Available online at www.sciencedirect.com

SCIENCE DIRECT

Information and Computation 204 (2006) 853–919

Information and Computation

www.elsevier.com/locate/ic

Recognizability, hypergraph operations, and logical types

A. Blumensath [a,1], B. Courcelle [b,*]

[a] *Universität Darmstadt, Fachbereich Mathematik, AG 1, Schloßgartenstraße 7, 64289 Darmstadt, Germany*
[b] *LaBRI Bordeaux 1 University, 351, Cours de la Libération, 33405 Talence cedex, France*

Received 16 July 2004; revised 27 October 2005
Available online 4 May 2006

Abstract

We study several algebras of graphs and hypergraphs and the corresponding notions of equational sets and recognizable sets. We generalize and unify several existing results which compare the associated equational and recognizable sets. The basic algebra on relational structures is based on disjoint union and *quantifier-free definable* operations. We expand it to an equivalent one by adding operations definable with "few quantifiers," i.e., operations that take into account *local information* about elements or tuples. We also consider monadic second-order transductions and we prove that the inverse image of a recognizable set under such a transduction is recognizable.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Monadic second-order logic; Graph operation; Equational set; Recognizable set; Monadic second-order transduction; Logical type

1. Introduction

Formal language theory studies sets of finite and infinite words and terms (usually called trees) that are finitely described by means of grammars, automata, or logical formulas. It also investigates transformations of words and terms in a similar perspective. Its scope now extends to descriptions of sets of graphs, hypergraphs, partial orders, and related combinatorial structures, and to that of transformations

* Corresponding author.
E-mail addresses: blumensath@mathematik.tu-darmstadt.de (A. Blumensath), courcell@labri.fr (B. Courcelle).
1 Work done during a postdoctoral stay at LaBRI supported by the European RTN GAMES.

0890-5401/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2005.11.006

of these objects, which we will call, as for words and terms, *transductions.* Universal algebra and logic are fundamental for developing this extension, and this article contributes to showing why.

1.1. Algebras, equational and recognizable sets

Context-free languages can be characterized as least solutions of systems of recursive equations, while regular languages can be characterized as unions of classes of finite congruences on the free monoid. Both characterizations are based on the algebraic structure on words associated with concatenation. As observed by Mezei and Wright in [30] the two notions of least solution of a system of recursive equations and of a congruence with finitely many classes are meaningful in every algebra, not only in the monoid of finite words and in the algebra of finite terms. In every algebra, they yield two families of sets, the family of *equational sets* and the family of *recognizable sets*. These notions generalize those of context-free languages and of regular languages, respectively.

The advantage of this algebraic approach, especially for describing sets of graphs, is that it depends neither on rewriting rules nor on automata. This is essential because graph rewriting rules are complicated to define and to study, and graph automata satisfying good closure and decidability properties do not exist, except for very particular classes of graphs. By contrast, the families of recognizable and equational sets of any algebra satisfy useful closure properties: the family of recognizable sets is closed under union, intersection, and difference, and the intersection of an equational set with a recognizable one is equational.

A class of graphs is made into an *algebra* by equipping it with graph operations. These operations form the *signature* of the algebra. A graph operation linking two graphs can be considered as a generalized concatenation. However, graphs can be concatenated in several ways, and different operations are specified in terms of labellings of the vertices. We will also use unary graph operations that manipulate labellings. In every algebra of graphs, we have thus equational sets and recognizable sets. Their definitions only use concepts of universal algebra and need not deal with the specific combinatorial properties of the graphs under consideration.

In the above description, we have written "graphs" for simplicity, but it equally applies to hypergraphs, partial orders, and actually all combinatorial objects represented by relational structures with a finite set of relations. For example, a graph is represented by the relational structure whose domain is the set of vertices and that has a binary relation describing the edges. (The multiplicity of edges is lost in this representation. There exists another one for graphs with multiple edges, see [4].)

Several signatures can be defined on the same class of relational structures. However, in many cases, "small" variations of the signature do not modify the classes of equational and recognizable sets, a fact indicating the robustness of the algebraic framework. We will say that two signatures are *equivalent* if the corresponding classes of equational and recognizable sets are the same. One of the purposes of this article is to investigate equivalences of signatures. Another one is to relate these algebraic notions with monadic second-order logic. We now explain the role of logic in this theory.

1.2. The role of logic

Logic is used for three purposes: first to specify the operations on relational structures in the signatures, second to define recognizable sets of relational structures, and third to specify transformations of relational structures. Let us comment each of these uses.

The basic signature of operations, denoted by $\mathcal{QF}$, consists of disjoint union, of all unary operations that can be defined by quantifier-free formulas (called *quantifier-free operations*), and of constants denoting structures with a single element. The edge complement is an example of a quantifier-free operation: the edge relation of the output graph is just the complement of the edge relation of the input graph, hence the former is definable by a formula without quantifiers in terms of the latter. Quantifier-free operations can be combined with disjoint union to form various kinds of graph concatenations.

This definition generalizes and unifies previously defined algebras, the algebra of graphs called $\mathcal{VR}$, and the algebra of hypergraphs called $\mathcal{HR}$. They have been defined in such a way that their equational sets are the sets of graphs and hypergraphs defined by certain *context-free graph grammars*, based respectively on *vertex replacement* and on *hyperedge replacement* (see [4] and other chapters of the same book on graph grammars). Many results proved independently for these two algebras can now be proved as particular instances of more general results relative to $\mathcal{QF}$.

Monadic second-order logic (MSO) is the fundamental language for defining recognizable sets and transductions of relational structures. That MSO is useful is not too surprizing given that, for sets of words and terms, MSO-definability is equivalent to definability by finite-state automata, and that many types of tree transductions can also be described by MSO-formulas (see [1,5,21]). A fundamental result says that every set of relational structures that is the set of finite models of an MSO-formula is *$\mathcal{QF}$-recognizable* (i.e., is recognizable with respect to the algebra of relational structures defined by the signature $\mathcal{QF}$). On the other hand, it is much easier to check that a property is definable by an MSO-formula than to construct a finite congruence saturating the corresponding set. In the cases of words and trees, finite-state automata offer such a convenient specification language for recognizable sets, but they do not work on graphs and, a fortiori, on relational structures. Hence MSO takes their place in a natural way. *Transducers* which define transformations of words or terms into words or terms are finite-state automata with outputs. Hence, they cannot be generalized to graphs on the basis of automata, and MSO, again, offers a powerful and easy to use specification language.

Furthermore, there are quite close connections between equational sets and recognizable sets of relational structures, and MSO-transductions: for example, a set is $\mathcal{QF}$-equational iff it is the image of a recognizable set of finite terms under an MSO-transduction, and it follows that the class of $\mathcal{QF}$-equational sets is stable under MSO-transductions. Further, we prove in this article that the inverse image of a $\mathcal{QF}$-recognizable set under an MSO-transduction is $\mathcal{QF}$-recognizable.

1.3. The main results

We will only consider *finite* terms, graphs, hypergraphs, and relational structures. Furthermore, we will consider relational structures only up to isomorphism. There are several reasons for doing so. First, we have no use for distinguishing isomorphic relational structures. This is also a requirement for applying logic since logical formulas cannot distinguish between isomorphic structures. To derive algorithms from this theory as done in [6], we need to use whenever possible finite relational signatures and we do not want to introduce infinitely many constants to describe infinitely many isomorphic structures. Hence a term will not define a single relational structure but the isomorphism class of some relational structure.

Our starting point is the signature $\mathcal{QF}$ of operations on relational structures consisting of disjoint union, quantifier-free operations (there are countably many, the use of infinite signatures for dealing with graphs, even finite ones, is unavoidable), and constants denoting relational structures having a singleton domain.

We prove in Section 5 that the inverse image of a $\mathcal{QF}$-recognizable set under an MSO-transduction is $\mathcal{QF}$-recognizable. This result, of which weak forms are already known, confirms the robustness of the formal framework associating graph operations and MSO.

In Section 6, we prove that the signature $\mathcal{QF}$ can be restricted to an equivalent subsignature. This "small" (although still countably infinite) signature is based on quantifier-free operations of three types: we can *forget* a relation R (i.e., delete all tuples in R without modifying the domain of the considered structure), *rename* a relation R into S (where R and S have same arity; if R and S are both present in the input structure then this operation merges them into a single relation), and we can *add* either a new relation T or tuples to an existing relation T (roughly, given two relations R and S we concatenate the tuples of R with those of S and add the resulting tuples to T). If the relational signature Σ contains only relations of arity at most n then we can define an equational set of Σ-structures by a system of equations where the operations only use the relations of Σ and auxiliary relations of arity at most $n-1$. In the case of graphs, that is for $n = 2$, we obtain known results about the signature $\mathcal{VR}$ (cf. [7,8]) where the auxiliary relations are unary, i.e., they encode vertex labels.

In Section 7, we develop a method for enlarging the signature $\mathcal{QF}$ to an equivalent one, and we apply this method to the *fusion* operation considered by Courcelle and Makowsky in [9]. This operation fuses all elements satisfying a given unary relation. It is not quantifier-free. Roughly speaking, we prove that adding it to $\mathcal{QF}$ yields an equivalent signature. This generalizes the results of [9].

In Section 8, we consider the algebra $\mathcal{HR}$ whose equational sets are those defined by *hyperedge replacement* context-free graph grammars. This is an algebra of relational structures with distinguished elements called *sources*. The operations consist of constants for singleton structures and *parallel composition* which combines two structures with sources into the one obtained from their disjoint union by fusing the sources with same label. One can replace a relational structure with sources by a purely relational one by introducing, for each constant c, a unary relation lab_c which contains as single element the value of c. However, if we do so, quantifier-free definable operations on relational structures with sources are no longer quantifier-free definable on the corresponding relational structures without sources. We overcome this difficulty by showing that nevertheless the operations of $\mathcal{HR}$ can be handled in the general framework of purely relational structures.

These results contribute to build a robust foundation for the extension of formal language theory to sets of graphs, hypergraphs, and relational structures. Let us say a few words about the tools we use for establishing them. The main one is the classical notion of a *logical type* used, e.g., in [2,25,26,28]. Given a finite set Φ of formulas with n free variables (for instance, the set of MSO-formulas of quantifier height at most k, up to logical equivalence), we define the Φ-*type* of an n-tuple of elements $\bar{a}$ of a relational structure as the set of those formulas of Φ that are satisfied by $\bar{a}$. There are thus finitely many possible Φ-types. If the formulas in Φ are quantifier-free or if their quantifications are restricted to a "neighbourhood" of $\bar{a}$, then the Φ-type of $\bar{a}$ encodes *local information* associated with $\bar{a}$. Given a structure $\mathfrak{A}$, its Φ-*annotation* is the structure $\mathcal{M}_\Phi(\mathfrak{A})$ with same domain where, for each Φ-type p, we have a new n-ary relation T_p containing all n-tuples of $\mathfrak{A}$ with type p. The annotation $\mathcal{M}_\Phi(\mathfrak{A})$ provides information about $\mathfrak{A}$ that is immediately available from the relations without the need to use formulas with quantifiers. In the language of database theory, this construction builds an extensional database out of an intensional one. In this article, a typical use is the following: a transduction of structures $\mathfrak{A}$, defined by MSO-formulas of quantifier height at most k can be replaced by a quantifier-free transduction acting on the annotated structures $\mathcal{M}_\Phi(\mathfrak{A})$ where Φ is the set of MSO-formulas of quantifier height at most k.

1.4. Related works

This article develops the algebraic and logical extension of formal language theory to sets of relational structures initiated by Courcelle and presented in [10] (its algebraic background) and [4] (its application to graphs and hypergraphs, and its relationships with graph grammars). This theory also uses results from [1,11,22]. Sections 6 and 8 elaborate the definition given in [7] of an algebra for relational structures with constants. Section 7 generalizes the definition of fusion given in [9] and establishes new results. Closure properties of the family of $\mathcal{HR}$-recognizable sets of hypergraphs have been studied in [12], and Section 5 continues this work. The stability of the family of recognizable sets under modifications of signatures is studied in [8], and the notion of equivalence of signatures investigated in Sections 6, 7, and 8 extends this stability requirement to also include the family of equational sets.

1.5. Summary of the article

The article is organized as follows. Section 2 reviews algebras, equational and recognizable sets, and it introduces an extension of the notion of derived operation closely related to linear deterministic bottom-up tree transductions. It also extends the notion of a homomorphism to that of a *heteromorphism,* making it possible to relate algebras of different signatures. Section 3 reviews relational structures, monadic second-order logic, monadic second-order transductions, and operations on relational structures defined by quantifier-free formulas. Section 4 introduces *monadic types* (sets of monadic second-order formulas of bounded quantifier height) as a first form of type information, and establishes several technical results. Section 5 establishes the preservation of recognizability under inverse MSO-transductions. Section 6 shows the equivalence of the basic signature $\mathcal{QF}$ on relational structures with a proper subsignature that generalizes the signature $\mathcal{VR}$ to relational structures and, hence, to hypergraphs. Section 7 takes the opposite direction. Its objective is to extend $\mathcal{QF}$ by operations that are not quantifier-free definable, but to obtain nevertheless an equivalent signature. A method for doing so is introduced and applied to the fusion operation. Section 8 shows how the operations defining the $\mathcal{HR}$-equational and $\mathcal{HR}$-recognizable sets can be studied in terms of relational structures without constants.

1.6. Notation, conventions, and general facts

In this article we only consider equational and recognizable sets of *finite* structures. The reason for this limitation is that the algebraic definitions of these notions are not well suited to infinite objects. In particular, the recognizable sets of infinite trees are not those defined by tree automata. However, our technical constructions of transformations of structures based on logical formulas work for infinite structures as well. But their algebraic consequences are only meaningful in the finite case.

All proofs in this article are effective. Hence every statement of the form "For every m, n, there exists an MSO-transduction such that ..." can be read as "There exists an algorithm that, given m, n, constructs an MSO-transduction such that ..."

Let us fix notation and introduce some conventions. The set $\mathbb{N}$ of natural numbers contains 0. We set $[k] := \{1, \dots, k\}$ and $[0] := \emptyset$. We denote by $\mathcal{P}(X)$ the power set of a set X. For an n-tuple $\bar{a} = a_1 \dots a_n$, we sometimes also write $\bar{a}$ for the set $\{a_1, \dots, a_n\}$ of its components. In particular, we sometimes write

$\bar{a} \subseteq A$ instead of $\bar{a} \in A^n$. The empty tuple is denoted by $\langle\rangle$. We will denote by $|x|$ both, the cardinality of a set x and the length of a word x. (No ambiguity will arise.)

2. Equational and recognizable sets in arbitrary algebras

The notions of an equational set and a recognizable set are due to Mezei and Wright [30]. While they were originally defined for algebras over one sort, we adapt them to the many-sorted case with infinitely many sorts. We begin with definitions concerning such algebras. We refer the reader to [10] for more about recognizable and equational sets.

2.1. Algebras

Let S be a set whose elements we call *sorts*. An S-*signature* is a set F of function symbols each of which has a *type* $s_1 \times s_2 \times \cdots \times s_n \to s$ where $s_1, \ldots, s_n, s \in S$. We may have $n = 0$; in this case the symbol is called a *constant*. We denote by $T(F, X)$ the set of finite well-formed terms built with functions from F and variables from X. They will simply be called *terms* in the following. In the case $X = \emptyset$, we simply write $T(F)$. Automata defining sets of terms are usually called *tree automata,* and multivalued mappings from terms to terms are called *tree transductions.* We will keep this standard terminology, although trees in the sense of graph theory do not coincide with terms.

An F-*algebra* is an object $M = \langle (M_s)_{s\in S}, (f_M)_{f\in F}\rangle$ where each set M_s, called the *domain* of M of sort s, is nonempty and, for every symbol $f \in F$ of type $s_1 \times \cdots \times s_n \to s$, we have a total function $f_M : M_{s_1} \times \cdots \times M_{s_n} \to M_s$. These mappings are called the *operations* of M. We assume that $M_s \cap M_{s'} = \emptyset$, for $s \neq s'$. We denote the set $\bigcup \{ M_s \mid s \in S \}$ also by M. We assume that the notions of a *homomorphism, subsignature, subalgebra,* etc. are well-known. See [10] or [8] for details.

We can define a canonical F-algebra (the *free* F-*algebra*) on the set of terms $T(F)$ such that, for every F-algebra M, there exists a unique homomorphism $\mathrm{val}_M : T(F) \to M$. For $t \in T(F)_s$, the image of t under val_M is an element of M_s, called the *value* of t in M. A term t with variables $x_1, \ldots, x_n$ of sort $s_1, \ldots, s_n$ defines a function $t_M : M_{s_1} \times \cdots \times M_{s_n} \to M$ which is obtained by replacing all function symbols f in t by the corresponding operations f_M of M. In the special case that $n = 0$ we obtain $t_M = \mathrm{val}_M(t)$.

A *derived operation of the algebra* M is an n-ary operation defined by a term in $T(F, \{x_1, \ldots, x_n\})$ where each variable x_i occurs at most once. Such terms are called *linear.* Let F and G be S-signatures and M an F-algebra. If N is a G-algebra with the same domains as M such that each operation of N is a derived operation of M then we say that N is a *derived algebra*, and that it is *derived of* M. We call G a *derived signature* of F. The signature of all derived operations of F is denoted by F^{der}.

Our notion of a derived operation is restricted to linear terms to guarantee that the class of equational sets is not changed by adding derived operations to a signature. The class of recognizable sets stays the same even if we add derived operations built from nonlinear terms.

If G is a derived signature of F every term $t \in T(G)$ can be translated into a term $\delta(t) \in T(F)$ such that $\delta(t)_N = t_M$, for all algebras M and N as above. The mapping δ is a tree transducer of a particular type, namely a deterministic, bottom-up, linear tree transducer with a single state. By a *regular* set of terms we mean a subset $K \subseteq T(F)$, for some finite signature F, that is defined by a finite-state tree automaton. Generalizing the notion of a regular set we will define below the notion of a recognizable set in an arbitrary algebra. It is an easy exercise to show that a set of terms in $T(F)$ is regular if and only if it is recognizable in the free F-algebra $T(F)$.

For definitions and basic results concerning tree automata and tree transducers, we refer the reader to the books [23] or [3], and to the surveys [24,31]. In the following we will only use finite-state deterministic, bottom-up, linear tree transducers and we will call them simply *tree transducers.* Among the basic facts we recall that the image of a regular set of terms under such a tree transducer is again regular.

Lemma 1. *If C is a regular set of terms then so is $\delta(C)$, for every tree transducer δ.*

Let us stress that, by our definition, a tree transducer is always linear. Without this condition Lemma 1 would not hold.

2.2. Recognizable and equational sets

Let F be an S-signature. We say that an F-algebra M is *locally finite* if each domain M_s is finite. (Note that in universal algebra the term "locally finite" has a different meaning.)

A *congruence* on M is an equivalence relation $\approx$ on $\bigcup\{ M_s \mid s \in S \}$ such that each set M_s is a union of equivalence classes and such that $\approx$ is stable under all operations of M. It is said to be *finite* if, for each sort s, the restriction $\approx_s$ of $\approx$ to M_s is finite, i.e., has finitely many classes. A congruence *saturates* a set $X \subseteq M$ if X is a union of equivalence classes.

Definition 2. Let M be an F-algebra and $s \in S$. A subset $X \subseteq M_s$ is *M-recognizable* if it is saturated by a finite congruence on M. We denote the set of all M-recognizable subsets of M_s by $\mathrm{Rec}(M)_s$, and the union of the sets $\mathrm{Rec}(M)_s$ by $\mathrm{Rec}(M)$.

An equivalent definition can be given in terms of homomorphisms. A subset $X \subseteq M_s$ is M-recognizable if and only if there exists a homomorphism $h : M \to A$ into a locally finite F-algebra A and a (finite) subset $Y \subseteq A_s$ such that $X = h^{-1}(Y)$. The class $\mathrm{Rec}(M)_s$ forms a boolean algebra. We have $\emptyset, M_s \in \mathrm{Rec}(M)_s$, and $X, Y \in \mathrm{Rec}(M)_s$ implies that $X \cup Y, X \cap Y, X \setminus Y \in \mathrm{Rec}(M)_s$ (see [10]).

Note that in the definition of a congruence constants play no role. Hence, a set X is recognizable with respect to an F-algebra M if and only if it is recognizable with respect to the F_{-}-reduct of M where F_{-} consists of all operations of F except for the constant symbols.

Definition 3. A subset $L \subseteq M_s$ is *M-equational* if it is a component of the least solution of a finite system of recursive equations using as operations union and the extension of the operations of F to subsets of M. We denote the class of equational subsets of M by $\mathrm{Equat}(M)$, and by $\mathrm{Equat}(M)_s$ the subclass of those included in M_s.

For instance, the equational sets of a free monoid are exactly the context-free languages. Similarly, the equational subsets of graph algebras are exactly those that are *context-free.* See [4] for the relationship between graph grammars and equational sets. Instead of the above definition we will mainly use the following characterization of M-equational sets.

Proposition 4 ([30,10])**.** *Let M be an F-algebra. A set $L \subseteq M_s$ is M-equational if and only if there exists a regular set $K \subseteq T(F)_s$ such that $L = \mathrm{val}_M(K)$.*

Note that, by definition, if $K \subseteq T(F)_s$ is a regular set of terms then there is a finite subsignature $F_0 \subseteq F$ with $K \subseteq T(F_0)_s$.

Corollary 5. *A set $K \subseteq T(F)_s$ is regular if and only if it is equational.*

In particular, if F is a finite signature that *generates* M, i.e., such that every element of M is the value of a term in $T(F)$, then every recognizable set is equational. This condition is satisfied for the usual algebras of finitely generated monoids, but not for the algebras of graphs that we will consider. See [10] for a thorough treatment of the basic results about recognizable and equational sets.

In certain cases, for instance when considering graphs, there is a canonical choice for the domains M_s, $s \in S$, while there are several possible signatures F. To simplify terminology and notation we will speak in such cases of F-equational and F-recognizable sets instead of introducing a separate name M_F for the structure obtained from the signature F and using the terms "M_F-equational" and "M_F-recognizable." Similarly we will write $\mathrm{Equat}(F)$ and $\mathrm{Rec}(F)$ instead of, respectively, $\mathrm{Equat}(M_F)$ and $\mathrm{Rec}(M_F)$.

2.3. Finite-state derived operations and homomorphisms

We will need some extensions of the classical notions of a derived operation and a homomorphism that are closely related to tree transducers.

Definition 6. Let M be an F-algebra. A mapping $\alpha : M \to X$ from M into an arbitrary set X is *M-computable* if the sets $A_s := \alpha(M_s) \subseteq X$, for $s \in S$, are finite and pairwise disjoint, and there exists an F-algebra A with domains A_s, for each s, such that $\alpha : M \to A$ is a homomorphism. In other words, the latter condition means that, for every $f \in F$ of arity n and all $a_1, \dots, a_n \in M$ of appropriate sorts, the value $\alpha(f_M(\bar{a}))$ can be computed from $\alpha(a_1), \dots, \alpha(a_n)$.

Definition 7. Let M be an F-algebra and $\alpha : M \to A$ be M-computable. An n-ary mapping $g : M_{s_1} \times \cdots \times M_{s_n} \to M_s$, $n \geqslant 1$, is a *finite-state derived operation* (*based on* α) if, for each $\bar{a} \in A^n$, there is an n-ary derived operation $t[\bar{a}]$ of M such that we have

$$g(x_1, \dots, x_n) = t[\alpha(x_1), \dots, \alpha(x_n)]_M(x_1, \dots, x_n)\,,$$

for all elements $x_1, \dots, x_n \in M$ of sorts, respectively, $s_1, \dots, s_n$.

Example 8. Let X be a set and F the signature consisting of one binary operation $\cdot$ and constant symbols ε and a, for every $a \in X$. Let M be the free monoid over X, that is, the F-algebra with domain X^* where $\cdot_M$ is concatenation, ε_M the empty word, and $a_M := a$, for $a \in X$. Fix some element $a \in X$. We define a binary operation $\circledast$ on X^* by

$$u \circledast v := \begin{cases} uv & \text{if neither } u \text{ nor } v \text{ contains an occurrence of } a, \\ a & \text{otherwise.} \end{cases}$$

We claim that $\circledast$ is a finite-state derived operation. We define an F-algebra N on $[2]$ by setting

$$i \cdot_N k := \begin{cases} 2 & \text{if } i = k = 2\,, \\ 1 & \text{otherwise}\,, \end{cases}$$

$$\varepsilon_N := 2\,, \quad a_N := 1\,, \quad \text{and} \quad b_N := 2\,, \text{ for } b \neq a\,.$$

Let $\alpha : M \to N$ be the homomorphism

$$\alpha(u) := \begin{cases} 1 & \text{if } u \text{ contains an occurrence of } a, \\ 2 & \text{otherwise.} \end{cases}$$

Then we can define $\circledast$ by the terms

$$t[1,1](x,y) := a\,, \qquad t[1,2](x,y) := a\,,$$

$$t[2,1](x,y) := a\,, \qquad t[2,2](x,y) := x \cdot y\,.$$

If M is an F-algebra and G a set of finite-state derived operations we obtain a G-algebra N with the same sorts and domains as M. We call G a *signature of finite-state derived operations,* and we call N a *finite-state derived algebra of* M. If the operations of G are all based on the same M-computable mapping α then we say that G and N are *based on* α.

For each M-computable mapping α, we denote by F^{der}_α the signature of all finite-state derived operations based on α. If F is countable then so is F^{der}_α since we require that the sets A_s are finite. Clearly, F^{der}_α contains F^{der} because the operations $t[\bar{a}]$ in the above definition may actually not depend on $\bar{a}$. Note that the operations of F^{der}_α depend on M via α, whereas those of F^{der} do not: they are defined in a purely syntactic way without reference to any algebra.

Remark 9. Let F be a finite signature, M an F-algebra, and G a finite signature of finite-state derived operations based on some M-computable mapping $\alpha : M \to A$. Let N be the associated $(F \cup G)$-algebra. For every $t \in T(F \cup G)$, there exists a term $\delta(t) \in T(F)$ with $t_N = \delta(t)_M$. This mapping δ can be defined by a tree transducer.

We will see below that adding finite-state derived operations does not change the notions of an equational or a recognizable set. Hence, when we want to compare algebras with respect to such sets we need a kind of homomorphism that is invariant under this operation. Furthermore, we will need to relate algebras with different signatures.

Definition 10. Let M be an F-algebra with set of sorts S and N a G-algebra with set of sorts S'.

(a) A *heteromorphism* $h : M \to N$ is a collection of mappings consisting of $h_{\mathrm{sort}} : S \to S'$ and $h_s : M_s \to N_{h_{\mathrm{sort}}(s)}$, for each $s \in S$, such that, for every $f \in F$ of type $s_1 \times \cdots \times s_n \to s$, there exists a linear term $t^f \in T(G, \{x_1, \ldots, x_n\})$ such that

$$h_s(f_M(b_1, \ldots, b_n)) = t^f_N(h_{s_1}(b_1), \ldots, h_{s_n}(b_n))\,,$$

for all $b_1, \ldots, b_n \in M$ of sorts $s_1, \ldots, s_n$.

(b) Let $\alpha : M \to A$ be an M-computable mapping. A collection h as above is a *finite-state heteromorphism based on* α if, for every $f \in F$ of type $s_1 \times \cdots \times s_n \to s$, there exist linear terms $t^f[\bar{a}] \in T(G, \{x_1, \ldots, x_n\})$, for $\bar{a} \in A^n$, such that

$$h_s(f_M(b_1, \ldots, b_n)) = t^f[\alpha(b_1), \ldots, \alpha(b_n)]_N(h_{s_1}(b_1), \ldots, h_{s_n}(b_n))\,,$$

for all $b_1, \ldots, b_n \in M$ of sorts $s_1, \ldots, s_n$.

In the following we will write in both cases h instead of h_{sort} or h_s, without risk of ambiguity.

Remark 11. An important special case of a (finite-state) heteromorphism consists of a function $h : M \to N$ from an F-algebra M to a G-algebra N such that there exists a set G' of (finite-state) derived operations of N that turns $h : M \to N$ into a homomorphism from M to the G'-algebra N.

Example 12. Let M be the free monoid as in the previous example.

(a) The function $h : u \mapsto \tilde{u}$ that maps every word to its mirror image is a heteromorphism. Since $\widetilde{uv} = \tilde{v}\tilde{u}$ we can choose the term $t^{\cdot}(x, y) := y \cdot x$.
(b) An example of a finite-state heteromorphism is the function

$$g(u) := \begin{cases} \tilde{u} & \text{if } u \text{ contains no occurrence of } a, \\ a^n & \text{if } u \text{ contains } n > 0 \text{ occurrences of } a. \end{cases}$$

If we again choose $\alpha : M \to [2]$ to be the homomorphism with

$$\alpha(u) := \begin{cases} 1 & \text{if } u \text{ contains an occurrence of } a, \\ 2 & \text{otherwise,} \end{cases}$$

then we can define g by the terms

$$t^g[1,1](x, y) := x \cdot y\,, \qquad t^g[1,2](x, y) := x\,,$$
$$t^g[2,1](x, y) := y\,, \qquad t^g[2,2](x, y) := y \cdot x\,.$$

Remark 13. Let $h : M \to N$ be a finite-state heteromorphism. For every term $t \in T(G)$, there exists a term $\delta(t) \in T(G)$ such that $h(t_M) = \delta(t)_N$. If the signature F of M is finite then this mapping δ can be defined by a tree transducer.

Lemma 14. *Let $h : M \to N$ be a finite-state heteromorphism based on α between an F-algebra M and a G-algebra N.*

(*a*) $L \in \mathrm{Rec}(N)$ *implies* $h^{-1}(L) \in \mathrm{Rec}(M)$.
(*b*) $L \in \mathrm{Equat}(M)$ *implies* $h(L) \in \mathrm{Equat}(N)$.

Proof. (a) Let $L \in \mathrm{Rec}(N)$ and $\approx$ be a finite G-congruence saturating L. We define a relation $\equiv$ on M by setting

$$x \equiv y \quad :\text{iff } x \text{ and } y \text{ have the same sort,} \\ h(x) \approx h(y)\,, \text{ and } \alpha(x) = \alpha(y)\,.$$

It is clear that $\equiv$ is an equivalence relation. For each sort s, it has at most $|N_{h(s)}/{\approx}| \cdot |A_s|$ classes. If $x \equiv y$ then $h(x) \in L$ implies $h(y) \in L$ since $h(x) \approx h(y)$ and $\approx$ saturates L. Consequently, $\equiv$ saturates $h^{-1}(L)$.

It remains to prove that $\equiv$ is a congruence. Let $f \in F$ be of arity n and let $\bar{x}, \bar{y} \in M^n$ with $x_i \equiv y_i$, for all i. By the definition of $\equiv$, we have $\alpha(x_i) = \alpha(y_i)$, and since α is a homomorphism it follows that $\alpha(f_M(\bar{x})) = \alpha(f_M(\bar{y}))$.

It remains to prove that $h(f_M(\bar{x})) \approx h(f_M(\bar{y}))$. We have

$$\begin{aligned} h(f_M(\bar{x})) &= t^f[\alpha(x_1), \dots, \alpha(x_n)]_N(h(x_1), \dots, h(x_n)) \\ &= t^f[\alpha(y_1), \dots, \alpha(y_n)]_N(h(x_1), \dots, h(x_n)) \\ &\qquad\qquad\qquad\qquad\qquad\qquad (\text{since } \alpha(x_i) = \alpha(y_i)) \\ &\approx t^f[\alpha(y_1), \dots, \alpha(y_n)]_N(h(y_1), \dots, h(y_n)) \\ &\qquad\qquad\qquad\qquad\qquad\qquad (\text{since } h(x_i) \approx h(y_i)) \\ &= h(f_M(\bar{y})), \end{aligned}$$

which completes the proof.

(b) Each set $L \in \mathrm{Equat}(M)$ can be written $L = \mathrm{val}_M(K)$, for some regular set of terms $K \subseteq T(F)$ (see Proposition 4). We have remarked that there exists a tree transducer δ associated with h such that

$$\mathrm{val}_N(\delta(t)) = h(\mathrm{val}_M(t)), \qquad \text{for all } t \in T(F).$$

Hence $h(L) = \mathrm{val}_N(\delta(K))$. Since, by Lemma 1, tree transducers preserve regularity it follows that $h(L)$ is N-equational. □

Definition 15. Let F and G be S-signatures for some set of sorts S and $M = (M_s)_{s \in S}$ a family of domains. Let M_F and M_G be algebras with the same family of domains M and signatures F and G, respectively. We say that M_F and M_G are *equivalent* if

$$\mathrm{Equat}(M_F) = \mathrm{Equat}(M_G) \quad \text{and} \quad \mathrm{Rec}(M_F) = \mathrm{Rec}(M_G).$$

If M_F and M_G are understood from the context we will simply say that F and G are *equivalent signatures.*

Remark 16. For $F \subseteq G$ we obviously always have

$$\mathrm{Equat}(F) \subseteq \mathrm{Equat}(G) \quad \text{and} \quad \mathrm{Rec}(G) \subseteq \mathrm{Rec}(F).$$

Hence, when testing for equivalence we only need to check the converse inclusions.

Consider an F-algebra M and let G be a signature of finite-state derived operations of M that are all based on the same M-computable mapping α (cf. Definition 7). It follows from the next lemma that $F \cup G$ is equivalent to F.

Lemma 17. *Let M be an F-algebra. For every M-computable function $\alpha : M \to A$, the signature F_α^{der} is equivalent to F.*

Proof. If $\approx$ is a finite F-congruence on M then the equivalence relation defined by

$$x \equiv y \quad :\text{iff} \quad x \text{ and } y \text{ are of the same sort, } \alpha(x) = \alpha(y), \text{ and } x \approx y$$

is a finite F_α^{der}-congruence. (The proof is the same as in Lemma 14 (a).) Hence, if $\approx$ witnesses the F-recognizability of some set L then $\equiv$ witnesses the F_α^{der}-recognizability of L. It follows that $\text{Rec}(F_\alpha^{\text{der}}) = \text{Rec}(F)$.

Suppose that L is F_α^{der}-equational. Then we have $L = \text{val}_M(K)$ for some regular subset $K \subseteq T(F_\alpha^{\text{der}})$. We have noted that there exists a tree transducer δ such that

$$\text{val}_M(\delta(t)) = \text{val}_{M'}(t)\,, \qquad \text{for all } t \in T(F_\alpha^{\text{der}})\,,$$

where M' is the F_α^{der}-algebra with same domains as M. Hence, $L = \text{val}_M(\delta(K))$ and since tree transducers preserve regularity it follows that L is F-equational. Consequently, we have $\text{Equat}(F_\alpha^{\text{der}}) = \text{Equat}(F)$. $\square$

3. Relational structures and monadic second-order logic

A *relational signature* is a finite set $\Sigma = \{R, S, T, \dots\}$ of relation symbols each of which is given with an *arity* $\text{ar}(R) \geqslant 1$. We denote by $\text{STR}[\Sigma]$ the set of all finite Σ-*structures* $\mathfrak{A} = \langle A, (R_\mathfrak{A})_{R\in\Sigma}\rangle$ where $R_\mathfrak{A} \subseteq A^{\text{ar}(R)}$. The set A is called the *domain* of $\mathfrak{A}$. The arity of Σ is the maximal arity of a symbol in Σ. We denote it by $\text{ar}(\Sigma)$. The arity of a Σ-structure $\mathfrak{A}$ is the arity of its signature Σ.

Intuitively, a Σ-structure $\mathfrak{A}$ can be seen as a *directed hypergraph* where A is the set of vertices and, for every tuple $\bar{a} \in R$, we have a *hyperedge* with label R and sequence of vertices $\bar{a}$.

For a relational Σ-structure $\mathfrak{A}$ and a set $X \subseteq A$, we denote by $\mathfrak{A}[X]$ the substructure of $\mathfrak{A}$ *induced* by X. This is the structure with domain X and relations

$$R_{\mathfrak{A}[X]} = R_\mathfrak{A} \cap X^{\text{ar}(R)}\,, \qquad \text{for } R \in \Sigma\,.$$

A graph G is defined as an $\{edg\}$-structure $G = \langle V_G, edg_G\rangle$ where V_G is the set of vertices of G and $edg_G \subseteq V_G \times V_G$ is a binary relation representing the directed edges. For undirected graphs, the relation edg_G is symmetric. In particular, graphs are always *simple,* i.e., without parallel edges.

A term $t \in T(F)$ where F is a finite signature of symbols of arity at most k can be seen as a directed labelled tree. We encode such a tree by a relational structure of the form $\mathfrak{S}(t) := \langle N, (\text{suc}_i)_{1\leqslant i\leqslant k}, \text{rt}, (\text{lab}_f)_{f\in F}\rangle$ where

- N is the set of nodes,
- $\text{suc}_i(x, y)$ holds iff y is the ith successor of x,
- $\text{rt}(x)$ holds iff x is the root, and
- $\text{lab}_f(x)$ holds iff the node x has label f.

We denote by $\Delta(F)$ the relational signature of this structure.

We recall that *monadic second-order logic* extends first-order logic by set variables, quantification over set variables, and new atomic formulas of the form $x \in X$ that express the membership of an element x in the set X. We will denote by $\text{MSO}[\Sigma, W]$ the set of all MSO-formulas over the signature Σ with free variables in W. Similarly, $\text{FO}[\Sigma, W]$ is the set of first-order formulas and $\text{QF}[\Sigma, W]$ denotes the set of *quantifier-free formulas*. Frequently, we will omit the parameters Σ and W if their values are obvious from the context.

The *quantifier height* of a formula φ, either first-order or monadic second-order, is the maximal number of nested quantifiers in φ. We denote it by $\mathrm{qh}(\varphi)$. The quantifier-free formulas are those of quantifier height 0.

A subset $C \subseteq \mathrm{STR}[\Sigma]$ is MSO-*definable* if there is some formula $\varphi \in \mathrm{MSO}[\Sigma, \emptyset]$ such that

$$C = \{\, \mathfrak{A} \in \mathrm{STR}[\Sigma] \mid \mathfrak{A} \models \varphi \,\}\,.$$

3.1. Transductions of relational structures

We will use logic for several purposes. First, we use formulae to define transformations on structures and second, we label structures by logical types that encode properties of tuples. Let C and D be sets of structures. A *transduction* $g : C \to D$ is a binary relation $g \subseteq C \times D$ that we consider as a multivalued partial mapping associating with certain structures in C one or more structures in D.

An MSO-*transduction* is a transduction specified by MSO-formulas. Given a structure $\mathfrak{A}$ and a tuple of *parameters* $W_1, \ldots, W_n \subseteq A$ it constructs a new structure $\mathfrak{B}$ whose domain is a subset of $A \times [k]$, for some $k \geqslant 1$. Such a transduction g has an associated *backwards translation,* a mapping that effectively transforms an MSO-formula φ over $\mathfrak{B}$ (possibly with free variables) into an MSO-formula φ^g over $\mathfrak{A}$ whose free variables correspond to those of φ (k times as many actually) together with those for the parameters. The formula φ^g expresses in $\mathfrak{A}$ the property of $\mathfrak{B}$ defined by φ. We now give some details. See also [5,4].

Definition 18.

Let Σ and Γ be two relational signatures and let W be a finite set of set variables called *parameters*.

(a) A *definition scheme* (from Σ to Γ) is a tuple of formulas of the form

$$\mathcal{D} = (\varphi, \psi_1, \ldots, \psi_k, (\vartheta_w)_{w \in \Gamma \boxtimes k})$$

where $k > 0$:

$$\begin{aligned}
&\Gamma \boxtimes k := \left\{ (R, \bar{\imath}) \;\middle|\; R \in \Gamma,\ \bar{\imath} \in [k]^{\mathrm{ar}(R)} \right\}, && \\
&\varphi \in \mathrm{MSO}[\Sigma, W]\,, && \\
&\psi_i \in \mathrm{MSO}[\Sigma, W \cup \{x_1\}]\,, && \text{for } i = 1, \ldots, k\,, \\
\text{and}\quad &\vartheta_w \in \mathrm{MSO}[\Sigma, W \cup \{x_1, \ldots, x_{\mathrm{ar}(R)}\}]\,, && \text{for } w = (R, \bar{\imath}) \in \Gamma \boxtimes k\,.
\end{aligned}$$

(b) Let $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ and let γ be a W-assignment in $\mathfrak{A}$. We say that $\mathcal{D}$ *defines* the Γ-structure $\mathfrak{B}$ in $(\mathfrak{A}, \gamma)$ if

(i) $(\mathfrak{A}, \gamma) \models \varphi$,

(ii) $B = \left\{ (a, i) \in A \times [k] \;\middle|\; (\mathfrak{A}, \gamma) \models \psi_i(a) \right\}$,

(iii) for each $R \in \Gamma$,

$$R_{\mathfrak{B}} = \left\{ ((a_1, i_1), \ldots, (a_n, i_n)) \in B^n \;\middle|\; (\mathfrak{A}, \gamma) \models \vartheta_{R, \bar{\imath}}(a_1, \ldots, a_n) \right\},$$

where $\bar{\imath} = i_1 \ldots i_n$ and $n = \mathrm{ar}(R)$.

(By $(\mathfrak{A}, \gamma) \models \vartheta(a_1, \dots, a_n)$ we mean $(\mathfrak{A}, \gamma') \models \vartheta$ where γ' is the assignment extending γ such that $\gamma'(x_i) = a_i$, for all $i \leqslant n$.) Note that we do not redefine equality (in contrast to, e.g. [2]). Two elements of B are equal if they are equal as elements of $A \times [k]$.

The structure $\mathfrak{B}$ is uniquely determined by $\mathfrak{A}$, γ, and $\mathcal{D}$ whenever it is defined, i.e., whenever $(\mathfrak{A}, \gamma) \models \varphi$. Therefore, we can use a functional notation and we write $\mathfrak{B} = \hat{\mathcal{D}}(\mathfrak{A}, \gamma)$. The relation

$$\left\{ \left(\mathfrak{A},\ \hat{\mathcal{D}}(\mathfrak{A}, \gamma)\right) \,\middle|\, \gamma \text{ is some } W\text{-assignment in } \mathfrak{A} \right\} \subseteq \mathrm{STR}[\Sigma] \times \mathrm{STR}[\Gamma]$$

is called the *transduction defined by* $\mathcal{D}$.

Let L be some fragment of MSO. A transduction $g \subseteq \mathrm{STR}[\Sigma] \times \mathrm{STR}[\Gamma]$ is an *L-transduction* if it is defined (up to isomorphism) by some definition scheme $\mathcal{D}$ consisting of formulas from L. In the case where $W = \emptyset$, we say that g is *parameterless*. (Note that parameterless transductions are functional.) We will refer to the integer k by saying that $\mathcal{D}$ is *k-copying*. If $k = 1$ we will call $\mathcal{D}$ and $\hat{\mathcal{D}}$ *noncopying.* A noncopying definition scheme has the simple form $(\varphi, \psi, (\vartheta_R)_{R \in \Gamma})$.

The *quantifier height* of a definition scheme is the maximal quantifier height of the formulas it consists of. Since, up to logical equivalence, there are only finitely many MSO-formulas of a given quantifier height $k \in \mathbb{N}$, it follows that the number of MSO-transductions (defined by schemes) of quantifier height k is finite.

Note that since logical equivalence is not decidable one cannot effectively select representatives of each class of logically equivalent formulas. However, one can replace logical equivalence by a decidable finer equivalence relation that still has only finitely many classes. A construction is given in [8].

Example 19. As an example we recall from [9], Lemma 2.1, that if we have an MSO-definable equivalence relation $\approx$ on $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ then there is an MSO-transduction mapping $\mathfrak{A} = \langle A, (R_{\mathfrak{A}})_{R \in \Sigma} \rangle$ to its quotient structure

$$\mathfrak{A}/{\approx} := \langle A/{\approx}, (R_{\mathfrak{A}/\approx})_{R \in \Sigma} \rangle ,$$

where $R_{\mathfrak{A}/\approx} := \left\{ ([a_1], \dots, [a_n]) \,\middle|\, (a_1, \dots, a_n) \in R_{\mathfrak{A}} \right\}$ and $[a]$ denotes the equivalence class of a. Note that $\mathfrak{A}/{\approx}$ can be defined from $\mathfrak{A}$ with the help of any set $X \subseteq A$ containing exactly one representative of every $\approx$-class. Therefore, we can write down a noncopying definition scheme with one parameter X where the formula φ states that X contains one representative of every $\approx$-class and $\psi(x)$ is the formula $x \in X$. We omit routine details.

Let F and G be finite signatures. By encoding terms as labelled trees we can consider a mapping from $T(F)$ to $T(G)$ as a transduction between relational structures. Similarly, mappings from $T(F)$ to $\mathrm{STR}[\Sigma]$ can also be given by transductions.

Every transduction defined by a tree transducer can be represented by a parameterless MSO-transduction (see [1,21]). The fact that we only consider linear tree transducers is here essential.

On several occasions we will use transductions that transform a structure into the substructure induced by a definable subset X of its domain. If $\psi(x)$ is a formula with a single free variable we denote by del_ψ the transduction that eliminates all elements satisfying ψ.

3.2. *The fundamental property of* MSO*-transductions*

Every definition scheme $\mathcal{D}$ does not only define an MSO-transduction between structures but it also gives rise to a translation of formulas. The following proposition says that if $\mathfrak{B} = \hat{\mathcal{D}}(\mathfrak{A}, \gamma)$ then all monadic second-order definable properties of $\mathfrak{B}$ can be expressed by monadic second-order formulas over $\mathfrak{A}$. The usefulness of MSO-transductions is based on this fact.

Let $\mathcal{D} = (\varphi, \psi_1, \ldots, \psi_k, (\vartheta_w)_{w\in\Gamma\boxtimes k})$ be a definition scheme with a set of parameters W. Given a set V of set variables disjoint from W we introduce new variables $X^{(i)}$, for $X \in V$ and $i \in [k]$, and we set $V^{(k)} := \{ X^{(i)} \mid X \in V,\ i \in [k] \}$.

Let $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ be a structure. For every mapping $\eta : V^{(k)} \to \mathcal{P}(A)$, we define $\eta^k : V \to \mathcal{P}(A \times [k])$ by

$$\eta^k(X) := \eta(X^{(1)}) \times \{1\} \cup \cdots \cup \eta(X^{(k)}) \times \{k\}\,.$$

Let $Y = \{y_1, \ldots, y_r\}$ be a set of first-order variables. For a mapping $\mu : Y \to A$ and an r-tuple $\bar{\imath} = i_1 \ldots i_r \in [k]^r$, we denote by $\mu_{\bar{\imath}} : Y \to A \times [k]$ the function with

$$\mu_{\bar{\imath}}(y_j) := (\mu(y_j), i_j)\,.$$

If $k = 1$ then we identify $A \times [1]$ with A and $\mu_{1\ldots1}$ with μ.

Proposition 20 ([13,12])**.** *Let $\mathcal{D}$ be a k-copying definition scheme from Σ to Γ of quantifier height m with set of parameters W. Let V be a finite set of set variables and $Y = \{y_1, \ldots, y_r\}$ a set of first-order variables.*

For every formula $\beta \in \mathrm{MSO}[\Gamma, V \cup Y]$ and all $\bar{\imath} \in [k]^r$, one can effectively construct a formula $\beta^{\mathcal{D}}_{\bar{\imath}} \in \mathrm{MSO}[\Sigma, V^{(k)} \cup Y \cup W]$ of quantifier height

$$\mathrm{qh}(\beta^{\mathcal{D}}_{\bar{\imath}}) \leqslant k \cdot \mathrm{qh}(\beta) + m$$

such that, for each $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ and all assignments $\gamma : W \to \mathcal{P}(A)$, $\eta : V^{(k)} \to \mathcal{P}(A)$, and $\mu : Y \to A$, we have

$$\begin{aligned}(\mathfrak{A}, \eta \cup \gamma \cup \mu) \models \beta^{\mathcal{D}}_{\bar{\imath}} \text{ iff } & \hat{\mathcal{D}}(\mathfrak{A}, \gamma) \textit{ is defined, } \eta^k \cup \mu_{\bar{\imath}} \textit{ is a} \\ & (V \cup Y)\textit{-assignment in } \hat{\mathcal{D}}(\mathfrak{A}, \gamma), \textit{ and} \\ & \big(\hat{\mathcal{D}}(\mathfrak{A}, \gamma), \eta^k \cup \mu_{\bar{\imath}}\big) \models \beta\,.\end{aligned}$$

Proof. Let $\mathcal{D} = (\varphi, \psi_1, \ldots, \psi_k, (\vartheta_w)_{w\in\Gamma\boxtimes k})$. For every monadic second-order formula $\beta(y_1, \ldots, y_r, X_1, \ldots, X_s)$ and all tuples $\bar{\imath} \in [k]^r$, we define a formula $\beta^*_{\bar{\imath}}$ with first-order variables $y_1, \ldots, y_r$ and set variables $X^{(i)}_j$, for $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant s$, by induction on β. W.l.o.g. we may assume that β does not contain universal quantifiers and conjunctions. In the atomic case we set

$$\begin{aligned}(x = y)^*_{ij} &:= x = y\,,\\ (x \in X)^*_i &:= x \in X^{(i)}\,,\\ (R\bar{x})^*_{\bar{\imath}} &:= \vartheta_{R,\bar{\imath}}(\bar{x})\,,\end{aligned}$$

boolean operations remain unchanged

$$(\neg\beta)^*_{\bar{\imath}} := \neg\beta^*_{\bar{\imath}}\,,$$
$$(\beta \vee \gamma)^*_{\bar{\imath}} := \beta^*_{\bar{\imath}} \vee \gamma^*_{\bar{\imath}}\,,$$

and for quantifiers we define

$$(\exists y_{r+1}\beta)^*_{\bar{\imath}} := \bigvee_{j\in[k]} \exists y_{r+1}(\psi_j(y_{r+1}) \wedge \beta^*_{\bar{\imath}j})\,,$$
$$(\exists X\beta)^*_{\bar{\imath}} := \exists X^{(1)} \cdots \exists X^{(k)} \beta^*_{\bar{\imath}}\,.$$

Note that in the case of a second-order quantifier $\exists X\beta$ we do not need to add the condition that every $x \in X^{(i)}$ satisfies ψ_i since set variables X are only used in atomic formulas of the form $y \in X$ and we require that every y satisfies the corresponding ψ_i.

To conclude the proof we can set $\beta^{\mathcal{D}}_{\bar{\imath}} := \beta^*_{\bar{\imath}} \wedge \varphi \wedge \bigwedge_{j=1}^{r} \psi_{i_j} y_j$. The construction ensures that $\mathrm{qh}(\beta^{\mathcal{D}}_{\bar{\imath}}) \leqslant k \cdot \mathrm{qh}(\beta) + m$. (We can slightly improve this bound to

$$\mathrm{qh}(\beta^{\mathcal{D}}_{\bar{\imath}}) \leqslant k \cdot \mathrm{qh}_2(\beta) + \mathrm{qh}_1(\beta) + m\,,$$

by distinguishing between the quantifier heights $\mathrm{qh}_1(\beta)$ and $\mathrm{qh}_2(\beta)$ of first-order and second-order quantifiers.) □

Note that, even if $\mathfrak{B} = \hat{\mathcal{D}}(\mathfrak{A}, \gamma)$ is well-defined, the mapping η^k is not necessarily a V-assignment in $\mathfrak{B}$ because $\eta^k(X)$ may not be a subset of the domain of $\mathfrak{B}$.

We call $\beta^{\mathcal{D}}_{\bar{\imath}}$ the *backwards translation* of β relative to the transduction $\mathcal{D}$. If g is the transduction defined by $\mathcal{D}$ then we also write β^g instead of $\beta^{\mathcal{D}}$. For $k = 1$ and $r \geqslant 1$, we abbreviate $\beta^{\mathcal{D}}_{1\dots1}$ by $\beta^{\mathcal{D}}$. Similarly, we write $\beta^{\mathcal{D}}$ instead of $\beta^{\mathcal{D}}_{\langle\rangle}$.

Proposition 21 ([13,12])**.**

(1) *The inverse image of an* MSO*-definable class of structures under an* MSO*-transduction is* MSO*-definable. The domain of an* MSO*-transduction is* MSO*-definable.*
(2) *The composition of two* MSO*-transductions is an* MSO*-transduction.*

We prove a special case of the second statement.

Lemma 22. *Let* $f : \mathrm{STR}[\Gamma] \to \mathrm{STR}[\Delta]$ *and* $g : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ *be* MSO*-transductions of quantifier height m and n, respectively, and suppose that g is noncopying.*

Then $f \circ g$ *is an* MSO*-transduction of quantifier height at most* $m + n$*. Furthermore, if both f and g are parameterless and noncopying then so is* $f \circ g$*.*

Proof. Let $\mathcal{D} = (\varphi, \psi_1, \dots, \psi_k, (\vartheta_w)_{w\in\Delta\boxtimes k})$ be the definition scheme of f. We obtain a definition scheme of $f \circ g$ consisting of

$$\big(\varphi^g,\ \psi^g_1, \dots, \psi^g_k,\ (\vartheta^g_w)_{w\in\Delta\boxtimes k}\big)\,.$$

By Proposition 20, the quantifier height of these formulas is bounded by $m + n$. The second claim also follows easily. □

3.3. Operations on relational structures

Let us introduce the basic operations on relational structures that constitute the standard signature $\mathcal{QF}$ to which we will compare other signatures.

Disjoint union. The *disjoint union* $\mathfrak{A} \oplus \mathfrak{B}$ of two structures $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ and $\mathfrak{B} \in \mathrm{STR}[\Gamma]$ is the structure $\mathfrak{C} \in \mathrm{STR}[\Sigma \cup \Gamma]$ whose domain $C := A \cup B$ is the disjoint union of A and B and, for each relation $R \in \Sigma \cup \Gamma$, we have $R_{\mathfrak{C}} := R_{\mathfrak{A}} \cup R_{\mathfrak{B}}$ where we set $R_{\mathfrak{A}} := \emptyset$ for $R \in \Gamma \setminus \Sigma$, and $R_{\mathfrak{B}} := \emptyset$ for $R \in \Sigma \setminus \Gamma$. (We are only interested in properties of structures up to isomorphism. Hence we can freely replace structures by isomorphic copies.)

Quantifier-free operations. A *quantifier-free definition scheme* is a parameterless noncopying definition scheme $\mathcal{D} = (\varphi, \psi, (\vartheta_R)_{R \in \Gamma})$ where $\varphi = \text{true}$ and the formulas ψ and ϑ_R, for $R \in \Gamma$, are quantifier-free. The transduction $\hat{\mathcal{D}} : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ defined by such a scheme is total and functional. When considered to be part of a signature, we will call functions of this form *quantifier-free operations.* (We keep the term transduction for transformations of structures that are, typically, encodings relating different classes of relational structures.)

Note that since we require $\varphi = \text{true}$ not every parameterless noncopying definition scheme of quantifier height 0 defines a quantifier-free operation. By inspecting the proof of Lemma 22, one easily sees that the composition of two quantifier-free operations is again a quantifier-free operation.

Example 23. The *edge complement* for simple, loop-free, undirected graphs can be defined as the quantifier-free operation where

$$\vartheta_{edg}(x_1, x_2) := x_1 \neq x_2 \wedge \neg edg(x_1, x_2)\,.$$

Another edge complement could be defined for graphs with loops by deleting $x_1 \neq x_2$ in the above formula.

Remark 24. To shorten notation we will usually omit defining formulas ϑ_R of the form $\vartheta_R = R\bar{x}$ $(= Rx_1 \ldots x_{\mathrm{ar}(R)})$ that do not modify the relations R.

If we have a quantifier-free definition scheme of the form $\mathcal{D} = (\text{true}, \psi, (\vartheta_R)_{R \in \Sigma})$ where $\Gamma = \Sigma$ and ϑ_R is $Rx_1 \ldots x_{\mathrm{ar}(R)}$, for all $R \in \Sigma$, then we say that $\hat{\mathcal{D}}$ is a (*quantifier-free*) *domain restriction.* In this case we have $\hat{\mathcal{D}} = \mathrm{del}_{\neg\psi}$ and $\hat{\mathcal{D}}(\mathfrak{A})$ is the substructure of $\mathfrak{A}$ induced by the set of elements satisfying ψ.

If, on the other hand, $\mathcal{D} = (\text{true}, \text{true}, (\vartheta_R)_{R \in \Gamma})$, then we call $\hat{\mathcal{D}}$ *nondeleting*. Then the structure $\hat{\mathcal{D}}(\mathfrak{A})$ has the same domain as $\mathfrak{A}$ but its relations are redefined by the formulas ϑ_R. Other examples will be given in Section 3.5 below.

Lemma 25. *Every quantifier-free operation is the composition of a quantifier-free domain restriction and a nondeleting quantifier-free operation.*

Proof. For every quantifier-free definition scheme $\mathcal{D} = (\text{true}, \psi, (\vartheta_R)_{R\in\Gamma})$ from Σ to Γ we have $\hat{\mathcal{D}} = \hat{\mathcal{D}}' \circ \text{del}_{\neg\psi}$ where

$$\text{del}_{\neg\psi} := (\text{true}, \psi, (R\bar{x})_{R\in\Sigma}) \quad \text{and} \quad \mathcal{D}' := (\text{true}, \text{true}, (\vartheta_R)_{R\in\Gamma}) . \qquad \square$$

3.4. The many-sorted algebra of relational structures

We define an algebra STR of relational structures as follows. Suppose that Σ_∞ is a fixed relational signature with countably many symbols of each arity. We assume that every relational signature Σ is a subset of Σ_∞. We regard every signature $\Sigma \subseteq \Sigma_\infty$ as a sort of STR. The corresponding domain (of sort Σ) is the set $\text{STR}[\Sigma]$ of all finite Σ-structures.

The operations consist of the disjoint union $\oplus$ and all quantifier-free operations. Furthermore, we add constant symbols for all *singleton structures,* that is, structures whose domain contains exactly one element. Note that every set $\text{STR}[\Sigma]$ contains only finitely many of them (up to isomorphism).

This signature, which we denote by $\mathcal{QF}$, will be our reference signature for the algebra STR. We will construct alternative equivalent signatures.

If $\Sigma \subseteq \Gamma$ we could regard structures $\mathfrak{A} \in \text{STR}[\Sigma]$ as elements of $\text{STR}[\Gamma]$ where all relations $R \in \Gamma \setminus \Sigma$ are empty. However we will distinguish $\mathfrak{A}$ from its expansions, so the sets $\text{STR}[\Sigma]$ are pairwise disjoint. The natural inclusion $i : \text{STR}[\Sigma] \to \text{STR}[\Gamma]$ is a quantifier-free operation. In particular, $i \in \mathcal{QF}$. The operation symbol $\oplus$ is overloaded. It actually represents countably many binary operations, one for each pair of sorts.

According to our general definitions we obtain the classes Equat(STR) and Rec(STR) of all $\mathcal{QF}$-equational and $\mathcal{QF}$-recognizable sets. Since $\mathcal{QF}$ is our standard signature we will call such sets simply *equational* and *recognizable.*

Proposition 26 ([7,4])**.** *Let* $C \subseteq \text{STR}[\Sigma]$.

(*a*) *If* C *is* MSO-*definable then* $C \in \text{Rec}(\text{STR})_\Sigma$.
(*b*) *If* $C \in \text{Rec}(\text{STR})_\Sigma$ *and* $D \subseteq \text{STR}[\Sigma]$ *is* MSO-*definable then* $C \cap D \in \text{Rec}(\text{STR})_\Sigma$.
(*c*) *If* $\Sigma \subseteq \Gamma$ *and* $i : \text{STR}[\Sigma] \to \text{STR}[\Gamma]$ *is the inclusion map then we have* $C \in \text{Rec}(\text{STR})_\Sigma$ *iff* $i(C) \in \text{Rec}(\text{STR})_\Gamma$.

Proposition 27 ([7,4,22])**.** *Let* $C \subseteq \text{STR}[\Sigma]$. *The following statements are equivalent:*

(*i*) $C \in \text{Equat}(\text{STR})_\Sigma$.
(*ii*) $C = \text{val}_{\text{STR}}(K)$, *for some* $K \in \text{Rec}(T(\mathcal{QF})_\Sigma)$.
(*iii*) $C = \tau(L)$, *for some* MSO-*transduction* $\tau : \text{STR}[\Delta(F)] \to \text{STR}[\Sigma]$ *and some regular set of terms* $L \subseteq T(F)$ (*over an arbitrary finite signature* F).

Corollary 28. *Let* $C \in \text{Equat}(\text{STR})_\Sigma$.

(*a*) *If* $\tau : \text{STR}[\Sigma] \to \text{STR}[\Gamma]$ *is an* MSO-*transduction then* $\tau(C) \in \text{Equat}(\text{STR})_\Gamma$.
(*b*) *If* $D \subseteq \text{STR}[\Sigma]$ *is* MSO-*definable then* $C \cap D \in \text{Equat}(\text{STR})_\Sigma$.
(*c*) *If* $\Sigma \subseteq \Gamma$ *and* $i : \text{STR}[\Sigma] \to \text{STR}[\Gamma]$ *is the inclusion map then we have* $C \in \text{Equat}(\text{STR})_\Sigma$ *iff* $i(C) \in \text{Equat}(\text{STR})_\Gamma$.

Proof.

(a) If $C \in \mathrm{Equat}(\mathrm{STR})_\Sigma$ then there exists a regular set of terms L and an MSO-transduction σ such that $C = \sigma(L)$. Hence, $\tau(C) = (\tau \circ \sigma)(L)$ and Proposition 21 implies that $\tau(C) \in \mathrm{Equat}(\mathrm{STR})_\Gamma$.
(b) If D is MSO-definable then the identity function $\mathrm{id}_D : D \to D$ is an MSO-transduction. Since $C \cap D = \mathrm{id}_D(C)$ the claim follows from (a).
(c) follows immediately from (a) since i and its inverse are MSO-transductions. □

3.5. $\mathcal{VR}$-operations on graphs

Let us consider the special case of graphs. We recall the definitions of two algebras of graphs, called VR and VR^p, which are connected to certain context-free graph grammars and to the graph complexity measure called *clique width* (see [4,14,6]). We show that these algebras can be considered as subalgebras of STR. In addition to the edge relation *edg* we fix a countable set Π_∞ of unary relation symbols that we will use as vertex labels. The algebra of graphs VR has domains of the form $\mathrm{STR}[\{edg\} \cup \Pi]$, for finite $\Pi \subseteq \Pi_\infty$. The corresponding structures are labelled graphs $G = \langle V_G, edg_G, (P_G)_{P \in \Pi}\rangle$ where a vertex v has label P iff it belongs to the set P_G. Hence a vertex may have no, one, or several labels.

We define a signature $\mathcal{VR}$ that, apart from the disjoint union $\oplus$ and constant symbols for the basic graphs with a single vertex, contains the following particular quantifier-free operations. The mapping $\mathrm{ren}_{P\to Q}$ changes every label P to Q, the operation fgt_P (forget P) deletes every label P, and $\mathrm{add}_{P,Q}$, for $P \neq Q$, is defined by the quantifier-free definition where

$$\vartheta_{edg}(x_1, x_2) := edg(x_1, x_2) \vee (Px_1 \wedge Qx_2)\,.$$

Hence $\mathrm{add}_{P,Q}$ adds a new directed edge from each vertex labelled by P to each vertex labelled by Q – unless there exists already one (we deal with simple directed graphs, possibly with loops).

A more restricted algebra of labelled graphs is VR^p. A Π-*graph* is a structure $G = \langle V_G, edg_G, (P_G)_{P\in\Pi}\rangle$ in $\mathrm{STR}[\{edg\} \cup \Pi]$ such that the unary relations form a partition of the domain. (The superscript p refers to this fact.) Hence every vertex has one and only one label. The above defined operations, except fgt_P, preserve this property. (Of course, we have to omit those constant symbols which define labelled graphs that are not Π-graphs.)

For each set Π, we denote by $\mathcal{VR}^\mathrm{p}_\Pi$ the signature

$$\{\,\mathbf{P}, \mathbf{P}^{\mathrm{loop}}, \oplus, \mathrm{add}_{P,Q}, \mathrm{ren}_{P\to Q} \mid P, Q \in \Pi, P \neq Q\,\},$$

where $\mathbf{P}$ is a single vertex labelled by P, and $\mathbf{P}^{\mathrm{loop}}$ is the same with an incident loop. We obtain in this way the $\mathcal{VR}^\mathrm{p}_\Pi$-algebra of Π-graphs which was first introduced in [11].

Remark 29. The algebra VR is obtained from STR by deleting certain sorts, the corresponding domains, all operations involving them, and certain unary operations between sorts kept in VR. For VR^p, we additionally remove those structures from the remaining domains where the relations of Π do not partition the set of vertices.

Every term $t \in T(\mathcal{VR}^\mathrm{p}_\Pi)$ defines a Π-graph, and every Π-graph is the value of some $t \in T(\mathcal{VR}^\mathrm{p}_\Psi)$, for a sufficiently large set $\Psi \supseteq \Pi$. The *clique width* of G is defined as the smallest cardinality of Ψ such that

G is the value of some term in $T(\mathcal{VR}^{\mathrm{p}}_{\Psi})$ (see [15,6]). We recall that trees have clique width at most 3. This signature originates from context-free graph grammars defined by vertex replacement (see [4,11]).

To generate undirected graphs we can make the definition of $\mathrm{add}_{P,Q}$ symmetric by setting

$$\vartheta_{edg}(x_1,x_2) := edg(x_1,x_2) \vee (Px_1 \wedge Qx_2) \vee (Px_2 \wedge Qx_1)\,.$$

The notion of clique width of an undirected graph follows immediately. Every clique has clique width 2. We recall the following result from [4,22].

Proposition 30. *A set of finite graphs has bounded clique width if and only if it is contained in the image of a set of finite trees under an* MSO*-transduction.*

We have defined a many-sorted algebra $\mathcal{VR}$ of graphs. The notion of a *$\mathcal{VR}$-recognizable set of graphs* follows from the general definitions. This notion is robust as proved in [8] Theorem 4.5: a set of graphs is $\mathcal{VR}$-recognizable iff it is recognizable w.r.t. $\mathcal{VR}^+$ (the signature consisting of the operations from $\mathcal{VR}_\Pi$ and all quantifier-free operations) iff it is $\mathcal{QF}$-recognizable. We will establish further robustness results below.

Example 31. Recall that, for a finite signature F, we denote by $\Delta = \Delta(F)$ the signature used to encode terms $t \in T(F)$ as labelled trees $\mathfrak{S}(t) \in \mathrm{STR}[\Delta]$. We show that the function $\mathrm{STR}[\Delta] \times \mathrm{STR}[\Delta] \to \mathrm{STR}[\Delta]$ that corresponds to the mapping $T(F) \times T(F) \to T(F) : (t_1, t_2) \mapsto f(t_1, t_2)$, for fixed $f \in F$, can be expressed in terms of $\oplus$, some quantifier-free operations, and one constant. Let $\mathbf{rt}$ be a constant symbol denoting a single element labelled by rt and no other relation. In addition to the relation of Δ we will use unary relations rt_1 and rt_2, and a constant symbol $\mathbf{rt}$. If $t_1, t_2 \in T(F)$ are represented by $\mathfrak{S}(t_1), \mathfrak{S}(t_2) \in \mathrm{STR}[\Delta]$ with disjoint domains then we have

$$\begin{aligned}\mathfrak{S}(f(t_1,t_2)) = (&\mathrm{fgt}_{\mathrm{rt}_1} \circ \mathrm{fgt}_{\mathrm{rt}_2} \circ \mathrm{add}_{\mathrm{rt},\mathrm{rt}_1,\mathrm{suc}_1} \circ \mathrm{add}_{\mathrm{rt},\mathrm{rt}_2,\mathrm{suc}_2})\\ &[\mathbf{rt} \oplus \mathrm{ren}_{\mathrm{rt}\to\mathrm{rt}_1}(\mathfrak{S}(t_1)) \oplus \mathrm{ren}_{\mathrm{rt}\to\mathrm{rt}_2}(\mathfrak{S}(t_2))]\,,\end{aligned}$$

where the operation $\mathrm{add}_{\mathrm{rt},\mathrm{rt}_i,\mathrm{suc}_i}$ adds all pairs (x, y) with $\mathrm{rt}(x)$ and $\mathrm{rt}_i(y)$ to the relation suc_i. This operation can be defined by the quantifier-free transduction where

$$\vartheta_{\mathrm{suc}_i}(x, y) := \mathrm{suc}_i(x, y) \vee (\mathrm{rt}(x) \wedge \mathrm{rt}_i(y))\,.$$

4. Annotated structures

A central notion in many of our proofs is that of a *type annotation* which we use to encode *information* about a tuple of elements of the considered structure. We define finite sets Φ_n of formulas by certain syntactic restrictions such that all formulas in Φ_n have free variables among $x_1, \ldots, x_n$. With every n-tuple $\bar{a}$ we associate the set of those formulas in Φ_n that are satisfied by $\bar{a}$. Such sets are called *logical n-types* (see, e.g. [28,26,25]). The syntactic restrictions defining Φ_n (we will consider several variants) ensure that each type is finite and that there are finitely many types of the considered form.

We enrich a relational structure $\mathfrak{A}$ by adding, for every n-type, a new n-ary relation containing all tuples of that type. This operation is called *annotating the structure* $\mathfrak{A}$. We will examine the relationship between annotations and MSO-transductions and their effect on recognizability.

4.1. Monadic types

The monadic type of a tuple $\bar{a}$ is just the set of all MSO-formulas of a given maximal quantifier height satisfied by $\bar{a}$. In particular, since it contains all quantifier-free formulas that hold for $\bar{a}$, such a type completely describes, up to isomorphism, the substructure induced by $\bar{a}$.

Definition 32. Let $\mathfrak{A}$ be a Σ-structure and $\bar{a} \in A^n$ an n-tuple, $n \geqslant 0$. The *monadic n-type of quantifier height k* of $\bar{a}$ is the set

$$\mathrm{tp}_k(\bar{a}/\mathfrak{A}) := \{ \varphi(\bar{x}) \in \mathrm{MSO}[\Sigma, \{x_1, \dots, x_n\}] \mid \mathrm{qh}(\varphi) \leqslant k,\ \mathfrak{A} \models \varphi(\bar{a}) \}.$$

We denote by $S_{\mathrm{M}}^{n,k}(\Sigma)$ the set of all such monadic n-types realized in some Σ-structure,[2] and we write $S_{\mathrm{M}}^{\leqslant m,k}(\Sigma) := \bigcup_{1 \leqslant n \leqslant m} S_{\mathrm{M}}^{n,k}(\Sigma)$ for the union over all n with $1 \leqslant n \leqslant m$. (We need the subscript M to distinguish monadic types from other kinds of types which we will introduce in Section 7.)

Types of quantifier height 0 are also called *atomic* or *quantifier free.* They contain local information about the given n-tuple. For the empty tuple $\bar{a} = \langle\rangle$, we use the abbreviation $\mathrm{tp}_k(\mathfrak{A}) := \mathrm{tp}_k(\langle\rangle/\mathfrak{A})$.

We will treat the monadic type of the empty tuple differently from the monadic n-types with $n > 0$. For $n > 0$, we can introduce n-ary relations to label tuples of the corresponding type whereas we do not allow relations of arity 0. This is the reason why we exclude the case $n = 0$ in the union defining $S_{\mathrm{M}}^{\leqslant m,k}(\Sigma)$. A type $\mathrm{tp}_k(\mathfrak{A})$ contains a finite amount of global information concerning $\mathfrak{A}$ which, according to Lemma 45 below, is $\mathcal{QF}$-computable.

As stated in the next lemma types are MSO-definable because we only consider finite relational signatures. Furthermore, for finite structures we can effectively compute the type $\mathrm{tp}_k(\bar{a}/\mathfrak{A})$ from $\bar{a}$ and $\mathfrak{A}$.

Definition 33. Let $p \in S_{\mathrm{M}}^{n,k}(\Sigma)$ be a monadic n-type. The *Hintikka-formula* of p is defined by

$$\psi_p(\bar{x}) := \bigwedge p\,.$$

(By convention we do not distinguish between logically equivalent formulas so that the above conjunction is finite, cf. Section 3.1.)

It follows immediately from the definition that a type is defined by its Hintikka-formula.

Lemma 34. *For every monadic n-type* $p \in S_{\mathrm{M}}^{n,k}(\Sigma)$, *we have* $\mathrm{qh}(\psi_p) = k$ *and*

$$\mathfrak{A} \models \psi_p(\bar{a}) \quad \text{iff} \quad \mathrm{tp}_k(\bar{a}/\mathfrak{A}) = p\,,$$

for every structure $\mathfrak{A}$ *and each tuple* $\bar{a} \in A^n$.

Finally, let us remark that quantifier-free operations induce a map on the set of types.

[2] The reader may worry about the fact that $S_{\mathrm{M}}^{n,k}(\Sigma)$ is not recursive (only recursively enumerable). Instead of $S_{\mathrm{M}}^{n,k}(\Sigma)$ we could use the larger set of all sets of formulas over the signature Σ. This will not affect our proofs.

Lemma 35. *For every quantifier-free operation* $f : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$, *there exist mappings* $f_k^n : S_{\mathrm{M}}^{n,k}(\Sigma) \to S_{\mathrm{M}}^{n,k}(\Gamma)$ *such that*

$$\mathrm{tp}_k(\bar{a}/f(\mathfrak{A})) = f_k^n(\mathrm{tp}_k(\bar{a}/\mathfrak{A})),$$

for every structure $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ *and every n-tuple* $\bar{a}$ *in* $f(\mathfrak{A})$.

Proof. For every formula $\varphi(\bar{x})$ of quantifier height at most k, we have

$$\varphi(\bar{x}) \in \mathrm{tp}_k(\bar{a}/f(\mathfrak{A})) \quad \text{iff} \quad \mathfrak{A} \models \varphi^f(\bar{a}) \quad \text{iff} \quad \varphi^f(\bar{x}) \in \mathrm{tp}_k(\bar{a}/\mathfrak{A}).$$

Note that $\mathrm{qh}(\varphi^f) = \mathrm{qh}(\varphi)$, by Proposition 20. Therefore, f_k^n can be defined by

$$f_k^n(p) := \{\, \varphi \mid \varphi^f \in p \,\}. \qquad \square$$

4.2. Monadic annotations

Sometimes it is useful to have all monadic information available via a single relation. To make the full monadic type accessible we add new relations T_p, for every type p. After adding all these relations T_p the original relations are superfluous, and we can delete them.

Definition 36. Let $\mathfrak{A}$ be a Σ-structure, $m > 0$, and $k \geqslant 0$. The *monadic annotations* of $\mathfrak{A}$ are the structures

$$\mathcal{M}_k^m(\mathfrak{A}) := \left\langle A, (T_p)_{p \in S_{\mathrm{M}}^{\leqslant m,k}(\Sigma)} \right\rangle$$

with the same domain as $\mathfrak{A}$ where, for each monadic n-type $p \in S_{\mathrm{M}}^{\leqslant m,k}(\Sigma)$, we add the n-ary relation

$$T_p := \{\, \bar{a} \in A^n \mid \mathrm{tp}_k(\bar{a}/\mathfrak{A}) = p \,\}$$

of all tuples of type p. We denote the relational signature of $\mathcal{M}_k^m(\mathfrak{A})$ by

$$\Sigma_{\mathrm{M}}^{m,k} := \{\, T_p \mid p \in S_{\mathrm{M}}^{\leqslant m,k}(\Sigma) \,\}.$$

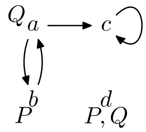
For $m = \mathrm{ar}(\Sigma)$, we simply write $\mathcal{M}_k(\mathfrak{A})$ and Σ_{M}^k.

Definition 37. Let $\mathfrak{A}$ be a structure. The *rank* of an n-tuple $\bar{a} \in A^n$ is the size of the set $\{a_1, \dots, a_n\}$. An n-tuple is a *loop* if its rank is less than n.

By $\mathfrak{A}|_m$ we denote the structure obtained from $\mathfrak{A}$ by removing from all relations every tuple of rank greater than m. Let $\mathrm{STR}_m[\Sigma]$ be the set of all structures $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ such that $\mathfrak{A}|_m = \mathfrak{A}$.

Remark 38. If $m \geqslant \mathrm{ar}(\Sigma)$ then we can reconstruct $\mathfrak{A}$ from $\mathcal{M}_k^m(\mathfrak{A})$. For $m < \mathrm{ar}(\Sigma)$, we can only recover the atomic information about tuples of rank at most m.

Example 39. We consider the following vertex labelled graph $\mathfrak{G} \in \mathrm{STR}[edg, P, Q]$ with domain (i.e., set of vertices) $\{a, b, c, d\}$ and labels P and Q.



The annotated structure $\mathcal{M}_0^2(\mathfrak{G})$ is the complete graph where each vertex x has a unique label $\mathrm{tp}_0(x/\mathfrak{G})$ and every edge (x, y) is labelled by $\mathrm{tp}_0(xy/\mathfrak{G})$. For instance:

$$
\begin{aligned}
\mathrm{tp}_0(a) &= \{\neg Px,\ Qx,\ \neg edg(x,x),\ldots\}\,,\\
\mathrm{tp}_0(b) &= \{Px,\ \neg Qx,\ \neg edg(x,x),\ldots\}\,,\\
\mathrm{tp}_0(c) &= \{\neg Px,\ \neg Qx,\ edg(x,x),\ldots\}\,,\\
\mathrm{tp}_0(d) &= \{Px,\ Qx,\ \neg edg(x,x),\ldots\}\,,\\
\mathrm{tp}_0(ab) &= \{edg(x,y),\ edg(y,x),\ x \neq y,\ldots\} \cup \mathrm{tp}_0(a) \cup \mathrm{tp}_0(b)[y/x]\,,\\
\mathrm{tp}_0(ac) &= \{edg(x,y),\ \neg edg(y,x),\ x \neq y,\ldots\} \cup \mathrm{tp}_0(a) \cup \mathrm{tp}_0(c)[y/x]\,.
\end{aligned}
$$

Note that every type contains a lot of redundant formulas. For the purpose of clarity we have omitted in the above list all formulas that are logical consequences of those shown. To improve readability we also have used the variables x and y instead of x_1 and x_2. Finally, $[y/x]$ denotes the substitution of y for x.

The Hintikka-formula $\psi_{\mathrm{tp}_0(a)}(x)$ of a is thus equivalent to

$$\neg Px \wedge Qx \wedge \neg edg(x,x)\,.$$

If we delete from $\mathcal{M}_0^2(\mathfrak{G})$ the vertex labels we obtain a symmetric labeled 2-structure as defined by Ehrenfeucht et al. [20]. Our results show that equational and recognizable sets of graphs can be defined in an algebraic framework based on vertex and edge labeled complete graphs that are quite close to 2-structures.

Monadic annotations are compatible with MSO-transductions. First of all, the operation $\mathcal{M}_k^m$ is itself an MSO-transduction.

Lemma 40. *Let Σ be a relational signature.*

(*a*) *The mapping $\mathcal{M}_k^m : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Sigma_{\mathrm{M}}^{m,k}]$ is a noncopying parameterless* MSO-*transduction of quantifier height k.*

(*b*) *There exists a quantifier-free noncopying parameterless transduction $g : \mathrm{STR}[\Sigma_{\mathrm{M}}^{m,k}] \to \mathrm{STR}_m[\Sigma]$ such that*

$$g(\mathcal{M}_k^m(\mathfrak{A})) = \mathfrak{A}|_m\,, \qquad \textit{for all } \mathfrak{A} \in \mathrm{STR}[\Sigma]\,.$$

(*c*) *The restriction of $\mathcal{M}_k^m$ to $\mathrm{STR}_m[\Sigma]$ is injective. Its inverse $(\mathcal{M}_k^m)^{-1} : \mathrm{STR}[\Sigma_{\mathrm{M}}^{m,k}] \to \mathrm{STR}_m[\Sigma]$ is an* MSO-*transduction.*

Proof.

(a) We have already seen in Lemma 34 that one can define the relation T_p by the formula ψ_p of quantifier height k.
(b) For $n \leqslant m$, we can write an n-ary relation $R \in \Sigma$ as

$$R_{\mathfrak{A}} = \{\, \bar{a} \in A^n \mid \bar{a} \in T_p \text{ for some } p \text{ with } R\bar{x} \in p \,\}\,.$$

Hence, we obtain a definition scheme for g by setting

$$\vartheta_R(x_1, \ldots, x_n) := \bigvee \{\, T_p x_1 \ldots x_n \mid p \in S^{n,k}_{\mathrm{M}}(\Sigma), R x_1 \ldots x_n \in p \,\}\,.$$

For $n > m$, we need some notation to write down ϑ_R. With an n-tuple $\bar{a}$ of rank r we can associate a surjective function $\sigma : [n] \to [r]$ such that $a_i = a_l$ iff $\sigma(i) = \sigma(l)$. Given such a function σ we set $\mu_i(\sigma) := \min \sigma^{-1}(i)$, for $i \in [r]$, and

$$\chi_\sigma(x_1, \ldots, x_n) := \bigwedge_{i \in [r]} \; \bigwedge_{k,l \in \sigma^{-1}(i)} x_k = x_l\,.$$

Then we can define R by

$$\begin{aligned}\vartheta_R(x_1, \ldots, x_n) := \bigvee \{\; & T_p x_{\mu_1(\sigma)} \ldots x_{\mu_r(\sigma)} \wedge \chi_\sigma(x_1, \ldots, x_n) \mid \\ & 1 \leqslant r \leqslant m\,,\ \sigma : [n] \to [r] \text{ surjective with} \\ & \mu_1(\sigma) < \cdots < \mu_r(\sigma)\,, \text{ and} \\ & p \in S^{r,k}_{\mathrm{M}}(\Sigma) \text{ with } R x_{\sigma(1)} \ldots x_{\sigma(n)} \in p \,\}\,.\end{aligned}$$

For example, if $\sigma : [6] \to [3]$ maps $[6]$ to the sequence $1, 2, 2, 1, 2, 3$ then the above disjunction includes the formula

$$T_p x_1 x_2 x_6 \wedge x_1 = x_4 \wedge x_2 = x_3 \wedge x_2 = x_5 \wedge x_3 = x_5$$

if and only if we have $R x_1 x_2 x_2 x_1 x_2 x_3 \in p$.

Note that the above disjunctions are finite since there are only finitely many types in $S^{\leqslant m,k}_{\mathrm{M}}(\Sigma)$.

(c) In light of (b) we only need to prove that the range of $\mathcal{M}^m_k$ restricted to $\mathrm{STR}_m[\Sigma]$ is MSO-definable. Then we can restrict the transduction g of (b) appropriately. Let $\mathfrak{A} \in \mathrm{STR}[\Sigma^{m,k}_{\mathrm{M}}]$. If $\mathfrak{A} = \mathcal{M}^m_k(\mathfrak{B})$, for some $\mathfrak{B} \in \mathrm{STR}_m[\Sigma]$, then we have

$$\mathfrak{B} = \mathfrak{B}|_m = g(\mathcal{M}^m_k(\mathfrak{B})) = g(\mathfrak{A})\,,$$

which implies that $\mathfrak{A} = \mathcal{M}^m_k(g(\mathfrak{A}))$. Conversely, if $\mathfrak{A} = \mathcal{M}^m_k(g(\mathfrak{A}))$ then $\mathfrak{A}$ is in the range of $\mathcal{M}^m_k$. We can express that $\mathfrak{A} = \mathcal{M}^m_k(g(\mathfrak{A}))$ by the formula

$$\bigwedge_{p \in S^{\leqslant m,k}_{\mathrm{M}}(\Sigma)} \forall \bar{x} (T_p \bar{x} \leftrightarrow (\psi_p)^g(\bar{x}))$$

where ψ_p is the Hintikka-formula for p and $(\psi_p)^g$ its backwards translation via g. This formula can be used in the definition scheme of the transduction $(\mathcal{M}_k^m)^{-1} : \mathrm{STR}[\Sigma_{\mathrm{M}}^{m,k}] \to \mathrm{STR}_m[\Sigma]$ to define the domain. $\square$

Since, by Corollary 28, $\mathcal{QF}$-equational sets are closed under MSO-transductions it follows immediately that $\mathcal{M}_k^m$ preserves equationality.

Corollary 41. *A set $C \subseteq \mathrm{STR}_m[\Sigma]$ is $\mathcal{QF}$-equational if and only if $\mathcal{M}_k^m(C)$ is $\mathcal{QF}$-equational.*

Each noncopying parameterless MSO-transduction of quantifier height k factors through $\mathcal{M}_k^m$.

Lemma 42. *Let $g : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ be a noncopying parameterless* MSO*-transduction of quantifier height k and $m := \mathrm{ar}(\Gamma)$. There exists a noncopying parameterless quantifier-free transduction $f : \mathrm{STR}[\Sigma_{\mathrm{M}}^{m,k}] \to \mathrm{STR}[\Gamma]$ such that*

$$g(\mathfrak{A}) = f\big(\mathcal{M}_k^m(\mathfrak{A})\big), \quad \textit{for all } \mathfrak{A} \in \mathrm{STR}[\Sigma] \textit{ such that } g(\mathfrak{A}) \textit{ is defined}.$$

Proof. Given a definition scheme $(\varphi, \psi, (\vartheta_R)_{R\in\Gamma})$ of g, we construct a quantifier-free logical definition scheme $(\mathrm{true}, \psi', (\vartheta'_R)_{R\in\Gamma})$ for f by setting

$$\psi' := \bigvee \{\, T_p x_1 \mid p \models \psi \,\} \quad \text{and} \quad \vartheta'_R := \bigvee \{\, T_p \bar{x} \mid p \models \vartheta_R \,\}.$$

($\models$ is the logical entailment relation.) $\square$

4.3. Operations on annotated structures

It turns out that the mapping $\mathrm{tp}_k : \mathrm{STR}[\Sigma] \to S_{\mathrm{M}}^{0,k}(\Sigma)$ is $\mathcal{QF}$-computable (cf. Definition 6). One part of the proof is given by the following (special case of a) theorem of Shelah [33] (see also the thorough study by Makowsky [29]).

Proposition 43. *Let $k, m, n \geqslant 0$. For every formula $\varphi \in \mathrm{MSO}[\Sigma \cup \Gamma, \{x_1, \dots, x_{m+n}\}]$ of quantifier height k, one can effectively construct finite sequences of formulas*

$$\begin{aligned} &\psi_1, \dots, \psi_l \in \mathrm{MSO}[\Sigma, \{x_1, \dots, x_m\}] \\ \textit{and } &\vartheta_1, \dots, \vartheta_l \in \mathrm{MSO}[\Gamma, \{x_{m+1}, \dots, x_{m+n}\}] \end{aligned}$$

of quantifier height at most k such that, for all structures $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ and $\mathfrak{B} \in \mathrm{STR}[\Gamma]$, and all tuples $\bar{a} \in A^m$ and $\bar{b} \in B^n$, we have

$$\mathfrak{A} \oplus \mathfrak{B} \models \varphi(\bar{a}, \bar{b}) \quad \text{iff} \quad \mathfrak{A} \models \psi_i(\bar{a}) \textit{ and } \mathfrak{B} \models \vartheta_i(\bar{b}) \quad \textit{for some } 1 \leqslant i \leqslant l.$$

Corollary 44. *For all $k, n \in \mathbb{N}$ and every set $I \subseteq [n]$, there exists a binary function $\oplus_{k,I}$ such that*

$$\mathrm{tp}_k(\bar{c}/\mathfrak{A} \oplus \mathfrak{B}) = \mathrm{tp}_k(\bar{c}|_I \,/\, \mathfrak{A}) \oplus_{k,I} \mathrm{tp}_k(\bar{c}|_{[n]\setminus I} \,/\, \mathfrak{B}),$$

for all structures $\mathfrak{A}$ and $\mathfrak{B}$ and all tuples $\bar{c} \in (A \cup B)^n$ such that $\bar{c}|_I \subseteq A$ and $\bar{c}|_{[n]\setminus I} \subseteq B$. (By $\bar{c}|_I$ we denote the subtuple of all components c_i with $i \in I$.)

Lemma 45. *The function* $\mathrm{tp}_k : \mathrm{STR}[\Sigma] \to S^{0,k}_{\mathrm{M}}(\Sigma)$ *is* $\mathcal{QF}$*-computable.*

Proof. It is sufficient to find operations on $S^{0,k}_{\mathrm{M}}(\Sigma)$ such that $\mathrm{tp}_k : \mathrm{STR}[\Sigma] \to S^{0,k}_{\mathrm{M}}(\Sigma)$ becomes a $\mathcal{QF}$-homomorphism. For the disjoint union, we can use the operation $\oplus_{k,\emptyset}$ introduced in Corollary 44. And, if $g : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ is a quantifier-free operation then we have shown in Lemma 35 that

$$\mathrm{tp}_k(g(\mathfrak{A})) = g^0_k(\mathrm{tp}_k(\mathfrak{A}))\,, \qquad \text{for all structures } \mathfrak{A}\,. \qquad \square$$

Lemma 46. *For every* $m \in \mathbb{N}$, *the mapping* $\mathcal{M}^m_k : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Sigma^{m,k}_{\mathrm{M}}]$ *is a finite-state heteromorphism based on* tp_k.

Proof. We have to show that, for every operation $f \in \mathcal{QF}$ of arity $0 \leqslant n \leqslant 2$, there exist linear terms $t[p_1,\dots,p_n] \in T(\mathcal{QF},\{x_1,\dots,x_n\})$, for $p_1,\dots,p_n \in S^{0,k}_{\mathrm{M}}(\Sigma)$, such that

$$\mathcal{M}^m_k(f(\mathfrak{A}_1,\dots,\mathfrak{A}_n)) = t[\mathrm{tp}_k(\mathfrak{A}_1),\dots,\mathrm{tp}_k(\mathfrak{A}_n)]\big(\mathcal{M}^m_k(\mathfrak{A}_1),\dots,\mathcal{M}^m_k(\mathfrak{A}_n)\big)\,,$$

for all structures $\mathfrak{A}_1,\dots,\mathfrak{A}_n \in \mathrm{STR}[\Sigma]$.

First, we consider a quantifier-free operation $f : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$. Recall the mappings $f^i_k : S^{i,k}_{\mathrm{M}}(\Sigma) \to S^{i,k}_{\mathrm{M}}(\Gamma)$ defined in Lemma 35. We have

$$\mathcal{M}^m_k(f(\mathfrak{A})) = g(\mathcal{M}^m_k(\mathfrak{A}))$$

where the definition scheme of the quantifier-free operation g consists of the formulas:

$$\begin{aligned}
\psi(x) &:= \bigvee \{\, T_q x \mid q \in S^{1,k}_{\mathrm{M}}(\Sigma),\ \psi' \in q \,\}\,,\\
\vartheta_{T_p}(\bar{x}) &:= \bigvee \{\, T_q \bar{x} \mid q \in (f^i_k)^{-1}(p) \,\}\,, \quad \text{for every } p \in S^{i,k}_{\mathrm{M}}(\Gamma)\,,\ 1 \leqslant i \leqslant m\,,
\end{aligned}$$

where ψ' is the formula of the definition scheme for f that specifies the domain of the output structure. Note that in this case the term $t[\mathrm{tp}_k(\mathfrak{A})] = g(x_1)$ does not depend on $\mathrm{tp}_k(\mathfrak{A})$.

Second, we consider the case where $f = \oplus$. We define quantifier-free operations h_0, h_1, and g depending on $\mathrm{tp}_k(\mathfrak{A})$ and $\mathrm{tp}_k(\mathfrak{B})$ such that

$$\mathcal{M}^m_k(\mathfrak{A} \oplus \mathfrak{B}) = g\big(h_0(\mathcal{M}^m_k(\mathfrak{A})) \oplus h_1(\mathcal{M}^m_k(\mathfrak{B}))\big)\,.$$

The operations h_0 and h_1 just add a new unary relation $P \notin \Sigma$ to their argument such that $P = \emptyset$ for h_0 whereas, for h_1, P contains every element. These functions are only needed so that we can tell the elements of the two structures apart. The main work is done by g which updates the type annotation. Recall from Corollary 44, that there exists a binary operation $\oplus_{k,I}$ on $S^{\leqslant m,k}_{\mathrm{M}}(\Sigma)$, for $n \leqslant m$ and $I \subseteq [n]$, such that

$$\mathrm{tp}_k(\bar{c}/\mathfrak{A} \oplus \mathfrak{B}) = \mathrm{tp}_k(\bar{c}|_I/\mathfrak{A}) \oplus_{k,I} \mathrm{tp}_k(\bar{c}|_{[n]\setminus I}/\mathfrak{B})\,,$$

for all structures $\mathfrak{A}$ and $\mathfrak{B}$ and all tuples $\bar{c} \in (A \cup B)^n$ with $\bar{c}|_I \subseteq A$ and $\bar{c}|_{[n]\setminus I} \subseteq B$. Hence, we can define the definition scheme of g by the formulas

$$\psi(x) := \text{true}\,,$$

$$\begin{aligned}\text{and}\quad \vartheta_{T_p}(\bar{x}) := \bigvee \Big\{ \textstyle\bigwedge_{i\in I} \neg P x_i \wedge \bigwedge_{i\notin I} P x_i \wedge T_q \bar{x}|_I \wedge T_r \bar{x}|_{[n]\setminus I} \;\Big|\; & \\ I \subseteq [n],\ I \notin \{\emptyset, [n]\},\ q \oplus_{k,I} r = p \Big\} & \\ \vee \bigvee \Big\{ \textstyle\bigwedge_{i\in[n]} \neg P x_i \wedge T_q \bar{x} \;\Big|\; q \oplus_{k,[n]} \mathrm{tp}_k(\mathfrak{B}) = p \Big\} & \\ \vee \bigvee \Big\{ \textstyle\bigwedge_{i\in[n]} P x_i \wedge T_r \bar{x} \;\Big|\; \mathrm{tp}_k(\mathfrak{A}) \oplus_{k,\emptyset} r = p \Big\}, &\end{aligned}$$

for $p \in S^{n,k}_{\mathrm{M}}(\Sigma)$. (In the case where $\mathfrak{A}$ and $\mathfrak{B}$ have different signatures the argument is adapted in the obvious way.)

Finally, we consider the case where f is a constant. Then the value of f is a singleton structure $\mathfrak{A}$. Consequently, its annotation $\mathcal{M}^m_k(\mathfrak{A})$ is also a singleton structure that can be denoted by a constant. □

Recall that we write $\mathcal{M}_k(\mathfrak{A})$ for $\mathcal{M}^{\mathrm{ar}(\Sigma)}_k(\mathfrak{A})$. As usual we set

$$\mathcal{M}_k(C) := \{\, \mathcal{M}_k(\mathfrak{A}) \mid \mathfrak{A} \in C \,\},$$

for classes $C \subseteq \mathrm{STR}[\Sigma]$.

Theorem 47. *A set $C \subseteq \mathrm{STR}[\Sigma]$ is $\mathcal{QF}$-recognizable if and only if $\mathcal{M}_k(C)$ is $\mathcal{QF}$-recognizable.*

Proof. ($\Leftarrow$) By Lemma 46, $\mathcal{M}_k$ is a finite-state heteromorphism based on tp_k. We have seen in Lemma 40 that $\mathcal{M}_k$ is injective. Therefore, we have $C = (\mathcal{M}_k)^{-1}(\mathcal{M}_k(C))$ and, by Lemma 14, it follows that C is $\mathcal{QF}$-recognizable.

($\Rightarrow$) Suppose that $C \subseteq \mathrm{STR}[\Sigma]$ is $\mathcal{QF}$-recognizable. Let $\approx$ be a $\mathcal{QF}$-congruence witnessing this fact.

By Lemma 40 (c), the range $\mathcal{D} := \mathcal{M}_k(\mathrm{STR}[\Sigma]) \subseteq \mathrm{STR}[\Sigma^k_{\mathrm{M}}]$ of $\mathcal{M}_k$ is MSO-definable and, therefore, $\mathcal{QF}$-recognizable by Proposition 26. We denote the corresponding $\mathcal{QF}$-congruence by $\simeq$.

To show that $\mathcal{M}_k(C)$ is $\mathcal{QF}$-recognizable we define

$$\mathfrak{A} \equiv \mathfrak{B} \quad \text{iff} \quad \mathfrak{A} \simeq \mathfrak{B} \text{ and } \mathfrak{A} \approx \mathfrak{B}.$$

Clearly, $\equiv$ is a finite $\mathcal{QF}$-congruence.

It remains to show that $\equiv$ saturates $\mathcal{M}_k(C)$. Let $\mathfrak{A} \in \mathcal{M}_k(C)$, that is, $\mathfrak{A} = \mathcal{M}_k(\mathfrak{C})$, for some $\mathfrak{C} \in C$. If $\mathfrak{B} \equiv \mathfrak{A}$ then $\mathfrak{A} \simeq \mathfrak{B}$ implies that $\mathfrak{B} = \mathcal{M}_k(\mathfrak{D})$, for some $\mathfrak{D} \in \mathrm{STR}[\Sigma]$. We have seen in Lemma 40 (b) that there exists a left-inverse g of $\mathcal{M}_k$ that is a quantifier-free operation. Hence, $\mathfrak{A} \approx \mathfrak{B}$ implies

$$\mathfrak{C} = g(\mathfrak{A}) \approx g(\mathfrak{B}) = \mathfrak{D}.$$

Consequently, we have $\mathfrak{D} \in C$ and $\mathfrak{B} = \mathcal{M}_k(\mathfrak{D}) \in \mathcal{M}_k(C)$, as desired. □

4.4. Annotating the leaves of a binary tree

We state some definitions and lemmas that we will use in Section 6. Let F be a set of binary function symbols and C a set of constants. As remarked at the beginning of Section 3 we can represent every term $t \in T(F \cup C)$ by a tree

$$\mathfrak{S}(t) := \big\langle N(t), \mathrm{suc}_1, \mathrm{suc}_2, \mathrm{rt}, (\mathrm{lab}_a)_{a \in F \cup C} \big\rangle \in \mathrm{STR}[\Delta(F \cup C)],$$

where $N(t)$ is the set of nodes of t. Let $\Delta := \Delta(F \cup C)$ be the corresponding signature. We denote the set of *leaves* by $L(t) \subseteq N(t)$ and by $\leqslant$ the usual linear left-right order on $L(t)$.

Definition 48. Let t be a term, $m > 0$, and $k \geqslant 0$. A tuple $\bar{a} \in L(t)^n$ is *increasing* if $a_1 < \cdots < a_n$. The *restricted monadic annotations* of $\mathfrak{S}(t)$ are the $\Delta^{m,k}_{\mathrm{M}}$-structures

$$\mathcal{R}^m_k(t) := \left\langle L(t), (T_p)_{p \in S^{\leqslant m,k}_{\mathrm{M}}(\Delta)} \right\rangle$$

with domain $L(t)$ where, for each monadic n-type $p \in S^{\leqslant m,k}_{\mathrm{M}}(\Delta)$, we add the n-ary relation

$$T_p := \left\{ \bar{a} \in L(t)^n \;\middle|\; \bar{a} \text{ increasing}, \operatorname{tp}_k(\bar{a}/\mathfrak{S}(t)) = p \right\}.$$

Remark 49. There are formulae $\varphi(x)$ and $\psi(x, y)$ of quantifier height $\mathrm{qh}(\varphi) = 1$ and $\mathrm{qh}(\psi) = 5$ such that φ defines the set of leaves and ψ defines the ordering $<$:

$$\begin{aligned} \varphi(x) &:= \neg\exists y[\mathrm{suc}_1(x, y) \vee \mathrm{suc}_2(x, y)], \\ \psi(x, y) &:= \exists z[\exists u_1(\mathrm{suc}_1(z, u_1) \wedge u_1 \preceq x) \wedge \exists u_2(\mathrm{suc}_2(z, u_2) \wedge u_2 \preceq y)], \end{aligned}$$

where the tree ordering $\preceq$ is defined by

$$x \preceq y \quad \text{iff} \quad \forall Z[y \in Z \wedge \forall u \forall v[v \in Z \wedge (\mathrm{suc}_1(u, v) \vee \mathrm{suc}_2(u, v)) \rightarrow u \in Z] \rightarrow x \in Z].$$

($x \preceq y$ can be read as "x is an ancestor of y".) Hence, there exists a formula $\vartheta_n(x_1, \dots, x_n)$ of quantifier height 5 expressing that $\bar{x}$ is an increasing tuple of leaves. It follows that, for $k \geqslant 5$, we can tell from $\operatorname{tp}_k(\bar{a}/\mathfrak{S}(t))$ whether $\bar{a}$ is such a tuple. Consequently, we can obtain $\mathcal{R}^m_k(t)$ from $\mathcal{M}^m_k(\mathfrak{S}(t))$ by

- deleting all nodes that are not leaves,
- removing all relations T_p such that $p \not\models \vartheta_n$.

For $t \in T(F \cup C)$ and $u \in N(t)$, we denote by $t/u \in T(F \cup C)$ the subterm of t rooted at the node u. Let $*$ be a new constant symbol. We denote by $t \setminus u \in T(F \cup C \cup \{*\})$ the term obtained from t by replacing the subterm t/u by the constant $*$. Hence, the unique occurrence of $*$ in t/u is u.

Lemma 50. *Let $k \in \mathbb{N}$.*

(*a*) *For every $f \in F$ and all numbers $0 \leqslant m \leqslant n$, there exists a mapping*

$$\odot^f_{m,n} : S^{m,k}_{\mathrm{M}}(\Delta) \times S^{n-m,k}_{\mathrm{M}}(\Delta) \rightarrow S^{n,k}_{\mathrm{M}}(\Delta)$$

such that we have

$$\operatorname{tp}_k\big(\bar{a}\bar{b}/\mathfrak{S}(f(t_1, t_2))\big) = \operatorname{tp}_k\big(\bar{a}/\mathfrak{S}(t_1)\big) \odot^f_{m,n} \operatorname{tp}_k\big(\bar{b}/\mathfrak{S}(t_2)\big),$$

for all $t_1, t_2 \in T(F \cup C)$ and all increasing tuples $\bar{a} \in L(t_1)^m$ and $\bar{b} \in L(t_2)^{n-m}$.

(*b*) *For every $f \in F$ and all numbers $0 \leqslant m \leqslant n$, there exists a mapping*

$$\hat{\odot}^{f}_{m,n} : S^{0,k}_{\mathrm{M}}(\Delta) \times S^{m,k}_{\mathrm{M}}(\Delta) \times S^{n-m,k}_{\mathrm{M}}(\Delta) \to S^{n,k}_{\mathrm{M}}(\Delta)$$

such that we have

$$\mathrm{tp}_k\big(\bar{a}\bar{b}/\mathfrak{S}(t)\big) = \hat{\odot}^{f}_{m,n}\big(\mathrm{tp}_k(\mathfrak{S}(t \setminus u)),\ \mathrm{tp}_k(\bar{a}/\mathfrak{S}(t_1)),\ \mathrm{tp}_k(\bar{b}/\mathfrak{S}(t_2))\big),$$

for every $t \in T(F \cup C)$ such that $t/u = f(t_1, t_2)$ and all increasing tuples $\bar{a} \in L(t_1)^m$ and $\bar{b} \in L(t_2)^{n-m}$.

Proof. (a) We recall from the example after Proposition 30 that the mapping $\langle \mathfrak{S}(t_1), \mathfrak{S}(t_2)\rangle \mapsto \mathfrak{S}(f(t_1, t_2))$ is a $\mathcal{QF}$-derived operation. Consequently, the result follows from Lemma 35 and Corollary 44.

(b) The claim follows as in (a) since we have

$$\begin{aligned}\mathfrak{S}(t) = \big(&\mathrm{ren}_{*\to f} \circ \mathrm{fgt}_{\mathrm{rt}_1} \circ \mathrm{fgt}_{\mathrm{rt}_2} \circ \mathrm{add}_{*,\mathrm{rt}_1,\mathrm{suc}_1} \circ \mathrm{add}_{*,\mathrm{rt}_2,\mathrm{suc}_2}\big)\\ &\big(\mathfrak{S}(t \setminus u) \oplus \mathrm{ren}_{\mathrm{rt}\to\mathrm{rt}_1}(\mathfrak{S}(t_1)) \oplus \mathrm{ren}_{\mathrm{rt}\to\mathrm{rt}_2}(\mathfrak{S}(t_2))\big) \qquad \square\end{aligned}$$

5. Inverse MSO-transductions preserve recognizability

In this section we establish the following theorem which is one of the main results of the article.

Theorem 51. *If $L \in \mathrm{Rec}(\mathrm{STR})_\Gamma$ and $\tau : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ is an* MSO-*transduction then* $\tau^{-1}(L) \in \mathrm{Rec}(\mathrm{STR})_\Sigma$.

The special case where L is CMSO-*definable* (CMSO is the extension of monadic second-order logic by counting predicates which count the cardinality of a set modulo a fixed integer) follows from existing results. It is known that every CMSO-definable set is recognizable [7] and the inverse image of a CMSO-definable set under an MSO-transduction is CMSO-definable. The case where L is a recognizable set of (simple) graphs of bounded tree width is a consequence of a result by Lapoire [27] stating that such sets are CMSO-definable if we allow quantification over sets of edges (and not only over sets of vertices). It follows that L is also CMSO-definable by a result of [12] where it is shown that, in the case of finite graphs of bounded tree width, quantifiers over sets of edges can be eliminated.

On the other hand, in [8] it is shown that there are uncountably many $\mathcal{VR}$-recognizable sets of graphs. Hence, uncountably many of them are not definable in monadic-second order logic or in its extensions like CMSO, because these languages are countable. This shows that Theorem 51 cannot be proved by reduction to the special case of CMSO-definable sets.

The proof is based on the fact that a k-copying MSO-transduction τ with parameters $W_1, \dots, W_n$ can be written as $\tau = \varrho \circ \mathrm{copy}_k \circ \gamma$ where

- ϱ is a noncopying parameterless transduction,
- γ is a noncopying transduction guessing $W_1, \dots, W_n$, and
- copy_k is a k-copying parameterless transduction constructing the k-fold disjoint union of its argument, with some additional annotations to tell apart the different copies.

We will prove the theorem separately for these three special cases.

5.1. Transductions that replicate structures

The simplest MSO-transduction we consider is a parameterless k-copying transduction denoted by copy_k. It transforms a structure $\mathfrak{A}$ into the disjoint union of k copies of $\mathfrak{A}$, denoted by $\mathfrak{A}_1, \ldots, \mathfrak{A}_k$, expanded by

- new binary relations Y_i that encode the canonical isomorphisms $\mathfrak{A}_1 \to \mathfrak{A}_i$,
- new unary relations P_i that "mark" the elements of the ith copy $\mathfrak{A}_i$.

Definition 52. Let $\Upsilon_k := \{ P_i \mid 1 \leqslant i \leqslant k \} \cup \{ Y_i \mid 1 < i \leqslant k \}$. We assume that Υ_k is disjoint from every other relational signature $\Sigma, \Gamma, \Delta, \ldots$ that we will consider. For each relational signature Σ, we define an operation

$$\mathrm{copy}_k : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Sigma \cup \Upsilon_k]$$

that maps a structure $\mathfrak{A} = \langle A, (R_{\mathfrak{A}})_{R \in \Sigma} \rangle$ to the structure $\mathfrak{C} = \mathrm{copy}_k(\mathfrak{A})$ with domain $C = A \times [k]$ and relations:

$$\begin{aligned} R_{\mathfrak{C}} &:= \big\{ ((a_1, i), \ldots, (a_{\mathrm{ar}(R)}, i)) \bigm| (a_1, \ldots, a_{\mathrm{ar}(R)}) \in R_{\mathfrak{A}},\ i \in [k] \big\}, \\ (P_i)_{\mathfrak{C}} &:= A \times \{i\}, \\ (Y_i)_{\mathfrak{C}} &:= \big\{ ((a, 1), (a, i)) \bigm| a \in A \big\}. \end{aligned} \tag{1}$$

It is clear that copy_k is a parameterless k-copying MSO-transduction.

Lemma 53. *For every parameterless k-copying* MSO*-transduction* $\tau : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$, *there exists a parameterless noncopying* MSO*-transduction* $\varrho : \mathrm{STR}[\Sigma \cup \Upsilon_k] \to \mathrm{STR}[\Gamma]$ *such that* $\tau = \varrho \circ \mathrm{copy}_k$ *and* $\varrho(\mathfrak{B})$ *is undefined if the argument* $\mathfrak{B}$ *is not of the form* $\mathrm{copy}_k(\mathfrak{A})$, *for some* $\mathfrak{A}$.

Proof. Note that a structure $\mathfrak{C} \in \mathrm{STR}[\Sigma \cup \Upsilon_k]$ of the form $\mathrm{copy}_k(\mathfrak{A})$ satisfies the following conditions:

(1) The sets $(P_1)_{\mathfrak{C}}, \ldots, (P_k)_{\mathfrak{C}}$ form a partition of the domain.
(2) For every $R \in \Sigma$ and all tuples $\bar{a} \in R_{\mathfrak{C}}$, there is some i with $\bar{a} \subseteq (P_i)_{\mathfrak{C}}$.
(3) Each relation $(Y_i)_{\mathfrak{C}}$ defines an isomorphism between $\mathrm{fgt}_{P_1}(\mathfrak{C}[P_1])$ and $\mathrm{fgt}_{P_i}(\mathfrak{C}[P_i])$.

Conversely, every structure $\mathfrak{C} \in \mathrm{STR}[\Sigma \cup \Upsilon_k]$ satisfying these conditions is isomorphic to $\mathrm{copy}_k(\mathfrak{A})$ where $\mathfrak{A}$ is the Σ-reduct of $\mathfrak{C}[P_1]$. The conjunction of (1)–(3) can be expressed by a first-order formula χ.

We denote the relativization of a formula α to the set P_i by $\alpha^{(P_i)}$. Suppose that τ is defined by

$$\mathcal{D} = (\varphi, \psi_1, \ldots, \psi_k, (\vartheta_w)_{w \in \Gamma \boxtimes k}).$$

A definition scheme $\mathcal{E} = (\varphi', \psi', (\vartheta'_R)_{R \in \Gamma})$ for ϱ can be defined as follows. The formula φ' has to express in $\mathfrak{C}$ that there is some $\mathfrak{A}$ with $\mathfrak{C} = \mathrm{copy}_k(\mathfrak{A})$ and $\mathfrak{A} \models \varphi$. We can set

$$\varphi' := \chi \wedge \varphi^{(P_1)}.$$

The formula ψ' should define the set of all elements $(a, i) \in C$ such that $\mathfrak{A} \models \psi_i(a)$. This can be done by defining

$$\psi'(x) := \bigwedge_{i=1}^{k} (P_i x \to \psi_i^{(P_i)}(x)) .$$

Finally, we must construct formulas ϑ'_R, for $R \in \Gamma$. We use the relations Y_i to obtain a copy of a given tuple that lies in the first copy. We have

$$((a_1, i_1), \dots, (a_n, i_n)) \in R_{\hat{\mathcal{D}}(\mathfrak{A})} \quad \text{iff} \quad \mathfrak{A} \models \vartheta_{R, i_1 \dots i_n}(a_1, \dots, a_n) .$$

For fixed $i_1, \dots, i_n$, we can express this by the formula

$$\beta_{i_1 \dots i_n}(\bar{x}) := \exists y_1 \cdots \exists y_n \Big(\bigwedge_{k=1}^{n} Y_{i_k} y_k x_k \wedge \vartheta^{(P_1)}_{R, i_1 \dots i_n}(\bar{y}) \Big) .$$

(If $i_k = 1$ then instead of $Y_1 y_k x_k$ we use the formula $y_k = x_k \wedge P_1 x_k$.) Therefore, we can set

$$\vartheta'_R(\bar{x}) := \bigwedge_{i_1, \dots, i_n} \Big(\bigwedge_{k=1}^{n} P_{i_k} x_k \to \beta_{i_1 \dots i_n}(\bar{x}) \Big) . \qquad \square$$

Lemma 54.

(*a*) *For all structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{STR}[\Sigma]$ *and every* k, *we have*

$$\mathrm{copy}_k(\mathfrak{A} \oplus \mathfrak{B}) = \mathrm{copy}_k(\mathfrak{A}) \oplus \mathrm{copy}_k(\mathfrak{B}) .$$

(*b*) *For every* k *and every quantifier-free operation* $f : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ *there is a quantifier-free operation* $f' : \mathrm{STR}[\Sigma \cup \Upsilon_k] \to \mathrm{STR}[\Gamma \cup \Upsilon_k]$ *such that we have*

$$\mathrm{copy}_k(f(\mathfrak{A})) = f'(\mathrm{copy}_k(\mathfrak{A})) , \qquad \textit{for every } \mathfrak{A} \in \mathrm{STR}[\Sigma] .$$

Proof. (a) is clear. (b) Let $\mathcal{D} = (\mathrm{true}, \psi, (\vartheta_R)_{R \in \Gamma})$ be the definition scheme of f. We can define a definition scheme

$$\mathcal{D}' = \big(\mathrm{true}, \psi', (\vartheta'_R)_{R \in \Gamma}, (\vartheta'_{P_i})_{1 \leqslant i \leqslant k}, (\vartheta'_{Y_i})_{1 < i \leqslant k}\big)$$

of f' by

$$\begin{aligned} \psi'(x) &:= \psi^{(P_1)}(x) \vee \cdots \vee \psi^{(P_k)}(x) , \\ \vartheta'_R(\bar{x}) &:= (\vartheta_R)^{(P_1)}(\bar{x}) \vee \cdots \vee (\vartheta_R)^{(P_k)}(\bar{x}) , \\ \vartheta'_{P_i}(x) &:= P_i x , \\ \vartheta'_{Y_i}(x, y) &:= Y_i x y , \end{aligned}$$

where $\varphi^{(P_i)}(\bar{x})$ denotes the relativization of $\varphi(\bar{x})$ to P_i written in such a way that the formula $\varphi^{(P_i)}(\bar{x})$ implies $P_i x_l$, for all l. $\square$

Proposition 55. *Theorem 51 holds for* $\tau = \mathrm{copy}_k$.

Proof. By Lemma 54, the mapping copy_k is a heteromorphism for the subsignature of $\mathcal{QF}$ obtained by removing all constants. Therefore, the result follows from Lemma 14 and the remark that recognizability does not depend on the constants in the signature. $\square$

5.2. Parameterless noncopying transductions

Proposition 56. *Theorem 51 holds for parameterless noncopying* MSO*-transductions.*

Proof. Let $\tau : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ be a noncopying parameterless MSO-transduction of quantifier height k with definition scheme $(\varphi, \psi, (\vartheta_R)_{R\in\Gamma})$. Suppose that $L \in \mathrm{Rec}(\mathrm{STR})_\Gamma$ and let $\approx$ be a congruence witnessing the recognizability of L. Let $m := \mathrm{ar}(\Gamma)$. By Lemma 42, there is a quantifier-free operation $f : \mathrm{STR}[\Sigma^{m,k}_{\mathrm{M}}] \to \mathrm{STR}[\Gamma]$ such that, if $\tau(\mathfrak{A})$ is defined then $\tau(\mathfrak{A}) = f(\mathcal{M}^m_k(\mathfrak{A}))$. Consequently, we have

$$\tau^{-1}(L) = \{\, \mathfrak{A} \in \mathrm{STR}[\Sigma] \mid \mathfrak{A} \models \varphi \,\} \cap (\mathcal{M}^m_k)^{-1}(f^{-1}(L))\,.$$

Clearly, $\approx$ also witnesses the recognizability of $f^{-1}(L)$. By Lemmas 46 and 14, it follows that $(\mathcal{M}^m_k)^{-1}(f^{-1}(L))$ is also recognizable. Furthermore, by Proposition 26 (a) the set $\{\, \mathfrak{A} \in \mathrm{STR}[\Sigma] \mid \mathfrak{A} \models \varphi \,\}$ is recognizable. Since recognizable sets are closed under intersection (cf. the remark after Definition 2) the result follows. $\square$

5.3. Handling parameters

Let $\Pi_m := \{P_1, \dots, P_m\}$ be a set of unary relation symbols disjoint from the other signatures Σ, Γ, Υ etc. that we will consider. Let $\mathrm{fgt}_{\Pi_m} : \mathrm{STR}[\Sigma \cup \Pi_m] \to \mathrm{STR}[\Sigma]$ be the quantifier-free transduction that deletes all relations in Π_m. Its inverse is a noncopying MSO-transduction with m parameters that specify the values of the relations $P_1, \dots, P_m$.

Lemma 57. *Every* MSO*-transduction* $\tau : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ *with m parameters can be factorized as* $\varrho \circ \mathrm{fgt}^{-1}_{\Pi_m}$ *where* $\varrho : \mathrm{STR}[\Sigma \cup \Pi_m] \to \mathrm{STR}[\Gamma]$ *is a parameterless* MSO*-transduction.*

Proof. When we apply $\mathrm{fgt}^{-1}_{\Pi_m}$ to a structure $\mathfrak{A}$ we obtain all possible expansions of $\mathfrak{A}$ by m unary relations $P_1, \dots, P_m \subseteq A$. The transduction ϱ can simulate τ by replacing the parameters by these relations. If $\mathfrak{B} = (\mathfrak{A}, \bar{P}) \in \mathrm{fgt}^{-1}_{\Pi_m}(\mathfrak{A})$ is a structure such that $\bar{P}$ does not satisfy the first formula of the definition scheme of τ then $\varrho(\mathfrak{B})$ is undefined. $\square$

Proposition 58. *If* $L \in \mathrm{Rec}(\mathrm{STR})_{\Sigma\cup\Pi_m}$ *then* $\mathrm{fgt}_{\Pi_m}(L) \in \mathrm{Rec}(\mathrm{STR})_\Sigma$.

Proof. The following obvious facts will be used.

(1) For all structures $\mathfrak{A}_0$, $\mathfrak{A}_1$, and $\mathfrak{C}$ and every m, we have

$$\mathfrak{A}_0 \oplus \mathfrak{A}_1 = \mathrm{fgt}_{\Pi_m}(\mathfrak{C})$$

if and only if there exist structures $\mathfrak{B}_0$ and $\mathfrak{B}_1$ such that

$$\mathfrak{C} = \mathfrak{B}_0 \oplus \mathfrak{B}_1 , \quad \mathfrak{A}_0 = \mathrm{fgt}_{\Pi_m}(\mathfrak{B}_0) , \quad \text{and} \quad \mathfrak{A}_1 = \mathrm{fgt}_{\Pi_m}(\mathfrak{B}_1) .$$

(2) For every quantifier-free operation $f : \mathrm{STR}[\Gamma] \to \mathrm{STR}[\Delta]$ and every m, there exists a quantifier-free operation $g : \mathrm{STR}[\Gamma \cup \Pi_m] \to \mathrm{STR}[\Delta \cup \Pi_m]$ such that, for all structures $\mathfrak{A}$ and $\mathfrak{B}$, we have

$$f(\mathfrak{A}) = \mathrm{fgt}_{\Pi_m}(\mathfrak{B})$$

if and only if there exists a structure $\mathfrak{C}$ with

$$\mathfrak{B} = g(\mathfrak{C}) \quad \text{and} \quad \mathfrak{A} = \mathrm{fgt}_{\Pi_m}(\mathfrak{C}) .$$

We apply a technique which was used in [16] to prove that certain operations on hypergraphs preserve recognizability. We fix m and we will write Π instead of Π_m. Let $\approx$ be a congruence witnessing the recognizability of a set $L \in \mathrm{Rec}(\mathrm{STR})_{\Sigma\cup\Pi}$. To show that $\mathrm{fgt}_\Pi(L)$ is recognizable we define an equivalence relation on each set $\mathrm{STR}[\Delta]$ by

$$\begin{aligned}\mathfrak{A} \equiv \mathfrak{B} \quad \text{iff} \quad & \big\{ [\mathfrak{C}] \bigm| \mathfrak{C} \in \mathrm{STR}[\Delta \cup \Pi],\ \mathrm{fgt}_\Pi(\mathfrak{C}) = \mathfrak{A} \big\} \\ = \ & \big\{ [\mathfrak{C}] \bigm| \mathfrak{C} \in \mathrm{STR}[\Delta \cup \Pi],\ \mathrm{fgt}_\Pi(\mathfrak{C}) = \mathfrak{B} \big\},\end{aligned}$$

where $[\mathfrak{C}]$ denotes the equivalence class of $\mathfrak{C}$ w.r.t. $\approx$.

Since $\approx$ is an equivalence relation with finitely many classes of each sort so is $\equiv$. Furthermore, $\equiv$ saturates $\mathrm{fgt}_\Pi(L)$. If $\mathfrak{A} = \mathrm{fgt}_\Pi(\mathfrak{C})$ with $\mathfrak{C} \in L$ and $\mathfrak{B} \equiv \mathfrak{A}$ then, by definition, there is some structure $\mathfrak{D} \approx \mathfrak{C}$ such that $\mathfrak{B} = \mathrm{fgt}_\Pi(\mathfrak{D})$. Hence $\mathfrak{D} \in L$ and $\mathfrak{B} \in \mathrm{fgt}_\Pi(L)$.

It remains to verify that $\equiv$ is a congruence. Suppose that $\mathfrak{A}_0 \equiv \mathfrak{B}_0$ and $\mathfrak{A}_1 \equiv \mathfrak{B}_1$. We want to prove that $\mathfrak{A}_0 \oplus \mathfrak{A}_1 \equiv \mathfrak{B}_0 \oplus \mathfrak{B}_1$.

By symmetry, it is sufficient, for each $\mathfrak{C} \in \mathrm{fgt}_\Pi^{-1}(\mathfrak{A}_0 \oplus \mathfrak{A}_1)$, to construct a structure $\mathfrak{D} \in \mathrm{fgt}_\Pi^{-1}(\mathfrak{B}_0 \oplus \mathfrak{B}_1)$ such that $\mathfrak{D} \approx \mathfrak{C}$. By (1), there are structures $\mathfrak{C}_0 \in \mathrm{fgt}_\Pi^{-1}(\mathfrak{A}_0)$ and $\mathfrak{C}_1 \in \mathrm{fgt}_\Pi^{-1}(\mathfrak{A}_1)$ such that $\mathfrak{C} = \mathfrak{C}_0 \oplus \mathfrak{C}_1$. By definition of $\equiv$, we can find structures $\mathfrak{D}_0 \approx \mathfrak{C}_0$ and $\mathfrak{D}_1 \approx \mathfrak{C}_1$ such that $\mathfrak{B}_0 = \mathrm{fgt}_\Pi(\mathfrak{D}_0)$ and $\mathfrak{B}_1 = \mathrm{fgt}_\Pi(\mathfrak{D}_1)$. Then $\mathrm{fgt}_\Pi(\mathfrak{D}_0 \oplus \mathfrak{D}_1) = \mathfrak{B}_0 \oplus \mathfrak{B}_1$ and, since $\approx$ is a $\mathcal{QF}$-congruence, we have $\mathfrak{C}_0 \oplus \mathfrak{C}_1 \approx \mathfrak{D}_0 \oplus \mathfrak{D}_1$, as desired.

Let $f : \mathrm{STR}[\Gamma] \to \mathrm{STR}[\Delta]$ be a quantifier-free operation and suppose that $\mathfrak{A} \equiv \mathfrak{B}$. We want to prove that $f(\mathfrak{A}) \equiv f(\mathfrak{B})$. Let $\mathfrak{C} \in \mathrm{fgt}_\Pi^{-1}(f(\mathfrak{A}))$. We have to find a structure $\mathfrak{D} \in \mathrm{fgt}_\Pi^{-1}(f(\mathfrak{B}))$ such that $\mathfrak{D} \approx \mathfrak{C}$. By (2), there exists a transduction g and some structure $\mathfrak{C}'$ such that $\mathfrak{C} = g(\mathfrak{C}')$ and $\mathfrak{A} = \mathrm{fgt}_\Pi(\mathfrak{C}')$. By definition of $\equiv$, we can find some structure $\mathfrak{D}' \approx \mathfrak{C}'$ with $\mathfrak{B} = \mathrm{fgt}_\Pi(\mathfrak{D}')$. Hence $\mathfrak{D} := g(\mathfrak{D}') \approx g(\mathfrak{C}') = \mathfrak{C}$ and $\mathrm{fgt}_\Pi(\mathfrak{D}) = f(\mathfrak{B})$. By symmetry, it follows that $f(\mathfrak{A}) \equiv f(\mathfrak{B})$. □

Proof of Theorem 51. By Lemmas 53 and 57, it follows that every k-copying MSO-transduction $\tau : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ with m parameters can be written as

$$\tau = \varrho \circ \mathrm{copy}_k \circ \mathrm{fgt}_{\Pi_m}^{-1},$$

where $\varrho : \mathrm{STR}[\Sigma \cup \Pi_m \cup \Upsilon_k] \to \mathrm{STR}[\Gamma]$ is a parameterless noncopying MSO-transduction and $\mathrm{copy}_k : \mathrm{STR}[\Sigma \cup \Pi_m] \to \mathrm{STR}[\Sigma \cup \Pi_m \cup \Upsilon_k]$.

Let $L \in \mathrm{Rec}(\mathrm{STR})_\Gamma$. Then

$$\tau^{-1}(L) = \mathrm{fgt}_{\Pi_m}(\mathrm{copy}_k^{-1}(\varrho^{-1}(L)))\,.$$

By Proposition 56, $\varrho^{-1}(L)$ is recognizable. Thus, $\mathrm{copy}_k^{-1}(\varrho^{-1}(L))$ is recognizable by Proposition 55. Finally, $\tau^{-1}(L) \in \mathrm{Rec}(\mathrm{STR})_\Sigma$, by Proposition 58. □

6. A small signature for the algebra of relational structures

Our basic signature for defining recognizable and equational sets of structures (or hypergraphs) is $\mathcal{QF}$. To show that this is a natural and robust choice we present several other signatures that all turn out to be equivalent to $\mathcal{QF}$. We have already seen in Lemma 17that the larger signatures $\mathcal{QF}_\alpha^{\mathrm{der}}$ are equivalent to $\mathcal{QF}$ and in Section 7 we will introduce more interesting examples of larger signatures. Before doing so let us try the opposite. In this section we consider a proper subsignature that is equivalent to $\mathcal{QF}$.

Let us first state some general facts that will serve as guidelines for proving our results. We claim that, to prove that a subsignature $G \subseteq \mathcal{QF}_\alpha^{\mathrm{der}}$ is equivalent to $\mathcal{QF}$, it suffices to prove the following two properties:

(P1) If a subset $L \subseteq \mathrm{STR}[\Sigma]$ is the image $\tau(K)$ of a regular set K of terms (over any signature) under an MSO-transduction τ, then there exists a regular subset $K' \subseteq T(G)$ such that $L = \mathrm{val}_{\mathrm{STR}}(K')$.
(P2) If a subset $L \subseteq \mathrm{STR}[\Sigma]$ is G-recognizable then it is $\mathcal{QF}$-recognizable.

Proposition 59. *Let $G \subseteq \mathcal{QF}_\alpha^{\mathrm{der}}$.*

(*a*) *If G satisfies* (P1) *then* $\mathrm{Equat}(G) = \mathrm{Equat}(\mathcal{QF})$.
(*b*) *If G satisfies* (P2) *then* $\mathrm{Rec}(G) = \mathrm{Rec}(\mathcal{QF})$.

In particular, any signature $G \subseteq \mathcal{QF}_\alpha^{\mathrm{der}}$ satisfying (P1) *and* (P2) *is equivalent to $\mathcal{QF}$. Furthermore, all signatures H with $G \subseteq H_\beta^{\mathrm{der}} \subseteq \mathcal{QF}_\alpha^{\mathrm{der}}$ are equivalent to $\mathcal{QF}$.*

Proof. Since $G \subseteq \mathcal{QF}_\alpha^{\mathrm{der}}$ and $\mathcal{QF}$ is equivalent to $\mathcal{QF}_\alpha^{\mathrm{der}}$ we have

$$\begin{aligned} \mathrm{Rec}(\mathcal{QF}) &= \mathrm{Rec}(\mathcal{QF}_\alpha^{\mathrm{der}}) \subseteq \mathrm{Rec}(G) \\ \text{and } \mathrm{Equat}(G) &\subseteq \mathrm{Equat}(\mathcal{QF}_\alpha^{\mathrm{der}}) = \mathrm{Equat}(\mathcal{QF})\,. \end{aligned}$$

Therefore, if G satisfies (P2) then we have $\mathrm{Rec}(\mathcal{QF}) = \mathrm{Rec}(G)$.

To prove (a), suppose that $L \in \mathrm{Equat}(\mathcal{QF})$. By Proposition 27 (iii), L is the image of a regular set of terms under an MSO-transduction. Hence, (P1) and Proposition 4 imply that $L \in \mathrm{Equat}(G)$.

Finally, suppose that $G \subseteq H_\beta^{\mathrm{der}} \subseteq \mathcal{QF}_\alpha^{\mathrm{der}}$. Then we have

$$\begin{aligned}&\mathrm{Equat}(\mathcal{QF}) = \mathrm{Equat}(G) \subseteq \mathrm{Equat}(H_\beta^{\mathrm{der}}) \subseteq \mathrm{Equat}(\mathcal{QF}_\alpha^{\mathrm{der}}) = \mathrm{Equat}(\mathcal{QF})\\ &\text{and } \mathrm{Rec}(\mathcal{QF}) = \mathrm{Rec}(\mathcal{QF}_\alpha^{\mathrm{der}}) \subseteq \mathrm{Rec}(H_\beta^{\mathrm{der}}) \subseteq \mathrm{Rec}(G) = \mathrm{Rec}(\mathcal{QF})\,.\end{aligned}$$

Since, by Lemma 17, H is equivalent to H_β^{der}, the result follows. □

6.1. Auxiliary relations of small arity

We define a subsignature $\mathcal{QF}_0$ of $\mathcal{QF}$ by retaining from the unary operations the particular operations that forget some relation (delete the corresponding hyperedges), rename some relation (relabel the corresponding hyperedges), and build new relations from pairs of given relations of smaller arity (create new hyperedges by concatenation of existing ones).

Definition 60. The unary operations of $\mathcal{QF}_0$ are the following ones:

(1) The *forget* operation $\mathrm{fgt}_\Lambda : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Sigma \setminus \Lambda]$ deletes all R-hyperedges, for $R \in \Lambda \subseteq \Sigma$.
(2) For an arity-preserving map $h : \Sigma \to \Gamma$ between signatures, we have the *relabelling* $\mathrm{relab}_h : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ that replaces every hyperedge label R by $h(R)$.
(3) Let $R, S, T \in \Sigma$, $k := \mathrm{ar}(R)$, $l := \mathrm{ar}(S)$, $m := \mathrm{ar}(T)$, and suppose that $h : [m] \to [k + l]$ is surjective. The *hyperedge addition* $\mathrm{add}_{R,S,T,h}$ has a defining formula $\vartheta_T(\bar{x})$ of the form

$$T\bar{x} \vee \Big(Rx_{i_1} \cdots x_{i_k} \wedge Sx_{i_{k+1}} \cdots x_{i_{k+l}} \wedge \bigwedge \{\, x_j = x_{j'} \mid h(j) = h(j')\,\}\Big)$$

where i_j is the smallest element of $h^{-1}(j)$.

Remark 61. This operation adds a T-hyperedge of length m for each pair of an R-hyperedge and an S-hyperedge (which may have loops and common vertices). The resulting T-hyperedge may be a loop.

We denote by $\mathcal{QF}_0$ the signature consisting of the above operations, the disjoint union, and all constants for singleton structures. By $\mathcal{QF}_0[\Sigma]$ we denote the subsignature of all those operations that refer only to relations in Σ.

In the proposition below we will make use of the following normal form of MSO-transductions.

Lemma 62. *Given a finite signature F, a regular set of terms $K \subseteq T(F)$, and an* MSO*-transduction $\tau : \mathrm{STR}[\Delta(F)] \to \mathrm{STR}[\Sigma]$, we can construct a finite signature F', a regular set $K' \subseteq T(F')$, and an* MSO*-transduction $\tau' : \mathrm{STR}[\Delta(F')] \to \mathrm{STR}[\Sigma]$ such that $\tau(K) = \tau'(K')$ and F', K', and τ' have the following additional properties:*

(1) *F' contains only constants and binary function symbols.*
(2) *τ' is noncopying and parameterless.*
(3) *For every $t' \in K'$, the relational structure $\tau'(t')$ is defined and its domain consists only of leaves of t'.*

Proof. In three steps, we transform F, τ, K into F', τ', K' with the above properties. The same construction is used in the proof of Theorem 4.6 of [7]. Hence we only sketch the different steps.

Step 1: Eliminating parameters. Suppose that the transduction τ uses m parameters $X_1, \dots, X_m$. We replace F by the signature $F' := F \times \{0,1\}^m$ where the symbol $(f, \bar{b}) \in F'$ has the same arity as f. Every term $t' \in T(F')$ encodes a pair $(t, \langle P_1, \dots, P_m \rangle)$ where $t \in T(F)$ is the projection of t' to the first component and the set P_i consists of those nodes of t' that are labelled by a pair $(f, \bar{b})$ with $b_i = 1$. Thus, every term in $T(F')$ contains an F-term and the values of the parameters $X_1, \dots, X_m$. The set K'_0 of all those terms which encode a pair $(t, \bar{P})$ for which $\tau(t, \bar{P})$ is defined is regular. This is a standard construction, based on the result by Doner, Thatcher, and Wright stating that a set of terms is regular if and only if the corresponding set of structures encoding them is MSO-definable (see Chapter 3 of [3]). It follows that the subset $K' \subseteq K'_0$ of all terms encoding pairs $(t, \bar{P})$ with $t \in K$ is also regular.

Step 2: Making τ noncopying and satisfy condition (3). By the first step, we can assume that τ is parameterless. Suppose that it is k-copying for $k \geqslant 1$. We increase the arity of each symbol in F (including constants) by k and we add a new constant, say, $*$. Let F' be the resulting signature. We define a transformation $T(F) \to T(F') : t \mapsto t^*$ of terms by:

$$\begin{aligned} c^* &:= c(*, \dots, *)\,, \\ f^*(t_1, \dots, t_n) &:= f(t_1^*, \dots, t_n^*, *, \dots, *)\,, \end{aligned}$$

where we add k times $*$ in each case. Since * is a tree transduction it follows by Lemma 1 that the image $K^* \subseteq T(F')$ of K is regular. The nodes corresponding to the new constants $*$ are all leaves, and they offer enough space to define the domain of the output structure, without the need to use several copies of the term. Hence, we can construct an MSO-transduction τ' that is (still parameterless and) noncopying such that $\tau(t) = \tau'(t^*)$, for each $t \in K$.

Note that even if τ is noncopying we have to perform this transformation to satisfy the second part of condition (3).

Step 3: Removing non-binary function symbols. By the first two steps, we can assume that conditions (2) and (3) hold. We can satisfy condition (1) as follows. Let F' be the signature obtained from F by adding a new constant $\perp$ and changing the arity of all functions symbols to 2. The operation $T(F) \to T(F') : t \to t^\perp$ with:

$$\begin{aligned} c^\perp &:= c\,, \\ f(t)^\perp &:= f(t^\perp, \perp)\,, \\ f(t_1, t_2)^\perp &:= f(t_1^\perp, t_2^\perp)\,, \\ f(t_1, \dots, t_k)^\perp &:= f(t_1^\perp, f(t_2^\perp, (\dots f(t_{k-1}^\perp, t_k^\perp) \dots)))\,, \quad \text{for } k \geqslant 3\,, \end{aligned}$$

preserves regularity. In the same way as above it follows that the image of K under $^\perp$ is regular. □

The following result strengthens the implication (iii) ⇒ (ii) of Proposition 27. Recall the notion of rank introduced in Definition 37.

Proposition 63. *Let K be a regular set of terms and τ an* MSO*-transduction with $\tau(K) \subseteq \mathrm{STR}[\Sigma]$. There exists a finite set of relations Γ with $\mathrm{ar}(\Gamma) \leqslant \mathrm{ar}(\Sigma) - 1$ and a regular set $M \subseteq T(\mathcal{QF}_0[\Sigma \cup \Gamma])$ such that $\tau(K) = \mathrm{val}_{\mathrm{STR}}(M)$.*

Proof. Suppose that $K \subseteq T(F \cup C)$, $\Delta := \Delta(F \cup C)$, and $\tau : \mathrm{STR}[\Delta] \to \mathrm{STR}[\Sigma]$. We assume that K, τ, and $F \cup C$ satisfy conditions (1)–(3) of Lemma 62 where C is a set of constants and F a set of binary function symbols. Furthermore, we may assume that every structure in $\tau(K)$ contains at least 2 elements. Let k be the quantifier height of τ and set $n := \mathrm{ar}(\Sigma)$. Our aim is to construct a relational signature Γ with $\mathrm{ar}(\Gamma) = n - 1$ and a regular subset $M \subseteq T(\mathcal{QF}_0[\Sigma \cup \Gamma])$ such that $\tau(K) = \mathrm{val}_{\mathrm{STR}}(M)$.

1. Overview of the proof. The signature Γ will consist of three disjoint copies of $\Delta_{\mathrm{M}}^{n-1,k}$. We define a function $\kappa : K \to T(\mathcal{QF}_0[\Sigma \cup \Gamma]^{\mathrm{der}})$ such that

$$\mathrm{val}_{\mathrm{STR}}(\kappa(t)) = \tau(t)\,, \quad \text{for all } t \in K\,.$$

The mapping κ replaces every binary function symbol f at a node u of t by a binary derived operation of the form $\mu_u(x_1 \oplus x_2)$ where μ_u is a composition of unary $\mathcal{QF}_0[\Gamma]$-operations. Similarly, it replaces a constant c at a leaf u by a constant $\gamma_u \in \mathcal{QF}_0[\Gamma]$. Let us denote the set of these terms μ_u and γ_u by Π. The definition of μ_u and γ_u will depend only on f, c, and $\mathrm{tp}_{k+3}(u/\mathfrak{S}(t))$. This implies that Π is finite and, by Lemma 34, there exist MSO-formulas $\varphi_\alpha(x)$, for $\alpha \in \Pi$, such that, for every node u of t

$$\mu_u \text{ or } \gamma_u \text{ is equal to } \alpha \quad \text{iff} \quad \mathfrak{S}(t) \models \varphi_\alpha(u)\,.$$

Since the required information is expressible in MSO it follows that the transformation κ can be performed by a tree transducer. Using the fact that K is regular we conclude that $\kappa(K)$ is a regular subset of $T(\mathcal{QF}_0[\Gamma]^{\mathrm{der}})$. Furthermore, we have

$$\tau(K) = \mathrm{val}_{\mathrm{STR}}(\kappa(K)) = \mathrm{val}_{\mathrm{STR}}(M),$$

where M is obtained from $\kappa(K)$ by replacing each derived operation by its definition. By Lemma 17, it follows that M is a regular subset of $T(\mathcal{QF}_0[\Gamma])$. This completes the proof.

2. Definition of κ. It remains to define κ. Let $\Gamma := \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$ where

$$\Gamma_0 := \Delta_{\mathrm{M}}^{n-1,k} \quad \text{and} \quad \Gamma_i := \{\, T_p^i \mid T_p \in \Gamma_0 \,\}\,, \quad \text{for } i \in \{1,2\}\,.$$

Let $h^i : \Gamma_0 \to \Gamma_i$ be the canonical bijections $T_p \mapsto T_p^i$. Note that these mappings preserve arities. Recall that t/u denotes the subterm of t rooted at u and that $\mathcal{R}_k^m$ denotes the restricted monadic annotation (cf. Definition 48). The construction of κ will ensure that, for every $t \in K$,

(1) for every node u of t, we have

$$\mathrm{fgt}_\Gamma\big(\mathrm{val}_{\mathrm{STR}}(\kappa(t/u))\big) = \tau(t)[L(t/u) \cap D]\,,$$

where D denotes the domain of $\tau(t)$,

(2) and, for every node u of t that is not the root,

$$\mathrm{fgt}_\Sigma\big(\mathrm{val}_{\mathrm{STR}}(\kappa(t/u))\big) = \mathrm{relab}_{h^i}(\mathcal{R}_k^{n-1}(t/u))\,,$$

where

$$i := \begin{cases} 1 & \text{if } u \text{ is the left successor of its parent,} \\ 2 & \text{if } u \text{ is the right successor of its parent.} \end{cases}$$

Condition (2) specifies the values of the auxiliary relations in Γ at inner nodes u of t. We use the distinct copies Γ_1 and Γ_2 of the signature to distinguish between left and right successors.

Note that $\kappa(t)$ is obtained from t by replacing constants by constants and function symbols by function symbols of the same arity. Hence, $\kappa(t)$ and t have the same underlying trees and the same set of nodes.

3. Definition of γ_u. It is straightforward to define the constants γ_u such that condition (2) is satisfied. If u does not belong to the domain of the structure $\tau(t)$ then we set $\gamma_u := \emptyset$, where $\emptyset$ is a new constant denoting the empty structure (which we also denote by $\emptyset$ without risk of ambiguity). This constant is not in the signature $\mathcal{QF}_0[\Sigma \cup \Gamma]$ and we will eliminate it at the very last stage of our proof.

Otherwise, let γ_u be the constant that denotes the structure

$$\tau(t)[u] \cup \mathrm{relab}_{h^i}(\mathcal{R}_k^{n-1}(t/u)),$$

where $i := 1$ if u is a left successor and $i := 2$ if u is a right successor. This structure consists of the single element u, the incident Σ-hyperedges of rank 1 of $\tau(t)$ (they are defined by $\tau(t)[u]$) together with the Γ-hyperedge of arity 1 that defines the (i-copy of the) monadic 1-type of u in $\mathfrak{S}(t/u)$ (this is defined by $\mathrm{relab}_{h^i}(\mathcal{R}_k^{n-1}(t/u))$). It is the unique structure $\mathfrak{A} \in \mathrm{STR}[\Sigma \cup \Gamma]$ such that

$$\mathrm{fgt}_\Gamma(\mathfrak{A}) = \tau(t)[u] \quad \text{and} \quad \mathrm{fgt}_\Sigma(\mathfrak{A}) = \mathrm{relab}_{h^i}(\mathcal{R}_k^{n-1}(t/u))\,.$$

Note that the structure $\mathfrak{S}(t/u)$ consists of a single node labelled by some constant c. Hence, $\mathrm{tp}_k(u/\mathfrak{S}(t/u))$ can be computed from c. The Σ-hyperedges of rank 1 are determined by $\mathrm{tp}_k(u/\mathfrak{S}(t))$.

4. Definition of μ_u. To define the mappings μ_u, we recall that, by Lemma 50, there are functions $\odot^f_{m,n}$ and $\hat{\odot}^f_{m,n}$ such that

- for all $t_1, t_2 \in T(F \cup C)$ and all increasing tuples $\bar{a} \in L(t_1)^m$ and $\bar{b} \in L(t_2)^n$, we have

$$(*)\mathrm{tp}_k\big(\bar{a}\bar{b}/\mathfrak{S}(f(t_1,t_2))\big) = \mathrm{tp}_k\big(\bar{a}/\mathfrak{S}(t_1)\big) \odot^f_{m,n} \mathrm{tp}_k\big(\bar{b}/\mathfrak{S}(t_2)\big)\,,$$

- for every $t \in T(F \cup C)$ such that $t/u = f(t_1, t_2)$ and all increasing tuples $\bar{a} \in L(t_1)^m$ and $\bar{b} \in L(t_2)^n$, we have

$$(**)\mathrm{tp}_k\big(\bar{a}\bar{b}/\mathfrak{S}(t)\big) = \hat{\odot}^f_{m,n}\big(\mathrm{tp}_k(\mathfrak{S}(t \setminus u)),\ \mathrm{tp}_k(\bar{a}/\mathfrak{S}(t_1)),\ \mathrm{tp}_k(\bar{b}/\mathfrak{S}(t_2))\big)\,.$$

To satisfy condition (2) we define the operation μ_u such that, for all terms t_1 and t_2,

$$\mathrm{relab}_{h^i}\big(\mathcal{R}_k^{n-1}(f(t_1,t_2))\big) = \mu_u\big(\mathrm{relab}_{h^1}(\mathcal{R}_k^{n-1}(t_1)) \oplus \mathrm{relab}_{h^2}(\mathcal{R}_k^{n-1}(t_2))\big)\,,$$

where i is either 1 or 2 depending on whether u is a left successor or a right successor. (The case where u is the root will be treated separately below.)

Let $\bar{a} \in L(t_1)^{m_1}$ and $\bar{b} \in L(t_2)^{m_2}$ be increasing with $m_1, m_2 > 0$ and $m_1 + m_2 \leqslant n - 1$. The operation μ_u has to compute the type of $\bar{a}\bar{b}$ in $\mathfrak{S}(f(t_1,t_2))$ from the types $\mathrm{tp}_k(\bar{a}/\mathfrak{S}(t_1))$ and $\mathrm{tp}_k(\bar{b}/\mathfrak{S}(t_2))$. This can be done with the help of the operation $\odot^f_{m_1,m_2}$. Let ADD_Γ be the composition (in any order) of the operations $\mathrm{add}_{T^1_p,T^2_q,T_r}$ where $p \in S^{m_1,k}_{\mathrm{M}}(\Delta)$, $q \in S^{m_2,k}_{\mathrm{M}}(\Delta)$ and $r := p \odot^f_{m_1,m_2} q$.

Furthermore, μ_u also has to update the type of tuples $\bar{a} \in L(t_j)^m$, $j \in \{1,2\}$. Note that

$$\mathrm{tp}_k(\bar{a}/\mathfrak{S}(f(t_1,t_2))) = \mathrm{tp}_k(\bar{a}/\mathfrak{S}(t_1)) \odot^f_{m,0} \mathrm{tp}_k(\mathfrak{S}(t_2)), \quad \text{for } \bar{a} \in L(t_1)^m,$$
$$\mathrm{tp}_k(\bar{a}/\mathfrak{S}(f(t_1,t_2))) = \mathrm{tp}_k(\mathfrak{S}(t_1)) \odot^f_{0,m} \mathrm{tp}_k(\bar{a}/\mathfrak{S}(t_2)), \quad \text{for } \bar{a} \in L(t_2)^m.$$

Let $g : \Gamma_1 \cup \Gamma_2 \to \Gamma$ be the mapping with

$$g(T^1_p) := T_q \quad \text{with} \quad q := p \odot^f_{m,0} \mathrm{tp}_k(\mathfrak{S}(t_2)),$$
$$g(T^2_p) := T_q \quad \text{with} \quad q := \mathrm{tp}_k(\mathfrak{S}(t_1)) \odot^f_{0,m} p.$$

We can define

$$\mu_u := \mathrm{relab}_{h^i} \circ \mathrm{relab}_g \circ \mathrm{ADD}_\Gamma \circ \mathrm{ADD}_\Sigma,$$

where the term ADD_Σ is defined below to satisfy condition (1), and i is either 1 or 2 depending on whether u is a left successor or a right successor.

Note that ADD_Γ depends on f but not on $\mathrm{tp}_k(u/\mathfrak{S}(t))$. The mapping g depends on $\mathrm{tp}_k(\mathfrak{S}(t_1))$ and $tp_k(\mathfrak{S}(t_2))$ and, hence, on $\mathrm{tp}_{k+3}(u/\mathfrak{S}(t))$. (Since the tree ordering relation is expressed by an MSO-formula of quantifier height 3 (see Section 4.4) it follows that $\mathrm{tp}_k(\mathfrak{S}(t/u))$ can be computed from $\mathrm{tp}_{k+3}(u/\mathfrak{S}(t))$ by relativization to the formula defining the nodes below u in t.)

5. Satisfying condition (1). The incomplete definitions of γ_u and μ_u given above result in a structure $\kappa(t) \in \mathrm{STR}[\Gamma \cup \Sigma]$ with Γ-hyperedges of arity and rank at most $n-1$ where the only Σ-hyperedges are those of $\tau(t) \in \mathrm{STR}[\Sigma]$ that have rank 1. To complete the definition of μ_u we have to define the term ADD_Σ which adds the missing Σ-hyperedges.

Suppose that $\bar{a} \in L(t)^r$ has rank $s \leqslant n$. There exists a unique surjective map $\sigma : [r] \to [s]$ and a unique increasing s-tuple $\bar{b}$ such that $a_i = b_{\sigma(i)}$, for all $1 \leqslant i \leqslant r$. We will denote this tuple by $\bar{a}^\sigma := \bar{b}$.

Let $\vartheta_U(x_1,\dots,x_r)$ be the formula of the definition scheme of τ that defines the relation $U \in \Sigma$ and set $\vartheta^\sigma_U(x_1,\dots,x_s) := \vartheta_U(x_{\sigma(1)},\dots,x_{\sigma(r)})$. We have

$$\begin{aligned} \bar{a} \in U_{\tau(t)} \quad &\text{iff } \mathfrak{S}(t) \models \vartheta_U(\bar{a}) \\ &\text{iff } \mathfrak{S}(t) \models \vartheta^\sigma_U(\bar{a}^\sigma) \\ &\text{iff } \mathrm{tp}_k(\bar{a}^\sigma/\mathfrak{S}(t)) \models \vartheta^\sigma_U. \end{aligned}$$

Suppose that $t/u = f(t_1,t_2)$. The operation ADD_Σ will create all Σ-hyperedges $\bar{a}$ with $\bar{a} \cap L(t_1) \neq \emptyset$ and $\bar{a} \cap L(t_2) \neq \emptyset$. Note that, for such a tuple $\bar{a}$, we have $\bar{a}^\sigma = \bar{c}\bar{d}$ where $\bar{c}$ is an increasing tuple in $L(t_1)$ and $\bar{d}$ is an increasing tuple in $L(t_2)$.

For each $U \in \Sigma$ and σ, we have to choose pairs p, q of types such that the operation $\mathrm{add}_{T^1_p,T^2_q,U,\sigma}$ adds the right tuples to U. Hence, the situation is similar to that of ADD_Γ except that we are interested in the type $\mathrm{tp}_k(\bar{a}^\sigma/\mathfrak{S}(t))$ and not in $\mathrm{tp}_k(\bar{a}^\sigma/\mathfrak{S}(t/u))$. We can compute this type with the help of the operation $\hat{\odot}^f_{m_1,m_2}$. Thus, we define ADD_Σ as the composition (in any order) of all operations $\mathrm{add}_{T^1_p,T^2_q,U,\sigma}$ where $p \in S^{m_1,k}_{\mathrm{M}}(\Delta)$, $q \in S^{m_2,k}_{\mathrm{M}}(\Delta)$, $m_1, m_2 > 0$, $m_1 + m_2 \leqslant n-1$, $\sigma : [\mathrm{ar}(U)] \to [m_1 + m_2]$ is surjective, and

$$\hat{\odot}^{f}_{m_1,m_2}\big(\mathrm{tp}_k(\mathfrak{S}(t \setminus u)), p, q\big) \models \vartheta^{\sigma}_{U} .$$

Note that the definition of ADD_Σ depends on $\mathrm{tp}_k(\mathfrak{S}(t \setminus u))$. Since the tree ordering can be defined by an MSO-formula of quantifier height 3 (see Section 4.4) it follows that $\mathrm{tp}_k(\mathfrak{S}(t \setminus u))$ can be computed from $\mathrm{tp}_{k+3}(u/\mathfrak{S}(t))$ (by relativizing all formulas to the set of those nodes that are not below u).

6. Final steps. We have not yet defined μ_u when u is the root. In this case we set $\mu_u := \mathrm{fgt}_\Gamma \circ \mathrm{ADD}_\Sigma$ where ADD_Σ is defined as above. After these operations are performed all Σ-tuples are in the right place. The relations in Γ are not needed anymore and we remove them with fgt_Γ.

We have constructed a regular set

$$K' := \kappa(K) \subseteq T\big(\mathcal{QF}_0[\Sigma \cup \Gamma]^{\mathrm{der}} \cup \{\emptyset\}\big)$$

with $\tau(K) = \mathrm{val}_{\mathrm{STR}}(K')$. It remains to remove the constant $\emptyset$. Note that $f(\emptyset) = \emptyset$, for every quantifier-free operation f, and $\mathfrak{A} \oplus \emptyset = \emptyset \oplus \mathfrak{A} = \mathfrak{A}$, for every structure $\mathfrak{A}$. Using these equations we can eliminate all occurrences of $\emptyset$ in the terms of K'. (Since every structure in $\tau(K)$ is nonempty there is no term in K' which denotes the empty structure.) This is an easy task for a tree transducer. Hence K' can be replaced by a regular set $K'' \subseteq T(\mathcal{QF}_0[\Sigma \cup \Gamma]^{\mathrm{der}})$. Finally, we transform K'' into a set $M \subseteq T(\mathcal{QF}_0[\Sigma \cup \Gamma])$ as explained in part 1 above. This completes the proof. $\square$

Definition 64. We denote by $\mathcal{QF}_0[\Sigma, \Gamma]$ the subsignature of $\mathcal{QF}_0[\Sigma \cup \Gamma]$ that consists of disjoint union and

- the operations fgt_Λ, for $\Lambda \subseteq \Gamma$,
- only those relabellings relab_h where h is the identity on Σ,
- the operations $\mathrm{add}_{R,S,T,h}$ with $R, S \in \Gamma$, R not equal to S, and $T \in \Gamma \cup \Sigma$, and
- all constants.

Let $\mathcal{QF}^{\Sigma}_0$ be the union of all signatures of the form $\mathcal{QF}_0[\Sigma, \Gamma]$.

Remark 65. Note that the proof of the preceding proposition uses only the operations of $\mathcal{QF}_0[\Sigma, \Gamma]$. The set M we construct is a subset of $T(\mathcal{QF}_0[\Sigma, \Gamma])$. We have thus shown that we can construct every structure in $\mathrm{STR}[\Sigma]$ with the help of a set Γ of auxiliary symbols of arity $\mathrm{ar}(\Gamma) < \mathrm{ar}(\Sigma)$.

6.2. The case of graphs

As an example we apply the above result to graphs. Let $\Sigma = \{edg\}$. Since edg is a binary relation every equational set of graphs can be defined by a system of equations over a signature of the form $\mathcal{QF}_0[edg, \Pi]$ where Π contains only unary symbols. We compare such signatures with the signature $\mathcal{VR}$ reviewed in Section 3.5.

The operations in $\mathcal{QF}_0[edg, \Pi]$ are the disjoint union, constants, and the quantifier-free operations:

- fgt_Φ, for $\Phi \subseteq \Pi$,
- relab_h, for $h : \Pi \to \Pi$, and
- $\mathrm{add}_{P,Q,edg,h}$, with $P, Q \in \Pi$.

The mapping fgt_Φ, is the composition of the mappings fgt_P, for $P \in \Phi$. A mapping relab_h is a composition of mappings $\mathrm{ren}_{P\to Q}$. Depending on h, the mapping $\mathrm{add}_{P,Q,edg,h}$ is either $\mathrm{add}_{P,Q}$ or $\mathrm{add}_{Q,P}$. Hence, the signature $\mathcal{QF}_0^{\{edg\}}$ is, up to some details of writing, the one considered in Section 3.5.

We obtain Corollary 4.9 of [7] which states that equational sets of graphs need not be defined with operations that use relation symbols of arity more than 2 or operations that label edges. Only vertices must be labelled. More about this in Section 6.4.

6.3. The recognizable sets are also the same

Our objective is now to establish the result that both signatures $\mathcal{QF}_0^\Sigma$ and $\mathcal{QF}$ lead to the same notion of recognizability for subsets of $\mathrm{STR}[\Sigma]$. Recall Section 4 where we defined monadic types $\mathrm{tp}_k(\bar{a}/\mathfrak{A})$ and monadic annotations $\mathcal{M}_k^m(\mathfrak{A})$. In particular, k denotes the quantifier height and m is the maximal size of annotated tuples. We will make use of the following lemma which follows immediately from Lemma 35.

Lemma 66. *For every quantifier-free transduction $f : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ and each $m > 0$, there exists a mapping $f^m : \Sigma_{\mathrm{M}}^{m,0} \to \Gamma_{\mathrm{M}}^{m,0}$ such that, for all structures $\mathfrak{A} \in \mathrm{STR}[\Sigma]$ and all $D \subseteq A$, we have*

$$\mathcal{M}_0^m(f(\mathfrak{A})[D]) = \mathrm{relab}_{f^m}(\mathcal{M}_0^m(\mathfrak{A}[D]))\,.$$

Proof. Note that we have:

$$\begin{aligned} f(\mathfrak{A}[D]) &= f(\mathfrak{A})[D]\,, \\ \mathcal{M}_0^m(\mathfrak{A}[D]) &= \mathcal{M}_0^m(\mathfrak{A})[D]\,, \\ \text{and } \mathrm{relab}_{f^m}(\mathfrak{A}[D]) &= \mathrm{relab}_{f^m}(\mathfrak{A})[D]\,. \end{aligned}$$

Since f is nondeleting the mapping $\mathcal{M}_0^m(\mathfrak{A}) \mapsto \mathcal{M}_0^m(f(\mathfrak{A}))$ only manipulates the relations. For $p \in S_{\mathrm{M}}^{n,0}(\Sigma)$ with $n \leqslant m$, we can define the relabelling by

$$f^m(T_p) := T_{f_0^n(p)}\,,$$

where f_0^n is the function from Lemma 35. □

Proposition 67. *Every $\mathcal{QF}_0^\Sigma$-recognizable set $L \subseteq \mathrm{STR}[\Sigma]$ is $\mathcal{QF}$-recognizable.*

Before giving the proof let us state the following consequence of Propositions 63 and 67.

Theorem 68. *The signatures $\mathcal{QF}_0^\Sigma$ and $\mathcal{QF}$ yield the same equational sets and the same recognizable sets of structures in $\mathrm{STR}[\Sigma]$. Hence the signatures $\mathcal{QF}_0$ and $\mathcal{QF}$ are equivalent.*

Proof of Proposition 67. Suppose that $L \subseteq \mathrm{STR}[\Sigma]$ is $\mathcal{QF}_0^\Sigma$-recognizable and let $m := \mathrm{ar}(\Sigma)$. There exists a finite $\mathcal{QF}_0^\Sigma$-congruence saturating L. We denote the corresponding finite equivalence relations on $\mathrm{STR}[\Sigma \cup \Gamma]$ by $\simeq_\Gamma$ where Γ is a relational signature with $\mathrm{ar}(\Gamma) < m$.

For a relational signature Δ, let $\beta(\Delta) := \Delta_{\mathrm{M}}^{\leqslant m-1,0}$. With each quantifier-free operation $f : \mathrm{STR}[\Delta] \to \mathrm{STR}[\Sigma]$ we associate the function $\hat{f} : \mathrm{STR}[\Delta] \to \mathrm{STR}[\Sigma \cup \beta(\Delta)]$ with

$$\hat{f}(\mathfrak{A}) := f(\mathfrak{A}) \cup \mathcal{M}_0^{m-1}(\mathfrak{A})[D]$$

where $D \subseteq A$ is the domain of $f(\mathfrak{A})$. Note that the union above is not a disjoint one. The domain of $\hat{f}(\mathfrak{A})$ is that of $f(\mathfrak{A})$ and the relations are those of $f(\mathfrak{A})$ and those of $\mathcal{M}_0^{m-1}(\mathfrak{A})[D]$. We assume that $\beta(\Delta)$ is disjoint from Σ so there is no confusion. $\hat{f}$ is obviously a quantifier-free operation.

For $\mathfrak{A}, \mathfrak{B} \in \mathrm{STR}[\Delta]$ we define

$$\begin{aligned} \mathfrak{A} \approx \mathfrak{B} \ \text{ iff } \ & \mathrm{tp}_m(\mathfrak{A}) = \mathrm{tp}_m(\mathfrak{B}), \\ \text{and } \mathfrak{A} \equiv_\Delta \mathfrak{B} \ \text{ iff } \ & \mathfrak{A} \approx \mathfrak{B} \text{ and, for every quantifier-free operation} \\ & f : \mathrm{STR}[\Delta] \to \mathrm{STR}[\Sigma], \text{ we have } \hat{f}(\mathfrak{A}) \simeq_{\beta(\Delta)} \hat{f}(\mathfrak{B}). \end{aligned}$$

We claim that $\equiv_\Delta$ is a finite $\mathcal{QF}$-congruence, for all Δ, and that $\equiv_\Sigma$ saturates L. Clearly, $\equiv_\Delta$ is an equivalence relation. It is also finite since $\approx$ and $\simeq_{\beta(\Delta)}$ are finite and there are only finitely many quantifier-free operations $\mathrm{STR}[\Delta] \to \mathrm{STR}[\Sigma]$ (because Δ and Σ are finite).

To see that $\equiv_\Sigma$ saturates L assume that $\mathfrak{A} \in L$ and $\mathfrak{A} \equiv_\Sigma \mathfrak{B}$. Set $f := \mathrm{fgt}_{\beta(\Sigma)}$. We have $\hat{f}(\mathfrak{A}) \simeq_{\beta(\Sigma)} \hat{f}(\mathfrak{B})$, which implies that

$$\mathfrak{A} = f(\hat{f}(\mathfrak{A})) \simeq_\emptyset f(\hat{f}(\mathfrak{B})) = \mathfrak{B}.$$

Since $\simeq_\emptyset$ saturates L it follows that $\mathfrak{B} \in L$.

Next we check that $\approx$ is a congruence. In Corollary 44 we have shown this for the disjoint union. It is easy to see that for quantifier-free operations it can be derived from Lemma 35.

It remains to verify that $\equiv_\Delta$ is a congruence. Let $g : \mathrm{STR}[\Delta] \to \mathrm{STR}[\Delta']$ be a quantifier-free transduction and suppose that $\mathfrak{A} \equiv_\Delta \mathfrak{B}$. Since $\approx$ is a congruence we have $g(\mathfrak{A}) \approx g(\mathfrak{B})$. Let $f : \mathrm{STR}[\Delta'] \to \mathrm{STR}[\Sigma]$ be a quantifier-free operation. By definition, we have

$$\begin{aligned} (\hat{f} \circ g)(\mathfrak{A}) &= (f \circ g)(\mathfrak{A}) \cup \mathcal{M}_0^{m-1}(g(\mathfrak{A}))[D], \\ \text{and } (f \circ g)^\wedge(\mathfrak{A}) &= (f \circ g)(\mathfrak{A}) \cup \mathcal{M}_0^{m-1}(\mathfrak{A})[D], \end{aligned}$$

where D is the domain of the structure $(f \circ g)(\mathfrak{A})$. Therefore, it follows from Lemma 66 that there is some function $h : \Sigma \cup \beta(\Delta) \to \Sigma \cup \beta(\Delta')$ such that

$$(\hat{f} \circ g)(\mathfrak{A}) = \mathrm{relab}_h((f \circ g)^\wedge(\mathfrak{A}))$$

and h is the identity on Σ. Since $\mathrm{relab}_h \in \mathcal{QF}_0^\Sigma$ and

$$(f \circ g)^\wedge(\mathfrak{A}) \simeq_{\beta(\Delta)} (f \circ g)^\wedge(\mathfrak{B})$$

we have

$$\begin{aligned}\hat{f}(g(\mathfrak{A})) &= \mathrm{relab}_h((f \circ g)^\wedge(\mathfrak{A}))\\ &\simeq_{\beta(\Delta')} \mathrm{relab}_h((f \circ g)^\wedge(\mathfrak{B})) = \hat{f}(g(\mathfrak{B}))\,,\end{aligned}$$

which implies that $g(\mathfrak{A}) \equiv_\Delta g(\mathfrak{B})$.

It remains to consider the case of disjoint union. Suppose that $\mathfrak{A}_0 \equiv_\Delta \mathfrak{B}_0$ and $\mathfrak{A}_1 \equiv_\Delta \mathfrak{B}_1$. We have to prove that $\mathfrak{A}_0 \oplus \mathfrak{A}_1 \equiv_\Delta \mathfrak{B}_0 \oplus \mathfrak{B}_1$. We already know that $\mathfrak{A}_0 \oplus \mathfrak{A}_1 \approx \mathfrak{B}_0 \oplus \mathfrak{B}_1$. Let $f : \mathrm{STR}[\Delta] \to \mathrm{STR}[\Sigma]$ be a quantifier-free operation such that $f(\mathfrak{A}_0 \oplus \mathfrak{A}_1) \in \mathrm{STR}[\Sigma]$.

Claim 69. *Let $\beta'(\Delta)$ be a disjoint copy of $\beta(\Delta)$ and let h be the relabelling mapping $R \in \beta(\Delta)$ to $R' \in \beta'(\Delta)$. There exists a $\mathcal{QF}_0[\Sigma, \beta(\Delta) \cup \beta'(\Delta)]$-derived operation g such that*

$$\hat{f}(\mathfrak{A} \oplus \mathfrak{B}) = g(\hat{f}(\mathfrak{A}) \oplus h(\hat{f}(\mathfrak{B})))\,, \quad \textit{for all structures } \mathfrak{A} \textit{ and } \mathfrak{B}\,.$$

Assuming the claim to be true we continue the proof as follows. Since $\mathfrak{A}_0 \equiv_\Delta \mathfrak{B}_0$ and $\mathfrak{A}_1 \equiv_\Delta \mathfrak{B}_1$ we have

$$\hat{f}(\mathfrak{A}_0) \simeq_{\beta(\Delta)} \hat{f}(\mathfrak{B}_0) \quad \text{and} \quad h(\hat{f}(\mathfrak{A}_1)) \simeq_{\beta'(\Delta)} h(\hat{f}(\mathfrak{B}_1))\,.$$

As g is a $\mathcal{QF}_0[\Sigma, \beta(\Delta) \cup \beta'(\Delta)]$-derived operation it follows that

$$\begin{aligned}\hat{f}(\mathfrak{A}_0) \oplus h(\hat{f}(\mathfrak{A}_1)) &\simeq_{\beta(\Delta)\cup\beta'(\Delta)} \hat{f}(\mathfrak{B}_0) \oplus h(\hat{f}(\mathfrak{B}_1))\,,\\ \text{and } \hat{f}(\mathfrak{A}_0 \oplus \mathfrak{A}_1) &= g(\hat{f}(\mathfrak{A}_0) \oplus h(\hat{f}(\mathfrak{A}_1)))\\ &\simeq_{\beta(\Delta)} g(\hat{f}(\mathfrak{B}_0) \oplus h(\hat{f}(\mathfrak{B}_1)))\\ &= \hat{f}(\mathfrak{B}_0 \oplus \mathfrak{B}_1)\,.\end{aligned}$$

This completes the main proof.

Proof of the claim. To define g let us consider the action of $\hat{f}$ on $\mathfrak{A} \oplus \mathfrak{B}$. Since $\hat{f}$ is quantifier-free it adds tuples $\bar{a} \subseteq A$ to a relation R if and only if we have $\bar{a} \in R_{\hat{f}(\mathfrak{A})}$. The same holds for tuples $\bar{b} \subseteq B$. Therefore, we have

$$\hat{f}(\mathfrak{A} \oplus \mathfrak{B})[A] = \hat{f}(\mathfrak{A}) \quad \text{and} \quad \hat{f}(\mathfrak{A} \oplus \mathfrak{B})[B] = \hat{f}(\mathfrak{B})\,,$$

and the desired operation g only needs to add those tuples $\bar{c}$ to relations R that contain elements of both A and B. Since $\hat{f}$ is quantifier-free we can tell whether such a tuple $\bar{c}$ should be added to R by looking at the quantifier-free types

$$\mathrm{tp}_0(\bar{c}|_A/\mathfrak{A} \oplus \mathfrak{B}) = \mathrm{tp}_0(\bar{c}|_A/\mathfrak{A}) \quad \text{and} \quad \mathrm{tp}_0(\bar{c}|_B/\mathfrak{A} \oplus \mathfrak{B}) = \mathrm{tp}_0(\bar{c}|_B/\mathfrak{B})\,.$$

(By $\bar{c}|_A$ we denote the subtuple of $\bar{c}$ contained in A.) This information is available in $\mathcal{M}_0^{m-1}(\mathfrak{A})$ and $\mathcal{M}_0^{m-1}(\mathfrak{B})$. Hence, g can be written as $g = \mathrm{relab}_k \circ \mathrm{CREATE}$ where k is the canonical projection $\beta(\Delta) \cup \beta'(\Delta) \to \beta(\Delta)$ and CREATE is a composition of operations of the form $\mathrm{add}_{R,S,T,h}$ with $R \in \beta(\Delta)$, $S \in \beta'(\Delta)$, and $T \in \Sigma \cup \beta(\Delta) \cup \beta'(\Delta)$. This completes the proof of the claim. □

6.4. Optimality

These results prove that when dealing with equational or recognizable sets of hypergraphs of arity at most n, auxiliary relation symbols (like the labels from sets Π for dealing with graphs) can be limited to be of arity at most $n-1$.

The next example shows that, for equational sets, this bound is optimal. We define structures of arity 3 that cannot be defined without auxiliary symbols of arity 2.

Example 70. Let R be a ternary relation symbol and Π a set of unary predicates as in Section 3.5. Consider the signature

$$\mathcal{F}_\Pi := \{\, \oplus, \mathrm{ren}_{P\to Q}, \mathrm{fgt}_\Lambda, \mathrm{add}_{N,P,Q}, \mathbf{P} \mid N, P, Q \in \Pi,\ \Lambda \subseteq \Pi \,\},$$

where $\oplus$, $\mathrm{ren}_{P\to Q}$, fgt_Λ, and $\mathbf{P}$ are the usual $\mathcal{VR}$-operations of Section 3.5 and $\mathrm{add}_{N,P,Q}$ is the quantifier-free operation defined by the formula

$$\vartheta_R(x, y, z) := Rxyz \vee (Nx \wedge Py \wedge Qz)\,.$$

Every structure $\mathfrak{A} \in \mathrm{STR}[R]$ is of the form $\mathfrak{A} = \mathrm{val}_{\mathrm{STR}}(t)$, for some $t \in T(\mathcal{F}_\Pi)$, provided Π is large enough (say, $|\Pi| = |A|$). Let $\mathfrak{A}_n \in \mathrm{STR}[R]$ be the structure with domain $A = [n]$ and relation

$$R := \{\, (a, b, c) \in [n]^3 \mid a < b < c \,\},$$

and denote the set of all structures $\mathfrak{A}_n$ by C. There exists an MSO-transduction τ such that $C = \tau(K)$, where K is the set of all terms of the form $g^n(c)$, $n \in \mathbb{N}$, for some unary function symbol g and a constant c. Since K is regular it follows by Proposition 27 that C is equational. We claim that $C \nsubseteq \mathrm{val}(T(\mathcal{F}_\Pi))$, for any finite set Π.

Fix a finite set Π and set $n := 2^{|\Pi|}$. We will prove that $\mathfrak{A}_{2n+1} \notin \mathrm{val}(T(\mathcal{F}_\Pi))$. Suppose that there exists a term $t \in T(\mathcal{F}_\Pi)$ with value $\mathrm{val}(t) = \mathfrak{A}_{2n+1}$. Then $t = f(t_1 \oplus t_2)$ where f is a composition of unary operations that has to add all necessary hyperedges between $\mathfrak{B}_1 := \mathrm{val}(t_1)$ and $\mathfrak{B}_2 := \mathrm{val}(t_2)$.

For $a, b \in \mathrm{val}(t_1)$, we define

$$a \sim b \quad \text{iff} \quad \text{for all } P \in \Pi,\ a \in P_{\mathfrak{B}_1} \Leftrightarrow b \in P_{\mathfrak{B}_1}.$$

If f adds the tuple (a, b, c) to R, for $a \sim b$ in B_1 and $c \in B_2$, then it must also add the tuple (b, a, c). This is not possible. Therefore, each $\sim$-class of B_1 contains only one element and we have

$$|B_1| = |B_1/{\sim}| \leqslant 2^{|\Pi|} = n\,.$$

By symmetry, it follows that $|B_2| \leqslant n$ in contradiction to $|B_1 \cup B_2| = 2n + 1$.

7. Rich signatures with operations based on local information

7.1. The general framework

After investigating small signatures we will now look at the opposite problem of defining signatures that are as rich as possible while still being equivalent to $\mathcal{QF}$. Let $\mathcal{F}$ be a signature equivalent to $\mathcal{QF}$. We are interested in finding a set $\mathcal{G}$ of new operations on STR[Σ] that satisfy the following conditions:

(C1) Every $(\mathcal{F} \cup \mathcal{G})$-equational subset of STR[Σ] is $\mathcal{F}$-equational.
(C2) Every $\mathcal{F}$-recognizable subset of STR[Σ] is $(\mathcal{F} \cup \mathcal{G})$-recognizable.

Lemma 71. *If $\mathcal{G}$ satisfies* (C1) *and* (C2) *then $\mathcal{F} \cup \mathcal{G}$ is equivalent to $\mathcal{QF}$.*

Proof. Since $\mathcal{F} \subseteq \mathcal{F} \cup \mathcal{G}$, we have

$$\mathrm{Rec}(\mathcal{F} \cup \mathcal{G}) \subseteq \mathrm{Rec}(\mathcal{F}) \quad \text{and} \quad \mathrm{Equat}(\mathcal{F}) \subseteq \mathrm{Equat}(\mathcal{F} \cup \mathcal{G})\,.$$

By (C2), it follows that $\mathrm{Rec}(\mathcal{F} \cup \mathcal{G}) = \mathrm{Rec}(\mathcal{F}) = \mathrm{Rec}(\mathcal{QF})$, while (C1) impies that $\mathrm{Equat}(\mathcal{F} \cup \mathcal{G}) = \mathrm{Equat}(\mathcal{F}) = \mathrm{Equat}(\mathcal{QF})$. $\square$

Our approach is as follows. Suppose that, for each signature Σ, we have defined an injective mapping

$$\wedge : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\hat{\Sigma}] : \mathfrak{A} \mapsto \hat{\mathfrak{A}}$$

from Σ-structures to $\hat{\Sigma}$-structures, for some signature $\hat{\Sigma}$. Natural conditions implying both (C1) and (C2) are the following ones.

(H) The family of functions $\wedge : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\hat{\Sigma}]$ forms a finite-state heteromorphism from the $(\mathcal{F} \cup \mathcal{G})$-algebra STR to the $\mathcal{QF}$-algebra STR.
(M) The mapping $\wedge$ has a left-inverse $\hat{\mathfrak{A}} \mapsto \mathfrak{A}$ that is an MSO-transduction. Furthermore, for every Σ, there is an MSO-formula defining the image $D_\Sigma := (\mathrm{STR}[\Sigma])^\wedge \subseteq \mathrm{STR}[\hat{\Sigma}]$ of STR[Σ] under $\wedge$.

Remark 72. By Definition 10, to verify (H) we have to find

- a $(\mathcal{F} \cup \mathcal{G})$-computable mapping $\alpha : \mathrm{STR} \to A$, and
- for every n-ary operation $f \in \mathcal{F} \cup \mathcal{G}$, $\mathcal{QF}$-terms $t^f[\bar{a}]$, for $\bar{a} \in A^n$, that "emulate" f.

Note that the second step can be performed independently for every operation f. Below we will sometimes split it into two or more parts each dealing only with a subset of $\mathcal{F} \cup \mathcal{G}$.

Lemma 73. *Let $C \subseteq \mathrm{STR}[\Sigma]$ be a set of structures and $\hat{C}$ its image under $\wedge$. If* (H) *and* (M) *hold then the following conditions are equivalent*:

(*i*) *C is $\mathcal{QF}$-equational.*
(*ii*) *$\hat{C}$ is $\mathcal{QF}$-equational.*
(*iii*) *C is $(\mathcal{F} \cup \mathcal{G})$-equational.*

In particular, (H) *and* (M) *imply* (C1).

Proof. (iii) $\Rightarrow$ (ii) follows from Lemma 14 and (H), and (ii) $\Rightarrow$ (i) follows from Corollary 28 (a) and (M).
For (i) $\Rightarrow$ (iii), suppose that C is $\mathcal{QF}$-equational. Since $\mathcal{F}$ is equivalent to $\mathcal{QF}$ it is also $\mathcal{F}$-equational. Finally, $\mathcal{F} \subseteq \mathcal{F} \cup \mathcal{G}$ implies that C is $(\mathcal{F} \cup \mathcal{G})$-equational. □

Lemma 74. *Let $C \subseteq \mathrm{STR}[\Sigma]$ be a set of structures and $\hat{C}$ its image under* $\hat{}$. *If* (H) *and* (M) *hold then the following conditions are equivalent*:

(*i*) *C is $\mathcal{QF}$-recognizable.*
(*ii*) *$\hat{C}$ is $\mathcal{QF}$-recognizable.*
(*iii*) *C is $(\mathcal{F} \cup \mathcal{G})$-recognizable.*

In particular, (H) *and* (M) *imply* (C2).

Proof.

(i) $\Rightarrow$ (ii) Since $\hat{C} = D_\Sigma \cap (\hat{})^{-1}(C)$ this direction follows from (M), Proposition 26 (b), and Theorem 51.
(ii) $\Rightarrow$ (iii) follows from Lemma 14 and (H).
(iii) $\Rightarrow$ (i) Suppose that C is $(\mathcal{F} \cup \mathcal{G})$-recognizable. Since $\mathcal{F} \subseteq \mathcal{F} \cup \mathcal{G}$ it is also $\mathcal{F}$-recognizable. By assumption, $\mathcal{F}$ is equivalent to $\mathcal{QF}$ which implies that C is $\mathcal{QF}$-recognizable. □

Example 75.

(a) We can apply the above machinery to the mapping $\hat{\mathfrak{A}} := \mathcal{M}_k^m(\mathfrak{A})$. Condition (M) follows from Lemma 40, and in Lemma 46 we proved (H) for the case that $\mathcal{G} = \emptyset$ and $\mathcal{F} = \mathcal{QF}$. It follows that a class C is $\mathcal{QF}$-equational or $\mathcal{QF}$-recognizable if and only if its annotated version $\mathcal{M}_k^m(C)$ is. Hence, our framework provides an alternative proof of Corollary 41 and Theorem 47.
(b) It is not easy to find nontrivial signatures $\mathcal{G}$ that satisfy condition (H) for the annotation $\mathcal{M}_k^m$. We give an example of a simple operation that, for $k > 0$, violates condition (H). Consider the square operation $G \mapsto G^2$ where G^2 is the graph with the same vertices as G and edge relation

$$edg_{G^2} := \{ (x, y) \mid (x, y) \in edg_G \text{ or } (x,z), (z, y) \in edg_G \text{ for some } z \} .$$

The mapping $\mathcal{M}_1(G) \mapsto G^2$ is a quantifier-free operation. To satisfy (H) we have to lift it to a map $\mathcal{M}_1(G) \mapsto \mathcal{M}_1(G^2)$. But this cannot be done. We have $G^2 \models \exists z(edg(x,z) \wedge edg(z, y))$ iff $G \models \exists z\big[\big(edg(x,z) \vee \exists u(edg(x,u) \wedge edg(u,z))\big) \wedge \big(edg(z, y) \vee \exists u(edg(z,u) \wedge edg(u, y))\big)\big]$. By looking only at $\mathrm{tp}_1(xy/G)$ we cannot decide whether this formula holds in G.
(c) We give a last counterexample consisting of an operation defined by a very weak form of quantification such that the corresponding value mapping from terms to graphs is not an *MSO*-transduction. Let P, Q, R be unary relations and suppose that our signature contains the operations g and h where

$$h(x) := (\mathrm{relab}_{R \mapsto Q} \circ \mathrm{relab}_{Q \mapsto P} \circ \mathrm{add}_{Q,R,edg})(x \oplus \mathbf{R})$$

is a derived $\mathcal{QF}_2^{\{edg\}}$-operation, and g labels every vertex a by Q that has a neighbor labelled Q while the other relations remain unchanged. The term $t_{mn} := g^n h^m(\mathbf{Q})$ describes a path of length m where the last $n + 1$ vertices are labelled by Q and the remaining ones are labelled by P.

$$P \longrightarrow \cdots \longrightarrow P \longrightarrow Q \longrightarrow \cdots \longrightarrow Q$$

We claim that the function val mapping a term t_{mn} to its value is not an MSO-transduction. Note that the set

$$T := \{ \operatorname{val}(t_{mn}) \mid m \leqslant n \},$$

which consists of all finite paths where all vertices are labelled by Q, is MSO-definable and, hence, recognizable. If val were an MSO-transduction then the set

$$\operatorname{val}^{-1}(T) \cap \{ t_{mn} \mid m, n \in \mathbb{N} \} = \{ t_{mn} \mid m \leqslant n \}$$

would be recognizable as well. But, using pumping arguments, one can easily see that this is not the case. The set T is $\mathcal{QF}$-recognizable but not recognizable with respect to the operations used above.

7.2. Fusion and local types

Our main application of the approach described in the previous section concerns the fusion operation that merges all elements of a structure satisfying a given quantifier-free formula into a single element. We will show that one can augment the signature $\mathcal{QF}_0$ of Section 6.1 by this operation without changing the notions of recognizability and equationality. Let us first introduce the appropriate operation $\mathfrak{A} \mapsto \hat{\mathfrak{A}}$ on structures. Similarly to the operation $\mathcal{M}_k^m$ of Section 4.2, we use a labelling by a certain kind of types but with a more restricted form of quantification.

Definition 76.

(a) Let $n \in \mathbb{N}$. A formula $\varphi(x_1, \ldots, x_n)$ is *monadically existential*, *m.e.* for short, if

$$\varphi(x_1, \ldots, x_n) = \exists y_1 \cdots \exists y_m (R y_1 \ldots y_m \wedge \psi_1 \wedge \ldots \wedge \psi_m)$$
$$\text{or } \varphi = \exists y_1 \psi_1 ,$$

where each ψ_i is either the Hintikka-formula (cf. Definition 33) of a quantifier-free 1-type with free variable y_i, or it is of the form $y_i = x_k$, for some k. (Note that we do not require every variable x_i to appear in φ.)

(b) Let $\mathfrak{A}$ be a structure and $\bar{a} \in A^n$, for $n \in \mathbb{N}$. The *local n-type* of $\bar{a}$ is the set

$$\operatorname{ltp}(\bar{a}/\mathfrak{A}) := \{ \varphi(\bar{x}) \mid \varphi \text{ is m.e.}, \mathfrak{A} \models \varphi(\bar{a}) \}.$$

The set of all local n-types realized in some Σ-structure is denoted by $S^n_{\mathrm{L}}(\Sigma)$ and we set $S^*_{\mathrm{L}}(\Sigma) := \bigcup_{1 \leqslant n \leqslant \operatorname{ar}(\Sigma)} S^n_{\mathrm{L}}(\Sigma)$. As usual, we abbreviate $\operatorname{ltp}(\langle\rangle/\mathfrak{A})$ by $\operatorname{ltp}(\mathfrak{A})$. Note that $\operatorname{ltp}(\mathfrak{A})$ is included in all local n-types with $n \geqslant 0$.

Example 77. Suppose that $\Sigma = \{R, P\}$ where R is 4-ary and P is unary. The following formula is m.e.

$$\begin{aligned}\varphi(x_1, x_2) = \exists y_1 \exists y_2 \exists y_3 \exists y_4 \big(R y_1 y_2 y_3 y_4 \wedge{} & (P y_1 \wedge \neg R y_1 y_1 y_1 y_1) \\ \wedge{} & \; y_2 = x_1 \wedge y_3 = x_1 \\ \wedge{} & (\neg P y_4 \wedge \neg R y_4 y_4 y_4 y_4)\big)\,.\end{aligned}$$

Remark 78. Note that the local type $\mathrm{ltp}(\bar{a}/\mathfrak{A})$ of a tuple uniquely determines its quantifier-free type $\mathrm{tp}_0(\bar{a}/\mathfrak{A})$ since we have

$$\begin{aligned} & R x_{i_1} \dots x_{i_m} \in \mathrm{tp}_0(\bar{a}/\mathfrak{A}) \\ \text{iff } & \exists y_1 \cdots \exists y_m (R\bar{y} \wedge y_1 = x_{i_1} \wedge \cdots \wedge y_m = x_{i_m}) \in \mathrm{ltp}(\bar{a}/\mathfrak{A})\,. \end{aligned}$$

As for monadic types we can annotate a structure with local types. This annotation is an FO-transduction which satisfies condition (M).

Definition 79. Let $\mathfrak{A}$ be a Σ-structure. The *local annotation* of $\mathfrak{A}$ is the structure

$$\mathcal{L}(\mathfrak{A}) := \big\langle A, (T_p)_{p \in S^*_{\mathrm{L}}(\Sigma)} \big\rangle$$

with the same domain as $\mathfrak{A}$ where, for each local n-type $p \in S^*_{\mathrm{L}}(\Sigma)$, $1 \leqslant n \leqslant \mathrm{ar}(\Sigma)$, we add an n-ary relation

$$T_p := \{\, \bar{a} \in A^n \mid \mathrm{ltp}(\bar{a}/\mathfrak{A}) = p \,\}\,.$$

We denote the signature of $\mathcal{L}(\mathfrak{A})$ by Σ_{L}.

The following lemma is the analogue of Lemma 40.

Lemma 80. *Let Σ be a relational signature.*

(a) *The mapping $\mathcal{L} : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Sigma_{\mathrm{L}}]$ is an injective* FO*-transduction of quantifier height* $\mathrm{ar}(\Sigma)$.
(b) *The function $\mathcal{L}$ has a left-inverse that is a quantifier-free* FO*-transduction.*
(c) *$\mathcal{L}$ satisfies condition* (M).

Proof. (a) We can define the relation T_p by the formula

$$\bigwedge p \wedge \bigwedge \{\, \neg\varphi \mid \varphi \text{ is m.e., } \varphi \notin p \,\}\,.$$

This formula has quantifier height $\mathrm{qh}(\psi_p) = \mathrm{ar}(\Sigma)$.
(b) Conversely, we can write an n-ary relation $R \in \Sigma$ as

$$\begin{aligned} R_{\mathfrak{A}} = \{\, \bar{a} \in A^n \mid{} & \bar{a} \in T_p \text{ for some } p \text{ with} \\ & \exists\, \bar{y}(R\bar{y} \wedge y_1 = x_1 \wedge \ldots \wedge y_n = x_n) \in p \,\}\,. \end{aligned}$$

Since $S^*_{\mathrm{L}}(\Sigma)$ is finite this definition is equivalent to a finite disjunction of atomic formulas.

(c) Having proved (b) it remains to show that $\mathcal{L}(\mathrm{STR}[\Sigma])$ is MSO-definable. By composing the transductions of (a) and (b) we can construct a first-order formula φ such that $\mathfrak{A} \models \varphi$ if and only if $\mathfrak{A} = \mathcal{L}(\mathfrak{B})$, for some structure $\mathfrak{B}$. □

We have seen in Theorem 68 that the signature $\mathcal{F} := \mathcal{QF}_0$ is equivalent to $\mathcal{QF}$. Using the methods of Section 7.1 we extend it in two steps to a larger signature that is still equivalent to $\mathcal{QF}$. First, we add all domain restrictions del_ψ (cf. the end of Section 3.1). Let $\mathcal{QF}_*$ be the resulting signature. We start by proving an analogue to Lemma 35 for local types.

Lemma 81. *For every unary operation $f \in \mathcal{QF}_*$ of type $\Sigma \to \Gamma$, there exist functions $f_n : S^n_{\mathrm{L}}(\Sigma) \to S^n_{\mathrm{L}}(\Gamma)$, $n \in \mathbb{N}$, such that*

$$\mathrm{ltp}(\bar{a}/f(\mathfrak{A})) = f_n(\mathrm{ltp}(\bar{a}/\mathfrak{A}))\,,$$

for all structures $\mathfrak{A}$ and all n-tuples $\bar{a}$ in $f(\mathfrak{A})$.

Proof. Let $g = f^1_0 : S^{1,0}_{\mathrm{M}}(\Sigma) \to S^{1,0}_{\mathrm{M}}(\Gamma)$ be the function from Lemma 35. If ψ is the Hintikka-formula of an atomic 1-type q we denote by $g(\psi)$ the Hintikka-formula of $g(q)$, and, if ψ is $y_i = x_k$, then we set $g(\psi) := \psi$.

Let $p \in S^n_{\mathrm{L}}(\Sigma)$. For an m.e. formula of the form $\varphi = \exists y \psi(y)$ we have

$$\exists y \psi(y) \in f_n(p) \quad \text{iff} \quad \exists y \psi'(y) \in p \quad \text{for some } \psi' \in g^{-1}(\psi)\,.$$

Consider an m.e. formula of the form

$$\varphi(x_1, \ldots, x_n) = \exists y_1 \cdots \exists y_m (R y_1 \ldots y_m \wedge \psi_1 \wedge \cdots \wedge \psi_m)\,.$$

To define $f_n(p)$ we consider the following cases.

(1) $f = \mathrm{fgt}_\Lambda$. If $R \in \Lambda$ then $\varphi \notin f_n(p)$. Otherwise, $\varphi \in f_n(p)$ iff there are formulas $\psi'_i \in g^{-1}(\psi_i)$, $i \leqslant m$, such that

$$\exists y_1 \cdots \exists y_m (R y_1 \ldots y_m \wedge \psi'_1 \wedge \cdots \wedge \psi'_m) \in p\,.$$

(2) $f = \mathrm{relab}_h$. We set $\varphi \in f_n(p)$ iff there are a relation $S \in h^{-1}(R)$ and formulas $\psi'_i \in g^{-1}(\psi_i)$, $i \leqslant m$, such that

$$\exists y_1 \cdots \exists y_m (S y_1 \ldots y_m \wedge \psi'_1 \wedge \cdots \wedge \psi'_m) \in p\,.$$

(3) $f = \mathrm{add}_{S,T,U,h}$. If $R \neq U$ then we define $\varphi \in f_n(p)$ iff there are formulas $\psi'_i \in g^{-1}(\psi_i)$, $i \leqslant m$, such that

$$\exists y_1 \cdots \exists y_m (R y_1 \ldots y_m \wedge \psi'_1 \wedge \cdots \wedge \psi'_m) \in p\,.$$

For $R = U$, we have $\varphi \in f_n(p)$ iff one of the following two cases holds.

Case 1. There are formulas $\psi'_i \in g^{-1}(\psi_i)$, $i \leqslant m$, such that

$$\exists y_1 \cdots \exists y_m (U y_1 \dots y_m \wedge \psi'_1 \wedge \cdots \wedge \psi'_m) \in p\,.$$

Case 2. Otherwise, for all i, j with $h(i) = h(j)$, we have either

- $\psi_i = \psi_j$, or
- ψ_i is $y_i = x_k$ and ψ_j is the Hintikka-formula of the type $\mathrm{tp}_0(a_k/f(\mathfrak{A})) = g\big(\mathrm{tp}_0(a_k/\mathfrak{A})\big)$ (note that this type is determined by p), or
- vice versa.

Furthermore, there are formulas $\psi''_1, \dots, \psi''_{k+l}$, where $k := \mathrm{ar}(S)$, $l := \mathrm{ar}(T)$, such that

$$\begin{aligned} &\exists y_1 \cdots \exists y_k (S y_1 \dots y_k \wedge \psi''_1 \wedge \cdots \wedge \psi''_k) \in p \\ \text{and }\ &\exists y_1 \cdots \exists y_l (T y_1 \dots y_l \wedge \psi''_{k+1}(y_1) \wedge \cdots \wedge \psi''_{k+l}(y_l)) \in p\,, \end{aligned}$$

and, for all i, we either have

- ψ_i is a Hintikka-formula and $\psi''_{h(i)} \in g^{-1}(\psi_i)$, or
- ψ_i is $y_i = x_j$, for some j, and $\psi''_{h(i)}$ is $y_{h(i)} = x_j$.

(4) $f = \mathrm{del}_\vartheta$. We have $\varphi \in f_n(p)$ iff $\varphi \in p$ and $\psi_i \not\models \vartheta(y_i)$, for all $i \leqslant m$. $\square$

Example 82. Let us illustrate the case $f = \mathrm{add}_{S,T,U,h}$. Suppose that the arities of S, T, and U are 2, 3, and 7, respectively. Let $h : [7] \to [5]$ be the function mapping $1, \dots, 7$ to the sequence $1, 2, 3, 4, 4, 5, 5$. We consider a formula $\varphi(x_1, x_2, x_3)$ of the form

$$\exists \bar{y} (U \bar{y} \wedge y_1 = x_1 \wedge \psi_2(y_2) \wedge y_3 = x_2 \wedge y_4 = x_3 \wedge \psi_5(y_5) \wedge \psi_6(y_6) \wedge \psi_7(y_7))\,.$$

For $\bar{a} \in A^3$, we have $f(\mathfrak{A}) \models \varphi(\bar{a})$ iff either

$$\mathfrak{A} \models \exists \bar{y} (U \bar{y} \wedge y_1 = x_1 \wedge \psi'_2(y_2) \wedge y_3 = x_2 \wedge y_4 = x_3 \wedge \psi'_5(y_5) \wedge \psi'_6(y_6) \wedge \psi'_7(y_7))\,,$$

for some $\psi'_i \in g^{-1}(\psi_i)$, $i \in \{2, 5, 6, 7\}$, or ψ_5 is the Hintikka-formula of $g(\mathrm{tp}_0(a_3/\mathfrak{A}))$, we have $\psi'_6 = \psi_7$, and there are $\psi'_i \in g^{-1}(\psi_i)$, $i \in \{2, 6\}$, such that

$$\mathfrak{A} \models \exists y_1 \exists y_2 (S y_1 y_2 \wedge y_1 = x_1 \wedge \psi'_2(y_2)) \wedge \exists y_1 \exists y_2 \exists y_3 (T y_1 y_2 y_3 \wedge y_1 = x_2 \wedge y_2 = x_3 \wedge \psi'_6(y_3))\,.$$

The next lemma is analogous to Corollary 44.

Lemma 83. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures and $\bar{a} \in A^k$, $\bar{b} \in B^l$ with $k, l \geqslant 0$.*

$$\mathrm{ltp}(\bar{a}\bar{b}/\mathfrak{A} \oplus \mathfrak{B}) = \mathrm{ltp}(\bar{a}/\mathfrak{A}) \cup p\,,$$

where p is the type obtained from $\mathrm{ltp}(\bar{b}/\mathfrak{B})$ by replacing every variable x_i by x_{k+i}.

Corollary 84. *Every operation $f \in \mathcal{QF}_*$ satisfies condition* (H).

Proof. We claim that the function $\mathcal{L}$ is a finite-state heteromorphism based on ltp. The proof is analogous to that of Lemma 46. For unary operations the claim follows immediately from Lemma 81. It remains to consider the disjoint union. Lemma 83 implies that there exist $\mathcal{QF}$-terms $t[p,q]$, for $p,q \in S^0_\mathrm{L}(\Sigma)$, such that

$$\mathcal{L}(\mathfrak{A} \oplus \mathfrak{B}) = t[\mathrm{ltp}(\mathfrak{A}), \mathrm{ltp}(\mathfrak{B})]\big(\mathcal{L}(\mathfrak{A}), \mathcal{L}(\mathfrak{B})\big)\,.$$

(Note that the local type of a tuple $\bar{a}$ determines the type of any permutation of $\bar{a}$. Therefore, we only need Lemma 83 for tuples $\bar{a}\bar{b}$ with $\bar{a} \subseteq A$ and $\bar{b} \subseteq B$, not for arbitrary interleavings of elements of A and B.)

From Lemmas 81 and 83 we can deduce that the local 0-type of a structure is $\mathcal{QF}_*$-computable (cf. Definition 6). Consequently, the mapping $\mathcal{L}$ is a finite-state derived operation based on ltp. □

In the second step we extend $\mathcal{QF}_*$ by all fusion operations which are defined as follows. Recall the definition of quotient structures at the end of Section 3.1.

Definition 85. Let $\mathfrak{A}$ be a structure and $\varphi(x)$ a quantifier-free formula. We set $\mathrm{fuse}_\varphi(\mathfrak{A}) := \mathfrak{A}/{\sim}$ where $\sim$ is the equivalence relation

$$a \sim b \quad \text{iff} \quad a = b \quad \text{or} \quad \mathfrak{A} \models \varphi(a) \wedge \varphi(b)\,.$$

By $\mathcal{F}use$ we denote the signature consisting of all operations of the form fuse_φ.

We have seen that every operation of $\mathcal{QF}_*$ satisfies (H). To do the same for $\mathcal{F}use$ it therefore remains to prove (H) for fusion operations.

Lemma 86. *Let $\varphi(x)$ be a quantifier-free formula and $g : \mathfrak{A} \to \mathrm{fuse}_\varphi(\mathfrak{A})$ the canonical mapping. There exist functions $f_n : S^n_\mathrm{L}(\Sigma) \to S^n_\mathrm{L}(\Sigma)$, for $n \in \mathbb{N}$, such that*

$$\mathrm{ltp}\big(g(\bar{a})/\mathrm{fuse}_\varphi(\mathfrak{A})\big) = f_n(\mathrm{ltp}(\bar{a}/\mathfrak{A}))\,, \quad \textit{for all } \bar{a} \in A^n.$$

Proof. Let $p_1, \dots, p_s \in S^{1,0}_\mathrm{M}(\Sigma)$ be an enumeration of all quantifier-free 1-types p with $p \models \varphi$ that are realized in $\mathfrak{A}$. Let $q \in S^{1,0}_\mathrm{M}(\Sigma)$ be the quantifier-free 1-type with

$$Rx_1 \dots x_1 \in q \quad \text{iff} \quad Rx_1 \dots x_1 \in p_i\,, \text{ for some } i \leqslant s\,.$$

If $b \in A$ is some element of type $\mathrm{tp}_0(b/\mathfrak{A}) = p_i$ then $g(b)$ has the type

$$\mathrm{tp}_0(g(b)/\mathrm{fuse}_\varphi(\mathfrak{A})) = q\,.$$

To simplify notation we define a function $f : S^{1,0}_\mathrm{M}(\Sigma) \to S^{1,0}_\mathrm{M}(\Sigma)$ by

$$f(r) := \begin{cases} q & \text{if } r \in \{p_1, \dots, p_s\}\,, \\ r & \text{otherwise}\,. \end{cases}$$

For Hintikka-formulas ψ_r we set $f(\psi_r) := \psi_{f(r)}$, and for formulas ψ of the form $y_i = x_k$ we set $f(\psi) := \psi$.

For m.e. formulas of the form $\vartheta = \exists y\psi(y)$ we have

$$\exists y\psi(y) \in \mathrm{ltp}(g(\bar{a})/\mathrm{fuse}_\varphi(\mathfrak{A})) \text{ iff } \exists y\psi'(y) \in \mathrm{ltp}(\bar{a}/\mathfrak{A}) \quad \text{for some } \psi' \in f^{-1}(\psi)\,.$$

Let $\vartheta(x_1, \dots, x_n) = \exists y_1 \cdots \exists y_m (R y_1 \dots y_m \wedge \psi_1 \wedge \dots \wedge \psi_m)$ be a m.e. formula. We have

$$\vartheta \in \mathrm{ltp}(g(\bar{a})/\mathrm{fuse}_\varphi(\mathfrak{A}))$$

if and only if

$$\exists y_1 \cdots \exists y_m (R y_1 \dots y_m \wedge \psi'_1 \wedge \dots \wedge \psi'_m) \in \mathrm{ltp}(\bar{a}/\mathfrak{A})\,,$$

for some formulas $\psi'_i \in f^{-1}(\psi_i)$, $1 \leqslant i \leqslant m$. Since the types $p_1, \dots, p_r$ can be determined from $\mathrm{ltp}(\bar{a}/\mathfrak{A})$ this gives the desired definition of f_n. □

Corollary 87. *The signature $\mathcal{QF}_* \cup \mathcal{F}use$ satisfies condition* (H).

Proof. For the operations of $\mathcal{QF}_*$, we have already shown in Corollary 84 that $\mathcal{L}$ is a finite-state heteromorphism based on ltp. It remains to consider the operations $\mathrm{fuse}_\varphi \in \mathcal{F}use$. The preceding lemma implies that there exists a $\mathcal{QF}$-term t such that

$$\mathcal{L}(\mathrm{fuse}_\varphi(\mathfrak{A})) = t\big(\mathcal{L}(\mathfrak{A})\big)\,.$$

Together Lemmas 81, 83, and 86 show that the local 0-type of a structure is $(\mathcal{QF}_* \cup \mathcal{F}use)$-computable. Hence, the claim follows. □

By the results of the previous section, we immediately obtain the following theorem which is one of our main results.

Theorem 88. *The signatures $\mathcal{QF}$ and $\mathcal{QF}_* \cup \mathcal{F}use$ are equivalent.*

Let us compare this result with those of Courcelle and Makowsky [9] who show that the signature $\mathcal{F}$ consisting of the disjoint union $\oplus$, of certain restricted quantifier-free operations, and of the operations fuse_{P_X} satisfies the following properties. For every finite subsignature $\mathcal{F}_0 \subseteq \mathcal{F}$,

(1) the value mapping $\mathrm{val}_{\mathrm{STR}} : T(\mathcal{F}_0)_\Sigma \to \mathrm{STR}[\Sigma]$ is an MSO-transduction,
(2) every $\mathcal{F}_0$-equational set is $\mathcal{QF}$-equational, and
(3) each MSO-definable set of (hyper-)graphs contained in $\mathrm{val}_{\mathrm{STR}}(T(\mathcal{F}_0)_\Sigma)$ is $\mathcal{F}_0$-recognizable.

The restrictions imposed in [9] on quantifier-free operations and relational structures are the following ones:

- the sets $P_\mathfrak{A}$ form a partition of A,
- the only quantifier-free operations allowed to modify the vertex labellings are operations of the form $\mathrm{ren}_{P \to Q}$ as described in Section 6, and

• no quantifier-free operation restricts the domain of its argument.

In the present section we were able to remove the first and third restriction by using 1-types instead of vertex labels. Furthermore, we have shown that both signatures lead to the same notion of recognizability. Unfortunately, to do so we had to modify the second restriction by only allowing the quantifier-free operations of $\mathcal{QF}_*$. By the results of [9] and Theorem 88 we have

$$\begin{aligned}\mathrm{Equat}(\mathcal{QF}) &= \mathrm{Equat}(\mathcal{QF}_* \cup \mathcal{F}\mathit{use}) = \mathrm{Equat}(\mathcal{QF} \cup \mathcal{F}\mathit{use}),\\ \text{and } \mathrm{Rec}(\mathcal{QF}) &= \mathrm{Rec}(\mathcal{QF}_* \cup \mathcal{F}\mathit{use}) \supseteq \mathrm{Rec}(\mathcal{QF} \cup \mathcal{F}\mathit{use}).\end{aligned}$$

We currently do not know whether the last inclusion can be strengthened to an equality.

7.3. Fusion and complete local types for graphs

For graphs—or more generally for structures of arity at most 2—we can improve the above result by showing that the signatures $\mathcal{QF}$ and $\mathcal{QF} \cup \mathcal{F}\mathit{use}$ are equivalent. One would expect that this holds for arbitrary arities, but so far we have neither been able to prove such a statement, nor could we construct a counterexample. For the remainder of this section, we fix a signature Σ of arity $\mathrm{ar}(\Sigma) \leqslant 2$.

The reason why the above proof works only for $\mathcal{QF}_*$ is the fact that, if we use the labelling $\mathcal{L}$ then arbitrary quantifier-free operations do not satisfy condition (H). For arity 2, we are able to modify the notion of a local type such that all $\mathcal{QF}$-operations satisfy (H). The basic idea is to replace in an m.e. formula $\exists\bar{y}(R\bar{y} \wedge \psi_1 \wedge \cdots \wedge \psi_m)$ the atom $R\bar{y}$ by the Hintikka-formula of a quantifier-free 2-type. Though, to simplify notation we will not use such formulas but the quantifier-free 2-types themselves.

Definition 89. Let $\mathfrak{A}$ be a structure and $a, b \in A$. The *complete local* 2*-type* of a pair ab in $\mathfrak{A}$ is its quantifier-free type

$$\mathrm{ctp}(ab/\mathfrak{A}) := \mathrm{tp}_0(ab/\mathfrak{A}).$$

The *complete local* 1*-type* of a single element a in $\mathfrak{A}$ is the set of all complete local 2-types of pairs extending a

$$\mathrm{ctp}(a/\mathfrak{A}) := \{\, \mathrm{ctp}(ac/\mathfrak{A}) \mid c \in A \,\}.$$

Finally, we will also need the *complete local* 0*-type* of the empty tuple $\langle\rangle$ which is the set of all realized 1-types.

$$\mathrm{ctp}(\langle\rangle/\mathfrak{A}) := \{\, \mathrm{ctp}(a/\mathfrak{A}) \mid a \in A \,\}.$$

As usual, we abbreviate $\mathrm{ctp}(\langle\rangle/\mathfrak{A})$ by $\mathrm{ctp}(\mathfrak{A})$. For $0 \leqslant n \leqslant 2$, we denote by $S^n_{\mathrm{C}}(\Sigma)$ the set of all possible complete local n-types and we set $S^*_{\mathrm{C}}(\Sigma) := S^1_{\mathrm{C}}(\Sigma) \cup S^2_{\mathrm{C}}(\Sigma)$.

Remark 90. Since satisfiability is decidable for the 2-variable fragment of first-order logic it follows that the sets $S^0_{\mathrm{C}}(\Sigma)$, $S^1_{\mathrm{C}}(\Sigma)$, and $S^2_{\mathrm{C}}(\Sigma)$ are decidable.

As in the case of the other types one can define Hintikka formulas for complete local types.

Lemma 91. *For every complete local n-type $p \in S^n_{\mathrm{C}}(\Sigma)$, $0 \leqslant n \leqslant 2$, there exists a first-order formula $\psi_p(\bar{x})$ of quantifier height $2 - n$ such that*

$$\mathfrak{A} \models \psi_p(\bar{a}) \quad \text{iff} \quad \mathrm{ctp}(\bar{a}/\mathfrak{A}) = p\,,$$

for all structures $\mathfrak{A}$ and every tuple $\bar{a} \in A^n$.

Proof. We define ψ_p by reverse induction on n. The construction is analogous to that of Definition 33. For $n = 2$, we define

$$\psi_p(x_1, x_2) := \bigwedge p\,.$$

For $n = 1$, we have to express the back-and-forth property (cf. [26,19]). The formula

$$\psi_p(x_1) := \bigwedge_{q \in p} \exists x_2 \psi_q(x_1, x_2) \wedge \forall x_2 \bigvee_{q \in p} \psi_q(x_1, x_2)$$

states that every type $q \in p$ is realized and every realized type is contained in p. Similarly, for $n = 0$, we have

$$\psi_p := \bigwedge_{q \in p} \exists x_1 \psi_q(x_1) \wedge \forall x_1 \bigvee_{q \in p} \psi_q(x_1)\,. \qquad \square$$

Corollary 92. *The* 0*-type* $\mathrm{ctp}(\mathfrak{A})$ *is* $\mathcal{QF}$*-computable.*

Proof. As in Lemma 45, we can prove that first-order types of bounded quantifier depth are $\mathcal{QF}$-computable. Since $\mathrm{ctp}(\mathfrak{A})$ is logically equivalent to the first-order 0-type of quantifier depth 2 the claim follows. $\square$

We use Hintikka formulas to define the logical consequences of a local type.

Definition 93. For $p \in S^n_{\mathrm{C}}(\Sigma)$ and $\varphi \in \mathrm{FO}[\Sigma]$, we write $p \models \varphi$ iff $\models \psi_p \rightarrow \varphi$.

Remark 94. It follows that $p \models \varphi$ if and only if we have $\mathfrak{A} \models \varphi(\bar{a})$, for every structure $\mathfrak{A}$ and all tuples $\bar{a} \subseteq A$ of type $\mathrm{ctp}(\bar{a}/\mathfrak{A}) = p$.

Following the usual lines of our approach we annotate structures by types and we show that these annotations satisfy conditions (M) and (H).

Definition 95. Let $\mathfrak{A}$ be a Σ-structure with $\mathrm{ar}(\Sigma) \leqslant 2$. The *complete local annotation* of $\mathfrak{A}$ is the structure

$$\mathcal{C}(\mathfrak{A}) := \left\langle A, (T_p)_{p \in S^*_{\mathrm{C}}(\Sigma)} \right\rangle$$

with the same domain as $\mathfrak{A}$ where, for each local n-type $p \in S^*_{\mathrm{C}}(\Sigma)$, $n \in \{1, 2\}$, we add the relation

$$T_p := \{\, \bar{a} \in A^n \mid \mathrm{ctp}(\bar{a}/\mathfrak{A}) = p \,\}\,.$$

We denote the signature of $\mathcal{C}(\mathfrak{A})$ by Σ_{C}.

Lemma 96. *Let Σ be a relational signature.*

(*a*) *The mapping $\mathscr{C} : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Sigma_{\mathrm{C}}]$ is an injective* FO*-transduction of quantifier height* 1.
(*b*) *$\mathscr{C}$ has a left-inverse that is a quantifier-free transduction.*
(*c*) *$\mathscr{C}$ satisfies condition* (M).

Proof.

(a) The formula $\psi_p(\bar{x})$ from Lemma 91 can be used to define the relation T_p. For $p \in S^n_{\mathrm{C}}(\Sigma)$, this formula has quantifier height $\mathrm{qh}(\psi_p) = 2 - n$.
(b) Conversely, we can write an n-ary relation $R \in \Sigma$ as

$$R_{\mathfrak{A}} = \{\, \bar{a} \in A^n \mid \bar{a} \in T_p \text{ for some } p \text{ with } p \models Rx_1 \ldots x_n \,\}.$$

Since $S^*_{\mathrm{C}}(\Sigma)$ is finite this definition is equivalent to a finite disjunction of atomic formulas.
(c) Finally, by composing the transductions of (a) and (b) we can construct an FO-formula that defines the set $\mathcal{C}(\mathrm{STR}[\Sigma])$. $\square$

It remains to check condition (H). We start by considering the operations of $\mathcal{QF}$.

Lemma 97. *Let $\tau : \mathrm{STR}[\Sigma] \to \mathrm{STR}[\Gamma]$ be a quantifier-free operation with $\mathrm{ar}(\Gamma) \leqslant 2$. There exist functions $f_n : S^n_{\mathrm{C}}(\Sigma) \to S^n_{\mathrm{C}}(\Gamma)$, $0 \leqslant n \leqslant 2$, such that*

$$\mathrm{ctp}(\bar{a}/\tau(\mathfrak{A})) = f_n(\mathrm{ctp}(\bar{a}/\mathfrak{A})),$$

for every structure $\mathfrak{A}$ and every tuple $\bar{a}$ in $\tau(\mathfrak{A})$.

Proof. We decompose $\tau = \sigma \circ \mathrm{del}_\varphi$ into a domain restriction and a nondeleting quantifier-free operation (cf. Lemma 25), and we deal with the two cases separately. For $\tau = \mathrm{del}_\varphi$ and $a, b \in \mathrm{del}_\varphi(A)$, we have:

$$\begin{aligned}
&\mathrm{ctp}(ab/\mathrm{del}_\varphi(\mathfrak{A})) = \mathrm{ctp}(ab/\mathfrak{A}),\\
&\mathrm{ctp}(a/\mathrm{del}_\varphi(\mathfrak{A})) = \big\{\, p \in \mathrm{ctp}(a/\mathfrak{A}) \bigm| p \models \neg\varphi(x_2) \,\big\},\\
&\mathrm{ctp}(\langle\rangle/\mathrm{del}_\varphi(\mathfrak{A})) = \big\{\, f_1(p) \bigm| p \in \mathrm{ctp}(\langle\rangle/\mathfrak{A}),\ p \models \neg\varphi(x_1) \,\big\},
\end{aligned}$$

where f_1 in the last line is the function given by the second equation.
It remains to consider the case that $\tau = \sigma$. By Lemma 35, there exists a function g such that

$$\mathrm{tp}_0(ab/\sigma(\mathfrak{A})) = g(\mathrm{tp}_0(ab/\mathfrak{A})).$$

Hence, we can set $f_2 := g$. The functions f_1 and f_0 are defined by

$$\begin{aligned}
&\mathrm{ctp}(a/\sigma(\mathfrak{A})) = \{\, g(p) \mid p \in \mathrm{ctp}(a/\mathfrak{A}) \,\},\\
&\mathrm{ctp}(\langle\rangle/\sigma(\mathfrak{A})) = \{\, f_1(p) \mid p \in \mathrm{ctp}(\langle\rangle/\mathfrak{A}) \,\}. \qquad \square
\end{aligned}$$

We are interested in the fusion operation. It turns out that the annotation $\mathcal{C}$ can be used to treat an even stronger operation which we call the *gluing* of two structures.

Definition 98. A *gluing function* is a mapping

$$g : S^1_{\mathrm{C}}(\Sigma) \times S^1_{\mathrm{C}}(\Sigma) \to S^2_{\mathrm{C}}(\Sigma)\,,$$

such that, for all types $p, q \in S^1_{\mathrm{C}}(\Sigma)$ and every quantifier-free formula $\varphi(x)$ with one free variable, we have

$$\begin{aligned} \varphi(x_1) \in g(p,q) \quad &\text{iff} \quad p \models \varphi(x_1)\,, \\ \text{and } \varphi(x_2) \in g(p,q) \quad &\text{iff} \quad q \models \varphi(x_1)\,. \end{aligned}$$

For such a gluing function g and structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{STR}[\Sigma]$, we denote by $\mathfrak{A} \otimes_g \mathfrak{B}$ the following structure. Its domain is the disjoint union $A \cupdot B$. For unary relations P, we have

$$P_{\mathfrak{A}\otimes_g\mathfrak{B}} := P_{\mathfrak{A}} \cup P_{\mathfrak{B}}\,,$$

while binary relations R are defined by

$$\begin{aligned} R_{\mathfrak{A}\otimes_g\mathfrak{B}} := {} & R_{\mathfrak{A}} \cup R_{\mathfrak{B}} \\ & \cup \big\{ (a,b) \in A \times B \;\big|\; g\big(\mathrm{ctp}(a/\mathfrak{A}),\ \mathrm{ctp}(b/\mathfrak{B})\big) \models Rx_1x_2 \big\} \\ & \cup \big\{ (b,a) \in B \times A \;\big|\; g\big(\mathrm{ctp}(a/\mathfrak{A}),\ \mathrm{ctp}(b/\mathfrak{B})\big) \models Rx_2x_1 \big\}\,. \end{aligned}$$

Finally, we extend $\otimes_g$ to an operation $\mathrm{STR}[\Sigma] \times \mathrm{STR}[\Gamma] \to \mathrm{STR}[\Sigma \cup \Gamma]$ on structures of different signatures by defining $\mathfrak{A} \otimes_g \mathfrak{B} := \mathfrak{A}' \otimes_g \mathfrak{B}'$ where $\mathfrak{A}'$ is the $(\Sigma \cup \Gamma)$-structure obtained from $\mathfrak{A}$ by adding empty relations $R_{\mathfrak{A}'} := \emptyset$, for every $R \in \Gamma \setminus \Sigma$, and $\mathfrak{B}'$ is defined analogously.

By $\mathcal{G}lue$ we denote the signature consisting of all operations of the form $\otimes_g$.

Remark 99.

(a) Note that $\mathfrak{A} \otimes_g \mathfrak{B} = \mathfrak{A} \oplus \mathfrak{B}$ if we have $\neg Rx_1x_2, \neg Rx_2x_1 \in g(p,q)$, for all $p, q \in S^1_{\mathrm{C}}(\Sigma)$ and all binary relation symbols R.

(b) The conditions on a gluing function g ensure that

$$\mathrm{ctp}(ab/\mathfrak{A} \otimes_g \mathfrak{B}) = g\big(\mathrm{ctp}(a/\mathfrak{A}), \mathrm{ctp}(b/\mathfrak{B})\big)\,,$$

for all structures $\mathfrak{A}$ and $\mathfrak{B}$ and all elements $a \in A$ and $b \in B$. For instance, we have

$$\begin{aligned} Rx_1x_2 \in \mathrm{ctp}(ab/\mathfrak{A} \otimes_g \mathfrak{B}) \text{ iff } & (a,b) \in R_{\mathfrak{A}\otimes_g\mathfrak{B}} \\ \text{iff } & g(\mathrm{ctp}(a/\mathfrak{A}), \mathrm{ctp}(b/\mathfrak{B})) \models Rx_1x_2\,, \\ \text{and } Px_1 \in \mathrm{ctp}(ab/\mathfrak{A} \otimes_g \mathfrak{B}) \text{ iff } & a \in P_{\mathfrak{A}} \\ \text{iff } & \mathrm{ctp}(a/\mathfrak{A}) \models Px_1 \\ \text{iff } & g(\mathrm{ctp}(a/\mathfrak{A}), \mathrm{ctp}(b/\mathfrak{B})) \models Px_1\,. \end{aligned}$$

Example 100. Cunningham [18] studies graph decompositions, called *split decomposition,* that are based on the following operation (see also [14]). Given two undirected, simple, loop-free graphs $\mathfrak{G}$ and $\mathfrak{H}$ in $\mathrm{STR}[\{edg\} \cup \Pi]$ with labelled vertices as in Section 3.5 and some relations $P \in \Pi$, one forms the graph

$$\mathfrak{G} \diamond_P \mathfrak{H} := \mathrm{del}_{Px}(\mathfrak{G} \otimes_g \mathfrak{H}),$$

where del_{Px} deletes all vertices labelled P and g is the gluing function such that

$$g(p_1, p_2) \models edg(x_1, x_2) \quad \text{iff} \quad p_i \models \exists y(edg(x_1, y) \wedge Py) \text{ for both } i\,,$$

that is, $\otimes_g$ creates an edge (a, b) between a vertex a of $\mathfrak{G}$ and a vertex b of $\mathfrak{H}$ if and only if both a and b have a neighbour labelled P. Actually, in [18] this operation is used only on graphs where P contains a unique vertex.

The next lemma is analogous to Corollary 44 and Lemma 83.

Lemma 101. *Let g be a gluing function. There exist functions f_n, $0 \leqslant n \leqslant 2$, such that*

$$\mathrm{ctp}(\bar{a}/\mathfrak{A} \otimes_g \mathfrak{B}) = f_n\big(\mathrm{ctp}(\bar{a}|_A \;/\; \mathfrak{A}),\ \mathrm{ctp}(\bar{a}|_B \;/\; \mathfrak{B})\big)\,,$$

for all structures $\mathfrak{A}$ and $\mathfrak{B}$ and all tuples $\bar{a} \in (A \cup B)^n$, where $\bar{a}|_X$ denotes the subtuple of $\bar{a}$ consisting of all elements $a_i \in X$.

Proof. We start with the case $n = 2$. If $a, b \in A$ then

$$\mathrm{ctp}(ab/\mathfrak{A} \otimes_g \mathfrak{B}) = \mathrm{ctp}(ab/\mathfrak{A})\,.$$

The case that $a, b \in B$ is similar. If $a \in A$ and $b \in B$ then:

$$\begin{aligned} \mathrm{ctp}(ab/\mathfrak{A} \otimes_g \mathfrak{B}) &= g\big(\mathrm{ctp}(a/\mathfrak{A}), \mathrm{ctp}(b/\mathfrak{B})\big) \\ \text{and } \mathrm{ctp}(ba/\mathfrak{A} \otimes_g \mathfrak{B}) &= \sigma g\big(\mathrm{ctp}(a/\mathfrak{A}), \mathrm{ctp}(b/\mathfrak{B})\big)\,, \end{aligned}$$

where $\sigma(p)$ interchanges the variables x_1 and x_2 in every formula of p. (We have proved the first equation in the remark above. The second one follows from the fact that $\mathrm{ctp}(ba/\mathfrak{A} \otimes_g \mathfrak{B}) = \sigma(\mathrm{ctp}(ab/\mathfrak{A} \otimes_g \mathfrak{B}))$.)

For $a \in A$, we have

$$\mathrm{ctp}(a/\mathfrak{A} \otimes_g \mathfrak{B}) = \mathrm{ctp}(a/\mathfrak{A}) \cup \big\{\, g(\mathrm{ctp}(a/\mathfrak{A}), p) \;\big|\; p \in \mathrm{ctp}(\mathfrak{B}) \,\big\}\,,$$

and, for $b \in B$,

$$\mathrm{ctp}(b/\mathfrak{A} \otimes_g \mathfrak{B}) = \mathrm{ctp}(b/\mathfrak{B}) \cup \big\{\, \sigma g(p, \mathrm{ctp}(b/\mathfrak{B})) \;\big|\; p \in \mathrm{ctp}(\mathfrak{A}) \,\big\}\,.$$

Finally, for $n = 0$, we have

$$\mathrm{ctp}(\mathfrak{A} \otimes_g \mathfrak{B}) = \big\{\, f_1(p, \mathrm{ctp}(\mathfrak{B})) \;\big|\; p \in \mathrm{ctp}(\mathfrak{A}) \,\big\} \cup \big\{\, f_1(\mathrm{ctp}(\mathfrak{A}), p) \;\big|\; p \in \mathrm{ctp}(\mathfrak{B}) \,\big\}\,. \qquad \square$$

Corollary 102. *If we only consider structures of arity at most* 2 *then the signature* $\mathcal{QF} \cup \mathcal{G}lue$ *satisfies condition* (H).

Proof. We claim that the function $\mathcal{C}$ is a finite-state heteromorphism based on ctp. For quantifier-free operations and the gluing operation $\otimes_g$ this follows from the preceding lemmas. For the disjoint union $\oplus$, it is sufficient to note that $\oplus = \otimes_g$, for a suitable gluing function g (cf. the Remark after Definition 98).

It remains to show that ctp is $(\mathcal{QF} \cup \mathcal{G}lue)$-computable. We have already seen that it is $\mathcal{QF}$-computable in Corollary 92. Hence, Lemma 101 implies that ctp is $(\mathcal{QF} \cup \mathcal{G}lue)$-computable. $\square$

By the results of Section 7.1, it follows that, for structures of arity at most 2, the signature $\mathcal{QF} \cup \mathcal{G}lue$ is equivalent to $\mathcal{QF}$, i.e., the corresponding subalgebras of STR are equivalent.

Corollary 103. *For structures of arity at most* 2, *the signatures* $\mathcal{QF} \cup \mathcal{G}lue$ *and* $\mathcal{QF}$ *are equivalent.*

The signature we are actually interested in is $\mathcal{QF} \cup \mathcal{G}lue \cup \mathcal{F}use$. The following theorem, which is one of our main results, states that it is equivalent to $\mathcal{QF}$.

Theorem 104. *For structures of arity at most* 2, *the signatures* $\mathcal{QF} \cup \mathcal{G}lue \cup \mathcal{F}use$ *and* $\mathcal{QF}$ *are equivalent.*

Proof. By Corollary 103 and Lemma 17, it is sufficient to show that

$$\mathcal{QF} \cup \mathcal{G}lue \cup \mathcal{F}use \subseteq (\mathcal{QF} \cup \mathcal{G}lue)^{\mathrm{der}}.$$

We can express the operation fuse_φ as a derived $(\mathcal{QF} \cup \mathcal{G}lue)$-operation as follows. We add a new element c satisfying $\varphi(x)$ to the given structure by a suitable gluing operation that creates an R-edge from an element a to c iff there exists an R-edge (a, b) ending in an element b satisfying $\varphi(x)$. Then we delete all elements satisfying $\varphi(x)$ except for c. Formally, we have

$$\mathrm{fuse}_\varphi(x) = (\mathrm{fgt}_P \circ \mathrm{del}_\vartheta)(x \otimes_g \sigma(c))$$

where:

- c is a constant denoting a singleton structure whose only element b satisfies φ,
- σ creates a new unary relation $P \notin \Sigma$ and it adds all elements to it,
- g creates an R-edge between an element a and $\sigma(c)$ iff there is some element b satisfying φ such that $(a, b) \in R$. That is,

$$\begin{aligned} g(p, q) := &\{ Rx_1x_2 \mid p \models \exists y(Rx_1y \wedge \psi(y)) \text{ for some } \psi \in \Psi \} \\ \cup\ &\{ Rx_2x_1 \mid p \models \exists y(Ryx_1 \wedge \psi(y)) \text{ for some } \psi \in \Psi \} \\ \cup\ &\{ \psi(x_1) \mid p \models \psi(x_1), \psi \text{ quantifier free} \} \\ \cup\ &\{ \psi(x_2) \mid q \models \psi(x_1), \psi \text{ quantifier free} \}, \end{aligned}$$

where q is the complete local 1-type of the single element of the structure $\sigma(c)$ and Ψ is the set of all Hintikka-formulas ψ_r, $r \in S^1_{\mathrm{C}}(\Sigma)$, with $r \models \varphi$,

- $\vartheta := \varphi \wedge \neg Px_1$, i.e., del_ϑ deletes all elements satisfying φ except for the new one which is labelled by P, and
- fgt_P deletes the auxiliary relation P again. □

8. Sources in hypergraphs are not necessary

Equipping graphs and hypergraphs with distinguished vertices is useful for defining operations like series composition or parallel composition that generalize concatenation. These distinguished vertices are called *sources.* In terms of relational structures such distinguished elements can be defined as values of nullary symbols which are also called *constants.* They have been defined in this way in the general logical and algebraic framework of [7] which is further developed in [8]. Constants can be eliminated if one replaces them by unary relations containing single elements. However, the quantifier height of the definition scheme of a given transduction usually increases under this transformation. Take for example the quantifier-free definition

$$Rxy \quad \text{iff} \quad Sxa \wedge Tyb,$$

where a and b are constants. If we encode a and b by unary relations P_a and P_b, this definition becomes

$$Rxy \quad \text{iff} \quad \exists u \exists v (Sxu \wedge Tyv \wedge P_a u \wedge P_b v)\,,$$

which is no longer quantifier-free. Hence, after the transformation the signature $\mathcal{QF}$ may contain fewer operations. In this section, we show that quantifier-free operations using constants can be emulated by quantifier-free operations on relational structures without them. We will prove that the signature of quantifier-free operations using constants, denoted by $\mathcal{QF}^{\mathrm{c}}$, is "equivalent" to the signature $\mathcal{QF}$ on relational structures without constants (for the precise meaning of "equivalent" cf. Proposition 105 and Theorem 112).

8.1. Relational structures with constants

We recall definitions from [7,8]. We fix a countable set C_∞ of *constant symbols.* For a relational signature Σ and a finite subset $C \subseteq C_\infty$, we denote by $\mathrm{STR}[\Sigma, C]$ the set of all finite structures of the form

$$\mathfrak{A} = \big\langle A, (R_\mathfrak{A})_{R\in\Sigma}, (c_\mathfrak{A})_{c\in C}\big\rangle,$$

where $\langle A, (R_\mathfrak{A})_{R\in\Sigma}\rangle \in \mathrm{STR}[\Sigma]$ and $c_\mathfrak{A} \in A$, for every $c \in C$.

By $\mathfrak{A}[C]$ we denote the substructure of $\mathfrak{A}$ induced by the set of all elements that are denoted by some constant $c \in C$.

We call quantifier-free transductions between structures with constants QF^{c}*-transductions,* for short (the superscript $^{\mathrm{c}}$ indicates that we allow constants). A definition scheme for such a transduction $\mathrm{STR}[\Sigma, C] \to \mathrm{STR}[\Gamma, D]$ is of the form

$$\mathcal{D} = \big(\varphi,\ \psi,\ (\vartheta_R)_{R\in\Gamma},\ (\kappa_{cd})_{c\in C, d\in D}\big),$$

where:

- $\varphi = \text{true}$ (cf. Section 3.3),
- $\psi \in \text{QF}[\Sigma \cup C, \{x_1\}]$,
- $\vartheta_R \in \text{QF}[\Sigma \cup C, \{x_1, \dots, x_{\text{ar}(R)}\}]$, for $R \in \Gamma$, and
- $\kappa_{cd} \in \text{QF}[\Sigma \cup C, \emptyset]$, for each $c \in C$ and $d \in D$.

As usual, the formula ψ defines the domain of the new structure and the formulas ϑ_R define the new relations R. The new constants are determined by the formulas κ_{cd}. Given a structure $\mathfrak{A}$ we define the constant d in the new structure to denote that element $c_{\mathfrak{A}}$ such that κ_{cd} holds in $\mathfrak{A}$.

In order that a definition scheme defines a total mapping, the formulas κ_{cd} must satisfy the following conditions, for every structure in $\mathfrak{A} \in \text{STR}[\Sigma, C]$ and every $d \in D$:

- d denotes an element of the new structure, that is, $\mathfrak{A} \models \bigwedge_{c \in C}(\kappa_{cd} \to \psi(c))$.
- d has some value, that is, $\mathfrak{A} \models \bigvee_{c \in C} \kappa_{cd}$.
- d is unique, that is, $\mathfrak{A} \models \bigwedge_{c,c' \in C}(\kappa_{cd} \wedge \kappa_{c'd} \to c = c')$.

These conditions are given by quantifier-free formulas without free variables. Hence, they hold in a structure $\mathfrak{A} \in \text{STR}[\Sigma, C]$ iff they hold in $\mathfrak{A}[C]$. It is therefore decidable whether they are valid in every structure because we only need to check their validity in the finitely many structures of the form $\mathfrak{A}[C]$.

A definition scheme $\mathcal{D}$ as above defines a total mapping $\hat{\mathcal{D}} : \text{STR}[\Sigma, C] \to \text{STR}[\Gamma, D]$ where the domain and the relations of $\mathfrak{B} := \hat{\mathcal{D}}(\mathfrak{A})$ are defined in the same way as for structures without constants and, additionally, we have $d_{\mathfrak{B}} = c_{\mathfrak{A}}$ whenever $\mathfrak{A} \models \kappa_{cd}$.

We obtain thus an algebra STR^{c} of structures with constants where each pair (Σ, C) is a sort. The operations are the QF^{c}-transductions and the disjoint union $\oplus$ which we apply only to structures with disjoint sets of constants. (For structures $\mathfrak{A} \in \text{STR}[\Sigma, C]$ and $\mathfrak{B} \in \text{STR}[\Gamma, D]$ with $C \cap D = \emptyset$, the structure $\mathfrak{A} \oplus \mathfrak{B} \in \text{STR}[\Sigma \cup \Gamma, C \cup D]$ is well-defined). We denote by $\mathcal{QF}^{\text{c}}$ the corresponding signature.

We could define MSO-transductions between structures with constants in the same way as QF^{c}-transductions. But when we allow quantifiers then the formulas κ_{cd} are not needed. Therefore, we choose a simpler approach by reducing such transductions to MSO-transductions without constants.

Let $\Pi_C := \{ P_c \mid c \in C \}$ be a set of unary relations in bijection with C and disjoint from Σ. For $\mathfrak{A} \in \text{STR}[\Sigma, C]$, we denote by $\mathfrak{A}_\Pi \in \text{STR}[\Sigma \cup \Pi_C]$ the structure with the same domain as $\mathfrak{A}$ and the same Σ-relations. For every constant $c \in C$, we add a new unary relation $P_c := \{c_{\mathfrak{A}}\}$ to $\mathfrak{A}_\Pi$. Clearly, the mapping

$$\text{STR}[\Sigma, C] \to \text{STR}[\Sigma \cup \Pi_C] : \mathfrak{A} \mapsto \mathfrak{A}_\Pi$$

is an injective QF^{c}-transduction. (We identify $\text{STR}[\Sigma \cup \Pi_C, \emptyset]$ and $\text{STR}[\Sigma \cup \Pi_C]$.)

We define an MSO-*transduction* (of structures with constants) as a transduction $\tau : \text{STR}[\Sigma, C] \to \text{STR}[\Gamma, D]$ such that the relation $\{ (\mathfrak{A}_\Pi, \mathfrak{B}_\Pi) \mid \mathfrak{B} \in \tau(\mathfrak{A}) \}$ is an MSO-transduction. Routine arguments show that the composition of two MSO-transductions is an MSO-transduction, also when they use constants.

We now recall from [7] the following result, formulated with the terminology of the present article. It is the analogue of Proposition 27 for structures with constants.

Proposition 105. *Let $L \subseteq \mathrm{STR}[\Sigma, C]$. The following statements are equivalent*:

(*i*) *L is the image of a regular set of terms under an* MSO-*transduction.*
(*ii*) *L is $\mathcal{QF}^{\mathrm{c}}$-equational.*
(*iii*) *The set $L_\sqcap := \{\, \mathfrak{A}_\sqcap \mid \mathfrak{A} \in L \,\}$ is $\mathcal{QF}$-equational.*

Proof. The equivalence (i) $\Leftrightarrow$ (ii) is proved in [7]. Let us sketch the equivalence of (i) and (iii). With routine manipulations of MSO-transductions one can show that (i) is equivalent to the statement

$L_\sqcap$ is the image of a regular set of terms under an MSO-transduction.

Hence, the equivalence (i) $\Leftrightarrow$ (iii) follows from Proposition 27. $\square$

Our objective is to obtain a similar characterization of $\mathcal{QF}^{\mathrm{c}}$-recognizability of $L \subseteq \mathrm{STR}[\Sigma, C]$ in terms of the $\mathcal{QF}$-recognizability of $L_\sqcap$. Theorem 112 below achieves this goal. Following our general framework we will introduce a construction on structures that makes it possible to emulate the operations of $\mathcal{QF}^{\mathrm{c}}$ in terms of $\mathcal{QF}$-operations.

8.2. A second way of eliminating constants

The basic idea is to replace a structure $\mathfrak{A}$ by the structure $\hat{\mathfrak{A}}$ obtained by deleting all elements that are denoted by some constant and by adding new relations that memorize links with the deleted elements. For example, an edge from x (where x is not the value of any constant) to $c_\mathfrak{A}$ will be represented by a new unary relation $edg[*c]$. An essential fact is that $\mathfrak{A}$ can be reconstructed from $\hat{\mathfrak{A}}$ and $\mathfrak{A}[C]$. (Note that, up to isomorphism, there are only finitely many structures $\mathfrak{A}[C]$ for $\mathfrak{A} \in \mathrm{STR}[\Sigma, C]$.)

Definition 106.

(a) For every n-ary relation $R \in \Sigma$ and each word $w \in (C \cup \{*\})^n$, we introduce a new relation symbol $R[w]$ whose arity is the number of symbols $*$ occurring in w. Let $\Sigma(C)$ be the set of these symbols where we identify R with $R[* \ldots *]$, hence $\Sigma(C)$ contains Σ.
(b) For $\mathfrak{A} = \langle A, (R_\mathfrak{A})_{R\in\Sigma}, (c_\mathfrak{A})_{c\in C}\rangle \in \mathrm{STR}[\Sigma, C]$, we define a $\Sigma(C)$-structure $\hat{\mathfrak{A}} := \langle \hat{A}, (R_{\hat{\mathfrak{A}}})_{R\in\Sigma(C)}\rangle$ with domain $\hat{A} := A \setminus \{\, c_\mathfrak{A} \mid c \in C \,\}$ and the following relations. For $w = w_1 * w_2 \ldots w_k * w_{k+1}$ with $w_1, w_2, \ldots, w_{k+1} \in C^*$, we have

$$R[w]_{\hat{\mathfrak{A}}} := \left\{ (a_1, \ldots, a_k) \;\middle|\; \tilde{w}_1 a_1 \tilde{w}_2 \ldots \tilde{w}_k a_k \tilde{w}_{k+1} \in R_\mathfrak{A} \right\},$$

where $\tilde{w}_i$ is the sequence of elements of A denoted by the constants in $w_i \in C^*$.

Note that the substructure of $\mathfrak{A}$ induced by $\hat{A}$ is a substructure of $\hat{\mathfrak{A}}$. The following statements follow immediately from the definitions.

Lemma 107.

(1) *The structure $\mathfrak{A}$ can be reconstructed from $\hat{\mathfrak{A}}$ and $\mathfrak{A}[C]$.*
(2) *The mapping* $^\wedge : \mathrm{STR}[\Sigma, C] \to \mathrm{STR}[\Sigma(C)]$ *is a* QF^{c}-*transduction.*

(3) *For each structure* $\mathfrak{C} \in \mathrm{STR}[\Sigma, C]$ *with* $\mathfrak{C} = \mathfrak{C}[C]$, *there exists a* $(|C| + 1)$*-copying* MSO*-transduction of quantifier height* 0 *that maps every nonempty structure* $\mathfrak{B} \in \mathrm{STR}[\Sigma(C)]$ *to the unique structure* $\mathfrak{A} \in \mathrm{STR}[\Sigma, C]$ *such that* $\mathfrak{A}[C] = \mathfrak{C}$ *and* $\hat{\mathfrak{A}} = \mathfrak{B}$.

Definition 108. Let $L \subseteq \mathrm{STR}[\Sigma, C]$ and suppose that $\mathfrak{C} \in \mathrm{STR}[\Sigma, C]$ is a structure with $\mathfrak{C} \cong \mathfrak{C}[C]$. We denote by $L \bowtie \mathfrak{C}$ the set of structures $\mathfrak{A} \in L$ such that $\mathfrak{A}[C] \cong \mathfrak{C}$ and $\mathfrak{A} \neq \mathfrak{C}$ (so $\mathfrak{A}$ contains at least one element not denoted by a constant).

Proposition 109. *A set* $L \subseteq \mathrm{STR}[\Sigma, C]$ *is* $\mathcal{QF}^{\mathrm{c}}$*-equational iff* $(L \bowtie \mathfrak{C})^\wedge$ *is* $\mathcal{QF}$*-equational for each* $\mathfrak{C}$.

Proof. Let L be $\mathcal{QF}^{\mathrm{c}}$-equational. Since, for fixed $\mathfrak{C}$, the condition $\mathfrak{A}[C] \cong \mathfrak{C}$ is MSO-definable (even FO-definable) it follows by Proposition 105 and Corollary 28 (b) that each set $L \bowtie \mathfrak{C}$ is $\mathcal{QF}^{\mathrm{c}}$-equational. Hence, it is the image of a regular set of terms under an MSO-transduction and so is $(L \bowtie \mathfrak{C})^\wedge$, by Proposition 105 and Lemma 107 (2).

Conversely, since L is a finite union of sets $L \bowtie \mathfrak{C}$ and singletons $\{\mathfrak{C}\}$, it suffices to prove that each $L \bowtie \mathfrak{C}$ is $\mathcal{QF}^{\mathrm{c}}$-equational. This follows from Lemma 107 (3) by a similar argument as above. □

We will improve Lemma 107 (2) to have statements like Proposition 109 relating $\mathcal{QF}$- and $\mathcal{QF}^{\mathrm{c}}$-recognizability. Let us first state an immediate corollary of Lemma 107 (3) and Proposition 20.

Corollary 110. *Let* $\mathfrak{C} \in \mathrm{STR}[\Sigma, C]$ *be a structure such that* $\mathfrak{C} = \mathfrak{C}[C]$. *For every formula* $\varphi(x_1, \dots, x_n) \in \mathrm{QF}[\Sigma \cup C]$, *one can construct a formula* $\hat{\varphi}(x_1, \dots, x_n) \in \mathrm{QF}[\Sigma(C)]$, *such that we have*

$$\mathfrak{A} \models \varphi(\bar{a}) \quad \text{iff} \quad \hat{\mathfrak{A}} \models \hat{\varphi}(\bar{a}),$$

for every structure $\mathfrak{A} \in \mathrm{STR}[\Sigma, C]$ *with* $\mathfrak{A}[C] = \mathfrak{C}$ *and every* $\bar{a} \in \hat{A}^n$.

Proof. Let $\tau : \mathrm{STR}[\Sigma(C)] \to \mathrm{STR}[\Sigma, C]$ be the transduction of Lemma 107 (3). We can set $\hat{\varphi} := \varphi^\tau$. □

Among the $\mathcal{QF}^{\mathrm{c}}$-operations, it will be convenient to single out particular ones. If $d \in C$, we denote by fgt_d the operation $\mathrm{STR}[\Sigma, C] \to \mathrm{STR}[\Sigma, C \setminus \{d\}]$ that "forgets" the constant d. Nothing is changed except that some element of the domain is no longer denoted by d.

Proposition 111. *The function* $^\wedge : \mathrm{STR}[\Sigma, C] \to \mathrm{STR}[\Sigma(C)]$ *is a finite-state heteromorphism based on the mapping* $\mathfrak{A} \mapsto \mathfrak{A}[C]$.

Proof. We recall that on $\mathrm{STR}[\Sigma, C]$ we use the disjoint union and the QF^{c}-transductions as unary operations. We first observe that the mapping $\mathfrak{A} \mapsto \mathfrak{A}[C]$ is $\mathcal{QF}^{\mathrm{c}}$-computable. This follows from the following obvious facts.

(1) For all structures $\mathfrak{A} \in \mathrm{STR}[\Sigma, C]$ and $\mathfrak{B} \in \mathrm{STR}[\Gamma, D]$ with $C \cap D = \emptyset$, we have

$$(\mathfrak{A} \oplus \mathfrak{B})[C \cup D] = \mathfrak{A}[C] \oplus \mathfrak{B}[D].$$

(2) For every $\mathcal{QF}^{\mathrm{c}}$-operation $f : \mathrm{STR}[\Sigma, C] \to \mathrm{STR}[\Gamma, D]$, we have

$$f(\mathfrak{A})[D] = f(\mathfrak{A}[C])[D].$$

(This is true because $D_{f(\mathfrak{A})} \subseteq C_{\mathfrak{A}}$.)

Going back to the main proof, we consider the various operations. First it is clear that

$$(\mathfrak{A} \oplus \mathfrak{B})^\wedge = \hat{\mathfrak{A}} \oplus \hat{\mathfrak{B}} .$$

The case of a $\mathcal{QF}^{\mathrm{c}}$-operation $f : \mathrm{STR}[\Sigma, C] \to \mathrm{STR}[\Gamma, D]$ is more involved. Suppose that f is defined by the definition scheme

$$\mathcal{D} = (\varphi,\ \psi,\ (\vartheta_R)_{R\in\Gamma},\ (\kappa_{cd})_{c\in C, d\in D}) .$$

We consider a structure $\mathfrak{A}$. Our objective is to express $f(\mathfrak{A})^\wedge$ as $t(\hat{\mathfrak{A}})$ for some $\mathcal{QF}$-term t that may depend on $\mathfrak{A}[C]$. Let $C_{\mathfrak{A}}$ be the set of all elements of $\mathfrak{A}$ denoted by some constant $c \in C$. We denote by $N \subseteq C_{\mathfrak{A}}$ the set of all elements that are not deleted by f (i.e., that satisfy ψ) but that are not denoted by any constant $d \in D$ in $f(\mathfrak{A})$. (Note that we can compute N from $\mathfrak{A}[C]$.) The set $C_{\mathfrak{A}}$ is thus partitioned into $D_{f(\mathfrak{A})}$, N, and the set of all elements deleted by the transduction f. The domain of $f(\mathfrak{A})^\wedge$ consists of N and all elements of $\hat{A} = A \setminus C_{\mathfrak{A}}$ that are not deleted by f. We distinguish several cases.

(a) First, suppose that $N = \emptyset$. The domain of $f(\mathfrak{A})^\wedge$ is the set of elements of $\hat{A}$ that satisfy ψ in $\mathfrak{A}$. By Corollary 110, these are the elements that satisfy $\hat{\psi}$ in $\hat{\mathfrak{A}}$.

Now we consider a relation in $\Gamma(D)$, say $R[{*}c{*}dd{*}]$ to take a representative example. We have

$$\begin{aligned}
&(x, y, z) \in R[{*}c{*}dd{*}]_{f(\mathfrak{A})^\wedge} \\
\text{iff}\quad &(x, c_{f(\mathfrak{A})}, y, d_{f(\mathfrak{A})}, d_{f(\mathfrak{A})}, z) \in R_{f(\mathfrak{A})} \\
\text{iff}\quad &\mathfrak{A} \models \vartheta_R(x, c', y, d', d', z) \wedge \psi(x) \wedge \psi(y) \wedge \psi(z) \wedge \kappa_{c'c} \wedge \kappa_{d'd} , \\
&\text{for some } c', d' \in C \\
\text{iff}\quad &\hat{\mathfrak{A}} \models \hat{\vartheta}_{c',d'} \text{ for some } c', d' \in C \\
\text{iff}\quad &\hat{\mathfrak{A}} \models \hat{\vartheta} := \bigvee_{c',d'} \hat{\vartheta}_{c',d'} ,
\end{aligned}$$

where $\hat{\vartheta}_{c',d'}$ is the formula associated with

$$\vartheta_R(x, c', y, d', d', z) \wedge \psi(x) \wedge \psi(y) \wedge \psi(z) \wedge \kappa_{c'c} \wedge \kappa_{d'd}$$

according to Corollary 110.

The formula $\hat{\psi}$ which defines the domain of $f(\mathfrak{A})^\wedge$ and the formulas $\hat{\vartheta}$ as above yield a definition scheme for the transformation $\hat{\mathfrak{A}} \mapsto f(\mathfrak{A})^\wedge$. Hence, t is a quantifier-free operation.

(b) Next, we consider the case that $N \neq \emptyset$ and $f = \mathrm{fgt}_d$. Then $N = \{d\}$ and there is no $c \in C \setminus \{d\}$ such that $c_{\mathfrak{A}} = d_{\mathfrak{A}}$. The domain of $f(\mathfrak{A})^\wedge$ is that of $\hat{\mathfrak{A}}$ augmented with $d_{\mathfrak{A}}$. Hence we have $f(\mathfrak{A})^\wedge = t'(\hat{\mathfrak{A}} \oplus \mathfrak{D})$ where t' and $\mathfrak{D}$ are defined as follows.

$\mathfrak{D}$ is a structure with the single element $d_{\mathfrak{A}}$. The relations of $\mathfrak{D}$ either are empty or consist solely of the tuple $(d_{\mathfrak{A}}, \dots, d_{\mathfrak{A}})$ depending on whether the corresponding relation of $\mathfrak{A}[C]$ contains this tuple. For example, if $(d_{\mathfrak{A}}, b_{\mathfrak{A}}, c_{\mathfrak{A}}, d_{\mathfrak{A}}, d_{\mathfrak{A}}) \in R_{\mathfrak{A}}$, for $b, c \in C$, then we put the tuple $(d_{\mathfrak{A}}, d_{\mathfrak{A}}, d_{\mathfrak{A}})$ into the set $R[{*}bc{*}{*}]_{\mathfrak{D}}$. We also use a special new unary relation symbol to "mark" $d_{\mathfrak{A}}$, that is, to distinguish it from the elements of $\hat{\mathfrak{A}}$.

Let us call a relation $R[w]$ a *d-relation* if d occurs in w. The mapping t' is a quantifier-free operation that performs the following transformations:

(1) It preserves those relations of $\hat{\mathfrak{A}}$ and $\mathfrak{D}$ that are not d-relations.
(2) It removes all d-relations (they are all in $\hat{\mathfrak{A}}$).
(3) For every tuple in a d-relation, like $(x, y, z) \in R[{*}{*}abdd{*}d]$, it creates a corresponding tuple $(x, y, d_{\mathfrak{A}}, d_{\mathfrak{A}}, z, d_{\mathfrak{A}})$ in the relation $R[{*}{*}ab{*}{*}{*}{*}]$. The marking of $d_{\mathfrak{A}}$ is useful here.
(4) Finally, it removes the "marking" unary relation.

Hence, in this case we can take for t the $\mathcal{QF}$-term $t'(x \oplus \mathfrak{D})$.

(c) For the general case, we show that every $\mathcal{QF}^{\mathrm{c}}$-operation can be expressed as the composition of a bounded number of transformations of the above two forms.

Fix an enumeration $a_1, \ldots, a_k$ of N. (If it is empty case (a) applies.) Let $E = \{e_1, \ldots, e_k\} \subseteq C_\infty$ be a set of constants disjoint from C and D.

Let g be the QF^{c}-transduction that maps a structure $\mathfrak{C}$ with $\mathfrak{C}[C] = \mathfrak{A}[C]$ to the structure $g(\mathfrak{C}) \in \mathrm{STR}[\Sigma, D \cup E]$ obtained from $f(\mathfrak{C})$ by assigning the value a_i to the new constant e_i, for $i \leqslant k$.

The definition scheme of g can be constructed by adding to $\mathcal{D}$ the formulas $\kappa_{ce_i} :=$ true where, for each i, c is some element of C such that $c_{\mathfrak{A}} = a_i$. This choice can be made depending only on $\mathfrak{A}[C]$. The resulting QF^{c}-transduction g is of the type considered in case (a). Furthermore, for every structure $\mathfrak{B}$ with $\mathfrak{B}[C] = \mathfrak{A}[C]$, we have

$$f(\mathfrak{B}) = (\mathrm{fgt}_{e_1} \circ \cdots \circ \mathrm{fgt}_{e_k})(g(\mathfrak{B}))\,.$$

Hence the general case follows by combining the constructions of (a) and (b). □

The main result of this section is the following theorem.

Theorem 112. *Let* $L \subseteq \mathrm{STR}[\Sigma, C]$. *The following statements are equivalent*:

(*i*) *L is $\mathcal{QF}^{\mathrm{c}}$-recognizable.*
(*ii*) *$L_\sqcap$ is $\mathcal{QF}$-recognizable.*
(*iii*) *$(L \bowtie \mathfrak{C})^\wedge$ is $\mathcal{QF}$-recognizable, for every $\mathfrak{C}$ with $\mathfrak{C}[C] = \mathfrak{C}$.*

Proof. (ii) $\Leftrightarrow$ (iii) Note that, by Lemma 107, for every $\mathfrak{C}$, the sets $(L \bowtie \mathfrak{C})^\wedge$ and $(L \bowtie \mathfrak{C})_\sqcap$ are in bijection by an MSO-transduction the inverse of which is also an MSO-transduction. It follows from Theorem 51 that one is $\mathcal{QF}$-recognizable if and only if the other is. Furthermore, the set $\{\, \mathfrak{A} \mid \mathfrak{A} = \mathfrak{A}[C] \,\}$ is MSO-definable and hence recognizable. This proves (iii) $\Rightarrow$ (ii) since

$$L_\sqcap = \{\, \mathfrak{A} \mid \mathfrak{A} = \mathfrak{A}[C] \text{ or } \hat{\mathfrak{A}} \in (L \bowtie \mathfrak{C})^\wedge, \text{ for some } \mathfrak{C} \,\}$$

and a finite union of recognizable sets is recognizable.

For the other direction, note that, if $L_\sqcap$ is $\mathcal{QF}$-recognizable then so is $(L \bowtie \mathfrak{C})_\sqcap$ because the conditions $\mathfrak{A}[C] \cong \mathfrak{C}$ and $\mathfrak{A} \ncong \mathfrak{C}$ are MSO-definable.

(iii) $\Rightarrow$ (i) Suppose that $(L \bowtie \mathfrak{C})^\wedge$ is $\mathcal{QF}$-recognizable, for every $\mathfrak{C}$. Then $L \bowtie \mathfrak{C}$ is the inverse image of $(L \bowtie \mathfrak{C})^\wedge$ under the finite-state heteromorphism $^\wedge$ (Proposition 111). Hence it is $\mathcal{QF}^{\mathrm{c}}$-recognizable, by Lemma 14. It follows that L is $\mathcal{QF}^{\mathrm{c}}$-recognizable since L is a finite union of recognizable sets.

(i) $\Rightarrow$ (iii) We now assume that L is $\mathcal{QF}^c$-recognizable. Let $\approx$ be a finite congruence saturating L. By replacing it if necessary by a finer one, one can assume that $\mathfrak{A} \approx \mathfrak{A}'$ implies that $\mathfrak{A}[C] = \mathfrak{A}'[C]$ and the same relations from Σ are nonempty in $\mathfrak{A}$ and in $\mathfrak{A}'$. Hence this congruence saturates each set $L \bowtie \mathfrak{C}$.

Consider now the inverse mapping $(^\wedge)^{-1} : \mathrm{STR}[\Sigma(C)] \to \mathrm{STR}[\Sigma, C]$. For every $\mathfrak{C} \in \mathrm{STR}[\Sigma, C]$ such that $\mathfrak{C} = \mathfrak{C}[C]$, one can construct a $\mathcal{QF}^c$-term t, using both the relations of $\Sigma(C)$ (this set contains Σ) and the constants of C such that, for every structure $\mathfrak{A} \in \mathrm{STR}[\Sigma, C] \bowtie \mathfrak{C}$, we have $\mathfrak{A} = t(\hat{\mathfrak{A}} \oplus \mathfrak{C})$.

The effect of applying t to $\hat{\mathfrak{A}} \oplus \mathfrak{C}$ must be to replace a tuple like (x, y, u, v, w) in a relation $R[{*}{*}ab{*}{*}c{*}]$ by the tuple $(x, y, a_{\mathfrak{C}}, b_{\mathfrak{C}}, u, v, c_{\mathfrak{C}}, w) \in R$. This can be done by a QF^c-transduction $\tau : \mathrm{STR}[\Sigma(C)] \to \mathrm{STR}[\Sigma, C]$. Hence, we can set $t := \tau(x)$.

The restriction of the congruence $\approx$ to the sets $\mathrm{STR}[\Sigma]$ is a $\mathcal{QF}$-congruence since $\mathcal{QF}$ is a subsignature of $\mathcal{QF}^c$. It remains to check that it saturates $(L \bowtie \mathfrak{C})^\wedge$. Consider a structure $\mathfrak{A} \in (L \bowtie \mathfrak{C})^\wedge$, and suppose that $\mathfrak{A}' \approx \mathfrak{A}$. Let $\mathfrak{B} \in L \bowtie \mathfrak{C}$ be such that $\mathfrak{A} = \hat{\mathfrak{B}}$. Since $\mathfrak{A}'[C] = \mathfrak{A}[C] = \mathfrak{C}$, $\mathfrak{A}[C] \ncong \mathfrak{C}$, and the same relations from $\Sigma(C)$ occur in $\mathfrak{A}$ and $\mathfrak{A}'$, there exists a structure $\mathfrak{B}' \in \mathrm{STR}[\Sigma, C] \bowtie \mathfrak{C}$ such that $\mathfrak{A}' = \widehat{\mathfrak{B}'}$. Applying the term t defined above we obtain $\mathfrak{B} = t(\mathfrak{A} \oplus \mathfrak{C})$ and $\mathfrak{B}' = t(\mathfrak{A}' \oplus \mathfrak{C})$. Hence $\mathfrak{B} \approx \mathfrak{B}'$. But the congruence $\approx$ saturates $L \bowtie \mathfrak{C}$. Hence $\mathfrak{B}'$ belongs to $L \bowtie \mathfrak{C}$ and $\mathfrak{A}'$ belongs to $(L \bowtie \mathfrak{C})^\wedge$. It follows that each set $(L \bowtie \mathfrak{C})^\wedge$ is recognizable. □

Some variants of the operations of $\mathcal{QF}^c$ are considered in [8] where it is shown that one can use the following generalization of disjoint union. If $\mathfrak{A}$ and $\mathfrak{B}$ have a common set of constants C then their *parallel composition* $\mathfrak{A} \,//\, \mathfrak{B}$ is defined from their disjoint union by fusing those elements in $\mathfrak{A}$ and in $\mathfrak{B}$ that are denoted by the same constant. The results of this section extend to the corresponding variant of $\mathcal{QF}^c$.

9. Conclusion

The main results we have established above (Theorem 51 in Section 5, Theorem 68 in Section 6, Theorem 88 and 104 in Section 7 and Theorem 112 in Section 8) tighten even more the relationships between recognizability for algebras of relational structures, monadic second-order transductions, and operations on relational structures defined in terms of logical formulas—quantifier-free or with a limited form of quantification. We have extended older results on the fusion operation and we gave new uniform proofs in a wider algebraic setting.

Some questions remain open though. In particular, a uniform treatment of the fusion operation for relational structures would be desirable.

Open question 1. Are the signatures $\mathcal{QF}$ and $\mathcal{QF} \cup \mathcal{F}use$ equivalent?

Let us mention some other possible future research directions.

(1) Which quantifier-free operations on relational structures preserve recognizability?
(2) Is it true that, if a set of graphs of clique width at most k is $\mathcal{VR}_\Pi$-recognizable, for some set Π of size at most k (or $f(k)$, for some fixed function f), then it is recognizable?
(3) Using the signature $\mathcal{QF}_0^\Sigma$ and its distinction between auxiliary relations and those of Σ, one can define a complexity measure on relational structures that generalizes the notion of clique width: Given a structure $\mathfrak{A} \in \mathrm{STR}[\Sigma]$, let $w(\mathfrak{A})$ be the minimal number n such that there exists a signature

Γ and a term $t \in T(\mathcal{QF}_0[\Sigma, \Gamma])$ with $\mathfrak{A} = \mathrm{val}_{\mathrm{STR}}(t)$ and $\sum_{R\in\Gamma} \mathrm{ar}(R) \leqslant n$. By Proposition 63, it follows that a set $L \subseteq \mathrm{STR}[\Sigma]$ is the image of a set of terms under an MSO-transduction if and only if $w(L)$ is bounded.

For the case of the so-called HR-operations and HR-recognizability, questions related to (1) and (2) have been considered in [16,17]. A measure similar to (3) but based on a different signature is investigated in [2].

Acknowledgment

We thank J. Engelfriet and the referees for their critical reading and their numerous suggestions of improvements.

References

[1] R. Bloem, J. Engelfriet, A comparison of tree transductions defined by MSO logic and attribute grammars, Journal of Computer and System Sciences 61 (2000) 1–50.
[2] A. Blumensath, Structures of Bounded Partition Width, Ph.D. Thesis, RWTH Aachen, Aachen, 2003.
[3] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, M. Tommasi, Tree Automata Techniques and Applications, available online at http://www.grappa.univ-lille3.fr/tata/.
[4] B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic, in: [32], 1997, pp. 313–400.
[5] B. Courcelle, Monadic second-order graph transductions: a survey, Theoretical Computer Science 126 (1994) 53–75.
[6] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, Theory of Computing Systems 33 (2000) 125–150.
[7] B. Courcelle, The monadic second-order logic of graphs VII: graphs as relational structures, Theoretical Computer Science 101 (1992) 3–33.
[8] B. Courcelle, P. Weil, The recognizability of sets of graphs is a robust property, Theoretical Computer Science 342 (2005) 173–228.
[9] B. Courcelle, J.A. Makowsky, Fusion in relational structures and the verification of monadic second-order properties, Mathematical Structures in Computer Science 12 (2002) 203–235.
[10] B. Courcelle, Basic notions of Universal Algebra for Language Theory and Graph Grammars, Theoretical Computer Science 163 (1996) 1–54.
[11] B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-rewriting hypergraph grammars, Journal of Computer and System Science 46 (1993) 218–270.
[12] B. Courcelle, The monadic second-order logic of graphs VI: on several representations of graphs by relational structures, Discrete Applied Mathematics 54 (1994) 117–149.
[13] B. Courcelle, The monadic second-order logic of graphs V: on closing the gap between definability and recognizability, Theoretical Computer Science 80 (1991) 153–202.
[14] B. Courcelle, The monadic second-order logic of graphs XVI: canonical graph decompositions, Logical Methods in Computer Science 2 (2006) 1–46.
[15] B. Courcelle, S. Olariu, Upper bounds to the clique width of graphs, Discrete Applied Mathematics 101 (2000) 77–114.
[16] B. Courcelle, Recognizable sets of graphs: equivalent definitions and closure properties, Mathematical Structures in Computer Science 4 (1994) 1–32.
[17] B. Courcelle, J. Lagergren, Equivalent definitions of recognizability for sets of graphs of bounded tree-width, Mathematical Structures in Computer Science 6 (1996) 141–165.
[18] W. Cunningham, Decomposition of directed graphs, SIAM J. Alg. Discrete Methods 3 (1982) 214–228.

[19] H.-D. Ebbinghaus, J. Flum, Finite Model Theory, Springer Verlag, Berlin, 1995.
[20] A. Ehrenfeucht, T. Harju, G. Rozenberg, The Theory of 2-structures, A Framework for Decompositions and Transformations of Graphs, World Scientific, Singapore, 1999.
[21] J. Engelfriet, S. Maneth, Macro tree translations of linear size increase are MSO definable, SIAM Journal of Computers 32 (2003) 950–1006.
[22] J. Engelfriet, V. van Oostrom, Logical description of context-free graph-languages, Journal of Computer and System Sciences 55 (1997) 489–503.
[23] F. Gécseg, M. Steinby, Tree Automata, Akadémiai Kiado, Budapest, 1984.
[24] F. Gécseg, M. Steinby, Tree Languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of formal languages, Vol. 3, Springer Verlag, 1997, pp. 1–68..
[25] Y. Gurevich, Monadic second-order theories, in: J. Barwise, S. Feferman (Eds.), Model-Theoretic Logics, Springer, Berlin, 1985, pp. 479–506.
[26] W. Hodges, Model Theory, Cambridge University Press, Cambridge, 1993.
[27] D. Lapoire, Recognizability equals monadic second-order definability for sets of graphs of bounded tree-width, in: Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science, STACS, Lecture Notes in Computer Science, vol. 1373, 1998, pp. 618–628.
[28] L. Libkin, Elements of Finite Model Theory, Springer Verlag, Berlin, 2004.
[29] J.A. Makowsky, Algorithmic aspects of the Feferman–Vaught Theorem, Annals of Pure and Applied Logic 126 (2004) 159–213.
[30] J.E. Mezei, J. Wright, Algebraic automata and context-free sets, Information and Control 11 (1967) 3–29.
[31] J.-C. Raoult, A survey of tree-transduction, in: M. Nivat, A. Podelski (Eds.), Tree Automata and Languages, North-Holland, Amsterdam, pp. 311–325.
[32] G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformations, vol. 1: Foundations, World Scientific, Singapore, 1997.
[33] S. Shelah, The monadic theory of order, Annals of Mathematics 102 (1975) 379–419.