{tw}(\sigma) \leq k$  if and only if  $(\mathbb{U}, \sigma) \models_{\Delta(k)} A_k()$ .*

*Proof (Proof of Lemma 5).* “ $\Rightarrow$ ” If  $\sigma$  has tree decomposition of width at most  $k$ , then it also has a reduced tree decomposition  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  of width  $k$ , by Lemma 8. Let  $\lambda(r) = \{u_1, \dots, u_{k+1}\}$  and  $v$  be a store such that  $v(x_i) = u_i$ , for all  $i \in [1, k+1]$ . By Lemma 10, we get that  $\sigma \models_{\Delta(k)}^v A(x_1, \dots, x_{k+1}) * \mathcal{D}(x_1) * \dots * \mathcal{D}(x_{k+1})$ . Thus,  $\sigma \models_{\Delta(k)} A_k()$ , by rule (4).

“ $\Leftarrow$ ” By rule (4), there exists a store  $v$  with  $v(x_i) = u_i$ , for all  $1 \leq i \leq k+1$ , and a structure  $\sigma' \in \text{Str}(\Sigma, \mathcal{D})$  that agrees with  $\sigma$  on  $\Sigma$  such that  $\sigma' \models_{\Delta(k)}^v A(x_1, \dots, x_{k+1})$  and  $\sigma'(\mathcal{D}) = \sigma(\mathcal{D}) \setminus \{u_1, \dots, u_{k+1}\}$ . By Lemma 9, there exists a reduced tree decomposition  $\mathcal{T}$  of  $\sigma'$  of width  $k$ . Thus  $\text{tw}(\sigma) \leq k$ , because  $\sigma'$  agrees with  $\sigma$  on  $\Sigma$  and we have  $\sigma'(\mathcal{D}) = \sigma(\mathcal{D}) \setminus \{u_1, \dots, u_{k+1}\}$ .

**Lemma 6.** *For an integer  $r \geq 0$ , a store  $v$  and disjoint compatible structures  $(\mathbb{U}, \sigma_1), (\mathbb{U}, \sigma_2) \in \text{Str}(\Sigma \cup \Pi, \mathcal{D})$ , such that  $\text{Rel}(\sigma_1) \cap \text{Rel}(\sigma_2) \subseteq \{\sigma_1(c_{M+1}), \dots, \sigma_1(c_{M+k+1})\}$  and  $(\sigma_1(\mathcal{D}) \cup \sigma_2(\mathcal{D})) \cap \{v(x_i) \mid i \in [1, m]\} = \emptyset$ , we have:*

$$\text{type}^r(\text{encode}(\sigma_1 \bullet \sigma_2, v)) = \text{glue}^\sharp(\text{type}^r(\text{encode}(\sigma_1, v)), \text{type}^r(\text{encode}(\sigma_2, v)))$$

*Proof (Proof of Lemma 6).* Let us consider  $\sigma'_1 = \text{encode}(\sigma_1, v)$  and  $\sigma'_2 = \text{encode}(\sigma_2, v)$ . In order to apply glueing we will now consider two structures isomorphic to  $\sigma'_1$  and  $\sigma'_2$ , respectively. We note that  $\text{Dom}(\sigma'_1) \cap \text{Dom}(\sigma'_2) \subseteq \{\sigma'_2(c_{M_1}), \dots, \sigma'_2(c_{M+k+1})\}$  because of our assumption that  $\text{Rel}(\sigma_1) \cap \text{Rel}(\sigma_2) \subseteq \{v(x_1), \dots, v(x_{k+1})\}$ . We further note that  $\sigma'_1(c_{M_i}) = \sigma'_2(c_{M_i})$ , for all  $i \in [1, M+k+1]$ , because  $\sigma_1$  and  $\sigma_2$  are compatible and the interpretation of the additional constants  $c_{M+1}, \dots, c_{M+k+1}$  has been chosen w.r.t. the same store  $v$ . Hence, we can choose some partitioning  $\mathcal{U}_1 \uplus \mathcal{U}_2 = \mathbb{U}$  such that  $\mathcal{U}_1$  and  $\mathcal{U}_2$  are countably infinite,  $\text{Dom}(\sigma'_1) \subseteq \mathcal{U}_1$  and  $\text{Dom}(\sigma'_2) \setminus \{\sigma'_2(c_1), \dots, \sigma'_2(c_{M+k+1})\} \subseteq \mathcal{U}_2$ . We can now choose a structure  $\sigma''_2$  with  $\text{Dom}(\sigma''_2) \subseteq \mathcal{U}_2$  that is isomorphic to  $\sigma'_2$  and that agrees with  $\sigma'_2$  except for  $c_1, \dots, c_{M+k+1}$ , whose interpretation is chosen as  $\sigma''_2(c_1), \dots, \sigma''_2(c_{M+k+1}) \in \mathcal{U}_2 \setminus (\text{Dom}(\sigma'_1) \cup \text{Dom}(\sigma'_2))$ . Then,  $(\mathcal{U}_1, \sigma'_1)$  [resp.  $(\mathcal{U}_2, \sigma''_2)$ ] is isomorphic to  $(\mathbb{U}, \sigma_1)$  [resp.  $(\mathbb{U}, \sigma_2)$ ]. In particular, they have the same type i.e.,  $\text{type}^r(\mathcal{U}_1, \sigma'_1) = \text{type}^r(\mathbb{U}, \sigma_1)$  and  $\text{type}^r(\mathcal{U}_2, \sigma''_2) = \text{type}^r(\mathbb{U}, \sigma_2)$ . Moreover, we have that

$glue(encode((\mathcal{U}_1, \sigma'_1), \mathbf{v}), encode((\mathcal{U}_2, \sigma'_2), \mathbf{v})) = encode(\sigma_1 \bullet \sigma_2, \mathbf{v})$ . We compute:

$$\begin{aligned} & type^r(encode(\sigma_1 \bullet \sigma_2, \mathbf{v})) = \\ & type^r(glue(encode((\mathcal{U}_1, \sigma'_1), \mathbf{v}), encode((\mathcal{U}_2, \sigma'_2), \mathbf{v}))) = \\ & glue^\sharp(type^r(encode((\mathcal{U}_1, \sigma'_1), \mathbf{v})), type^r(encode((\mathcal{U}_2, \sigma'_2), \mathbf{v}))) = \\ & glue^\sharp(type^r(encode((\mathcal{U}_1, \sigma'_1), \mathbf{v})), type^r(encode((\mathcal{U}_2, \sigma'_2), \mathbf{v}))) = \\ & glue^\sharp(type^r(encode(\sigma_1, \mathbf{v})), type^r(encode(\sigma_2, \mathbf{v}))) \end{aligned}$$

**Lemma 11.** *Let  $k \geq 1$  be an integer,  $\phi$  be an MSO sentence,  $\sigma \in Str(\Sigma, \mathcal{D})$  be a structure and  $\mathbf{v}$  be a store such that  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} A^\tau(x_1, \dots, x_{k+1})$ ,  $\mathbf{v}(x_i) \notin \sigma(\mathcal{D})$  for all  $i \in [1, k+1]$ , and  $\mathbf{v}(x_i) \neq \mathbf{v}(x_j)$  for all  $i \neq j$ . Then,  $type^{qr(\phi)}(encode(\sigma, \mathbf{v})) = \tau$ .*

*Proof (Proof of Lemma 11).* We prove the claim by induction on the number of rule applications. The claim clearly holds for the base case, by a rule of type (7). We now consider a rule of type (5), i.e., we have  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} A^{\tau_1}(x_1, \dots, x_{k+1}) * A^{\tau_2}(x_1, \dots, x_{k+1})$ , such that  $\tau = glue^\sharp(\tau_1, \tau_2)$ . Then, there are structures  $\sigma_1$  and  $\sigma_2$  with  $\sigma_i \models_{\Delta(k, \phi)}^{\mathbf{v}} A^{\tau_i}(x_1, \dots, x_{k+1})$ , for  $i = 1, 2$  and  $\sigma_1 \bullet \sigma_2 = \sigma$ . By the inductive hypothesis, we have that  $type^{qr(\phi)}(encode(\sigma_i, \mathbf{v})) = \tau_i$ , for  $i = 1, 2$ . Because every derivation of  $\Delta(k, \phi)$  is also a derivation of  $\Delta(k)$ , obtained by removing the type annotations from the rules in  $\Delta(k, \phi)$ , we get that  $\sigma_i \models_{\Delta(k)}^{\mathbf{v}} A(x_1, \dots, x_{k+1})$ , for  $i = 1, 2$ . By Lemma 9 we have that  $Rel(\sigma_i) \subseteq \sigma_i(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ , for  $i = 1, 2$ . Since  $\sigma = \sigma_1 \bullet \sigma_2$ , we have that  $\sigma_1(\mathcal{D}) \cap \sigma_2(\mathcal{D}) = \emptyset$ . We compute

$$\begin{aligned} & type^{qr(\phi)}(encode(\sigma, \mathbf{v})) = \\ & glue^\sharp(type^{qr(\phi)}(encode(\sigma_1, \mathbf{v})), type^{qr(\phi)}(encode(\sigma_2, \mathbf{v}))) = \\ & glue^\sharp(\tau_1, \tau_2) = \tau \end{aligned}$$

by Lemma 6. We now consider rules of type (6) i.e., we have  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} \exists y . \mathcal{D}(y) * A^{\tau_1}(x_1, \dots, x_{k+1})[x_i/y]$ , for some  $i \in [1, k+1]$ , such that  $\tau = glue^\sharp(fgcst_{M+i}^\sharp(\tau_1), \rho_i)$  for the type  $\rho_i$  of some structure  $S \in Str(\{c_{M+i}\})$  with a singleton universe and  $S \Vdash \mathcal{D}(c_{M+i})$ . Then, there is an element  $u \in \mathbb{U}$ , such that  $\sigma' \models_{\Delta(k, \phi)}^{\mathbf{v}[x_i \leftarrow u]} A^{\tau_1}(x_1, \dots, x_{k+1})$ , where the structure  $\sigma'$  agrees with  $\sigma$ , except that  $\mathcal{D}$  does not hold for  $u$  in  $\sigma'$ . By the inductive hypothesis,  $type^{qr(\phi)}(encode(\sigma', \mathbf{v}[x_i \leftarrow u])) = \tau_1$ . Because every derivation of  $\Delta(k, \phi)$  is also a derivation of  $\Delta(k)$ , obtained by removing the type annotations from the rules in  $\Delta(k, \phi)$ , we get that  $\sigma' \models_{\Delta(k)}^{\mathbf{v}[x_i \leftarrow u]} A(x_1, \dots, x_{k+1})$ . By Lemma 9 we have that  $Rel(\sigma') \subseteq \sigma'(\mathcal{D}) \cup \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\} \setminus \{\mathbf{v}(x_i)\} \cup \{u\}$ . Because of  $\mathbf{v}(x_i) \neq u$  (due to the assumption  $\mathbf{v}(x_i) \notin \sigma(\mathcal{D})$ ) and because of  $\mathbf{v}(x_i) \neq \mathbf{v}(x_j)$  for all  $i \neq j$ , we get that  $encode(\sigma, \mathbf{v}) = glue(fgcst_{M+i}(encode(\sigma', \mathbf{v}[x_i \leftarrow u])), S')$ , where  $S' \in Str(\{c_{M+i}\})$  is the structure with singleton universe  $\{\mathbf{v}(x_i)\}$  and  $S' \Vdash \mathcal{D}(c_{M+i})$ . Because  $S$  is isomorphic to  $S'$ , we obtain:

$$\begin{aligned} & type^{qr(\phi)}(encode(\sigma, \mathbf{v})) = \\ & type^{qr(\phi)}(glue(fgcst_{M+i}(encode(\sigma', \mathbf{v}[x_i \leftarrow u])), S')) = \\ & glue^\sharp(type^{qr(\phi)}(fgcst_{M+i}(encode(\sigma', \mathbf{v}[x_i \leftarrow u])), type^{qr(\phi)}(S'))) = \\ & glue^\sharp(fgcst_{M+i}^\sharp(type^{qr(\phi)}(encode(\sigma', \mathbf{v}[x_i \leftarrow u])), type^{qr(\phi)}(S))) = \\ & glue^\sharp(fgcst_{M+i}^\sharp(\tau_1), \rho_i) = \tau \end{aligned}$$

**Lemma 12.** *Let  $k \geq 1$  be an integer,  $\tau$  be some  $r$ -type,  $\sigma \in \text{Str}(\Sigma, \mathcal{D})$  be a structure of treewidth  $\text{tw}(\sigma) \leq k$ , witnessed by some reduced tree decomposition  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$ , and  $\mathbf{v}$  be a store with  $\lambda(r) = \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$  and  $\mathbf{v}(x_i) \neq \mathbf{v}(x_j)$  for all  $i \neq j$ , such that  $\sigma(\mathcal{D}) = \bigcup_{n \in \mathcal{N}} \lambda(n) \setminus \{u_1, \dots, u_{k+1}\}$  and  $\text{type}^r(\text{encode}(\sigma, \mathbf{v})) = \tau$ . Then, we have  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} \mathbf{A}^\tau(x_1, \dots, x_{k+1})$ .*

*Proof (Proof of Lemma 12).* The proof proceeds by induction on the structure of  $\mathcal{T}$ . Clearly the claim holds for the base case, by a rule of type (7). For the inductive step, we assume first that the root of  $\mathcal{T}$  has two children. The subtrees  $\mathcal{T}_1$  and  $\mathcal{T}_2$  rooted in the two children induce substructures  $\sigma_1$  and  $\sigma_2$  of  $\sigma$ , where  $\langle v_1, \dots, v_{\#R} \rangle \in \sigma_i(R)$  if and only if  $\langle v_1, \dots, v_{\#R} \rangle$  is witnessed by some leaf of the respective subtree. Because  $\mathcal{T}$  is a reduced tree decomposition, there is at most one leaf that witnesses a tuple  $\langle v_1, \dots, v_{\#R} \rangle \in \sigma_i(R)$ . Hence, we have  $\sigma = \sigma_1 \bullet \sigma_2$ . Because the only elements that can appear as labels in  $\mathcal{T}_1$  and  $\mathcal{T}_2$  are  $\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})$  (as these are the labels of the root of the tree decomposition), we get that  $\text{Rel}(\sigma_1) \cap \text{Rel}(\sigma_2) \subseteq \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ . Let  $\tau_i = \text{type}^r(\text{encode}(\sigma_i, \mathbf{v}))$ , for  $i = 1, 2$ . From the inductive hypothesis, we obtain  $\sigma_i \models_{\Delta(k, \phi)}^{\mathbf{v}} \mathbf{A}^{\tau_i}(x_1, \dots, x_{k+1})$ , for  $i = 1, 2$ . Let  $\tau \stackrel{\text{def}}{=} \text{glue}^\#(\tau_1, \tau_2)$ . By Lemma 6, we obtain:

$$\begin{aligned} \text{type}^{\text{qr}(\phi)}(\text{encode}(\sigma, \mathbf{v})) &= \text{type}^{\text{qr}(\phi)}(\text{encode}(\sigma_1 \bullet \sigma_2, \mathbf{v})) = \\ \text{glue}^\#(\text{type}^{\text{qr}(\phi)}(\text{encode}(\sigma_1, \mathbf{v})), \text{type}^{\text{qr}(\phi)}(\text{encode}(\sigma_2, \mathbf{v}))) &= \text{glue}^\#(\tau_1, \tau_2) = \tau \end{aligned}$$

Hence, we now get that  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} \mathbf{A}^\tau(x_1, \dots, x_{k+1})$ , by a rule of type (5). We now assume that the root of  $\mathcal{T}$  has a single child which is not a leaf. We consider the subtree  $\mathcal{T}_1$  rooted at this child. Then, there is an element  $u \notin \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ , such that the root of  $\mathcal{T}_1$  is labeled by  $\{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{i-1}), u, \mathbf{v}(x_{i+1}), \dots, \mathbf{v}(x_{k+1})\}$ . Let  $\tau_1 \stackrel{\text{def}}{=} \text{type}^r(\text{encode}(\sigma, \mathbf{v}[x_i \leftarrow u]))$  and let  $\sigma'$  be the structure that agrees with  $\sigma$  except that we have  $\sigma'(\mathcal{D}) = \sigma(\mathcal{D}) \setminus \{u\}$ . By the inductive hypothesis, we obtain that  $\sigma' \models_{\Delta(k, \phi)}^{\mathbf{v}[x_i \leftarrow u]} \mathbf{A}^{\tau_1}(x_1, \dots, x_{k+1})$ .

Let  $\tau \stackrel{\text{def}}{=} \text{glue}^\#(\text{fgcst}_{M+i}^\#(\tau_1), \rho_i)$  for the type  $\rho_i$  of the structure  $S \in \text{Str}(\{c_{M+i}\})$  with singleton universe  $\{\mathbf{v}(x_i)\}$  and  $S \Vdash \mathcal{D}(c_{M+i})$ . Because of  $\mathbf{v}(x_i) \neq u$  and  $\mathbf{v}(x_i) \neq \mathbf{v}(x_j)$  for all  $i \neq j$ , we have  $\text{encode}(\sigma, \mathbf{v}) = \text{glue}(\text{fgcst}_{M+i}(\text{encode}(\sigma', \mathbf{v}[x_i \leftarrow u])), S)$ . We compute:

$$\begin{aligned} \text{type}^r(\text{encode}(\sigma, \mathbf{v})) &= \\ \text{type}^r(\text{glue}(\text{fgcst}_{M+i}(\text{encode}(\sigma', \mathbf{v}[x_i \leftarrow u])), S)) &= \\ \text{glue}^\#(\text{type}^r(\text{fgcst}_{M+i}(\text{encode}(\sigma', \mathbf{v}[x_i \leftarrow u])), \text{type}^r(S)) &= \\ \text{glue}^\#(\text{fgcst}_{M+i}^\#(\text{type}^r(\text{encode}(\sigma', \mathbf{v}[x_i \leftarrow u])), \text{type}^r(S)) &= \\ \text{glue}^\#(\text{fgcst}_{M+i}^\#(\tau_1), \rho_i) &= \tau \end{aligned}$$

Hence, we get that  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} \exists y. \mathcal{D}(y) * \mathbf{A}_\tau(x_1, \dots, x_{k+1})[x_i/y]$  i.e., that  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} \mathbf{A}^\tau(x_1, \dots, x_{k+1})$  by a rule of type (6).

**Proposition 6.** *Given  $k \geq 1$  and an MSO sentence  $\phi$ , for any guarded structure  $(\mathbb{U}, \sigma) \in \text{Str}(\Sigma, \mathcal{D})$ , we have: (1)  $(\mathbb{U}, \sigma) \Vdash \phi$  and  $\text{tw}(\sigma) \leq k$  if and only if (2)  $(\mathbb{U}, \sigma) \models_{\Delta(k, \phi)} \mathbf{A}_{k, \phi}()$ .*

*Proof (Proof of Proposition 6).* “(1)  $\Rightarrow$  (2)” Since  $\text{tw}(\sigma) \leq k$ , there exists a reduced tree decomposition  $\mathcal{T} = (\mathcal{N}, \mathcal{F}, r, \lambda)$  of width  $k$ , by Lemma 8. Let  $\mathbf{v}$  be a store such that  $\lambda(r) = \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$  and  $\mathbf{v}(x_i) \neq \mathbf{v}(x_j)$  for all  $i \neq j \in [1, k+1]$ . Let  $\sigma'$  be the structure that agrees with  $\sigma$  on  $\Sigma$ , and for which  $\sigma'(\mathcal{D}) = \sigma(\mathcal{D}) \setminus \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ .

We observe that  $encode(\sigma', \mathbf{v}) = \sigma$ . Thus, we obtain  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} \mathfrak{D}(x_1) * \dots * \mathfrak{D}(x_{k+1}) * A^\tau(x_1, \dots, x_{k+1})$ , by Lemma 12, for  $\tau \stackrel{\text{def}}{=} type^{qr(\phi)}(\sigma) = type^{qr(\phi)}(encode(\sigma', \mathbf{v}))$ . Since, moreover, we have assumed that  $\sigma \Vdash \phi$ , we have  $\phi \in \tau \stackrel{\text{def}}{=} type^{qr(\phi)}(\sigma)$ . Then, we obtain  $\sigma \models_{\Delta(k, \phi)} A_{k, \phi}()$ , by a rule of type (8).

“(2)  $\Rightarrow$  (1)” Since  $\sigma \models_{\Delta(k, \phi)} A_{k, \phi}()$ , there exists a store  $\mathbf{v}$  with  $\mathbf{v}(x_i) \neq \mathbf{v}(x_j)$  for all  $i \neq j \in [1, k+1]$ , such that  $\sigma \models_{\Delta(k, \phi)}^{\mathbf{v}} \mathfrak{D}(x_1) * \dots * \mathfrak{D}(x_{k+1}) * A^\tau(x_1, \dots, x_{k+1})$ , for some type  $\tau$  with  $\phi \in \tau$ , by a rule of type (8). Let  $\sigma'$  be the structure that agrees with  $\sigma$  on  $\Sigma$ , and for which  $\sigma'(\mathfrak{D}) = \sigma(\mathfrak{D}) \setminus \{\mathbf{v}(x_1), \dots, \mathbf{v}(x_{k+1})\}$ . We observe that  $encode(\sigma', \mathbf{v}) = \sigma$ . By Lemma 11, we obtain  $type^{qr(\phi)}(encode(\sigma', \mathbf{v})) = \tau$ . With  $encode(\sigma', \mathbf{v}) = \sigma$  and  $\phi \in \tau$  we then obtain  $\sigma \Vdash \phi$ . Moreover, we have  $tw(\sigma) \leq k$ , by Lemma 5, because each derivation of  $\sigma' \models_{\Delta(k, \phi)} A_{k, \phi}()$  corresponds to a derivation of  $\sigma \models_{\Delta(k)} A_k()$ , obtained by removing the type annotations from the rules in  $\Delta(k, \phi)$ .

**Lemma 7.** *Given  $r \geq 0$  and  $S = (\mathbb{U}, \sigma) \in Str(\Sigma)$ , we have  $type^r(S) = type^r(S^{2^r})$ .*

*Proof (Proof of Lemma 7).* We consider the  $r$ -round MSO Ehrenfeucht-Fraïssé game: In every round, Spoiler picks a vertex  $u$  or a set of vertices  $U$  in  $S$  or  $S^{2^r}$  and Duplicator answers with a vertex or a set of vertices in the other structure. Duplicator wins iff after  $r$ -rounds the substructures induced by the selected vertices are isomorphic. We now sketch a winning strategy for Duplicator (as the argument is standard, we leave the details to the reader): For vertices in  $Dom(\sigma)$  the Duplicator plays the same vertices in the respective other structure (this applies to single vertices as well as sets of vertices). For the vertices that do not belong to  $Dom(\sigma)$  the Duplicator selects vertices that do not belong to  $Dom(\sigma)$  in the other structure such that the selected vertices belong to the same subsets played in the previous rounds; care has to be taken for the involved cardinalities, if playing in  $S^{2^r}$  the Duplicator limits himself to at most  $2^{r-k-1}$  vertices that belong to any combination of previously played subsets or the complement of these subsets, where  $k$  is the number of previously played rounds; this choice is sufficient such that after  $r$ -rounds the substructures induced by the selected vertices are isomorphic.