



# Algorithmic uses of the Feferman–Vaught Theorem

J.A. Makowsky

*Department of Computer Science, Technion-Israel Institute of Technology, Haifa 32000, Israel*

On the occasion of Alfred Tarski's Centenary (1901–1983)  
In memoriam Robert Vaught (1926–2002)  
Dedicated to Solomon Feferman (1928–)<sup>1</sup>

---

## Abstract

The classical Feferman–Vaught Theorem for First Order Logic explains how to compute the truth value of a first order sentence in a generalized product of first order structures by reducing this computation to the computation of truth values of other first order sentences in the factors and evaluation of a monadic second order sentence in the index structure. This technique was later extended by Läuchli, Shelah and Gurevich to monadic second order logic. The technique has wide applications in decidability and definability theory.

Here we give a unified presentation, including some new results, of how to use the Feferman–Vaught Theorem, and some new variations thereof, algorithmically in the case of Monadic Second Order Logic *MSOL*.

We then extend the technique to graph polynomials where the range of the summation of the monomials is definable in *MSOL*. Here the Feferman–Vaught Theorem for these polynomials generalizes well known splitting theorems for graph polynomials. Again, these can be used algorithmically.

Finally, we discuss extensions of *MSOL* for which the Feferman–Vaught Theorem holds as well.

© 2003 Elsevier B.V. All rights reserved.

*MSC:* 03C; 05C; 05E; 68R; 68Q; 68W

*Keywords:* Monadic second order logic; Tree width; Clique width; Graph algorithms; Graph polynomials

---

*E-mail address:* [janos@cs.technion.ac.il](mailto:janos@cs.technion.ac.il) (J.A. Makowsky).

<sup>1</sup> At the occasion of his 75th birthday. It is also a tribute to Feferman's work in model theory, which went seemingly unnoticed at the celebrations of his 70th birthday. Feferman returned several times to topics related to the preservation of truth under various model theoretic constructions, so in [54–58,60]. The Feferman–Vaught Theorem can also be viewed as falling into this category.

## 1. Introduction

As the title indicates, this paper deals with algorithmic aspects of the Feferman–Vaught Theorem. It is both a survey paper and contains new results. It traces the history of the algorithmic uses of the Feferman–Vaught Theorem and it stresses unifying traits which led to the most recent applications and extensions. It is, last but not least, a tribute to Tarski and his two outstanding students, Feferman and Vaught, and to the environment Tarski had created in Berkeley, where early model theory flourished and some of its most seminal concepts were formulated.

### 1.1. Notation and prerequisites

A vocabulary (or a *similarity type*<sup>2</sup> as Tarski used to call it)  $\tau$  is a finite set of relation symbols, function symbols and constants. It may be many-sorted. In this paper, however, we shall only deal with vocabularies which do not contain any function symbols.<sup>3</sup>  $\tau$ -structures are interpretations of vocabularies. Sorts are mapped into non-empty sets,<sup>4</sup> the sort-universes. Relation symbols are mapped into relations over the sorts according to their specified arities. Constant symbols are mapped into elements of the corresponding sort-universes.  $FOL(\tau)$  denotes the set of  $\tau$ -formulas in first order logic.  $SOL(\tau)$  and  $MSOL(\tau)$  denote the set of  $\tau$ -formulas in second order and monadic second order logic, respectively. A *sentence* is a formula without free variables. For a class of  $\tau$ -structures  $K$ ,  $Th_{FOL}(K)$  is the set of sentences of  $FOL(\tau)$  true in all  $\mathfrak{A} \in K$ . We write  $Th_{FOL}(\mathfrak{A})$  for  $K = \{\mathfrak{A}\}$ . Similarly,  $Th_{SOL}(K)$  and  $Th_{MSOL}(K)$  denote the corresponding sets of sentences for  $SOL$  and  $MSOL$ . For a set of sentences  $\Sigma \subseteq SOL(\tau)$  we denote by  $Mod(\Sigma)$  the class of  $\tau$ -structures which are models of  $\Sigma$ .

We assume the reader is familiar with basic finite model theory and descriptive complexity theory as described in, say, [41,42,88]. In particular, the reader should be familiar with Ehrenfeucht–Fraïssé games, also called back and forth games, for first order and monadic second order logic, and with first order reductions (transductions, interpretations). The latter are nevertheless defined in detail in Section 2.

Finally, we treat free variables as *uninterpreted constants*. In particular, we tacitly assume that whenever we write  $\phi(\bar{x}, \bar{U}) \in MSOL(\tau)$  we think of  $\phi(\bar{a}, \bar{P}) \in MSOL(\tau \cup \{\bar{a}, \bar{P}\})$  where  $\bar{a}$  are the uninterpreted constants corresponding to  $\bar{x}$  and  $\bar{P}$  are the uninterpreted unary relation symbols corresponding to  $\bar{U}$ . This allows us to speak of theories with free variables without having to deal with the free variables separately.

<sup>2</sup> Originally Tarski called a similarity type an equivalence relation between structures, which are all interpretations of the same vocabulary. The vocabulary then can be viewed as a name for such an equivalence class.

<sup>3</sup> Function symbols can be replaced by relation symbols where the interpretation is the graph of the function which interprets the function symbol. The price one has to pay depends on the nesting depth of the terms, which translate into quantifier rank.

<sup>4</sup> In the one-sorted case the sort-universe is always required to be non-empty. However, certain authors allow empty sets as sort-universes in the many-sorted case, provided at least one sort-universe is non-empty. We shall not follow this convention. It requires too much exception handling and complicates matters unnecessarily. One can also replace sorts by unary predicates and think of one big universe instead.

## 1.2. Some history

The Feferman–Vaught Theorem stands at the beginnings of model theory. Hodges, in his delightful book [85], very carefully traces the history of early model theoretic developments. Most of the references in the sequel are taken from it.

Tarski published four short abstracts on model theory in 1949 in the Bulletin of the American Mathematical Society [156–159]. He also had sent his manuscript of *Contribution to the theory of models, I* to Beth for publication as [160]. Seemingly inspired by these, Beth published two papers on model theory [8,9]. In [9] he, and independently Fraïssé in [67], showed, among other things, that

**Theorem 1.1** (Beth [9] and Fraïssé [67]). *Let  $\mathfrak{A}, \mathfrak{B}$  be linear orders,  $\mathfrak{C} = \mathfrak{A} \sqcup_{<} \mathfrak{B}$  their ordered disjoint union. Then  $Th_{FOL}(\mathfrak{C})$  is uniquely determined by  $Th_{FOL}(\mathfrak{A})$  and  $Th_{FOL}(\mathfrak{B})$ , and can be computed from  $Th_{FOL}(\mathfrak{A})$  and  $Th_{FOL}(\mathfrak{B})$ .*

In the early 1950s Tarski had many young researchers gathered in Berkeley, among them Ehrenfeucht, Feferman, Fraïssé and Vaught. One of the many questions studied in this early period of logic, and the one which interests us here is the following:

**Question 1.** *Let  $\mathfrak{A}, \mathfrak{B}$  be  $\tau$ -structures,  $\mathfrak{A} \times \mathfrak{B}$  the cartesian product and  $\mathfrak{A} \sqcup \mathfrak{B}$  the disjoint union. Assume we are given  $Th_{FOL}(\mathfrak{A})$  and  $Th_{FOL}(\mathfrak{B})$ . What can we say about  $Th_{FOL}(\mathfrak{A} \times \mathfrak{B})$  and  $Th_{FOL}(\mathfrak{A} \sqcup \mathfrak{B})$ ? What happens in the case of infinite sums and products?*

This question triggered many landmark papers and also led to the study of ultraproducts.

Among Tarski’s pupils<sup>5</sup> dealing directly or indirectly with this question we have (with the year and place of their Ph.D.)

Andrzej Mostowski (Warsaw, 1938), Bjarni Jónsson (Berkeley, 1946), Wanda Szmielew (Warsaw, 1950), Anne Morel (Berkeley, 1952), Robert Vaught (Berkeley, 1954), Chen-Chung Chang (Berkeley, 1955) Solomon Feferman (Berkeley, 1957) Jerome Keisler (Berkeley, 1961).

Jónsson studied first order theories of products in group theory and what became later *Universal Algebra*, a field he had shaped single handedly [91–93]. Szmielew used the analysis of the first order theory of products in her decidability result for Abelian groups [155]. Morel, [69], Chang and Keisler were pioneers in generalized products and ultraproducts, cf. [7,20].

The Feferman–Vaught Theorem evolved as follows. In [120] Mostowski proves, among other things<sup>6</sup> the analogue of Theorem 1.1 for products

<sup>5</sup> Although perceived as a Tarski student, Scott finished his Ph.D. under Church in Princeton.

<sup>6</sup> In his own words: “The paper deals with the notion of direct product in the theory of decision problems. (...) [It discusses] a theory of which the primitive notions are representable as powers of certain base-relations and [reduces] all the problems concerning this theory (in particular the decision problem) to problems concerning the theory of the base relations”.

**Theorem 1.2** (Mostowski [120]). *Let  $\mathfrak{A}, \mathfrak{B}$  relational structures or algebras,  $\mathfrak{C} = \mathfrak{A} \times \mathfrak{B}$  their cartesian product. Then  $Th_{FOL}(\mathfrak{C})$  is uniquely determined by, and can be computed effectively from,  $Th_{FOL}(\mathfrak{A})$  and  $Th_{FOL}(\mathfrak{B})$ .*

Mostowski's proof already contains all the ingredients of the proof via reduction sequences, as in the proof of Theorem 1.6, below.

Finally, Feferman and Vaught answered the question in the outermost generality, but, as Feferman recalls,<sup>7</sup> Tarski did not really appreciate the answer given. A special case of this answer reads as follows.

**Theorem 1.3** (Feferman and Vaught [61]). *Let  $\mathfrak{A}_i, i \in I$  be structures of the same similarity type. Then the theory of the infinite cartesian product  $Th_{FOL}(\prod_{i \in I} \mathfrak{A}_i)$  and the theory of the disjoint union  $Th_{FOL}(\bigsqcup_{i \in I} \mathfrak{A}_i)$  are uniquely determined by  $Th_{FOL}(\mathfrak{A}_i)$ .*

Another version also allows for the index structure to vary.

**Theorem 1.4** (Mostowski [120] and Feferman and Vaught [61]). *Let  $\mathfrak{A}$  be structure and  $I$  an index set. Then the theory of the infinite cartesian product  $Th_{FOL}(\prod_{i \in I} \mathfrak{A})$  and the theory of the disjoint union  $Th_{FOL}(\bigsqcup_{i \in I} \mathfrak{A})$  are uniquely determined by  $Th_{FOL}(\mathfrak{A})$  and  $Th_{MSOL}(I)$ .*

The full generality is achieved by observing the following:

- By combining Theorems 1.3 and 1.4 with transductions and interpretations, similar results can be stated for a wide variety of *generalized products*. In the original paper [61] the transductions or interpretations are hidden in an unfortunate lengthy definition of generalized products. We shall see in the sequel, how one can, in the case of fixed finite index sets, successfully separate the issues.
- In a sequence of papers by Ehrenfeucht, Läuchli, Shelah and Gurevich, cf. [44,76,77,98,149] it emerged that Theorems 1.3 and 1.4 remain true for *MSOL* rather than *FOL* in the case of the sum  $Th_{MSOL}(\bigsqcup_{i \in I} \mathfrak{A}_i)$  and the multiple disjoint union  $Th_{MSOL}(\bigsqcup_{i \in I} \mathfrak{A})$  but not for products.

### 1.3. Proofs

An easy and transparent proof of Theorems 1.1 and 1.2 uses *Ehrenfeucht–Fraïssé Games* and was presented by Feferman as early as in 1957 [53], at the famous Summer Institute for Symbolic Logic at Cornell University.<sup>8</sup> Feferman knew Fraïssé's [67,68], and also was familiar with Ehrenfeucht's appealing presentation [43,44] even before its publication as [43,44].

This approach gives actually more.

<sup>7</sup> Personal communication, December 2000.

<sup>8</sup> For Tarski's influence on this Summer Institute, and Feferman's recollections, cf. [59].

The quantifier rank of a formula is defined inductively. Atomic formulas have rank 0. Boolean operations do not increase the rank, but quantification of one variable (first or second order) increases the rank by 1. Let  $q \in \mathbb{N}$  and  $FOL^q(\tau)$   $MSOL^q(\tau)$  denote the sentences of  $FOL$  and  $MSOL$  respectively of *quantifier rank at most  $q$* . Furthermore, put  $Th_{FOL}^q(\mathfrak{A}) = Th_{FOL}(\mathfrak{A}) \cap FOL^q(\tau)$ . and  $Th_{MSOL}^q(\mathfrak{A}) = Th_{MSOL}(\mathfrak{A}) \cap MSOL^q(\tau)$ .

**Theorem 1.5** (Ehrenfeucht, Fraïssé, Feferman and Vaught, cf. [53]).

- (i)  $Th_{FOL}^q(\mathfrak{A} \times \mathfrak{B})$  and  $Th_{FOL}^q(\mathfrak{A} \sqcup \mathfrak{B})$  are uniquely determined by  $Th_{FOL}^q(\mathfrak{A})$  and  $Th_{FOL}^q(\mathfrak{B})$ .
- (ii)  $Th_{MSOL}^q(\mathfrak{A} \sqcup \mathfrak{B})$  is uniquely determined by  $Th_{MSOL}^q(\mathfrak{A})$  and  $Th_{MSOL}^q(\mathfrak{B})$ .

**Proof.** The proof just consists of putting the winning strategies of the games for  $Th_{FOL}^q(\mathfrak{A})$  and  $Th_{FOL}^q(\mathfrak{B})$  together. In the case of  $FOL$  this works both for disjoint unions and Cartesian products. If we use  $Th_{MSOL}^q(\mathfrak{A})$  and  $Th_{MSOL}^q(\mathfrak{B})$  instead, it still works for the disjoint unions, as a subset of a disjoint union is uniquely determined by its restrictions to the contributing sets. In contrast to this, a subset of a Cartesian product of two sets is not uniquely determined by its two projection, so this approach does not work.  $\square$

A different, more computational proof is presented in [61], using *reduction sequences*, anticipated in the original proofs of Theorems 1.1 and 1.2.

Here one proves by induction, say for disjoint union and  $FOL$ .<sup>9</sup>

**Theorem 1.6** (Feferman and Vaught [61]).

- (i) For every  $q \in \mathbb{N}$  and for every formula  $\phi \in FOL^q(\tau)$  one can compute effectively a reduction sequence, i.e., a sequence of formulas

$$\langle \psi_1^A, \dots, \psi_m^A, \psi_1^B, \dots, \psi_m^B \rangle \in FOL^q(\tau)^{2m}$$

and a boolean function  $B_\phi : \{0, 1\}^{2m} \rightarrow \{0, 1\}$  such that

$$\mathfrak{A} \sqcup \mathfrak{B} \models \phi$$

iff

$$B_\phi(b_1^A, \dots, b_m^A, b_1^B, \dots, b_m^B) = 1$$

where  $b_j^A = 1$  iff  $\mathfrak{A} \models \psi_j^A$  and  $b_j^B = 1$  iff  $\mathfrak{B} \models \psi_j^B$ .

- (ii) The time complexity of the computation of the reduction sequence is bounded by an iterated exponential where the number of iterations depends on the quantifier rank  $q$  of  $\phi$  and is  $O(q)$ .

<sup>9</sup> In the literature one finds mostly proofs with infinite products or disjoint unions. The only proof in print of this special case I have found is in [27]. In [85] it is recommended to prove this special case as an enlightening exercise.

(iii) The corresponding theorem for MSOL is formulated in the same way. One just has to require  $\phi \in \text{MSOL}^q(\tau)$  and  $\langle \psi_1^A, \dots, \psi_m^A, \psi_1^B, \dots, \psi_m^B \rangle \in \text{MSOL}^q(\tau)^{2m}$ .

**Proof.** We exemplify the construction of such a reduction sequence in the case of ordered graphs  $G = \langle V, E, < \rangle$  with  $V$  the set of vertices,  $E$  a binary, not necessarily symmetric, edge relation and  $<$  a linear ordering of the vertices.

We are given  $G_1 = \langle V_1, E_1, <_1 \rangle$  and  $G_2 = \langle V_2, E_2, <_2 \rangle$  and their ordered sum  $G = G_1 \oplus_{<} G_2$  with  $V = V_1 \sqcup V_2$ ,  $E = E_1 \sqcup E_2$  and the order is defined by  $< = <_1 \sqcup <_2 \sqcup (V_1 \times V_2)$ .

We construct reduction sequences and boolean functions by induction. As we have free variables in the inductive construction, we assume that  $z: \text{Variables} \rightarrow V = V_1 \sqcup V_2$  is an assignment of the variables.

There are three types of atomic formulas:  $E(u, v)$ ,  $u < v$  and  $u \approx v$ . Using the definition of ordered sum we get

**For  $E(u, v)$**

Reduction sequence:  $\langle E_1(u, v), E_2(u, v) \rangle$

Boolean function:  $b_1^1 \vee b_1^2$ .

Here only the cases where  $z(u)$  and  $z(v)$  are both in  $V_1$  or both in  $V_2$  are relevant.

**For  $u \approx v$**

Reduction sequence:  $\langle u \approx_1 v, u \approx_2 v \rangle$

Boolean function:  $b_1^1 \vee b_1^2$ .

Again, only the cases where  $z(u)$  and  $z(v)$  are both in  $V_1$  or both in  $V_2$  are relevant.

**For  $u < v$**

Reduction sequence:  $\langle u <_1 v, u \approx_1 u, u <_2 v, v \approx_2 v \rangle$ .

Boolean function:  $b_1^1 \vee b_1^2 \vee (b_1^2 \wedge b_2^2)$ .

Here, the relevant cases are  $z(u)$  and  $z(v)$  are both in  $V_1$  or both in  $V_2$ , or  $z(u)$  is in  $V_1$  and  $z(v)$  is in  $V_2$ .

Let  $\Phi = \langle \phi_1^A, \dots, \phi_m^A, \phi_1^B, \dots, \phi_m^B \rangle$  and  $\Psi = \langle \psi_1^A, \dots, \psi_n^A, \psi_1^B, \dots, \psi_n^B \rangle$  be reduction sequences for  $\phi$  and  $\psi$  and  $B_\phi(\bar{b})$  and  $B_\psi(\bar{b}')$  the corresponding boolean functions with disjoint variables.

**$(\phi \wedge \psi)$**

Reduction sequence:  $\langle \Phi, \Psi \rangle$ .

Boolean function:  $B_\phi(\bar{b}) \wedge B_\psi(\bar{b}')$ .

**$\neg \phi$**

reduction sequences:  $\Phi$ .

Boolean function:  $\neg B_\phi(\bar{b})$ .

Each application of a propositional connectives results in linear growth of the reduction sequence. The case of quantification is considerably more complicated.

Let  $B_1$  be the disjunctive normal form of  $B_\phi(\bar{b})$  with

$$B_1 = \bigvee_{j \in J} C_j$$

with

$$C_j = \left( \bigwedge_{i \in J(i,A,pos)} b_i^A \wedge \bigwedge_{i \in J(i,A,neg)} \neg b_i^A \wedge \bigwedge_{i \in J(i,B,pos)} b_i^B \wedge \bigwedge_{i \in J(i,B,neg)} \neg b_i^B \right).$$

Now let

$$\theta_j^A = \exists x \left( \bigwedge_{i \in J(i,A,pos)} \phi_i^A(x) \wedge \bigwedge_{i \in J(i,A,neg)} \neg \phi_i^A(x) \right)$$

and

$$\theta_j^B = \exists y \left( \bigwedge_{i \in J(i,B,pos)} \phi_i^A(y) \wedge \bigwedge_{i \in J(i,B,neg)} \neg \phi_i^A(y) \right).$$

Finally we put

$$B_{\exists}(\bar{c}) = \bigvee_{j \in J} (c_j^A \vee c_j^B),$$

where  $c_j^A = 1$  iff  $\mathfrak{A} \models \theta_j^A$  and  $c_j^{B=1}$  iff  $\mathfrak{B} \models \theta_j^B$ .

With this notation, and  $m(J) = |J|$ , it is easy to verify that

$\exists x \phi$

Reduction sequence:  $\langle \theta_1^A, \dots, \theta_{m(J)}^A, \theta_1^B, \dots, \theta_{m(J)}^B \rangle$ .

Boolean function:  $B_{\exists}(\bar{c})$ .

For *MSOL* the proof is similar. Again, we need that every  $X \subseteq V$  has a *unique* decomposition  $X = X_1 \sqcup X_2$  with  $X_i \subseteq V_i$ .

The additional clause in the induction are:

$u \in X$

Reduction sequence:  $\langle u \in X_1, u \in X_2 \rangle$ .

Boolean function:  $b_1^1, \vee b_2^1$ .

$\exists X \phi$

With the same notation is in the case of first order existential quantification we first put

$$\theta_j^A = \exists X_1 \left( \bigwedge_{i \in J(i,A,pos)} \phi_i^A(X_1) \wedge \bigwedge_{i \in J(i,A,neg)} \neg \phi_i^A(X_1) \right)$$

and

$$\theta_j^B = \exists X_2 \left( \bigwedge_{i \in J(i,B,pos)} \phi_i^A(X_2) \wedge \bigwedge_{i \in J(i,B,neg)} \neg \phi_i^A(X_2) \right)$$

and we get:

Reduction sequence:  $\langle \theta_1^A, \dots, \theta_{m(J)}^A, \theta_1^B, \dots, \theta_{m(J)}^B \rangle$ .

Boolean function:  $B_{\exists}(\bar{c})$ .

The quantification step can be simultaneously performed for a block of existential quantifiers. Hence, if we have a formula  $\phi$  in prenex normal form, the time complexity of computing the reduction sequence, and its length, is an iterated exponential of the number of quantifier alternations, rather than of the quantifier rank.  $\square$

### Problem 1.

- (i) Find, if possible, a better algorithm for computing the reduction sequences of quantified formula, which avoids the computation of the disjunctive normal forms.
- (ii) Find sharp upper and lower bounds for the complexity of computing the reduction sequences.

#### 1.4. Algorithmic applications I: finite automata

For the algorithmic applications of Theorems 1.5, 1.6 we restrict our attention to finite structures.

The historically first algorithmic result which has a proof using the Feferman–Vaught theorem is the celebrated characterization of Büchi of regular languages in terms of definability by *MSOL*-sentences [17,18]. The theorem was also independently proved by Trakhtenbrot [163], and Elgot [45].

**Theorem 1.7** (Büchi [17], Elgot [45] and Trakhtenbrot [163]). *A set of words is definable by an MSOL-sentence iff it is recognizable by some finite automaton. Furthermore, the correspondence between automata and MSOL-sentences is computable.*

**Proof.** It is a straightforward induction, how to translate a regular expression into an *MSOL*-sentence, hence, using Kleene’s theorem, one direction is settled. The other direction uses the lookup table of Section 4.5 to define the automaton from an *MSOL*-sentence.  $\square$

**Corollary 1.8.** *MSOL-properties on words can be checked in polynomial time and constant space.*

In the papers [17,18,45,163] the model theoretic content of the proof was lost in the constructive details. A very transparent translation proof may be found in Straubing’s excellent monograph [154]. A clearly model theoretic proof using Ehrenfeucht–Fraïssé games is due to Ladner, [96], cf. also [41]. This proof does not give a computable translation, but is the key to further generalizations. A direct proof of Corollary 1.8 using the Feferman–Vaught theorem is given in Sections 4.5 and 4.6.



### 1.5. Algorithmic applications II: decidable MSOL-theories

Tarski and Mostowski proved before World War II that the first order theory of well-orderings is decidable. The manuscript, however was lost during the War and the result was announced in [121]. A full account of this result is given in [39]. The reconstruction of the lost proof benefited a lot from the later developments, including from the Feferman–Vaught Theorem. Explicit algorithmic use of the Feferman–Vaught theorem was made in many decidability results of *FOL* and *MSOL* theories dealing with infinite structures. We shall not discuss this line of results here, but refer the interested reader to the survey papers and monographs [22,23,50,62,85,119,139]. In [23,62] the Feferman–Vaught theorems also play an essential rôle in determining the complexity of decidable theories.

From our model theoretic point of view the developments which led to the decidability of the *MSOL* theory of finite labeled trees are particularly instructive.

A first step follows from Büchi’s celebrated Theorem 1.7:

**Theorem 1.9** (Büchi, 1961). *The MSOL-theory of finite linear orders is decidable.*

The decidability follows immediately from Theorem 1.7 and Moore’s theorem on the decidability of equivalence of finite automata, cf. e.g. [87]. There are two ingredients here, Theorem 1.7, which can be proven with model theoretic means, and an automata theoretic theorem. Following the model theoretic line, Läuchli [97] showed that weak monadic second order theory (with set quantification restricted to finite sets) of linear order is decidable, but Shelah [149], finally showed that the *MSOL* theory of all linear orders depends on set theory, and is, assuming the Continuum Hypothesis, undecidable. Doner announced in [38] that the weak second order theory of two successors is decidable. Rabin showed in [129], using automata theoretic methods, that the *MSOL* theory of binary trees is decidable. But here no purely model theoretic proof is known. From the theorem on binary trees Rabin then deduces the stronger result

**Theorem 1.10** (Rabin [129]). *The MSOL-theories of finite trees, infinite trees is decidable. The same holds for rooted trees.*

**Proof.** [Note on proofs:] Rabin’s classical theorem is about the full binary infinite tree. But the other cases follow easily. For rooted trees one has to add a constant. For arbitrary binary trees, one has to use relativization via a unary predicate. For finite trees one has to observe that finiteness is expressible by an *MSOL*-formula in the language of trees. For more details, cf. Thomas’ excellent survey [162]. □

A natural question to consider here is the following:

**Question 2.** *Which classes of finite structures have a decidable MSOL theory?*

As Gurevich notes in his [77]:

the most important tools for dealing with monadic theories are composition theorems. The term “composition” here means generalized products in the sense of Feferman–Vaught [61].

The notion of tree width of a graph, which is to play a key rôle in our story, emerged in this time in various contexts, cf. [82,136,171]. It is central in the deep combinatorial analysis of graph classes with forbidden minors, as pursued by Robertson and Seymour. We invite our reader to consult the excellent monograph of Diestel [96]. A concept related to the tree width, introduced by Courcelle, Engelfriet and Rozenberg, and, independently, by Wanke<sup>10</sup> [29,172], and further studied in [33,34,36,74] is the more general notion of clique width. We shall define and discuss tree width and clique width in a general context in Section 4.2. Tree width measures in sense how close a graph is to being a tree, and in many cases it also measures how small the largest clique-like induced subgraph or minor can be. The graphs of clique width at most 2 contain already all the cliques. Courcelle will forgive me, as I think the name is not a lucky choice, but no better name did appear in the literature. However, we do not yet really understand when graphs have a large clique width. Examples are studied in [74,115]. Two-dimensional  $(n \times n)$ -grids have both large tree width and clique width. A class of graphs of arbitrarily large tree width has, by a deep theorem due to Robertson and Seymour [137], arbitrarily large grids as its minors. No such characterization of classes of graphs of arbitrarily large clique width is known.

In a series of papers [145,146,148], Seese investigated classes of graphs which are tree-like. In [148] he finally proved, relying heavily on work by Robertson and Seymour [136], an amazing theorem.

**Theorem 1.11** (Seese [147]). *Let  $K$  be class of finite graphs  $G = \langle V, E, R \rangle$  represented as  $\tau_2$ -structures with two sorts of elements, vertices  $V$  and edges  $E$ , and an incidence relation  $R$ . If  $K$  has a decidable MSOL-theory, then  $K$  is of bounded tree width.*

The proof uses that, without loss of generality, one can assume that  $K$  is closed under minors, and that the MSOL-theory of grids is undecidable. Using the previous remarks, it is not too difficult to see that the corresponding MSOL-theory  $TW_k$  of all graphs (as  $\tau_2$ -structures) of tree width at most  $k$  is decidable.

If we restrict our presentation of graphs to  $\tau_1$ -structures  $G = \langle V, E \rangle$  which are one-sorted with vertices as elements and a binary symmetric edge relation  $E$ , the cliques have a decidable MSOL-theory. This is so, because the edge relation is just inequality, and the MSOL-theory of finite sets is decidable. More generally, the corresponding MSOL-theory  $CW_k$  of all graphs (as  $\tau_1$ -structures) of clique width at most  $k$  is decidable.

Seese conjectures that if a class  $K$  of finite graphs represented as  $\tau_1$ -structures has a decidable MSOL-theory, then  $K$  is contained in an MSOL-interpretable subset of finite

---

<sup>10</sup> Under the name  $k$ -NLC-graphs.

trees. Using a result from [30], Engelfriet reformulated the conjecture in terms of clique width:

**Conjecture 1** (Engelfriet–Seese Conjecture). *Let  $K$  be class of finite graphs  $G = \langle V, E \rangle$  represented as  $\tau_1$ -structures. If  $K$  has a decidable MSOL-theory, then  $K$  is of bounded clique width.*

A proof of this conjecture would very likely be based on a characterization of classes of graphs of arbitrarily large clique width.

### 1.6. Algorithmic applications III: checking MSOL-properties

Finite orders and words are constructed from one-element structures by repeated application of ordered disjoint unions or concatenation respectively. Words are just linear orders with unary predicates indicating which positions carry which letter. For both operations Theorem 1.1 applies. This gives a straightforward algorithm to check MSOL-properties of finite linear orders or words. We first observe that the set  $MSOL^q(\tau, \bar{x}, \bar{U})$  of MSOL-formulas of quantifier rank at most  $q$  is, up to logical equivalence, finite. For a more detailed discussion of this, cf. Section 4.5.

Let  $\phi$  be a  $MSOL^q$ -sentence for words. Now given a finite word  $w$  we look at it as a concatenation of two smaller words, i.e.  $w = w_1 \circ w_2$ . We can check  $w \models \phi$  by applying Theorem 1.1. This reduces to checking formulas  $\phi_1, \dots, \phi_m$  on  $w_1$  and  $w_2$  respectively, and then computing a fixed boolean function of the truth values of  $\phi_i$  on  $w_1$  or  $w_2$ , which is nothing else than the reduction sequence for  $\phi$  for concatenation. As  $MSOL^q$  is finite, we can precompute all the reduction sequences in advance and then repeat this step iteratively, till we only have to check formulas on one-letter words. This algorithm runs in linear time in the size of the word, but obviously is not practical, as the constants involved coming from the computation of all the possible reduction sequences for  $MSOL^q$  are prohibitively large.

A generalization of this algorithm will be shown to work in Section 4.6 for a wide class of graph grammars based on edge replacements (HR-grammars) and vertex replacements (VR-grammars) where each replacement operation satisfies a version of the Feferman–Vaught theorem as in Theorems 1.5 and 1.6.

In particular, the graphs of bounded tree width and of bounded clique width can be viewed this way, and we have

**Theorem 1.12** (Courcelle [26,27]). *Let  $K$  be class of finite graphs  $G = \langle V, E, R \rangle$  represented as  $\tau_2$ -structures and  $\phi$  be a  $MSOL(\tau_2)$  sentence. If  $K$  has of bounded tree width and  $G \in K$ , then checking whether  $G \models \phi$  can be done in linear time.*

Theorem 1.12 was reproven subsequently by several authors almost simultaneously, cf. [4,14,27,35].

Polynomial (linear) time is obtained, because there is a linear time algorithm, which, given  $G$  of tree width at most  $k$ , produces a parse tree  $T_G$  from which  $G$  can be computed, cf. [11]. This parse tree  $T_G$  is then used to check, whether  $\phi$  is true in  $G$ .

In the case of clique width at most  $k$ , it is open whether the parse tree  $T_G$  can be found in polynomial time. But we still have the following theorem.

**Theorem 1.13.** *Let  $K$  be class of finite graphs  $G = \langle V, E \rangle$  represented as  $\tau_1$ -structures and  $\phi$  be a  $MSOL(\tau_1)$  sentence. Assume  $K$  is of clique width at most  $k$ .*

- (i) *If  $G \in K$  is given as  $\tau_1$ -structure then checking whether  $G \models \phi$  is in **NP**.*
- (ii) *If  $G \in K$  is given as a parse tree  $T_G$ , then checking whether  $G \models \phi$  can be done in linear time in the size of  $T_G$ .*
- (iii) *If finding a parse tree for  $G \in K$  is in **P**, then so is checking whether  $G \models \phi$ .*

The second part is from [32–34]. One can view this as a form of fixed parameter feasibility (FPT) result in the sense of [40].

We shall show in Sections 4.5 and 4.6 that similar theorems hold for a wide class of graph grammars and inductively defined classes of structures.

In contrast to Theorems 1.12 and 1.13 it was shown by Pnueli and the author that checking  $MSOL$ -properties can be arbitrarily hard within the polynomial hierarchy, cf. [111]:

**Theorem 1.14** (Makowsky and Pnueli [111]). *For every level of the polynomial hierarchy  $\Sigma_k^P$  there are problems in  $\Sigma_k^P$  which are  $MSOL$ -definable and which are complete for  $\Sigma_k^P$  via polynomial reductions.*

The upper bound, namely that the complexity of checking  $MSOL$ -properties is in the polynomial hierarchy, follows from classical theorems due to Fagin [51] and Stockmeyer [153], cf. [70].

### 1.7. Algorithmic applications IV: graph polynomials

A graph polynomial is mapping

$$p : \text{Graphs} \rightarrow \mathcal{R}[X],$$

where  $\mathcal{R}$  is a commutative ring and  $X$  is a, possibly countably infinite, set of indeterminates. Furthermore,  $p$  has to be *invariant under graph isomorphisms*. In most cases  $\mathcal{R} = \mathbb{Z}$ . The polynomial can often be written as

$$p(G) = \sum_{\alpha} \prod_{\beta} m_{\alpha, \beta},$$

where  $m_{\alpha, \beta}$  is a term depending on  $\alpha$  and  $\beta$ ,  $\alpha$  ranges often over subgraphs  $H$  of  $G$  and  $\beta$  depends on  $\alpha$ . We shall see many examples in Appendix A.

In our final development we shall prove a Feferman–Vaught theorem for graph polynomials where the summation ranges over  $MSOL$ -definable sets of subgraphs. Here we exploit the following analogy between  $MSOL$ -properties  $\phi$  and the polynomial  $p$ .  $\phi$  is also invariant under graph isomorphisms and takes values in the ring  $\mathbb{Z}_2$  with two elements. Viewed like this, the boolean function of the reduction sequences of

Theorem 1.6 is a polynomial over  $\mathbb{Z}_2[X]$  where  $X$  is a set of indeterminates. In the case of graph polynomials one observes that many are multiplicative, i.e. if  $G = G_1 \sqcup G_2$  is the disjoint union of two graphs, then

$$p(G) = p(G_1) \cdot p(G_2).$$

The product is a special case of a two variable polynomial  $g(x, y) \in \mathcal{R}[x, y]$ . Furthermore, in some cases, if  $G = G_1 \cup_e G_2$ , where  $G_1$  and  $G_2$  have exactly one edge  $e$  in common, one can find graph polynomials  $p_1, p_2$ , and a polynomial  $g \in \mathcal{R}[x_1, x_2, x_3, x_4]$  such that

$$p(G) = g(p_1(G_1), p_2(G_1), p_1(G_2), p_2(G_2)),$$

which looks very much like a reduction sequence for graph polynomials. Our last result (Theorem 6.4) will make precise how the Feferman–Vaught Theorem can be generalized to graph polynomials. Although graph polynomials are in general hard to compute,  $\#P$ -hard to be precise, we shall see that for graphs of bounded tree width (clique width) many of them can be computed in polynomial time. These results extend previous work presented in [34,107–109], and generalize the results of Andrejak [2], Noble [123,124] and Mighton [117].

### 1.8. Outline of the paper

In Section 2 we introduce the formalism of *translation schemes* and its associated maps, the semantic transductions and syntactic translations. In its simplest forms, translation schemes can be motivated by viewing them as a special case of the Feferman–Vaught Theorem for the case where products or sums are taken over the same structure. But they also hold the key to further generalizations, both of the Feferman–Vaught Theorem as well as a strong tool of interpretability. Translation schemes unify formalisms due to Tarski, Mostowski and Robinson and considerably clarified by Rabin, cf. [128,161]. They allow us to recast the original Feferman–Vaught Theorem in its suitable generality.

In Section 3 we introduce a unary map between structures, the *Fusion*, which contracts the satisfaction set of a unary predicate to a single point. We show that this operation satisfies a Feferman–Vaught Theorem similar to Theorem 1.3. In its full generality the fusion operation is not known to be expressible as an *FOL*-transduction. However, several special cases can be expressed in this way.

In Section 4 we present our formalism of *MSOL*-inductive classes of structures. These are classes of structures built inductively using the disjoint union, quantifier free transductions, and the fusion operation in various forms. They generalize the classes of graphs of bounded tree width, bounded clique width, comprise many graph grammars, and extend to more general relational structures. Also in this section, we show how to construct lookup tables for reduction sequences associated with disjoint unions, quantifier free transductions and fusion, and how to use these lookup tables for linear time model checking of structures represented as parse trees of inductive classes.

In Section 5 we briefly discuss the already mentioned decidability results, and how they can be proved using this general framework.

In Section 6 we introduce graph polynomials and define the class of *MSOL*-definable graph polynomials. We formulate and prove a Feferman–Vaught Theorem for *MSOL*-definable graph polynomials, generalize known and generate new splitting theorems for graph polynomials, and how these splitting theorems can be used for calculations.

In Section 7 we discuss extensions of *MSOL* for which the Feferman–Vaught Theorem holds as well. Such extensions are abundant, even if we restrict our attention to finite structures. Courcelle has shown that the Feferman–Vaught Theorem holds if *MSOL* is augmented by countably many counting quantifiers  $C_{k,m}$  which say that a set has size  $k$  modulo  $m$ . We discuss, without a conclusive answer, the question, whether there is a maximal logic which satisfies the Feferman–Vaught Theorem. However, we think that it could be made precise that no such logic exists which is finitely generated.

In Appendix A, we discuss graph polynomials and we review the literature for examples of graph polynomials.

## 2. Translation schemes and transductions

There is a rich literature on various aspect of translation schemes. They are often rediscovered and named differently. They play important rôles in various branches of logic, which do not concern us here. But we would like to mention the booklet by Mycielski et al. [122], where a general theory of *logical (first order) theories* is developed, which is based on various uses of translation schemes. Here we look at translation schemes is a tool to analyze a Feferman–Vaught Theorem for *unary* operations.

### 2.1. Powers and disjoint unions of the same structure

Let us look for a moment at a special case of the cartesian product  $\mathfrak{A} \times \mathfrak{A}$  and the disjoint union  $\mathfrak{A} \sqcup \mathfrak{A}$  of two copies of one  $\tau$ -structure  $\mathfrak{A}$ , or in other words, we look at the cartesian product and the disjoint union of two copies of the same structure  $\mathfrak{A}$  as a unary operation performed on  $\mathfrak{A}$ . We want to check whether for a given *FOL*( $\tau$ )-sentence  $\phi$ ,  $\mathfrak{A} \times \mathfrak{A} \models \phi$  and  $\mathfrak{A} \sqcup \mathfrak{A} \models \phi$ . For both cases, Theorem 1.4 says that there are formulas  $\psi_1, \dots, \psi_k$  and  $\theta_1, \dots, \theta_m$  and boolean functions  $B_\times : 2^k \rightarrow 2$  and  $B_\sqcup : 2^m \rightarrow 2$  such that

$$\mathfrak{A} \times \mathfrak{A} \models \phi \text{ iff } B_\times(a_1, \dots, a_k) = 1$$

and

$$\mathfrak{A} \sqcup \mathfrak{A} \models \phi \text{ iff } B_\sqcup(b_1, \dots, b_m) = 1.$$

if we put  $a_i = 1$  iff  $\mathfrak{A} \models \psi_i$  and  $b_i = 1$  iff  $\mathfrak{A} \models \theta_i$ . But as  $B_\times$  and  $B_\sqcup$  depend only on the truth values of the  $\psi_i$  and  $\theta_i$  in  $\mathfrak{A}$ , there are formulas  $\Psi$  and  $\Theta$  such that

$$\mathfrak{A} \times \mathfrak{A} \models \phi \text{ iff } \mathfrak{A} \models \Psi$$

and

$$\mathfrak{A} \sqcup \mathfrak{A} \models \phi \text{ iff } \mathfrak{A} \models \Theta.$$

In other words the reduction sequence consists of one formula, and the boolean function can be incorporated into the formulas of the trivial reduction sequence. Both, the structures  $\mathfrak{A} \times \mathfrak{A}$  and  $\mathfrak{A} \sqcup \mathfrak{A}$  can be obtained from  $\mathfrak{A}$  by translation schemes, the product vectorized, and the disjoint union scalar but many-sorted. We shall explore this and make the notions precise in the sequel. We shall distinguish three related notions:

**Translation schemes.** A translation scheme  $\Phi$  consists just of a sequence of formulas which satisfies some formatting restrictions. Its meaning is given in two ways:

**Transductions.** A transduction  $\Phi^\star$  is a semantic map associated with a translation scheme  $\Phi$ .

**Translations.** A translation  $\Phi^\sharp$  is a syntactic map associated with a translation scheme  $\Phi$ .

## 2.2. Translation schemes

**Definition 2.1** (Translation schemes  $\Phi$ ).

- (i) Let  $\tau$  and  $\sigma = \{R_1, \dots, R_m\}$  be two vocabularies with  $\rho(R_i)$  be the arity of  $R_i$ . Let  $\mathcal{L}$  be a fragment of *SOL*, such as *FOL*, *MSOL*,  $\exists$ *MSOL*, etc. Let  $\Phi = \langle \phi, \psi_1, \dots, \psi_m \rangle$  be formulas of  $\mathcal{L}(\tau)$  such that  $\phi$  has exactly  $k$  distinct free first order variables and each  $\psi_i$  has  $k \cdot \rho(R_i)$  distinct free first order variables. In this situation we say that  $\Phi$  is *k-feasible* (for  $\sigma$  over  $\tau$ ).
- (ii) A *k-feasible*  $\Phi = \langle \phi, \psi_1, \dots, \psi_m \rangle$  is called a *k –  $\tau$  –  $\sigma$  –  $\mathcal{L}$ -translation scheme* or, in short, a *translation scheme*, if the parameters are clear in the context.
- (iii) If  $k = 1$  we speak of *scalar* or *non-vectorized* translation schemes.
- (iv) If  $k \geq 2$  we speak of *vectorized* translation schemes.
- (v) If  $\phi$  is such that  $\forall \bar{x} \phi(\bar{x})$  is a tautology (always true) the translation scheme is *not relativized* otherwise it is *relativized*.
- (vi) A translation scheme is *many-sorted* if  $\sigma$  is a many-sorted vocabulary. In this case there are defining formulas for each of the universes corresponding to the sorts.  
This generalizes Courcelle’s transduction which allow the disjoint union of a fixed number of copies of a structure, cf. [28].
- (vii) A translation scheme is *parametrized* (with parameters  $\bar{x}, \bar{U}$ ) if the formulas of  $\Phi$  all contain  $\bar{x}$  and  $\bar{U}$  as additional free variables.
- (viii) A translation scheme is *simple* if it is neither relativized nor vectorized. It is *many-sorted* if  $\sigma$  is a many-sorted vocabulary. In this case the domains are disjoint.

**Examples 1** (Products and sums). Let  $\tau = \{R\}$  consist of one binary relation symbol.

- (i) The  $2-\tau-\tau$ -*FOL*-translation scheme  $\Phi_{\text{power}}$  is given by  $\phi(x, y) =_{\text{def}} (x = x \wedge y = y)$ , and  $\psi(x_1, x_2, y_1, y_2) =_{\text{def}} R(x_1, y_1) \wedge R(x_2, y_2)$ .  $\Phi_{\text{power}}$  is vectorized.
- (ii) Let  $\sigma$  be like  $\tau$ , but two-sorted with sorts  $U_1$  and  $U_2$  (treated like unary predicates but with disjoint domains). The  $2-\tau-\sigma$ -*FOL*-translation scheme  $\Phi_{\text{dunion}}$  is given by  $\phi_{U_1}(x) =_{\text{def}} U_1(x)$ ,  $\phi_{U_2}(x) =_{\text{def}} U_2(x)$ , and  $\psi(x, y) =_{\text{def}} ((R(x, y) \wedge U_1(x) \wedge U_1(y)) \vee (R(x, y) \wedge U_2(x) \wedge U_2(y)))$ .  $\Phi_{\text{dunion}}$  is scalar but two-sorted.

**Examples 2** (From words to graphs).

- (i)  $\tau_{\text{words}_3}$  consists of  $\{R_<, P_0, P_1, P_2\}$  for three letters  $\{0, 1, 2\}$ .  $\tau_{\text{graphs}}$  consists of  $\{E\}$ . We put  $\phi_1(x) = (P_0(x) \vee P_1(x))$  and  $\psi_E(x, y) = (P_0(x) \wedge P_1(y))$ .

$$\Phi_1 = \langle \phi_1(x), \psi_E(x, y) \rangle$$

is a *scalar* (as  $k = 1$ ) and *relativized* translation scheme in *FOL*. If instead we look at  $\phi_2(x) = (x \approx x)$  then

$$\Phi_2 = \langle \phi_2(x), \psi_E(x, y) \rangle$$

is a *simple* translation scheme.

- (ii)  $\tau_{\text{words}_2}$  consists of  $\{R_<, P_0, P_1\}$ .  $\tau_{\text{grids}}$  consists of  $\{E_{NS}, E_{EW}\}$ , the North–South, and East–West relations. Put  $k = 2$ ,  $\phi(x) = ((x \approx x) \wedge (y \approx y))$ ,  $\psi_{E_{NS}}(x_1, x_2, y_1, y_2) = (R_<(x_1, x_2) \wedge y_1 \approx y_2)$  and  $\psi_{E_{EW}}(x_1, x_2, y_1, y_2) = (R_<(y_1, y_2) \wedge x_1 \approx x_2)$ .

$$\Phi_3 = \langle \phi(x, y), \psi_{E_{NS}}(x_1, x_2, y_1, y_2), \psi_{E_{EW}}(x_1, x_2, y_1, y_2) \rangle$$

is a *vectorized* but *not* relativized translation scheme in *FOL*.

### 2.3. Transductions

We now define the semantic map associated with  $\Phi$ , the transductions.

**Definition 2.2** (The induced transduction  $\Phi^\star$ ). Given a translation scheme  $\Phi$ , the function  $\Phi^\star : \text{Str}(\tau) \rightarrow \text{Str}(\sigma)$  is a *partial* function from  $\tau$ -structures to  $\sigma$ -structures defined by  $\Phi^\star(\mathcal{A}) = \mathcal{A}_\Phi$  and

- (i) the universe of  $\mathcal{A}_\Phi$  is the set

$$A_\Phi = \{\bar{a} \in A^k : \mathcal{A} \models \phi(\bar{a})\}$$

- (ii) the interpretation of  $R_i$  in  $\mathcal{A}_\Phi$  is the set

$$\mathcal{A}_\Phi(R_i) = \{\bar{a} \in A_\Phi^{\rho(R_i)-k} : \mathcal{A} \models \psi_i(\bar{a})\}.$$

$\mathcal{A}_\Phi$  is a  $\sigma$ -structure of cardinality at most  $|A|^k$ .

As  $\Phi$  is  $k$ -feasible for  $\sigma$  over  $\tau$ ,  $\Phi^\star(\mathcal{A})$  is defined iff  $\mathcal{A} \models \exists \bar{x} \phi$ , i.e., the universe defined by  $\phi$  is not empty.



In the case of parametrized translation schemes  $\Phi^\star(\mathcal{A})$  depends on the value of the parameters.

**Examples 3.**

- (i) Computing  $\Phi_{\text{prod}}^\star$  for  $\mathfrak{A}$  gives  $\mathfrak{A}^2$ .
- (ii) Computing  $\Phi_{\text{dunion}}^\star$  for  $\mathfrak{A}$  gives  $\mathfrak{A} \sqcup \mathfrak{A}$ .
- (iii) Computing  $\Phi_1^\star$  for the word 1001020102001022111 gives a complete bipartite graph, the vertices of which are the positions of the word which have either letter 0 or 1, and each 0-position is connected to every 1-position.
- (iv) Computing  $\Phi_3^\star$  for the word 0110101001 gives a  $(10 \times 10)$  rectangular grid. Note that this is independent of the letters  $\{P_0, P_1\}$ .

*2.4. Translations*

Next we define the syntactic map associated with  $\Phi$ , the translations. Translations are sometimes also called interpretations, especially when theories and provability are the subject of discourse.

**Definition 2.3** (The induced translation  $\Phi^\sharp$ ). Given a translation scheme  $\Phi$  we define a function  $\Phi^\sharp: \mathcal{L}(\sigma) \rightarrow \mathcal{L}(\tau)$  from  $\mathcal{L}(\sigma)$ -formulas to  $\mathcal{L}(\tau)$ -formulas inductively as follows:

- (i) For  $R_i \in \sigma$  and  $\theta = R_i(x_1, \dots, x_m)$  let  $x_{j,h}$  be new variables with  $j \leq m$  and  $h \leq k$  and denote by  $\bar{x}_i = \langle x_{i,1}, \dots, x_{i,k} \rangle$ . We put

$$\Phi^\sharp(\theta) = \left( \psi_i(\bar{x}_1, \dots, \bar{x}_m) \wedge \bigwedge_i \phi(\bar{x}_i) \right)$$

- (ii) This also works for equality and relation variables  $U$  instead of relation symbols  $R$ .
- (iii) For the boolean connectives, the translation distributes, i.e.

(iii.a) if  $\theta = (\theta_1 \vee \theta_2)$  then

$$\Phi^\sharp(\theta) = (\Phi^\sharp(\theta_1) \vee \Phi^\sharp(\theta_2))$$

(iii.b) if  $\theta = \neg\theta_1$  then

$$\Phi^\sharp(\theta) = \Phi^\sharp(\neg\theta_1)$$

(iii.c) similarly for  $\wedge$  and  $\rightarrow$ .

- (iv) For the existential quantifier, we use relativization to  $\phi$ :  
If  $\theta = \exists y\theta_1$ , let  $\bar{y} = \langle y_1, \dots, y_k \rangle$  be new variables. We put

$$\theta_\phi = \exists \bar{y}(\phi(\bar{y}) \wedge (\theta_1)_\phi).$$

This concludes the inductive definition for first order logic *FOL*.

- (v) For second order quantification of variables  $U$  of arity  $\ell$  and  $\bar{a}$  a vector of length  $\ell$  of first order variables or constants, we translate  $U(\bar{a})$  by treating  $U$  as a relation

symbol above and put

$$\theta_\Phi = \exists V(\forall \bar{v}(V(\bar{v}) \rightarrow (\phi(v_1) \wedge \dots \wedge \phi(v_l) \wedge (\theta_1)_\Phi)))$$

- (vi) For parametrized translation schemes  $\Phi$  with parameters  $\bar{x}, \bar{U}$ , we first compute  $\theta_{\Phi(\bar{x}, \bar{U})}$  by treating the parameters as uninterpreted constants, and then quantify the variables to get

$$\theta_\Phi = \exists \bar{x} \exists \bar{U} \theta_{\Phi(\bar{x}, \bar{U})}.$$

This gives immediately

**Proposition 2.4.** *Let  $\Phi$  be a  $k - \tau - \sigma$  translation scheme which is either in FOL or in MSOL.*

- (i) *If  $\Phi$  is in MSOL and scalar and  $\theta$  is in MSOL then  $\Phi^\#(\theta)$  is in MSOL.*  
(ii) *If  $\Phi$  is of quantifier rank  $q$  and has  $p$  parameters, and  $\theta$  is a  $\sigma$ -formula of quantifier rank  $r$ , then the quantifier rank of  $\Phi^\#(\theta)$  is bounded by  $r + k \cdot q + p$ .*

**Example 2.5.** We now look again at the examples of Section 2.2 and defined in detail in Examples 1. The reduction formulas for the product  $\Psi$  can be obtained as  $\Phi_{\text{prod}}^\#(\phi)$  and the reduction formulas for the disjoint union  $\Theta$  can be obtained as  $\Phi_{\text{dunion}}^\#(\phi)$ . If  $\phi \in \text{MSOL}$ ,  $\Phi_{\text{prod}}^\#(\phi)$  is not necessarily in MSOL, because  $\Phi_{\text{prod}}$  is vectorized. However,  $\Phi_{\text{dunion}}^\#(\phi)$  is in MSOL. In both cases the quantifier rank does not increase.

## 2.5. Properties of translation schemes

Translations schemes are useful tools providing us with various reduction techniques. Without further assumptions we have the following theorem:

**Theorem 2.6** (Fundamental property). *Let  $\Phi = \langle \phi, \psi_1, \dots, \psi_m \rangle$  be a  $k - (\tau - \sigma)$ -translation scheme in MSOL. Then the transduction  $\Phi^\star$  and the translation  $\Phi^\#$  are linked as follows: Given a  $\tau$ -structure  $\mathfrak{A}$  such that  $\Phi^\star(\mathfrak{A})$  is defined, an  $\mathcal{L}(\sigma)$ -formula  $\theta(\bar{x})$  with free variables  $\bar{x} = x_1, \dots, x_m$ , and an assignment  $z: \text{Variables} \rightarrow A$ , then*

$$\mathfrak{A} \models \Phi^\#(\theta)(\bar{y}_1, \dots, \bar{y}_m) \text{ iff } \Phi^\star(\mathfrak{A}) \models \theta(\bar{x}),$$

where  $\bar{y}_i$  is the vector of variables corresponding to  $x_i$  in the computation of  $\Phi^\#$ .

From this, together with Proposition 2.4, we get

**Theorem 2.7.** *Let  $\Phi$  be a  $k - \tau - \sigma$  translation scheme of quantifier rank  $q$  with  $p$  parameters, and let  $\mathfrak{A}$  be a  $\tau$ -structure. Then  $\text{Th}_{\text{MSOL}}^m(\Phi^\star(\mathfrak{A}))$  depends only on  $\text{Th}_{\text{MSOL}}^{m+k \cdot q + p}(\mathfrak{A})$ .*

If the induced maps  $\Phi^\star$  and  $\Phi^\#$  have additional properties we get more.

**Definition 2.8** ( $\mathcal{L}$ -Reductions). Let  $\mathcal{L}$  be a logic, say *FOL*, *MSOL* or some fragment of *SOL*, and  $\Phi$  a  $(\tau_1 - \tau_2)$  translation scheme. Furthermore, let  $K_1, K_2$  be two classes of  $\tau_1$  and  $\tau_2$ -structures, respectively. We say

- (i)  $\Phi^\star$  is a *weak reduction* from  $K_1$  to  $K_2$  if for every  $\tau_1$ -structure  $\mathfrak{A}$  with  $\mathfrak{A} \in K_1$  we have  $\Phi^\star(\mathfrak{A}) \in K_2$ .
- (ii)  $\Phi^\star$  is a *reduction* from  $K_1$  to  $K_2$  if for every  $\tau_1$ -structure  $\mathfrak{A}$ ,  $\mathfrak{A} \in K_1$  iff  $\Phi^\star(\mathfrak{A}) \in K_2$ .
- (iii)  $\Phi^\star$  from  $K_1$  to  $K_2$  is *onto* if (additionally) for every  $\mathfrak{B} \in K_2$  there is an  $\mathfrak{A} \in K_1$  with  $\Phi^\star(\mathfrak{A})$  isomorphic to  $\mathfrak{B}$ .
- (iv) By abuse of language we say  $\Phi^\star$  is a *translation from  $K_1$  onto  $K_2$*  also if  $\Phi^\star$  is not a weak reduction but only  $K_2 \subseteq \Phi^\star(K_1)$ .
- (v) We say that  $\Phi$  induces a reduction (a weak reduction) from  $K_1$  to  $K_2$ , if  $\Phi^\star$  is a reduction (a weak reduction) from  $K_1$  to  $K_2$ . For simplicity, we also say  $\Phi$  is a reduction (a weak reduction) instead of saying that  $\Phi$  induces a reduction (a weak reduction).

*2.6. Feferman–Vaught Theorem for translation schemes and generalized sums and products*

Proposition 2.4 and Theorem 2.6, gives us analogues to Theorems 1.5 and 1.6 for translation schemes  $\Phi$  where all the formulas of  $\Phi$  are quantifier free. We call those shortly *quantifier free translation schemes*. We note that the cartesian power  $\mathfrak{A}^k$  of  $k$  copies of a  $\tau$ -structure  $\mathfrak{A}$ , and the  $k$ -fold disjoint union of  $k$  copies of a  $\tau$ -structure  $\mathfrak{A}$  are operations which can be obtained as  $\Phi^\star$  for suitably chosen translation schemes  $\Phi$  which are quantifier free. Other such examples are given in Examples 2.

**Theorem 2.9.** *Let  $\Phi$  be a quantifier free  $\tau_1 - \tau_2$ -translation scheme in *SOL*. Let  $\mathfrak{A}$  and  $\mathfrak{B}$  two  $\tau_1$ -structures with  $Th_{SOL}^q(\mathfrak{A}) = Th_{SOL}^q(\mathfrak{B})$ . Then*

- (i)  $Th_{SOL}^q(\Phi^\star(\mathfrak{A})) = Th_{SOL}^q(\Phi^\star(\mathfrak{B}))$  and for each  $\theta \in SOL(\tau_2)$   $\Phi^\sharp(\theta)$  is a reduction sequence of length one for  $\theta$ .
- (ii) If each formula of  $\Phi$  has quantifier rank at most  $k$  and  $p$  parameters, the same holds with the stronger hypothesis  $Th_{SOL}^{q+k+p}(\mathfrak{A}) = Th_{SOL}^{q+k+p}(\mathfrak{B})$ .
- (iii) The same holds if we replace *SOL* by *FOL* or *MSOL*.

We combine this now with cartesian products and disjoint unions.

**Definition 2.10** (Product-like and sum-like operations). Let  $\mathcal{F}$  be a  $k$ -ary operation

$$\mathcal{F} : Str(\tau_1) \times \cdots \times Str(\tau_k) \rightarrow Str(\tau).$$

$\mathcal{F}$  is

- (i) *product-like* if there is a quantifier free *FOL*-translation scheme  $\Phi$  such that for all structures (over the appropriate vocabularies) we have

$$\mathcal{F}(\mathfrak{A}_1, \dots, \mathfrak{A}_k) = \Phi^\star(\mathfrak{A}_1 \times \cdots \times \mathfrak{A}_k)$$

- (ii) *sum-like* if there is a quantifier free *MSOL*-translation scheme  $\Phi$  such that for all structures (over the appropriate vocabularies) we have

$$\mathcal{F}(\mathfrak{A}_1, \dots, \mathfrak{A}_k) = \Phi^*(\mathfrak{A}_1 \sqcup \dots \sqcup \mathfrak{A}_k)$$

We note that it is convenient to regard  $\mathfrak{A}_1 \times \dots \times \mathfrak{A}_k$  and  $\mathfrak{A}_1 \sqcup \dots \sqcup \mathfrak{A}_k$  as *ordered*, i.e., non-commutative.

The finite generalized products in the original paper of Feferman and Vaught are defined like our product-like operations, but not necessarily with quantifier free translation schemes. It is now clear how to formulate Theorems 1.5 and 1.6 for product-like and sum-like operations. The restriction to *FOL* (*MSOL*) in the case of product-like (sum-like) operations stems from the restrictions on the validity of these theorems. Sum-like and product-like operations were first introduced in this way in [112].

### 2.7. Preservation of decidability under translation schemes

It is well known that the consequence relation for *FOL* is semi-computable (in old terminology recursive enumerable), but for *MSOL* it is undecidable. By Lindström's celebrated theorem, *FOL* is the only recursively presented fragment of *MSOL* which has the countable model property, [101,42] and has semi-computable consequence relation. When we study decidable theories, this difference is important. In *FOL* a complete finitely axiomatizable theory is always decidable, but in many fragments *MSOL* the theory of the arithmetic structure of the natural numbers is finitely axiomatizable but not decidable. However, there are many theories in fragments  $\mathcal{L}$  of *MSOL* which have a decidable theory. We have seen examples in Section 1.5.

For the sequel we fix some more terminology. A class of  $\tau$ -structures  $K$  is  $\mathcal{L}$ -closed if for every two  $\tau$ -structures  $\mathfrak{A}$  and  $\mathfrak{B}$  which satisfy the same  $\mathcal{L}$ -sentences, whenever  $\mathfrak{A} \in K$  then also  $\mathfrak{B} \in K$ . If  $K$  contains only finite structures, and  $\mathcal{L}$  contains all of *FOL*,  $K$  is  $\mathcal{L}$ -closed iff it is closed under  $\tau$ -isomorphisms.

$K$  is  $\mathcal{L}$ -decidable if the set of  $\mathcal{L}(\tau)$ -sentences  $Th_{\mathcal{L}}(K)$  true in all structures of  $K$  is computable (recursive). We write also  $Th(K)$  for  $Th_{\mathcal{L}}(K)$  if  $\mathcal{L}$  is understood from the context. Note that  $K$  is  $\mathcal{L}$ -closed iff  $\mathfrak{A} \in Th_{\mathcal{L}}(K)$  implies that  $\mathfrak{A} \in K$ . The converse is always true by definition.

Decidable theories were first studied systematically by Tarski in collaboration with Mostowski and Robinson in [161]. Translation schemes were first used explicitly by Rabin in [128] in his simplification of [161]. We present here in a nutshell the general results of [128] relating to decidability and definability.

**Theorem 2.11** (Tarski, Mostowski and Robinson [161] and Rabin [128]). *Let  $\mathcal{L}$  be a fragment of *MSOL*. Let  $K_1, K_2$  be  $\mathcal{L}$ -closed classes of  $\tau_1$  and  $\tau_2$ -structures respectively, and  $\Phi$  be a translation scheme in  $\mathcal{L}$  such that for every  $\theta \in \mathcal{L}(\tau_2)$  the formula  $\Phi^{\#}(\theta)$  is in  $\mathcal{L}(\tau_1)$ .*

- (i) *If  $\Phi^*$  is a weak reduction from  $K_1$  to  $K_2$  which is onto,  $K_1$  is decidable, then  $K_2$  is decidable.*

- (ii) Assume additionally  $\mathcal{L}$  has a recursive enumerable consequence relation. If  $\Phi^*$  is a transduction from  $K_1$  to  $K_2$  which is onto and  $K_1$  is decidable and  $K_2$  definable by a single  $\mathcal{L}$ -sentence, then  $K_2$  is decidable.
- (iii) Similarly, we can conclude that  $K_2$  is decidable in  $\text{TIME}(f)$ ,  $\text{SPACE}(g)$  for  $K_1$  decidable in  $\text{TIME}(f')$ ,  $\text{SPACE}(g')$  for suitable  $f, f'$  and  $g, g'$ , depending on the complexity of substitution in  $\mathcal{L}$ .
- (iv) The same results also hold for parametrized transformation schemes which re-define equality (even with second order variables).

**Proof.** (i) Let  $\theta \in \mathcal{L}(\tau_2)$ . We want to check whether  $\theta \in \text{Th}(K_2)$ .

First we check whether  $\Phi^\#(\theta) \in \text{Th}(K_1)$ . As  $\text{Th}(K_1)$  is computable, we get an answer.

If  $\Phi^\#(\theta) \in \text{Th}(K_1)$ , every  $\mathfrak{A} \in K_1$  satisfies  $\Phi^\#(\theta)$ . Hence for every  $\mathfrak{A} \in K_1$   $\Phi^*(\mathfrak{A}) \models \theta$ . But, as  $\Phi^*$  is onto  $\Phi^*(K_1) = K_2$ , hence every  $\mathfrak{B} \in K_2$  satisfies  $\theta$ , and we have that  $\theta \in \text{Th}(K_2)$ .

If  $\Phi^\#(\theta) \notin \text{Th}(K_1)$ , there is  $\mathfrak{A} \in K_1$  with  $\mathfrak{A} \models \neg\Phi^\#(\theta)$ . So we have also that  $\mathfrak{A} \models \Phi^\#(-\theta)$ , and by Theorem 2.6,  $\Phi^*(\mathfrak{A}) \models \neg\theta$ . As  $\Phi^*$  is a weak reduction,  $\Phi^*(\mathfrak{A}) \in K_2$ ,  $\theta \notin \text{Th}(K_2)$ .

(ii) Assume that  $K_2 = \text{Mod}(\alpha)$ . Now  $\theta \in \text{Th}(K_2)$  iff  $\alpha \rightarrow \theta$  is valid. As  $\mathcal{L}$  has an r.e. consequence relation,  $\text{Th}(K_2)$  is r.e.

On the other hand:  $\theta \notin \text{Th}(K_2)$  iff there is  $\mathfrak{B} \in K_2$  with  $\mathfrak{B} \models \neg\theta$  iff (here we use that  $\Phi^*$  is onto) there is  $\mathfrak{A} \in K_1$  with  $\mathfrak{A} \models \Phi^\#(-\theta)$  iff  $\Phi^\#(\theta) \notin \text{Th}(K_1)$ . This shows that  $\text{Th}(K_2)$  is co-r.e.

(iii) In both cases above we have reduced the problem of checking  $\theta \in \text{Th}(K_1)$  to checking  $\Phi^\#(\theta) \in \text{Th}(K_2)$ . But the cost of this transformation depends solely on the cost of computing  $\Phi^\#(\theta)$ .

(iv) We leave this as an exercise.  $\square$

Together with Theorem 1.10 Theorem 2.11 becomes a very powerful tool. We do pursue this further in Section 5. But we first have to prepare the grounds and present our approach to tree width and clique width of graphs and its generalizations.

### 3. The fusion operation

In Section 4 we present our model theoretic approach to tree width and clique width based on the Feferman–Vaught Theorem. To do this we study first the operation  $Fuse_P$ . Although  $Fuse_P$  can be defined using translation schemes, the translation scheme which defines it is not quantifier free. It seems that no quantifier free translation schemes exists which defines  $Fuse_P$  in its full generality, but nevertheless  $Fuse_P$  satisfies a Feferman–Vaught Theorem in the style of Theorem 1.5, provided we restrict our attention to vocabularies with only *unary* and *binary* relations, i.e.  $\tau = \{P_1, \dots, P_m, R_1, \dots, R_n\}$ .

#### 3.1. Fusion of a unary predicate

Let  $\mathfrak{A}$  be a  $\tau$ -structure with universe  $A$ . We denote the interpretations of the symbols in  $\tau$  by  $P_1^A, \dots, P_m^A, R_1^A, \dots, R_n^A$ , respectively.

**Definition 3.1.** Let  $\rho$  be the maximal arity of the relation symbols in  $\tau$ , and  $\mathfrak{A}$  be a  $\tau$ -structure. Assume  $P_i^A \neq \emptyset$ . The structure  $Fuse_i(\mathfrak{A}) = \mathfrak{B}$  is defined as follows

- (i)  $B = (A - P_i^A) \cup \{p_i\}$  where  $p_i$  is an element not in  $A$ .
- (ii)  $P_i^B = \{p_i\}$
- (iii) For  $j \neq i$   $P_j^B = (P_j^A \cap B) \cup \{p_i\}$  if  $P_i^A \cap P_j^A \neq \emptyset$ .  
Otherwise,  $P_j^B = P_j^A \cap B = P_j^A$ .
- (iv) For binary relation symbols  $R_j$ ,  
 $R_j^B = (R_j^A \cap B^2) \cup \{(b, p_i) : b \in B \wedge \exists x \in P_i^A \text{ with } (b, x) \in R_j^A\}$   
 $\cup \{(p_i, b) : b \in B \wedge \exists x \in P_i^A \text{ with } (x, b) \in R_j^A\}$
- (v) For  $r$ -ary relation symbols  $R_j$ ,  
 $R_j^B = (R_j^A \cap B^r) \cup \bigcup_{I \subseteq \{1, \dots, r\}} X_I$ , where  
 $X_I = \{vec_I(\vec{b}, \vec{p}_i) : \vec{b} \in B^{r-|I|} \wedge \exists \vec{x} \in (P_i^A)^{|I|} \text{ with } vec_I(\vec{b}, \vec{x}) \in R_j^A\}$ .  
Here  $vec_I(\vec{c}, \vec{d}) \in C^r$  is the shuffling of  $\vec{c} \in C^{r-|I|}$  and  $\vec{d} \in C^{|I|}$ , without changing the order of the coordinates of  $\vec{c}$  respectively  $\vec{d}$ .

If  $P_i^A = \emptyset$ ,  $Fuse_i(\mathfrak{A}) = \mathfrak{A}$ .

We note that this definition is of the form  $\Phi_{Fuse_i}^*$  for a many-sorted translation scheme of quantifier rank  $\rho - 1$ . Hence we have

**Proposition 3.2.**  $Th_{MSOL}^m(Fuse_{P_i}(\mathfrak{A}))$  depends only on  $Th_{MSOL}^{m+\rho-1}(\mathfrak{A})$ .

Furthermore,  $\theta \in Th_{MSOL}^m(Fuse_{P_i}(\mathfrak{A}))$  can be computed by checking whether  $\Phi_{Fuse_i}^\#(\theta) \in Th_{MSOL}^{m+\rho-1}(\mathfrak{A})$ .

It is a somewhat surprising fact that Proposition 3.2 can be improved if we have only unary and binary relation symbols.

**Proposition 3.3.** Assume  $\tau$  contains only unary and binary relation symbols. Then for every  $\tau$ -structure  $\mathfrak{A}$  the theory  $Th_{MSOL}^m(Fuse_{P_i}(\mathfrak{A}))$  depends only on the theory  $Th_{MSOL}^m(\mathfrak{A})$ .

**Proof.** Assume we have a winning strategy  $W$  of the Ehrenfeucht–Fraïssé games for  $Th_{MSOL}^m(\mathfrak{A})$  for  $m$  moves, as defined in [41,42,44]. We define a winning strategy  $W'$  for  $Th_{MSOL}^m(Fuse_{P_i}(\mathfrak{A}))$ . Assume  $k$  moves for  $W'$  have been defined.

- (i) Assume  $p_i$ , respectively  $p_i'$  was not yet chosen. If in the move  $(k + 1)$  player  $I$  chooses an element, say  $a \in A - P_i^A$ , player  $II$  answers with the same element as prescribed by  $W$  for move  $(k + 1)$ .
- (ii) If  $I$  chooses  $p_i$  or  $p_i'$  then  $II$  replies always with  $p_i'$ , respectively  $p_i$ , independently of the previous choices.
- (iii) Assume  $p_i$ , respectively  $p_i'$  was already chosen in a previous move and now it is move  $(k + 1)$ . If in the move  $(k + 1)$  player  $I$  chooses an element, say  $a \in A - P_i^A$ , player  $II$  answers with the same element as prescribed by  $W$  for move  $k$ , disregarding the choice of  $p_i$  and  $p_i'$ .

- (iv) For set moves  $U \subseteq A - P_i^A$  we again use  $W$ .
- (v) For set moves  $U$  containing  $p_i$ , respectively  $p'_i$  we split accordingly.

We have to show that  $W'$  is indeed a winning strategy. W.l.o.g. we can assume that  $p_i$  and  $p'_i$  were chosen only once and in fact in the last move. This is so, as  $W'$  is not affected by the choice of  $p_i$  and  $p'_i$ . The only way,  $W'$  could not be a partial isomorphism is that for some  $R_j$ ,  $a_k, a'_k$  chosen in move  $k$  we have  $(a_k, p_i) \in R_j^A$  but  $(a'_k, p'_i) \notin R_j^{A'}$ . But then we can show that  $W$  is not a winning strategy by choosing as the last move some  $a_m$  such that  $(a_k, a_m) \in R_j^A$  for which  $II$  has no reply, as no such  $a'_m$  exists.  $\square$

However, it is not clear how to compute (efficiently)  $Th_{MSOL}^m(Fuse_{P_i}(\mathfrak{A}))$  from  $Th_{MSOL}^m(\mathfrak{A})$ .

### 3.2. General vocabularies

We now discuss briefly the operation  $Fuse_{P_i}$  in the case that the vocabulary  $\tau$  contains finitely many relation symbols, at least one of which of arity  $\rho$ ,  $3 \leq \rho \in \mathbb{N}$ , but none of which of arity bigger than  $\rho$ . We also allow finitely many constant symbols. Obviously, Proposition 3.2 now becomes

**Proposition 3.4.**  $Th_{MSOL}^m(Fuse_{P_i}(\mathfrak{A}))$  depends only on  $Th_{MSOL}^{m+\rho-1}(\mathfrak{A})$ .

Furthermore, Proposition 3.3 can be improved, using the same type of argument as in the proof of Proposition 3.3, to the following proposition.

**Proposition 3.5.**  $Th_{MSOL}^m(Fuse_{P_i}(\mathfrak{A}))$  depends only on  $Th_{MSOL}^{m+\rho-2}(\mathfrak{A})$ .

For  $\rho = 2$  this is Proposition 3.3.

### 3.3. Fusions of constants

Now assume  $\tau$  contains at least two constant symbols  $a, b$ , and  $\sigma = \tau - \{b\}$ . We define the operation  $Fuse_{a=b}$  using a  $1 - \tau - \sigma - FOL$  quantifier free translation scheme  $\Phi_{Fuse_{a=b}}$  defined as follows

- (i) The universe is defined by the formula  $\phi(x) : x \neq b$ .
- (ii) For an  $n$ -ary relation symbol  $R$ , variables  $x_1, \dots, x_n$ , a term  $t$  and  $J \subseteq \{1, \dots, n\}$ , we denote by  $R_{J,t}(x_1, \dots, x_n)$  the formula obtained from  $R(x_1, \dots, x_n)$  by substituting  $t$  for each  $x_i$ ,  $i \in J$ .
- (iii) For every  $n$ -ary relation symbol  $R \in \tau$ ,  $\psi_R^J(x_1, \dots, x_n)$  is the formula

$$\left( R(x_1, \dots, x_n) \wedge \bigwedge_{i \in J} x_i \neq b \right) \vee \left( R_{J,b}(x_1, \dots, x_n) \wedge \bigwedge_{i \in J} x_i = a \right)$$

and

$$\psi_R(x_1, \dots, x_n) = \bigvee_{J \subseteq \{1, \dots, n\}} \psi_R^J(x_1, \dots, x_n).$$

Clearly, for every  $\tau$ -structure  $\mathfrak{A}$ ,  $\Phi_{Fuse_{a=b}}^\star(\mathfrak{A})$  is the  $\sigma$ -structure obtained from  $\mathfrak{A}$  by identifying  $a$  and  $b$ . Hence we get, using Theorem 2.7, the following:

**Proposition 3.6.**  *$Th_{MSOL}^m(Fuse_{a=b}(\mathfrak{A}))$  depends only on  $Th_{MSOL}^m(\mathfrak{A})$  and can be effectively computed.*

An alternative proof is that of [31, Lemma 5.2].

### 3.4. Sum-like operations and fuse

We have defined in Section 2.6, Definition 2.10 sum-like operations. As we do not know whether  $Fuse_P$  can be defined using a quantifier free translation scheme, it is not clear whether applying  $Fuse_P$  to a sum-like operation gives again a sum-like operation. In the next section we therefore introduce an extension of sum-like operations for which Feferman–Vaught Theorems still hold.

## 4. MSOL-classes of structures

### 4.1. MSOL-smooth operations on structures

Instead of the disjoint union one can prove Theorem 1.5 and 1.6 for the following operations on coloured (un)directed graphs:

- (i) Concatenation of words,  $v \circ w$ .
- (ii) Joining two trees at a new common root,  $T_1 \bullet T_2$ .
- (iii)  $H$ -sums of graphs:  
For  $i = 1, 2$  let  $G_i = \langle V(G_i), E(G_i) \rangle$  and  $V(G_1) \cap V(G_2) = V(H)$  and  $E(H) = E(G_1) \cap V(H)^2 = E(G_2) \cap V(H)^2$ . Then  $G = G_1 \oplus_H G_2$  is given by  $V(G) = V(G_1) \cup V(G_2)$  and  $E(G) = E(G_1) \cup E(G_2)$ .
- (iv)  $H$ -sums of edge and vertex coloured graphs are defined similarly.

Given two graphs  $G_1, G_2$  with distinguished induced subgraphs  $H_1, H_2$  which are isomorphic to  $H$  with isomorphisms  $h_1, h_2$ , the  $H$ -sum of  $G_1$  and  $G_2$  is an almost disjoint union of the two graphs where the intersection contains exactly  $H$  as induced subgraph (using the isomorphisms  $h_1$  and  $h_2$  to fix it).<sup>11</sup>

Both proofs of Theorem 1.5 and 1.6 for these operations, the one using Ehrenfeucht–Fraïssé games, and the one using reduction sequences, generalize. We shall take this

<sup>11</sup> Strictly speaking we should write  $G_1 \oplus_{H, h_1, h_2} G_2$ , but we shall drop the isomorphisms when there is no risk of confusion.



observation as a starting point for our further treatment. However we note that it is not clear whether  $H$ -sum of graphs are sum-like operations.

Instead of the previous operations we now allow any operation satisfying Theorem 1.5 or 1.6.

**Definition 4.1.**

- (i) A  $n$ -ary operation  $Op$  on  $\tau$ -structures is *MSOL-smooth* if for every  $q \in \mathbb{N}$  and every  $\mathfrak{A}_1, \mathfrak{A}_2, \dots, \mathfrak{A}_n$

$$Th_{MSOL}^q(Op(\mathfrak{A}_1, \dots, \mathfrak{A}_n))$$

depends only on  $Th_{MSOL}^q(\mathfrak{A}_i)$  for  $1 \leq i \leq n$ .

- (ii)  $Op$  is *effectively MSOL-smooth* if there is an algorithm which computes for every  $\phi \in MSOL(\tau)$  a *reduction sequence*, i.e. a sequence of formulas as described in Theorem 1.6.

Clearly, by the above, we have the following examples.

**Example 4.2.** The following are *MSOL-smooth* operations.

- (i) The disjoint union of structures, concatenation of words, root joining of trees,  $H$ -sums of graphs (and its generalization to arbitrary structures) are all effectively *MSOL-smooth* operations by proofs similar to the proof of Theorem 1.6.
- (ii) Quantifier free *MSOL-transductions* are effectively *MSOL-smooth* by Theorem 2.7.
- (iii) The fusion operation  $\mathbf{Fuse}_{a=b}$  was shown to be *MSOL-smooth* in the previous section, Proposition 3.6.
- (iv) The fusion operation  $\mathbf{Fuse}_P$  was shown to be *MSOL-smooth* in the previous section, Proposition 3.3, provided the relation symbols are all unary or binary.

However, parametrized quantifier free *MSOL-transductions* are not known to be *MSOL-smooth*, as Theorem 2.7 does not give us that much.

**Question 3.** *Are there more examples of MSOL-smooth operations? Are there MSOL-smooth operations which are not effectively MSOL-smooth?*

4.2. *MSOL-inductive classes*

*MSOL-inductive* classes of structures are similar to graph grammars. The exact relationship with graph grammars is studied in [31,72]. Mahajan and Peters, [104], look at a similar, but more restricted concept. Also [15, Chapter 11] is close in spirit to our notion of inductive classes.

**Definition 4.3.**

- (i) A class  $K$  of  $\tau$ -structures is *MSOL-inductive* if it is defined inductively using a finite set of *MSOL-smooth* operations as follows: we are given a finite set  $K_0$  of finite  $\tau$ -structures and a finite set  $\mathcal{O}$  of *MSOL-smooth* operations. Now,  $K$  is

defined as the smallest class of  $\tau$ -structures containing  $K_0$ , and which is closed under isomorphisms and the operations in  $\mathcal{O}$ .

- (ii)  $K$  is *effectively MSOL-inductive* if it is defined inductively using a finite set of effectively MSOL-smooth operations.

**Question 4.** *Are there MSOL-inductive classes  $K$  which are not effectively MSOL-inductive?*

The following are some examples of effectively MSOL-inductive classes of structures.

**Examples 4** (Tree width and clique width).

- (i) Words  $\Sigma^*$  are defined inductively by
- (i.a) the empty word is a word
  - (i.b) one letter words are words
  - (i.c) words are closed under concatenation
- (ii) Coloured trees (forests) are defined similarly:
- (ii.a) one leaf trees are trees
  - (ii.b) trees are closed under root joining
  - (ii.c) forests are closed under disjoint unions
- (iii) Series-parallel (*SP*) graphs are defined by<sup>12</sup>
- (iii.a) one edge graphs are *SP*.
  - (iii.b) *SP* graphs are closed under disjoint unions
  - (iii.c) *SP* graphs are closed under *H*-sums for all *H* with at most two vertices. Series-parallel graphs are exactly the graphs of tree width at most 2 defined below.
- (iv) Graphs of *tree width at most k*  $TW_k$  can be defined inductively by looking at vertex coloured graphs with at most  $k + 1$  colours:
- (iv.a) All graphs with at most  $k + 1$  vertices are in  $TW_k$ .
  - (iv.b)  $TW_k$  is closed under disjoint union.
  - (iv.c)  $TW_k$  is closed under renaming of colours.
  - (iv.d)  $TW_k$  is closed under *fusion*, i.e. for every coloured graph  $G \in TW_k$ , and for every unary predicate symbol representing a vertex colour  $P$ , also  $fuse_P(G) \in TW_k$ .
- This definition is not the standard definition given, say in [37], but is equivalent to it, cf. [31].
- (v) Cogroups are inductively defined as follows, cf. [15, Chapter 11.3].
- (v.a) One vertex graphs are cogroups.
  - (v.b) The disjoint union of two cogroups is a cogroup.
  - (v.c) For a graph  $G = \langle V, E \rangle$  the complement graph  $\bar{G} = \langle V, V^2 - D_V - E \rangle$  is a cogroup. Here  $D_V$  is the diagonal relation  $\{(v, v) \in V^2 : v \in V\}$ .

<sup>12</sup> In [15] an equivalent definition is given. In this definition a graph is series-parallel iff it is of tree-width at most 2. This should not be confused with 2-terminal series-parallel graphs, also discussed in [15], which sometimes are also called series-parallel. Under this definition, a graph is of tree-width at most 2 iff all its 2-connected components are 2-terminal series-parallel.

Cographs are exactly the graphs of clique width at most 2 defined below.

- (vi) Similarly, graphs of *clique width at most k*  $CW_k$  are defined inductively as follows, cf. [36]:
  - (vi.a) All graphs with at most 1 vertex are in  $CW_k$ .
  - (vi.b)  $CW_k$  is closed under disjoint union.
  - (vi.c)  $CW_k$  is closed under renaming of colours.
  - (vi.d)  $CW_k$  is closed under adding all possible edges between two sets of differently coloured vertices.

In [15, Chapter 11] series–parallel graphs, partial  $k$ -trees and cographs are discussed, but differently. Cographs are exactly the graphs without an induced  $P_4$ , i.e. path with four vertices (or three edges). Partial  $k$ -trees are exactly the graphs of tree width at most  $k$ . There also an inductive definition of threshold graphs is given which is effectively *MSLO*-inductive. Threshold graphs have also a definition using forbidden induced subgraphs. The forbidden graphs are  $P_4$ ,  $C_4$ , i.e. cycles of length 4, and pairs of  $K_2$ , i.e. two disjoint edges.

As we said in the introduction, graphs of tree width (or clique width) at most  $k$  are widely studied in the literature. But the unifying view which led to the above equivalent definitions is recent and was introduced in [34,36] and discussed in detail in [31]. The notions in the above examples are related as indicated in the following proposition.

**Proposition 4.4.**

- (i) *Words and trees are of tree width at most 1.*
- (ii) *Series parallel graphs are of tree width at most 2.*
- (iii) *The clique  $K_m$  is of tree width  $m - 1$  but of clique width 2.*
- (iv) *Every graph of tree width at most  $k$  is of clique width at most  $2^{k+1} + 1$ .*

(i)–(iii) follow from the definitions and (iv) is basically a sharp bound, cf. [25,36].

**4.3. Parse trees for MSOL-inductive classes**

Given an *MSOL*-inductive class  $K$  of graphs, we want to establish that a graph  $G$  belongs to  $K$ . One way of doing this is by exhibiting explicitly the way the graph was obtained inductively. In other words, we exhibit the construction tree or parse term of the graph  $G$ . We now have a closer look at this.

**Definition 4.5.** Given an (effectively) *MSOL*-inductive class of  $\tau$ -structures  $K$ , we define a *parse term*  $t_{\mathfrak{A}}$  of  $\mathfrak{A}$ , which, for a  $\tau$ -structure  $\mathfrak{A}$ , describes how it was obtained according to the inductive definition of  $K$ . The parse term is built from constants denoting the elements of  $K_0$ , and function symbols for each operation in  $\mathcal{O}$ .

Given a parse term  $t_{\mathfrak{A}}$ , it is clear how to construct the structure  $\mathfrak{A}$ . The converse asks for a given structure  $\mathfrak{A}$ , whether  $\mathfrak{A} \in K$ , i.e. whether there is a parse term  $t_{\mathfrak{A}}$ . What one would like to have is, that every structure  $\mathfrak{A} \in K$  has a parse term of size polynomial

in the size of  $\mathfrak{A}$ , where the degree of the polynomial depends only on  $K$ . There are various monotonicity conditions which can guarantee this. If at every construction step at least an edge or a vertex is added, and no vertices or edges are removed, this is the case. But weaker conditions suffice as well. We leave it here open, how to formulate a reasonable condition for the existence of polynomial size parse terms, and state it as an abstract property. We also do not have an example of an (effectively) *MSOL*-inductive class  $K$  where structures in  $K$  have necessarily an exponential size parse term. But we are quite sure that with some ingenuity such a class can be constructed.

**Definition 4.6.**

- (i) Let  $K$  be an (effectively) *MSOL*-inductive class given by  $K_0$  and a finite set of operations  $\mathcal{O}$ . We say that  $\mathcal{O}$ , and by abuse of notation  $K$ , has *small parse terms*, if there exists a polynomial  $p_K(n) \in \mathbb{Z}[x]$  such that for each  $\mathfrak{A} \in K$  there is a parse term (over  $\mathcal{O}$ ) of size  $p_K(|\mathfrak{A}|)$  representing  $\mathfrak{A}$ .
- (ii) Additionally, we say that  $K$  has *polynomial time evaluation of parse terms*, if for each parse term  $t_{\mathfrak{A}}$  over  $\mathcal{O}$  the structure  $\mathfrak{A}$  is computable in polynomial time in the size of the  $t$ .

It is easy to see that, if we only allow quantifier free scalar non-relativizing transductions and disjoint unions, we only get *MSOL*-inductive classes with small parse terms from which the structure is computable in polynomial time. Non-monotonicity can be caused by relativization and the fusion operation. The case of tree width at most  $k$  is interesting, as it involves, in our definition, the fusion operation. Nevertheless, it is effectively *MSOL*-inductive with small parse terms which can be evaluated in polynomial time. If we replace the unary predicates subject to fusion by named elements and use fusion of constants paired with disjoint unions, we can even achieve monotonicity.

**Proposition 4.7.** *Let  $K$  be an (effectively) *MSOL*-inductive class of structures with small parse terms which can be evaluated in polynomial time. Then membership in  $K$  is in **NP**.*

**Proof.** We can guess a parse term whose size is polynomial in the size of the structure. Then we construct the structure from the parse term, which can be done in polynomial time, and then we have to check whether the resulting structure is isomorphic to our candidate structure, which is again in **NP**.  $\square$

**Proposition 4.8.** *The following are *MSOL*-inductive classes recognizable in polynomial time:*

- (i) *The series-parallel graphs, the graphs of tree width at most 2.*
- (ii) *More generally, for every  $k$ , the class of graphs of tree width at most  $k$ .*
- (iii) *The cographs, i.e. the graphs of clique width at most 2.*
- (iv) *More generally, for  $k \leq 3$ , the class of graphs of clique width at most  $k$ .*

**Proof.** (i) is due to [3]. Linear time algorithms were designed later, cf. [4] for the best results.

(ii) is from [24]. For (iii) cf. [15], and (iv) is from [24].  $\square$

It remains a challenging open problem to determine the complexity of finding parse trees for clique width  $k \geq 4$ . Clearly, this is in **NP**.

**Problem 2.** Is there a  $k$  such that finding parse trees for clique width at most  $k$  is **NP**-complete?

Another natural question to ask is whether there are other *MSOL*-inductive classes with **NP**-complete recognition problem. A simple graph  $\langle V, E \rangle$  has (cyclic) bandwidth at most  $k$  if there is a linear ordering (cyclic ordering)  $E'$  of its vertices such that for every edge  $e = (v_1, v_2) \in E$  there is a  $E'$ -path of length at most  $k$  between  $v_1$  and  $v_2$ . Let  $B_k$  and  $CB_k$  be the class of simple graphs of bandwidth and cyclic bandwidth at most  $k$  respectively.

**Proposition 4.9** (Saxe [144] and Leung et al. [100]).  $B_k$  is in **P** but  $CB_2$  is **NP**-complete.

Turán in [164] has shown that an **NP**-complete modification  $\tilde{CB}_2$  of  $CB_2$  can be represented as an *NLC*-grammar. Given a graph  $G$  in  $CB_2$ , the modification adds two dangling edges (and hence two more vertices) to each original vertex of  $G$ . Analyzing his proof we get

**Proposition 4.10** (Turán [164]).  $\tilde{CB}_2$  is an *MSOL*-inductive class of bounded tree-width with polynomial time computable small parse terms where parsing is **NP**-complete.

#### 4.4. *MSOL*-inductive classes and graph grammars

Besides the examples given above, many graph grammars from the vast literature, cf. the handbooks [26,140–142], can be shown to be *MSOL*-inductive classes of labeled graphs. Actually, we view our approach to graph languages via *MSOL*-inductive classes as a model theoretic alternative to graph grammars. Production rules are replaced by closure conditions of *MSOL*-smooth operations.

The reader not at all acquainted with graph grammars should skip this section. Unfortunately the various notations and definitions in the literature vary confusingly and the notation carries too much information. For surveys on graph grammars, the reader may want to consult Habel's monograph [81] and the more updated and excellently written [47] by Engelfriet. More recently, Kim clarified the relative expressive power of various graph grammars, cf. [94,95].

The most studied graph grammars are the *HR*-grammars (Hyperedge replacement grammars). The production rules are all of the form

$$A_i \rightarrow H_i.$$

These grammars are context free (confluent). It is well known that

**Proposition 4.11.**

- (i) Every HR-language of graphs<sup>13</sup> has bounded tree width.
- (ii) The graph language  $TW_k$  is an HR-language.

Lautemann has explicitly described how to compute an upper bound to the tree width of an HR-language from the production rules, cf. [99].

**Proposition 4.12.** *Let  $K$  be an HR-language of graphs generated by an HR-grammar  $\Gamma$ . Let  $k$  be the maximal number of vertices in the right-hand side hypergraph of the production rules of  $\Gamma$ . Then the tree width of the graphs in  $K$  is bounded by  $k$ .*

Other graph grammars are the NCE-grammars (Neighborhood Controlled Embedding), which are all VR-grammars (Vertex replacement grammars) or, as they are called in [47], NR-grammars (Node replacement grammars). Here the production rules also specify an embedding  $emb_i$  and are of the form

$$A_i \rightarrow (H_i, emb_i).$$

There are three main cases we want to mention.

- (i) *A-NCE*-grammars (Apex NCE),  
where the embedding connects only terminals;
- (ii) *B-NCE*-grammars (Bounded NCE)  
where there are no edges between non-terminals; and
- (iii) *C-NCE*-grammars (Confluent NCE),  
the most general context-free VR-grammar.
- (iv) We have  $A\text{-NCE} \subseteq B\text{-NCE} \subseteq C\text{-NCE}$ .
- (v) If the edges of the graphs are labeled and directed we speak of *A-edNCE*, *B-edNCE* and *C-edNCE* grammars respectively. In model theoretic terms this extensions amounts to an extension of the vocabulary by unary predicates.

From [29,48,49] we get

**Proposition 4.13.** *Let  $K$  be a context free VR-language of graphs.*

- (i)  $K$  is *A-NCE* iff  $K$  is of bounded degree.
- (ii)  $K$  is *B-edNCE* with bounded non-terminal degree iff  $K$  has bounded tree width iff  $K$  is an HR-language.
- (iii)  $K$  is *C-edNCE* iff  $K$  has bounded clique width.

More generally we have

**Proposition 4.14.** *If  $K$  is a context free VR-language of graphs, then  $K$  is MSOL-inductive.*

<sup>13</sup> Strictly speaking, an HR-grammar  $\Gamma$  produces a class of hypergraphs. An HR-language of graphs refers to the case where the HR-grammar  $\Gamma$  generates a class of graphs.

**Proof.** [Sketch of proof:] One has to show that the production rules are *MSOL*-smooth operations on graphs. This is often verifiable by replacing the production rules by a composition of quantifier free transductions, disjoint unions and possibly fusions. However, many-sorted transductions are needed here, and the details are tedious.  $\square$

*NCE*-grammars which have a polynomial time parsing are studied by Flasiński in [65], in the case of *NLC*-grammars (Node label controlled grammars, a special case of *NCE*-grammars), and by Kim in [95] for general *NC*-grammars.

**Problem 3.** Characterize graph grammars and *MSOL*-inductive classes which do have polynomial time parsing.

In [72] Proposition 4.12 has been improved and extended to context sensitive grammars, the *NCE*-grammars and *edNCE*-grammars.<sup>14</sup> The main result there is

**Theorem 4.15.** *Let  $K$  be a graph language generated by an *NCE*-grammar. Then  $K$  is of bounded clique width and the bound can be computed from properties of the grammar.*

#### 4.5. Computing a look-up table and dynamic programming

Given a finite set of effectively *MSOL*-smooth operations  $\mathcal{O}$ , we want to exploit the reduction sequences for all the operations in  $\mathcal{O}$  for various computations. What we obtain is a finite look-up table which allows us to check properties of graphs in *MSOL*-inductive classes using the parse tree of the graph. This resembles a finite automaton, and Courcelle in [27] calls classes of graphs where properties can be checked in this way *recognizable classes of graphs*.

We shall stress here the model theoretic point of view and generalize previous results for graphs of bounded tree width and clique width. We note that, like in Theorem 1.7, both the automata theoretic and the model theoretic approach coexist. But we think that the model theoretic approach is more flexible and covers more cases.

We first determine the number of formulas in *FOL*( $\tau$ ) and *MSOL*( $\tau$ ) of fixed quantifier rank. Let  $\bar{x}$  and  $\bar{U}$  be fixed finite sequences of first order and second order variables respectively. For  $q \in \mathbb{N}$ , *FOL*<sup>*q*</sup>( $\tau, \bar{x}, \bar{U}$ ) and *MSOL*<sup>*q*</sup>( $\tau, \bar{x}, \bar{U}$ ) denote the formulas of *FOL* and *MSOL* respectively with all free variables among  $\bar{x}, \bar{U}$  and which are of quantifier rank at most *q*.

**Proposition 4.16.**  *$FOL^q(\tau, \bar{x}, \bar{U})$  and  $MSOL^q(\tau, \bar{x}, \bar{U})$  are finite of size*

$$|FOL^q(\tau, \bar{x}, \bar{U})| = \alpha_{FOL}(\tau, q, \bar{x}, \bar{U})$$

and

$$|MSOL^q(\tau, \bar{x}, \bar{U})| = \alpha_{MSOL}(\tau, q, \bar{x}, \bar{U})$$

respectively.

---

<sup>14</sup> *e* for edge labels and *d* for directed edges.

The proof is straightforward, but gives very large numbers. A good estimate of the numbers  $\alpha_{FOL}(\tau, q, \bar{x}, \bar{U})$  and  $\alpha_{MSOL}(\tau, q, \bar{x}, \bar{U})$  may be found in [96, Lemma 3.2]. The upper bound given there is at least an  $q$  times iterated exponential. The semantic difference between *FOL* and *MSOL* is lost in this count.

**Problem 4.** Improve the estimates on how many formulas there are in  $FOL^q(\tau, \bar{x}, \bar{U})$  or  $MSOL^q(\tau, \bar{x}, \bar{U})$  up to logical equivalence over all structures (equivalence over finite structures).

Exact computation is hopeless, as logical equivalence is undecidable. One may opt for a weaker form of equivalence, where formulas are, say, in prenex normal form, and two formulas are equivalent if they have the same quantifier prefix and the boolean parts are equivalent boolean function of atomic formulas.

In the sequel we restrict our discussion to *MSOL* and leave it to the reader to formulate the corresponding analogues for *FOL*.

Given a formula  $\phi$  of  $MSOL^q(\tau, \bar{x}, \bar{U})$ , and a fixed finite set  $\mathcal{O}$  of *MSOL*-smooth operations, we define inductively a set  $\mathcal{R}(\phi, \mathcal{O})$  as follows:

**Definition 4.17.** Let  $q$  be the quantifier rank of  $\phi$ .  $\mathcal{R}(\phi, \mathcal{O})$  is the smallest set of formulas in  $MSOL^q(\tau, \bar{x}, \bar{U})$  such that

- (i)  $\phi \in \mathcal{R}(\phi, \mathcal{O})$ ;
- (ii) If  $\psi \in \mathcal{R}(\phi, \mathcal{O})$ ,  $Op \in \mathcal{O}$  is an  $m$ -ary operation and

$$Red(\psi, Op) = \langle \psi_1^1, \dots, \psi_n^1, \dots, \psi_1^m, \dots, \psi_n^m \rangle$$

is a reduction sequence for  $Op$  and  $\psi$  then each  $\psi_i^j \in \mathcal{R}(\phi, \mathcal{O})$ .

$\mathcal{R}(\phi, \mathcal{O})$  is called the *reduction set* of  $\phi$  for  $\mathcal{O}$ .

**Observation 5.** For every  $\phi \in MSOL$  of quantifier rank  $k$  and every finite set of *MSOL*-smooth operations  $\mathcal{O}$ ,  $\mathcal{R}(\phi, \mathcal{O})$  is finite and consists of formulas of quantifier rank at most  $k$ . Furthermore, if the operations are effectively smooth,  $\mathcal{R}(\phi, \mathcal{O})$  is computable as a function of  $\phi$  and  $\mathcal{O}$ .

We now define a *look-up table* for  $\mathcal{R}(\phi, \mathcal{O})$  inductively as follows:

**Definition 4.18.** For a formula  $\psi \in \mathcal{R}(\psi, \mathcal{O})$  and  $Op \in \mathcal{O}$  we denote by  $B_{Op}(\psi)$  the boolean function and by  $Red(\psi, Op)$  the corresponding reduction sequence. The *look-up table*  $Look(\phi, \mathcal{O})$  consists of all the quadruples

$$(\psi, Op, B_{Op}(\psi), Red(\psi, Op))$$

with  $\psi \in \mathcal{R}(\psi, \mathcal{O})$  and  $Op \in \mathcal{O}$ .

From Theorem 1.6 we see immediately:



**Observation 6.** *For every  $\phi \in MSOL$  of quantifier rank  $k$  and every finite set of  $MSOL$ -smooth operations  $\mathcal{O}$ ,  $Look(\phi, \mathcal{O})$  is finite. Furthermore, if the operations are effectively  $MSOL$ -smooth,  $Look(\phi, \mathcal{O})$  is computable as a function of  $\phi$  and  $\mathcal{O}$ .*

#### 4.6. Model checking for $MSOL$ -inductive classes

Model checking is the problem to check whether

$$\mathfrak{A} \models \phi$$

is true for a finite  $\tau$ -structure  $\mathfrak{A}$  and sentence  $\phi$  in a fragment of Second Order Logic  $SOL(\tau)$ . The combined model checking problem takes both a structure and a formula as input. Other variants consider the model checking with fixed formula or fixed structure.

The fragment may be  $MSOL$  or some temporal logic not discussed in this paper, cf. [21]. We measure the problem in the size of  $\mathfrak{A}$  and  $\phi$  (combined case) or for specific  $\phi$ .

Model checking has found many applications in hardware and software verification and enormous efforts have been made successfully in turning this complex problem into an industrially feasible tool. The reader may consult the excellent book by Clarke et al. [21] and the literature cited therein. Here we only briefly discuss the complexity theoretic aspect of the problem for  $MSOL$ . But we predict that the systematic exploration of how to use Feferman–Vaught-type techniques in industrial model checking will add yet another tool to increase its feasibility. A first step was initiated in Ravve’s M.Sc. Thesis [133] and in [112]. Theorem 1.14 can be restated in this context as follows:

**Theorem 4.19** (Makowsky and Pnueli [111]). *For every level  $\Sigma_i^P$  in the polynomial hierarchy  $\mathbf{PH}$  there are vocabularies  $\tau$  and formulas  $\phi \in MSOL(\tau)$  such that checking  $\mathfrak{A} \models \phi$  (for  $\phi$  fixed) is complete for  $\Sigma_i^P$ .*

**Theorem 4.20** (Vardi [169]). *The combined problem is  $\mathbf{PSPACE}$ -complete even for  $FOL$ .*

We want to do model checking on  $MSOL$ -inductive classes  $K$ . We can represent the relational structure  $\mathfrak{A}$  by its *relation tables* or by a *parse term*  $t_{\mathfrak{A}}$  which serves as a certificate for  $\mathfrak{A} \in K$ . In general, by Proposition 4.10, finding  $t_{\mathfrak{A}}$  is  $\mathbf{NP}$ -hard.

**Theorem 4.21** (Courcelle and Makowsky [31]). *Let  $K$  be an  $MSOL$ -inductive class of  $\tau$ -structures given by a finite  $K_0$  set of  $\tau$ -structures and  $MSOL$ -smooth operations  $\mathcal{O}$ , and let  $\phi \in MSOL(\tau)$ . Given a parse term  $t_{\mathfrak{A}}$  for  $\mathfrak{A} \in K$  of size  $n_t$ , the problem of deciding, for fixed  $\phi$ ,*

$$\mathfrak{A} \models \phi$$

*can be solved in linear time in  $n_t$ .*

**Proof.** Using Observation 6 we use the look-up table  $Look(\phi, \mathcal{O})$ . We compute the truth values of the formulas in  $\mathcal{R}(\phi, \mathcal{O})$  on all the structures in  $K_0$ . This uses a constant

amount of time and does not depend on  $\mathfrak{A}$ . Now we use the boolean functions of the look-up table and  $t_{\mathfrak{A}}$  to compute the truth value of  $\phi$  in  $\mathfrak{A}$  bottom up. This uses  $O(n_t)$  time.  $\square$

Note that finding the parse term  $t_{\mathfrak{A}}$  for  $\mathfrak{A} \in K$  may be **PSpace**-hard already for  $K$  a graph language generated by an *NC*-grammar, cf. [94].

## 5. Decidable *MSOL*-theories

We have seen in the introduction, Theorems 1.9 and 1.10, that the *MSOL* theory of words and the *MSOL* theory of trees are decidable. We have also seen in Section 1.5 Seese's Theorem 1.11 which states that if an *MSOL*( $\tau_2$ )-theory of a class of graphs  $K$  is decidable, then  $K$  is of bounded tree width. In this section we want to explore this further.

To use the Feferman–Vaught Theorem for products and disjoint unions, we shall need some more classical decidability results:

**Theorem 5.1** (Classical).

- (i) *The FOL theory of infinite atomic boolean algebras is decidable.*
- (ii) *The MSOL theory of infinite sets is decidable.*
- (iii) *The FOL theory of atomic boolean algebras is decidable.*
- (iv) *The MSOL theory of sets is decidable.*

**Proof.** Our point here is to show how the general machinery allows to get (ii)–(iv) from (i).

- (i) is well known, cf. [119, Chapter 21, Theorem 21.34].
- (ii) is obtained from (i) by translation.
- (iii) is obtained from (i) by realizing that the theory of atomic boolean algebras has a recursive set of decidable completions, which by a criterion due to Ershov, [50] and [119, Chapter 15, Theorem 15.6], implies decidability.
- (iv), finally, is obtained from (iii) again by translations.  $\square$

For historic references the reader should consult [50]. Here our emphasis is on using (i) and model theoretic arguments to derive (ii)–(iv).

Using the Feferman–Vaught Theorem 1.4 for infinite products or disjoint unions, and Theorem 5.1, one can show the following:

**Theorem 5.2** (Classical). *Let  $K$  be class of  $\tau$ -structures and denote by  $\mathbf{P}(K)$  the closure of  $K$  under products,  $\mathbf{DU}(K)$  the closure of  $K$  under disjoint unions.*

- (i) *If  $K$  has a decidable FOL theory, so do  $\mathbf{P}(K)$  and  $\mathbf{DU}(K)$ .*
- (ii) *If  $K$  has a decidable MSOL theory, so does  $\mathbf{DU}(K)$ .*

**Proof.** We use the same methods as in [119, Chapter 23], who presents proofs for the corresponding results for products and *FOL*.  $\square$

As forest are just disjoint unions of trees we get:

**Corollary 5.3.** *The MSOL theory of forests  $\mathcal{F}$  is decidable.*

There is another way of showing that the *MSOL* theory of forests is decidable, using Proposition 2.11 for transductions, i.e. the Feferman–Vaught Theorem for unary operations. We define a scalar *MSOL*-reduction  $\Phi^\star$  from rooted trees  $\mathcal{RT}$  to forest  $\mathcal{F}$ , by cutting of the root of a tree. The new universe is defined as all the nodes different from the root. For every rooted tree this gives a forest, and every forest can be obtained in this way, Hence  $\Phi^\star : \mathcal{RT} \rightarrow \mathcal{F}$  is a weak reduction which is onto (surjective). Now the *MSOL* theory of rooted trees is also decidable, cf. Theorem 1.10. Hence, using Proposition 2.11, the *MSOL* theory of  $\mathcal{F}$  is decidable.

We can use Theorem 1.5 to show the following result by Seese [147]:

**Proposition 5.4** (Seese [147]). *The following theories are decidable:*

- (i) *The MSOL-theory of series–parallel graphs (SP-graphs).*
- (ii) *For each  $k \in \mathbb{N}$ , the MSOL-theory graphs of tree width at most  $k$ .*

**Proof.** One shows that in both cases the graphs can be obtained as a straightforward *MSOL*-transductions of certain labeled trees and then applies Proposition 2.11 and Theorem 1.10.  $\square$

In [31] this was, based on ideas from [30], generalized to

**Theorem 5.5** (Courcelle and Makowsky [31]). *Let  $K$  be a class of structures which is MSOL-inductive using disjoint unions, fusions and quantifier free MSOL-transductions. Then  $Th_{MSOL}(K)$  is decidable.*

**Proof.** [Proof idea:] One shows that an *MSOL*-inductive class  $K$  is always an *MSOL*-transduction of a class of labeled trees. But the proof of this fact is rather involved. Then one applies again Proposition 2.11 and Theorem 1.10.  $\square$

The same proof technique also gives

**Theorem 5.6** (Courcelle and Makowsky [31]). *Let  $K$  be a class of labeled graphs which is MSOL-inductive using disjoint unions, fusions and quantifier free MSOL-transductions. Then  $K$  is of bounded clique width.*

We can now reformulate the Engelfriet–Seese’s Conjecture 1 from the introduction as an open problem for finite relational structures. Instead of graphs or hypergraphs we look at relational structures. As clique width for relational structures has no standard definition, we replace it in the conclusion by being contained in an *MSOL*-inductive class.

**Problem 5** (Generalized Seese conjecture). Let  $K$  be a class of finite  $\tau$ -structures which has a decidable *MSOL*-theory. Is  $K$  contained in some class of  $\tau$ -structures  $K_1$  which is *MSOL*-inductive?<sup>15</sup>

As we noted before, cf. Question 3 in Section 4.1, it is not clear whether every *MSOL*-smooth operation is definable using disjoint unions, quantifier free transduction and fusions.

So here is another challenge:

**Problem 6.** Does every *MSOL*-inductive class of structures  $K$  have a decidable *MSOL*-theory?

## 6. A Feferman–Vaught theorem for graph polynomials

The results in this section are due to the author, continuing work initiated with his co-authors, Courcelle, Mariño and Rotics.

### 6.1. Graph polynomials

A graph polynomial is mapping

$$p : \text{Graphs} \rightarrow \mathcal{R}[X],$$

where  $\mathcal{R}$  is a commutative ring and  $X$  is a, possibly countably infinite, set of indeterminates. Furthermore,  $p$  has to be *invariant under graph isomorphisms*. Very often  $\mathcal{R} = \mathbb{Z}$ .

As  $p(G) \in \mathcal{R}[X]$ ,  $p$  is not one polynomial, but gives a polynomial for each graph  $G$  for which it is defined. Nevertheless, by abuse of notation, both  $p$  and  $p(G)$  are called graph polynomials. We write sometimes  $p(G; X)$  or  $p(G; x_1, \dots, x_m)$  if  $X = \{x_1, \dots, x_m\}$  when we want to indicate explicitly the indeterminates.

In Appendix A we give many examples of graph polynomials. Among the graph polynomials with a fixed finite set of indeterminates over  $\mathbb{Z}$  we find: The chromatic polynomial, the matching polynomials, and the Tutte polynomial. The Farrell polynomials and the coloured Tutte polynomial, discussed next, may have the number of indeterminates dependent on the graph  $G$ . Finally we shall see in Appendix A how the determinant, the permanent and the hamiltonian of a matrix over  $\mathbb{Z}_2$  can be viewed as graph polynomials. However, for matrices over other fields, none of these are, strictly speaking, graph polynomials. This is so, because the entries of the matrix may be different from 0 or 1 and hence the matrix can only be viewed as referring to a weighted graph.

<sup>15</sup> In Section 7 the logic *CMSOL* is discussed, which contains modular counting quantifiers, and for which the Feferman–Vaught Theorem was proved by Courcelle. It is reasonable to ask the same question for *CMSOL* instead for *MSOL*. This also concerns Problem 6 below.

### 6.2. Graph properties

A graph property is a class of graphs closed under graph isomorphisms. To stress an analogy between graph properties and graph polynomials, we can look at a graph property as a mapping

$$p : \text{Graphs} \rightarrow 2,$$

where 2 is the two-element boolean algebra. Furthermore,  $p$  has to be *invariant under graph isomorphisms*. Let  $\tau_{\text{graphs}}$  be a vocabulary of graphs. Clearly, a sentence  $\phi$  in  $\text{FOL}(\tau_{\text{graphs}})$  or  $\text{MSOL}(\tau_{\text{graphs}})$  defines a graph property  $p_\phi$  with

$$p_\phi(\mathfrak{A}) = 1 \text{ iff } \mathfrak{A} \models \phi.$$

Let us now reformulate the Feferman–Vaught Theorem (Theorem 1.6), in a way suggestive for generalization to graph polynomials.

**Theorem 6.1** (Feferman–Vaught for graph properties). *For every graph property  $p_\phi$  with  $\phi \in \text{FOL}^q(\tau)$  one can compute in polynomial time a sequence of graph properties*

$$p_1, p_2, \dots, p_m$$

associated with the formulas

$$\psi_1, \psi_2, \dots, \psi_m \in \text{FOL}^q(\tau)^m$$

and a boolean polynomial  $B_\phi \in [Y_1, Y_2, \dots, Y_m]$  such that

$$p(\mathfrak{A} \sqcup \mathfrak{B}) = 1 \text{ iff } B_\phi(p_1(\mathfrak{A}), \dots, p_m(\mathfrak{A}), p_1(\mathfrak{B}), \dots, p_m(\mathfrak{B})) = 1.$$

### 6.3. Splitting properties for $H$ -sums

Let  $G = \langle V(G), E(G) \rangle$  be a graph with an edge  $e = (v_1, v_2)$ . We denote by  $G - e$  the graph  $\langle V(G), E(G) - \{e\} \rangle$ , and by  $G/e$  the graph  $\text{Fuse}_{v_1, v_2}(G - e)$ .

In Section 4.1 we introduced the  $H$ -sum  $G_1 \sqcup_H G_2$  of two graphs  $G_1$  and  $G_2$  with respect to a shared induced subgraph  $H$ . This includes the disjoint union,  $G_1 \sqcup G_2$ , if  $H$  is empty. If  $H$  is a single vertex  $v$ , a single edge  $e$ , or consists of two not connected vertices  $v_1, v_2$  we write  $G_1 \sqcup_v G_2$ ,  $G_1 \sqcup_e G_2$ ,  $G_1 \sqcup_{v_1, v_2} G_2$  respectively. Many graph polynomials  $p$  satisfy the following property:

$$p(G_1 \sqcup G_2) = p(G_1) \cdot p(G_2) \tag{+}$$

or even

$$p(G_1 \sqcup_v G_2) = p(G_1) \cdot p(G_2). \tag{++}$$

The case of  $p(G_1 \sqcup_e G_2)$  was analyzed only for the Tutte polynomial by Oxley and Welsh in [127]. Abstractly formulated they found numbers  $\alpha, \alpha_{0,0}, \alpha_{0,1}, \alpha_{1,0}, \alpha_{1,1} \in \mathbb{Z}$  and

a polynomial  $g_e \in \mathbb{Z}[x_1, \dots, x_4]$  all independent of  $G_1$  and  $G_2$ , such that

$$\begin{aligned} \alpha \cdot \mathfrak{p}(G_1 \sqcup_e G_2) &= g_e(\mathfrak{p}(G_1 - e), \mathfrak{p}(G_2 - e), \mathfrak{p}(G_1/e), \mathfrak{p}(G_2/e)) \\ &= \alpha_{0,0} \cdot \mathfrak{p}(G_1 - e) \cdot \mathfrak{p}(G_2 - e) \\ &\quad + \alpha_{1,0} \cdot \mathfrak{p}(G_1/e) \cdot \mathfrak{p}(G_2 - e) \\ &\quad + \alpha_{0,1} \cdot \mathfrak{p}(G_1 - e) \cdot \mathfrak{p}(G_2/e) \\ &\quad + \alpha_{1,1} \cdot \mathfrak{p}(G_1/e) \cdot \mathfrak{p}(G_2/e). \end{aligned} \tag{+++}$$

A similar formula holds also for  $\mathfrak{p}(G_1 \sqcup_{v_1, v_2} G_2)$ . These formulas were extended for the Tutte polynomial to arbitrary  $H$ -sums in [123,124,2]. To us the formulas (+), (++) and (+++) suggest some similarity with Theorem 1.6. We put  $\mathfrak{p}(G - e) = \mathfrak{p}_1(G)$  and  $\mathfrak{p}(G/e) = \mathfrak{p}_2(G)$  and write (+++) as

$$\begin{aligned} \mathfrak{p}(G_1 \sqcup_e G_2) &= \hat{g}_e(\mathfrak{p}_1(G_1), \mathfrak{p}_1(G_2), \mathfrak{p}_2(G_1), \mathfrak{p}_2(G_2)) \\ &= \frac{g_e(\mathfrak{p}_1(G_1), \mathfrak{p}_1(G_2), \mathfrak{p}_2(G_1), \mathfrak{p}_2(G_2))}{\alpha}. \end{aligned} \tag{*}$$

Now  $\hat{g}_e \in \mathbb{Q}[x_1, \dots, x_4]$  rather than in  $\mathbb{Z}[x_1, \dots, x_4]$ .

This leads us to the following definition:

**Definition 6.2** (Splitting set). Let  $\mathfrak{p}$  be a graph polynomial with values in a polynomial ring  $\mathcal{R}$ , and  $H$  be a fixed graph. A finite set of graph polynomials

$$\mathfrak{P} = \{\mathfrak{p}_1, \mathfrak{p}_2, \dots, \mathfrak{p}_m\}$$

and a polynomial

$$g_H^{\mathfrak{p}_i}(x_1, \dots, x_m, y_1, \dots, y_m) \in \mathcal{R}[y_1, \dots, y_m, y_{m+1}, \dots, y_{2m}]$$

is called a *splitting set* for  $\mathfrak{p}$  and  $H$ -sums if for every  $G$ , with  $G = G_1 \oplus_H G_2$  we have

$$\mathfrak{p}_i(G) = g_H^{\mathfrak{p}_i}(\mathfrak{p}_1(G_1), \dots, \mathfrak{p}_m(G_1), \mathfrak{p}_1(G_2), \dots, \mathfrak{p}_m(G_2)).$$

The definition can also be stated for arbitrary  $\tau$ -structures and operations other than  $H$ -sums of graphs. Our goal is to state (and prove) a general theorem which says that for a large class of graph polynomials and for any *MSOL*-smooth operation on graphs, a finite splitting set exists.

#### 6.4. *MSOL-definable polynomials*

Here we follow [34] and [108,109]. We look at polynomials with *MSOL*-definable ranges of summation over *MSOL*-definable monomials. More precisely:

**Definition 6.3.** A polynomial is *MSOL-definable* if it is of the form

$$\sum_{\phi(U)} \prod_{(i,j) \in U \subseteq E} t(i,j),$$

where  $\phi(U)$  is an *MSOL*-formula, and  $t(i,j)$  is a weight function<sup>16</sup> with finite range on edges which is obtained by an *MSOL*-definable case distinction. The *quantifier rank* of  $\mathfrak{p}$  is the maximum of the quantifier ranks of  $\phi$  and the quantifier rank of the case distinction.

The definition in [34,108,109] is a bit more general, and allows also order on the edges to be used in the case distinction, provided that the resulting polynomial does not depend on the order of the edges.

**Examples 7.**

(i) The matching polynomial is given by

$$g(G, \lambda) = \sum_r p(G,r) \lambda^r = \sum_{\phi(E')} \prod_{e \in E'} \lambda$$

where  $p(G,r)$  denotes the number of partial  $r$ -matchings of  $G$ . It is a *MSOL*-definable polynomial where  $\phi(U)$  says that  $U$  is a partial matching and  $t(i,j) = \lambda$ .

(ii) Denote by  $\chi(G,n)$  the number of proper  $n$ -colourings of a graph  $G$ . By a classical theorem, cf. [12, Chapter V, p. 151ff], this defines a polynomial  $\chi(G,X) \in \mathbb{N}[X]$ , called the *chromatic polynomial*. In its given form, it is not *MSOL*-defined, but it can be obtained as a substitution instance of the Tutte polynomial, which will turn out to be *MSOL*-definable (using an ordering on the edges), cf. [107].

We shall give a small catalogue of examples in Appendix A.

Now we can state the Feferman–Vaught theorem for graph polynomials.

**Theorem 6.4.** Let  $Op(G_1, \dots, G_k)$  be a  $k$ -ary *MSOL-smooth operation* on graphs. Let  $\mathcal{R}$  be a ring. For every *MSOL-definable graph polynomial*  $\mathfrak{p}$  over  $\mathcal{R}$  there is a finite set of *MSOL-definable graph polynomials*  $\mathfrak{P}$  over  $\mathcal{R}$  containing  $\mathfrak{p}$  and a polynomial  $g_{Op} \in \mathcal{R}[\bar{X}]$  which is a *splitting set* for  $\mathfrak{p}$  and  $Op$ .

**Proof.** [Proof idea:] We compute reduction sequences for the defining formulas of  $\mathfrak{p}$  and use them to define the polynomials of  $\mathfrak{P}$  and the polynomial  $g_{Op}$ . The full proof is given in [34].  $\square$

Using Proposition 4.16 we get

**Proposition 6.5.** There are only a finite number of *MSOL-definable graph polynomials* of given quantifier rank  $k$ .

<sup>16</sup>  $t(i,j)$  may depend also on indeterminates.

Using Theorem 6.1 we can now compute look-up tables for *MSOL*-definable graph polynomials exactly as in Section 4.5.

### 6.5. Computing *MSOL*-definable graph polynomials

Computing graph polynomials is in general  $\#P$ -hard, cf. [173]. However, similarly to Theorem 4.21 we have

**Theorem 6.6.** *Let  $K$  be an *MSOL*-inductive class of  $\tau$ -structures given by a finite  $K_0$  set of  $\tau$ -structures and *MSOL*-smooth operations  $\mathcal{O}$ , and let  $p$  be an *MSOL*-definable graph polynomial over the polynomial ring  $\mathcal{R}$ . Given a parse term  $t_{\mathfrak{A}}$  for  $\mathfrak{A}$ , computing  $p(\mathfrak{A})$  can be solved in polynomial time (in the size  $n_t$  of  $t_{\mathfrak{A}}$ ) if we assume unit cost for the ring operations of  $\mathcal{R}$ .*

**Corollary 6.7.** *Computing *MSOL*-definable graph polynomials on graphs of tree width at most  $k$  can be done in polynomial time.*

## 7. Beyond *MSOL*

The Feferman–Vaught Theorem was also intensively studied in abstract model theory, initiated by two landmark papers by Feferman, [57,58]. For a survey of abstract model theory, cf. [6], especially the chapter [106], where, among other issues, the relationship of various forms of abstract Feferman–Vaught type theorems to interpolation properties were studied.

### 7.1. Cardinality quantifiers and infinitary logics

Let  $\kappa \geq \lambda$  be infinite cardinals. Wojciechowska [174] considered the cardinality quantifiers  $Q_\kappa$  where  $Q_\kappa x \phi(x)$  is interpreted as “there are at least  $\kappa$  many  $x$  such that  $\phi(x)$ ”. She showed that one can add these quantifiers to *FOL* and still have a Feferman–Vaught Theorem for products. But her proof easily shows that one can add these quantifiers to *MSOL* and still have a Feferman–Vaught Theorem for disjoint unions.

Malitz [116] showed various Feferman–Vaught Theorems for infinitary logics  $\mathcal{L}_{\infty, \infty}$  and  $\mathcal{L}_{\kappa, \lambda}$  where  $\kappa$  is strongly inaccessible.

So, if we look at infinite structures, there is a proper class of logics with a Feferman–Vaught Theorem. But for our algorithmic perspective we restrict the logics to finite structures. For this case, Courcelle looked at the quantifiers  $C_{k,m}$  where  $k, m \in \mathbb{N}$  and  $C_{k,m} x \phi(x)$  is interpreted as “there are, modulo  $m$ , exactly  $k$  elements  $x$  satisfying  $\phi(x)$ ”. In [27], he showed that Theorem 1.6 holds for *CMSOL*, which is *MSOL* augmented by all the quantifiers  $C_{k,m}$  for  $k, m \in \mathbb{N}$ .  $CMSOL^q(\tau)$  denotes the set of *CMSOL*( $\tau$ )-sentences of quantifier rank  $q$ . Free variables can be treated as uninterpreted constants in  $\tau$ .



**Theorem 7.1** (Courcelle [26]). *For every  $q \in \mathbb{N}$  and every sentences  $\phi \in \text{CMSOL}^q(\tau)$  one can compute in polynomial time in the size of  $\phi$  a sequence of sentences*

$$\langle \psi_1^A, \dots, \psi_m^A, \psi_1^B, \dots, \psi_m^B \rangle \in \text{CMSOL}^q(\tau)^{2m}$$

and a boolean function  $B_\phi : \{0, 1\}^{2m} \rightarrow \{0, 1\}$  such that

$$\mathfrak{A} \sqcup \mathfrak{B} \models \phi$$

if and only if

$$B_\phi(b_1^A, \dots, b_m^A, b_1^B, \dots, b_m^B) = 1,$$

where  $b_j^A = 1$  iff  $\mathfrak{A} \models \psi_j^A$  and  $b_j^B = 1$  iff  $\mathfrak{B} \models \psi_j^B$ .

A detailed proof is found in [27, Lemma 4.5, p. 46ff]. Theorem 7.1 allows us to extend all algorithmic results from Sections 5 and 6 to the logic *CMSOL*. However, one has to be careful to verify that the operations are *CMSOL*-smooth, rather than just *MSOL*-smooth. Especially for the case of the *Fuse<sub>P</sub>*-operation this has not been verified, whereas for *Fuse<sub>a=b</sub>* it has been verified in [27].

If, instead of considering all the quantifiers  $C_{k,m}$  for  $k, m \in \mathbb{N}$ , we choose a computable set  $A \subseteq \mathbb{N}$  and form the logic *CMSOL(A)* with all the quantifiers  $C_{k,m}$  for  $k \leq m$ ,  $m \in A$ , the validity of sentences in a finite structure is still computable. But Theorem 1.12 has to be modified, such as to accommodate the complexity of the set  $A$ .

### 7.2. Logics with a Feferman–Vaught theorem

We finally discuss how one could obtain a characterization of logics which satisfy Theorem 1.6. For a precise definition of logics the reader should consult the books [42] or [6]. But here very little is needed. A logic has the finite occurrence property, if every  $\tau$ -sentence depends only on a finite subset of  $\tau$ . If  $\mathcal{L}$  is a logic and  $K$  a class of  $\tau$ -structures, we denote by  $\mathcal{L}[K]$  the smallest logic  $\mathcal{L}'$  in which  $K$  and all the  $\mathcal{L}$ -definable classes of structures are definable. The following definition is from [106].

**Definition 7.2.** A logic  $\mathcal{L}$  has the *uniform reduction for pairs*, which we denote by *URP*( $\mathcal{L}$ ), if for every sentence  $\phi \in \mathcal{L}(\tau)$  there exists a pair of finite sequences of sentences  $\psi_1^1, \dots, \psi_m^1$  and  $\psi_1^2, \dots, \psi_m^2$  all in  $\mathcal{L}(\tau)$  and a boolean function  $B \in 2^{2^m}$  such that for every two  $\tau$ -structures  $\mathfrak{A}_1, \mathfrak{A}_2$   $\mathfrak{A}_1 \sqcup \mathfrak{A}_2 \models \phi$  iff  $B(a_1^1, \dots, l_m^1, a_1^2, \dots, l_m^2) = 1$ , where  $a_k^i$  is the truth value of  $\mathfrak{A}_i \models \psi_k^i$ .

Clearly, a logic  $\mathcal{L}$  has uniform reduction for pairs iff it satisfies Theorem 1.6 without the computability condition that the  $\psi_k^i$  can be computed in polynomial time from  $\phi$ .

The following definition was studied first by Gessel in [71], cf. also the forthcoming [63].

**Definition 7.3.** Let  $K$  be a class of  $\tau$ -structures.

- (i) Two  $\tau$ -structures  $\mathfrak{A}_1, \mathfrak{A}_2$  are  $DU_K$ -equivalent if for every  $\tau$ -structure  $\mathfrak{B}$ ,  $\mathfrak{A}_1 \sqcup \mathfrak{B} \in K$  iff  $\mathfrak{A}_2 \sqcup \mathfrak{B} \in K$ .
- (ii) The  $DU$ -index of  $K$  is the number of  $DU_K$ -equivalence classes.
- (iii) The  $DU$ -index of  $\phi$  is the number of  $DU_K$ -equivalence classes for  $K = Mod(\phi)$ .

The following is useful for analyzing the  $DU$ -index.

**Lemma 7.4.** Let  $K$  be a class of  $\tau$ -structures of  $DU$ -index  $m$  with equivalence classes  $K_1, \dots, K_m$ .

- (i) There is  $I \subset [m]$  with  $K = \bigcup_{i \in I} K_i$ .
- (ii)  $DU_K$ -equivalence is a refinement of  $DU_{K_i}$ -equivalence and the  $DU$ -index of each  $K_i$  is at most  $m$ .

**Proof.** Use the associativity of the disjoint union and the transitivity of the  $DU_K$ -equivalence and  $DU_{K_i}$ -equivalence.  $\square$

**Example 7.5.** Let  $\tau = \{P\}$  consist of one unary predicate.

- (i) Let  $K_\kappa$  consist of all  $\tau$ -structures  $\langle A, P^A \rangle$  where  $|P^A| \geq \kappa$ . The  $DU$ -index of  $K_\kappa$  is 2. Hence there is a proper class of classes of  $\tau$ -structures of  $DU$ -index 2.
- (ii) Let  $K_{k,m}$  for  $k, m \in \mathbb{N}$  consist of all  $\tau$ -structures  $\langle A, P^A \rangle$  where  $|P^A| = k \pmod{m}$ . The  $DU$ -index of  $K_\kappa$  is finite. None of these are  $MSOL$ -definable.

Using Theorem 1.6 it is not difficult to show

**Proposition 7.6.** For every  $\phi \in MSOL(\tau)$  the  $DU$ -index of  $\phi$  is finite.

However, Specker [152] pointed out, in a discussion of [10,151] and [63], that there are many more classes with finite  $DU$ -index.

**Definition 7.7.** Let  $C_n$  denote the cycle of size  $n$ , i.e. a regular connected graph of degree 2 with  $n$ -vertices. Let  $A \subseteq \mathbb{N}$  be any set of natural numbers and  $Cycle(A) = \{C_n : n \in A\}$ . Similarly we define  $Clique(A)$  as  $Cycle(A) = \{K_n : n \in A\}$ , where  $K_n$  denotes the complete graph on  $n$  vertices.

**Proposition 7.8** (Specker).  $Cycle(A)$  and  $Clique(A)$  have  $DU$  index at most 2.

**Proof.** The disjoint union of two non-empty graphs is never a cycle (clique).  $\square$

**Corollary 7.9** (Specker). There is a continuum of classes (of graphs, of  $\bar{R}$ -structures) of finite  $DU$  index which are not  $CMSOL$ -definable.

**Proof.** Clearly there is continuum of classes of the type  $Cycle(A)$ , and hence a continuum of classes that are not definable in  $CMSOL$  (or even in second order logic,  $SOL$ ).  $\square$

**Problem 7.** Can one characterize some natural recursively presented extension of *MSOL* as the largest logic (over finite structures, over arbitrary structures) for which some form of Feferman–Vaught theorem holds?

To solve the problem above one would have to give a necessary and sufficient condition (\*) on a class  $K$  of structures such that

If  $\mathcal{L}$  is a logic, which has the uniform reduction property, and we add to  $\mathcal{L}$  a Lindström quantifier defined by a class of structures  $K$  which has the property (\*), then the logic  $\mathcal{L}[K]$  has the uniform reduction property, too.

It would seem that being of finite  $DU$  index is such a property. However, this is wrong. To see this we look at the class  $Is(A)$  of graphs  $\langle A, R_A \rangle$  such that  $\langle A, A - R^A \rangle \in Clique(A)$ , i.e. the graph consists of isolated points only. In every logic where  $Clique(A)$  is definable, also  $Is(A)$  is definable. But it is easy to see that for continuum many  $A$ ,  $Is(A)$  has infinite  $DU$ -index, and therefore does not satisfy the uniform reduction property. We just need that for infinitely many  $k, m, n \in A$ , say  $k + n \in A$  but  $m + n \notin A$ .

We nevertheless dare to formulate the following

**Conjecture 2.** *There are continuum many recursively presented (even finitely presented) extension of *CMSOL* which satisfy a Feferman–Vaught Theorem for disjoint unions and which have a recursively presented (even finitely presented) syntax.*<sup>17</sup>

## 8. Conclusions and further challenges

### 8.1. Algorithmic aspects

We have presented algorithmic applications of the Feferman–Vaught Theorem for Monadic Second Order Logic in its restricted form for finite generalized sums and products of finite structures. These applications include well known results on graphs of bounded tree width or bounded clique width with unified proofs, as well as more recent and new results of the author and his coauthors Courcelle, Fischer, Glikson, Mariño, Ravve and Rotics.

Many of the applications are theoretical in nature, as the algorithms are theoretically feasible but involve large constants. But Ravve and the author have outlined applications in database design and system verification which have the potential of real applications, cf. [133,134] and [112–114]. Ravve used to work with INTEL’s verification group in Haifa, Israel, and her work has confirmed that  $H$ -sums for rather small  $H$  are the operations used to build big computer chips from small components. In the industrial parlance,  $H$ -sums are called *unions with identification by names*.

<sup>17</sup>  $Clique(A)$  could be represented by a single generalized quantifier symbol, where the meaning depends on  $A$ . Hence we have continuum many interpretations of this quantifier symbol. Actually, the infinite cardinality quantifier  $Qx\phi(x)$  has a proper class of interpretations: for each cardinal  $\kappa$ , we can consider the interpretation “there are at least  $\kappa$  many  $x$  such that  $\phi(x)$ ”.

Other applications involve matrix polynomials with prescribed zero-patterns, especially the permanent and hamiltonian of matrices, but even in the computation of determinants improvements are possible, cf. [34].

Finally, we have also applied these techniques to knot theory, especially the computation of knot polynomials such as the Kauffman bracket, Jones Polynomial and HOMFLY-PT polynomials, cf. [110,107].

### 8.2. Generalized sums and products

Shelah [150], has further generalized  $H$ -sums and used them in formulating very weak 0–1 laws for First Order Logic over ordered structures. Recently, Gurevich and Rabinovich have formulated various versions of the Feferman–Vaught Theorem for First Order Logic, stressing applications in temporal logic, model checking and systems verification, [78,79,118,131,132]. Other applications include composition theorems for recursively defined (infinite) structures [130].

### 8.3. Other logics

We have seen in Section 7 that one can go beyond  $MSOL$  and still have the Feferman–Vaught Theorem and its resulting tools available. It would be interesting to see exactly how far this can be pushed. Looking from another angle, it would be equally interesting to see whether there is a Lindström type characterization of  $MSOL$ . In particular, in spite of Conjecture 2, it may still be possible that  $CMSOL$  is distinguished by some form of the Feferman–Vaught Theorem.

**Problem 8.** Can one characterize  $CMSOL$  as the largest logic (over finite structures, over arbitrary structures) for which some form of Feferman–Vaught theorem holds?

### 8.4. Other structures

Many graph polynomials, in particular the Tutte polynomial and its variations, were originally defined for matroids, cf. [16,125,126]. The natural question arises, whether Feferman–Vaught-type theorems should not be formulated also for matroids, rather than relational structures.

Matroids are defined as sets with families of subsets, and representable matroids are special cases which allow matrix presentations. However, a complexity theory for matroid algorithms has yet to be developed. There are very few papers dealing with computational models for matroids. Robinson and Welsh, in [138], were the first to study algorithmic problems of matroids in a proper setting. Hausmann and Korte in [83] examine various extensionally equivalent definitions of matroids, and it is shown that they are not computationally equivalent. One of the problems arises from the fact that abstract matroids are not representable as relational structures. This has also its effect on the difficulty of defining a suitable logic for matroids. Very recently, Hliněný in [84] has approached this problem. He defines a version of Monadic Second Order Logic for matroids. But there is still ample room for further investigation and alternatives.

### 8.5. Checking vs. listing

The model checking problem asks to compute the truth value of  $\mathfrak{A} \models \phi$ , for given  $\mathfrak{A}$  and  $\phi$ . If  $\phi(\bar{x})$  has free variables  $\bar{x}$ , we can ask the same for a fixed assignment for the variables  $\bar{x}$ . But instead, we can also ask the more general questions, such as

**Construction:** to find a value  $\bar{a}$  such that  $\mathfrak{A} \models \phi(\bar{a})$ ;

**Listing:** to find all values  $\bar{a}$  such that  $\mathfrak{A} \models \phi(\bar{a})$ ;

**Counting:** to compute the number of values  $\bar{a}$  such that  $\mathfrak{A} \models \phi(\bar{a})$ .

In [4], an analogue of the Feferman–Vaught Theorem for counting is stated. More recently, Flum et al. [66,75] have studied the construction and listing versions of model checking. But the exact analogy to the techniques coming from the Feferman–Vaught Theorem have not yet been worked out. This should not be too difficult, but it will need some work.

### Acknowledgements

I am grateful to the organizers of the Tarski Centenary Conference at the International Banach Center in Warsaw, Z. Adamowicz, D. Niwiński and J. Woleński, for having invited me to present the lecture underlying this paper. A very preliminary version of this work was also presented at the Banach Center at the occasion of the memorial conference in honour of H. Rasiowa in 1997. But my first visit at the Banach Center was during the very first year of its functioning, during the Logic Year 1972–1973, organized by A. Mostowski and A. Grzegorzczak. During that period I also prepared my first paper, [105], on the Feferman–Vaught Theorem, or rather on Feferman’s version of it for abstract model theory [57]. My career as a logician was definitely shaped considerably by my Polish experiences.

I would also like to thank my co-authors of the papers preceding but leading to this presentation, B. Courcelle, J. Mariño, E. Ravve and U. Rotics, for their stimulating collaboration. I would like to thank J. Przytycki and B. Wajnryb for many discussion about knot polynomials, which did also shape my view on splitting theorems, and A. Glikson, A. Rabinovich and A. Slissenko, for valuable discussions on composition theorems and graph grammars. Discussions with A. Litman and V. Harnik helped clarifying my understanding of the fuse operation, discussions with E. Fischer contributed to Section 7. B. Courcelle, A. Glikson, W. Marek and A. Slissenko kindly read the (almost) final version of this paper and helped with many valuable suggestions and questions in the final preparation of this paper for printing. Last but not least, I must have exhausted the patience of D. Niwiński, with all the delays he granted me for finishing this paper.

As this paper covers research done over 30 years, it is appropriate to thank here my teachers of mathematics and mathematical logic, who all have, in one way or another, deeply influenced my own approach to this work. In temporal order from my undergraduate studies in Zürich, to my graduate studies in Zürich and Warsaw, and my postdoctoral experiences at Stanford and in Warsaw, Vancouver and Jerusalem,

these are E. Specker, P. Bernays, H. Läuchli, E. Engeler, V. Strassen, B. Eckmann, K. Chandrasekharan, A. Mostowski, W. Marek, G. Kreisel, S. Feferman, J. Stavi, A. Lachlan and S. Shelah.

This work was partially supported by the Fund for Promotion of Research of the Technion-Israeli Institute of Technology.

## Appendix A. Examples of graph polynomials

This appendix is meant to illustrate the abundance of graph polynomials. The list is not complete, and does not contain answers to the question whether all these polynomials are *MSOL*-definable or the like. It is a guided tour to some of the literature, and should convince the reader that there are more graph polynomials between heaven and earth than his imagination would expect.

### A.1. The chromatic polynomial and the Tutte polynomial

The chromatic polynomial defined in Section 6.1 is the graph polynomial with the oldest history. It is a univariate polynomial and is extensively discussed in [12]. It is a special case of the two-variable Tutte polynomial, which was considered an exotic object of mathematical study when it was introduced in 1954 by Tutte [165]. There are several equivalent definitions of the Tutte polynomial, one based on the rank-generating polynomial and one using spanning trees. The rank-generating polynomial of a graph  $G = \langle V, E \rangle$  is defined by

$$S(G; x, y) = \sum_{F \subseteq E} x^{r(E)-r(F)} y^{n(F)} = \sum_{F \subseteq E} x^{k(F)-k(E)} y^{n(F)},$$

where  $F$  ranges over induced subgraphs of  $G$  viewed as subsets of  $E$ ,  $k(F)$  is the number of connected components of  $F$ ,  $r(F) = |V| - k(F)$ , and  $n(F) = |F| - |V| + k(F)$ .

The Tutte polynomial is defined as

$$T(G; x, y) = S(G; x - 1, y - 1) = \sum_{F \subseteq E} (x - 1)^{r(E)-r(F)} (y - 1)^{n(F)}.$$

We do not get into more details here, and refer the reader to the excellent chapter in the book by Bollobas [12].

The reason why the Tutte polynomial rose to center stage in mathematics is to be found in its connection to knot theory and the knot invariants, such as the Jones polynomial, first noted by Jaeger [89]. For a good survey see also [12]. There are also connections to the Penrose polynomial, cf. [1]. Recently, Bollobas and Riordan [13] have introduced a generalization of the Tutte polynomial to edge coloured graphs, the coloured Tutte polynomial. The coloured Tutte polynomial was shown to be complete for Valiant's complexity class **VNP** in [102]. **VNP** is a non-uniform algebraic analogue of **NP** which was introduced in [168]. A good reference is Bürgisser's book [19]. Completeness here means that every other polynomial in this class is a substitution instance of the coloured Tutte polynomial. For more on the Tutte polynomial,

its complexity, and its various interpretations, the reader may consult also Welsh’s delightful [173]. There it is also shown that the Tutte polynomial, and therefore also the coloured Tutte polynomial, are hard to compute. However, for graphs of bounded tree width they can be computed and evaluated in polynomial time, cf. [2,107,123]. The latest graph polynomial to appear in the literature is the interlace polynomial introduced in [5].

### A.2. The matching polynomial

We follow [73,103]. The *generating matching polynomials*  $g(G, \lambda)$ , the *defect matching polynomials*  $m(G, \lambda)$  and the *rook polynomials*  $\rho(B, \lambda)$  of a graph  $G$ , respectively bipartite graph  $B \subset K_{n,n}$  are defined as

$$g(G, \lambda) = \sum_r p(G, r)\lambda^r = \sum_{E' \subseteq E(G)} \prod_{e \in E'} \lambda \tag{gen-m}$$

$$m(G, \lambda) = \sum_r (-1)^r p(G, r)\lambda^{n-2r} = \lambda^{-n} \sum_{E' \subseteq E(G)} \prod_{e \in E'} (-1) \cdot \lambda^{-2} \tag{def-m}$$

$$\rho(B, \lambda) = \sum_r (-1)^r p(B, r)\lambda^{n-r} = \lambda^{-n} \sum_{E' \subseteq E(B)} \prod_{e \in E'} (-1) \cdot \lambda^{-1} \tag{rook}$$

where  $p(G, r)$  denotes the number of (partial)  $r$ -matchings of  $G$ , and  $E'$  ranges over the partial matchings of  $G$ , respectively  $B$ .

Both matching polynomials and the rook polynomials are #P hard to compute. This follows from the fact that  $p(G, n/2)$  is already #P hard to compute for bipartite graphs, cf. [166,73]. Noble in his thesis, [123] has shown that on graphs of tree width  $k$  the matching polynomials can be computed in polynomial time.

### A.3. Clique and independent set polynomials

We follow [86]. Let  $a_k(G)$ ,  $\bar{a}_k(G)$  and  $b_k(G)$  be the number of  $k$ -cliques, maximal  $k$ -cliques and  $k$ -independent sets of, with  $a_0(G) = b_0(G) = 1$ . The *clique polynomials*  $C(G; x)$  the *maximal clique polynomials*  $MC(G; x)$  and *independent set polynomials*  $I(G; x)$  are defined as

$$C(G; x) = \sum_k a_k(G)x^k = \sum_{C \subseteq v(G)} \prod_{v \in C} x \tag{clique}$$

$$MC(G; x) = \sum_k \bar{a}_k(G)x^k = \sum_{C \subseteq v(G)} \prod_{v \in C} x \tag{m-clique}$$

$$I(G; x) = \sum_k b_k(G)x^k = \sum_{I \subseteq v(G)} \prod_{v \in I} x \tag{indep}$$

where the sum in the right most term ranges over the cliques, maximal cliques and independent sets, respectively.  $C(G; x)$  and  $I(G; x)$  were studied in various contexts by Fisher and Solow [64] and Gutman and Harary [80] and Hoede and Li [86].

It is easy to see that  $C(G; x)$  and  $I(G; x)$  are **NP** hard to compute and that  $MC(G; x)$  is **#P** hard. In [86] is noted that for  $k$ -trees  $G$  one has

$$C(G; x) = x(1 + x)^k(1 + (n - k)x).$$

Here  $n$  is the size of vertex set  $V(G)$ . Hence, for  $k$ -trees  $G$  the computation of the polynomial is trivially in **P**. In [108,109] this was extended to graphs of tree width at most  $k$ .

#### A.4. Farrell polynomials

We follow and extend the framework given in [52]. Let  $\mathcal{F}$  be a family of connected graphs,<sup>18</sup> closed under isomorphisms. For each  $D \in \mathcal{F}$  (up to isomorphism) let  $X_D$  be an indeterminate and let  $R_{\mathcal{F}} = \mathbb{Z}[X_D : D \in \mathcal{F}]$  be the polynomial ring associated with  $\mathcal{F}$ .

Given a graph  $G = \langle V, E \rangle$ , a *vertex disjoint  $\mathcal{F}$ -cover*  $C$  of  $G$  is a spanning subgraph of  $G$  where each connected component  $D$  of  $C$  is in  $\mathcal{F}$ . An *edge disjoint  $\mathcal{F}$ -cover*  $C$  of  $G$  is a spanning subgraph  $\langle V, E' \rangle$  of  $G$  where  $E' = \bigsqcup_i E_i$  is partitioned as the disjoint union of edge sets  $E_i$  (also called components) such that the underlying graph of  $E_i$  is in  $\mathcal{F}$ . Here different components may overlap at vertices. A  $\mathcal{F}$ -cover is either vertex disjoint or edge disjoint and we specify which if needed.

With each  $\mathcal{F}$ -cover  $C$  of  $G$  we associate a monomial  $\mu_G(C) = \prod_{D \subseteq_{\text{comp}} C} X_D$ , where the product ranges over all connected components  $D$  of  $C$ . The  $\mathcal{F}$ -polynomial of  $G$  is now defined as

$$\sum_{C \text{ is an } \mathcal{F}\text{-cover of } G} \mu_G(C) \quad (\text{Farrell})$$

A *Farrell polynomial* is a substitution instance of an  $\mathcal{F}$ -polynomial for suitable  $\mathcal{F}$  and weight function  $w$  mapping indeterminates into some value of  $R_{\mathcal{F}}$ . Often the value set of the weight function  $w$  is assumed to be finite.

Let  $\mathcal{F}$  consist of the degenerate cliques  $K_1$  (a single vertex) and  $K_2$  (a single edge). A vertex disjoint  $\mathcal{F}$ -cover  $C$  is an  $m$ -matching with  $m = m(C)$  the number of  $K_2$ 's in  $C$ . To see that the matching polynomial  $m(G, \lambda)$  is a Farrell polynomial we put  $X_{K_1} = 1$  and  $X_{K_2} = -\lambda^{-2}$ .

Other examples, with vertex disjoint covers, include:

- (i) Rook polynomials of bipartite graphs, cf. [135]; Here  $\mathcal{F}$  is like for the matching polynomial and contains two graphs.
- (ii) The circuit polynomials, where  $\mathcal{F}$  consists of all proper and improper circuits, hence  $\mathcal{F}$  is infinite.
- (iii) The characteristic polynomial of a graph can be obtained as a special case of the circuit polynomial with proper choice of the weight function, cf. [52,143].

<sup>18</sup> One could also consider non-connected graphs, but no reasonable examples occur in the literature.



- (iv) The subgraph polynomial, where  $\mathcal{F}$  consists of all connected finite graphs. The Tutte polynomials (also called dichromatic polynomials), are of this form for a properly chosen weight function,<sup>19</sup> cf. [12,52].
- (v) Using edge disjoint covers, we can also include the *Martin polynomials for undirected graphs*, cf. [49,170].

*A.5. Determinant, permanent and hamiltonian*

The determinant of an  $(n \times n)$  matrix  $M = (m_{i,j})$  over a field is defined as

$$\det(M) = \sum_{\pi \in S_n} (-1)^{\text{sign}(\pi)} \prod_i m_{i,\pi(i)},$$

where  $S_n$  is the set of permutations over  $n$  elements.

The permanent is defined as

$$\text{per}(M) = \sum_{\pi \in S_n} \prod_i m_{i,\pi(i)}$$

and the hamiltonian is defined as

$$\text{per}(M) = \sum_{\pi \in C_n} \prod_i m_{i,\pi(i)}$$

where  $C_n$  is the set of cyclic permutations over  $n$  elements.

For matrices over  $\mathbb{Z}_2$  the entries  $m_{i,j}$  can be interpreted as indicating the presence or absence of an edge between vertices  $i, j$  of a graph  $G$ . In this case the determinant is related via Kirchhoff’s theorem, cf. [12], to the number of spanning trees of  $G$ , the permanent to the number of perfect matchings, and the hamiltonian to the number of hamiltonian cycles.

None of these are, strictly speaking graph polynomials. If the graph is given by the values of  $m_{i,j} \in \mathbb{Z}_2$ , they compute numbers. If the values include, say  $-1$  these functions are not graph invariants. But if we replace  $m_{i,j}$  by  $m_{i,j} \cdot x$  with  $m_{i,j} \in \mathbb{Z}_2$  we do get graph polynomials.

The permanent and the hamiltonian do have an interesting complexity theory. In the Turing model of computation they are  $\#\mathbf{P}$  complete, cf. [90,166] and in Valiant’s model of algebraic circuits they are  $\mathbf{VNP}$ -complete, cf. [19,167,168].

**References**

- [1] M. Aigner, The Penrose polynomial of graphs and matroids, in: *Surveys in Combinatorics, 2001* (Sussex), London Mathematical Society Lecture Note Series, Vol. 288, Cambridge University Press, Cambridge, 2001, pp. 11–46.
- [2] A. Andrzejak, An algorithm for the Tutte polynomials of graphs of bounded treewidth, *Discrete Math.* 190 (1998) 39–54.

---

<sup>19</sup> This does not apply to the coloured Tutte polynomials.

- [3] S. Arnborg, D.G. Corneil, A. Proskurowski, Complexity of finding embedding in a  $k$ -tree, *SIAM. J. Algebraic Discrete Methods* 8 (1987) 277–284.
- [4] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree decomposable graphs, *J. Algorithms* 12 (1991) 308–340.
- [5] R. Arratia, B. Bollobas, G.B. Sorkin, The interlace polynomial: a new graph polynomial, *IBM Res. Rep. RC 21813 (98165)* (1990) 31.
- [6] J. Barwise, S. Feferman (Eds.), *Model-theoretic logics. Perspectives in Mathematical Logic*, Springer, Berlin, 1985.
- [7] J.L. Bell, A.B. Slomson, *Models and Ultraproducts: An Introduction*, North-Holland, Amsterdam, 1969.
- [8] E.W. Beth, On Padoa's method in the theory of definitions, *Indag. Math.* 15 (1953) 330–339.
- [9] E.W. Beth, Observations métamathématiques sur les structures simplement ordonnées, in: *Applications scientifiques de la logique mathématique*, Collection de Logique Mathématique, Serie A, Vol. 5, Paris and Louvain, 1954, pp. 29–35.
- [10] C. Blatter, E. Specker, Recurrence relations for the number of labeled structures on a finite set, in: E. Börger, G. Hasenjaeger, D. Rödding (Eds.), *In Logic and Machines: Decision Problems and Complexity*, Lecture Notes in Computer Science, Vol. 171, Springer, Berlin, 1984, pp. 43–61.
- [11] H. Bodlaender, Treewidth: algorithmic techniques and results, in: I. Privara, P. Ruzicka (Eds.), *Proc. 22nd Internat. Symp. on the Mathematical Foundation of Computer Science, MFCS'97*, Lecture Notes in Computer Science, Vol. 1295, Springer, Berlin, 1997, pp. 29–36.
- [12] B. Bollobás, *Modern Graph Theory*, Springer, Berlin, 1999.
- [13] B. Bollobás, O. Riordan, A Tutte polynomial for coloured graphs, *Combin. Probab. Comput.* 8 (1999) 45–94.
- [14] R.B. Borie, R.G. Parker, C.A. Tovey, Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families, *Algorithmica* 7 (1992) 555–581.
- [15] A. Brandstädt, V. Le, J. Spinrad, *Graph classes: a survey*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA, 1999.
- [16] T. Brylawski, J. Oxley, The Tutte polynomial and its applications, in: N. White (Ed.), *Matroid Applications*, Cambridge University Press, Cambridge, 1992, pp. 123–225.
- [17] J.R. Büchi, Weak second-order arithmetic and finite automata, *Z. Math. Logik Grundl. Math.* 6 (1960) 66–92.
- [18] J.R. Büchi, Mathematische Theorie des Verhaltens endlicher Automaten, *ZAMM* 42 (1962) 9–16.
- [19] P. Bürgisser, Completeness and Reduction in Algebraic Complexity, in: *Algorithms and Computation in Mathematics*, Vol. 7, Springer, Berlin, 2000.
- [20] C.C. Chang, H.J. Keisler, *Model theory*, Studies in Logic, Vol. 73, 3rd Edition, North-Holland, Amsterdam, 1990.
- [21] E.M. Clarke, O. Grumberg, D. Peled, *Model Checking*, MIT Press, Cambridge, MA, 1999.
- [22] K.J. Compton, C.W. Henson, A uniform method for proving lower bounds on the computational complexity of logical theories, *Ann. Pure Appl. Logic* 48 (1990) 1–79.
- [23] K.J. Compton, C.W. Henson, A uniform method for proving lower bounds and the computational complexity of logical theories, in: S. Abramsky, D. Gabbay, T. Maibaum (Eds.), *Handbook of Logic in Computer Science*, Vol. 5, Oxford University Press, Oxford, 2000.
- [24] D.G. Corneil, M. Habib, J.-M. Lanlignel, B. Reed, U. Rotics, Polynomial time recognition of clique-width  $\leq 3$  graphs, in: *Proc. LATIN'2000*, Lecture Notes in Computer Science, Vol. 1776, Springer, Berlin, 2000, pp. 126–134.
- [25] D. Corneil, U. Rotics, On the relationship between clique-width and tree-width, in: *Lecture Notes in Computer Science*, Vol. 2204, 2001, pp. 78–90.
- [26] B. Courcelle, Graph rewriting: an algebraic and logic approach, in: J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science*, Vol. 2, Elsevier, Amsterdam, 1990, pp. 142–193 (Chapter 5).
- [27] B. Courcelle, The monadic second-order logic of graphs I: recognizable sets of finite graphs, *Inform. Comput.* 85 (1990) 12–75.
- [28] B. Courcelle, Monadic second order graph transductions: a survey, *Theoret. Comput. Sci.* 126 (1994) 53–75.

- [29] B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-rewriting hypergraph grammars, *J. Comput. System Sci.* 46 (1993) 218–270.
- [30] B. Courcelle, J. Engelfriet, A logical characterization of the sets of hypergraphs defined by hyperedge replacement grammars, *Math. Systems Theory* 28 (1995) 515–552.
- [31] B. Courcelle, J.A. Makowsky, Fusion on relational structures and the verification of monadic second order properties, *Math. Struct. Comput. Sci.* 12 (2) (2002) 203–235.
- [32] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graph of bounded clique width, extended abstract, in: J. Hromkovic, O. Sykora (Eds.), *Graph Theoretic Concepts in Computer Science, 24th Internat. Workshop, WG'98, Lecture Notes in Computer Science, Vol. 1517*, Springer, Berlin, 1998, pp. 1–16.
- [33] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, *Theory Comput. Systems* 33 (2) (2000) 125–150.
- [34] B. Courcelle, J.A. Makowsky, U. Rotics, On the fixed parameter complexity of graph enumeration problems definable in monadic second order logic, *Discrete Appl. Math.* 108 (1–2) (2001) 23–52.
- [35] B. Courcelle, M. Mosbah, Monadic second-order evaluations on tree-decomposable graphs, *Theoret. Comput. Sci.* 109 (1993) 49–82.
- [36] B. Courcelle, S. Olariu, Upper bounds to the clique-width of graphs, *Discrete Appl. Math.* 101 (2000) 77–114.
- [37] R. Diestel, *Graph theory*, Graduate Texts in Mathematics, Springer, Berlin, 1996.
- [38] J.E. Doner, Decidability of the weak second-order theory of two successors, *Notices Amer. Math. Soc.* 12 (1965) 819 (Abstract).
- [39] J.E. Doner, A. Mostowski, A. Tarski, The elementary theory of well-ordering—a metamathematical study, in: A.J. Macintyre, L. Pacholski, J.B. Paris (Eds.), *Logic Colloq. '77, Studies in Logic*, 1978, pp. 1–54.
- [40] R.G. Downey, M.F. Fellows, *Parametrized Complexity*, Springer, Berlin, 1999.
- [41] H.D. Ebbinghaus, J. Flum, *Finite model theory, Perspectives in Mathematical logic*, Springer, Berlin, 1995.
- [42] H.D. Ebbinghaus, J. Flum, W. Thomas, *Mathematical logic, Undergraduate Texts in Mathematics*, Springer, Berlin, 1980.
- [43] A. Ehrenfeucht, Application of games to some problems of mathematical logic, *Bull. Acad. Polon. Sci. Cl. III* 5 (1957) 35–37.
- [44] A. Ehrenfeucht, An application of games to the completeness problem for formalized theories, *Fund. Math.* 49 (1961) 129–141.
- [45] C.C. Elgot, Decision problems of finite automata design and related arithmetics, *Trans. Amer. Math. Soc.* 98 (1961) 21–52.
- [46] J. Ellis-Monaghan, New results for the Martin polynomial, *J. Combin. Theory Ser. B* 74 (1998) 326–352.
- [47] J. Engelfriet, Context-free graph grammars, in: *Handbook of Formal Languages, Vol. 3: Beyond Words*, Springer, Berlin, 1997, pp. 125–213.
- [48] J. Engelfriet, L.M. Heyker, G. Leih, Context-free graph languages of bounded degree are generated by Apex graph grammars, *Acta Inform.* 31 (1994) 341–378.
- [49] J. Engelfriet, G. Rozenberg, A comparison of boundary graph grammars and context-free hypergraph grammars, *Inform. and Comput.* 84 (1990) 163–206.
- [50] Yu. Ershov, L. Lavrov, I.A. Taïmanov, M.A. Taïtslin, Elementary theories, *Russian Math. Surveys* 20 (4) (1965) 35–105.
- [51] R. Fagin, Generalized first-order spectra and polynomial time recognizable sets, *Amer. Math. Soc. Proc.* 7 (1974) 27–41.
- [52] E.J. Farrell, On a general class of graph polynomials, *J. Combin. Theory Ser. B* 26 (1979) 111–122.
- [53] S. Feferman, Some recent work of Ehrenfeucht and Fraïssé, *Proc. Summer Institute of Symbolic Logic, Ithaca, 1957*, pp. 201–209.
- [54] S. Feferman, Persistent and invariant formulas for outer extensions, *Compositio Math.* 20 (1968) 29–52.
- [55] S. Feferman, Infinitary properties, local functors, and systems of ordinal functions, in: W. Hodges (Ed.), *Conference in Mathematical Logic-London 1970, Lecture Notes in Mathematics, Vol. 255*, Springer, Berlin, 1972, pp. 63–97.

- [56] S. Feferman, Applications of many-sorted interpolation theorems, in: L. Henkin, et al. (Eds.), *Proc. Tarski Symp., Proc. Symp. in Pure Mathematics*, Vol. 25, American Mathematical Society, Providence, RI, 1974, pp. 205–223.
- [57] S. Feferman, Two notes on abstract model theory, I: properties invariant on the range of definable relations between structures, *Fund. Math.* 82 (1974) 153–165.
- [58] S. Feferman, Two notes on abstract model theory, II: languages for which the set of valid sentences is s.i.i.d, *Fund. Math.* 89 (1975) 153–165.
- [59] S. Feferman, Alfred Tarski and a watershed meeting in logic: Cornell 1957, preprint, to appear as part of a chapter for a biography of Alfred Tarski, under preparation with Anita Burdman Feferman, 2003.
- [60] S. Feferman, G. Kreisel, Persistent and invariant formulas relative to theories of higher order, *Bull. Amer. Math. Soc.* 72 (1966) 480–485.
- [61] S. Feferman, R. Vaught, The first order properties of algebraic systems, *Fund. Math.* 47 (1959) 57–103.
- [62] J. Ferrante, C.W. Rackoff, *The Computational Complexity of Logical Theories*, in: *Lecture Notes in Mathematics*, Vol. 718, Springer, Berlin, 1979.
- [63] E. Fischer, J.A. Makowsky, The Specker–Blatter theorem revisited, in preparation.
- [64] D.C. Fisher, A.E. Solow, Dependence polynomials, *Discrete Math.* 82 (1990) 251–258.
- [65] M. Flasiński, Power properties of NLC graph grammars with polynomial membership problem, *Theoret. Comput. Sci.* 201 (1998) 189–231.
- [66] J. Flum, M. Frick, M. Grohe, Query evaluation via tree-decompositions, in: J. van den Bussche, V. Vianu (Eds.), *Database Theory—ICDT 2001*, *Lecture Notes in Computer Science*, Vol. 1973, Springer, Berlin, 2001, pp. 22–38.
- [67] R. Fraïssé, Sur quelques classifications des relations, basées sur les isomorphismes restraints, I: Études générale, *Publ. Sci. Univ. Alger Ser. A 2* (1955) 15–60.
- [68] R. Fraïssé, Sur quelques classifications des relations, basées sur les isomorphismes restraints, II: applications aux relations d’ordre, et constructions d’exemples montrant que les classifications sont distinctes, *Publ. Sci. Univ. Alger Ser. A 2* (1955) 273–295.
- [69] T.E. Frayne, A.C. Morel, D.S. Scott, Reduced direct products, *Fund. Math.* 51 (1962) 195–228.
- [70] M.G. Garey, D.S. Johnson, *Computers and Intractability*, *Mathematical Series*, Freeman, New York, 1979.
- [71] I. Gessel, Combinatorial proofs of congruences, in: D.M. Jackson, S.A. Vanstone (Eds.), *Enumeration and Design*, Academic Press, New York, 1984, pp. 157–197.
- [72] A. Glikson, J.A. Makowsky, NCE graph grammars and clique-width, Accepted for presentation at WG’03, 2003.
- [73] C.D. Godsil, *Algebraic Combinatorics*, Chapman & Hall, London, 1993.
- [74] M.C. Golumbic, U. Rotics, On the clique-width of some perfect graph classes, *Internat. J. Foundations of Comput. Sci.* 11 (2000) 423–443.
- [75] M. Grohe, Generalized model-checking for first-order logic, in: A. Ferreira, H. Reichel (Eds.), 18th STACS, *Lecture Notes in Computer Science*, Vol. 2010, Springer, Berlin, 2001, pp. 12–26.
- [76] Y. Gurevich, Modest theory of short chains, I, *J. Symbolic Logic* 44 (1979) 481–490.
- [77] Y. Gurevich, Monadic second order theories, in: *Model-Theoretic Logics*, *Perspectives in Mathematical Logic*, Springer, Berlin, 1985 (Chapter 14).
- [78] Y. Gurevich, A. Rabinovich, Definability and undefinability with real order at the background, *J. Symbolic Logic* 65 (2) (2000) 946–958.
- [79] Y. Gurevich, A. Rabinovich, Definability in rationals with order in the background, *J. Logic Comput.* 12 (1) (2002) 1–11.
- [80] I. Gutman, F. Harary, Generalizations of the matching polynomial, *Utilitas Math.* 24 (1983) 97–106.
- [81] A. Habel, Hyperedge Replacement: Grammars and Languages, in: *Lecture Notes in Computer Science*, Vol. 643, Springer, Berlin, 1992.
- [82] R. Halin, S-functions for graphs, *J. Geom.* 8 (1976) 171–186.
- [83] D. Hausmann, B. Korte, Algorithmic versus axiomatic definitions of matroids, *Math. Programming Stud.* 14 (1981) 98–111.
- [84] P. Hliněný, Branch-width, parse trees, and monadic second order logic for matroids, 2002, preprint.

- [85] W. Hodges, Model Theory, in: *Encyclopedia of Mathematics and its Applications*, Vol. 42, Cambridge University Press, Cambridge, 1993.
- [86] C. Hoede, X. Li, Clique polynomials and independent set polynomials of graphs, *Discrete Math.* 125 (1994) 219–228.
- [87] J.E. Hopcroft, J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley Series in Computer Science, Addison-Wesley, Reading, MA, 1980.
- [88] N. Immerman, *Descriptive complexity*, Graduate Texts in Computer Science, Springer, Berlin, 1999.
- [89] F. Jaeger, Tutte polynomials and link polynomials, *Proc. Amer. Math. Soc.* 103 (1988) 647–654.
- [90] D.S. Johnson, A catalog of complexity classes, in: J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science*, Vol. 1, Elsevier, Amsterdam, 1990 (Chapter 2).
- [91] B. Jónsson, Universal relational systems, *Math. Scand.* 4 (1956) 193–208.
- [92] B. Jónsson, On isomorphism types of groups and other algebraic systems, *Math. Scand.* 5 (1957) 224–229.
- [93] B. Jónsson, Homogeneous universal relational systems, *Math. Scand.* 8 (1960) 137–142.
- [94] C. Kim, A hierarchy of ence families of graph languages, *Theoret. Comput. Sci.* 186 (1997) 157–169.
- [95] C. Kim, Efficient recognition algorithms for boundary and linear ence graph languages, *Acta Inform.* 37 (2001) 619–623.
- [96] R.E. Ladner, Application of model theoretic games to discrete linear orders and finite automata, *Inform. Control* 33 (1977) 281–303.
- [97] H. Läuchli, A decision procedure for the weak second order theory of linear order, in: *Logic Colloquium '66*, North-Holland, Amsterdam, 1968, pp. 189–197.
- [98] H. Läuchli, J. Leonhard, On the elementary theory of order, *Fund. Math.* 59 (1966) 109–116.
- [99] C. Lautemann, Decomposition trees: structured graph representation and efficient algorithms, in: *Proc. CAAP'88*, Lecture Notes in Computer Science, Vol. 299, Springer, Berlin, 1988, pp. 28–39.
- [100] J.Y.-T. Leung, O. Vornberger, J.D. Witthoff, On some variants of the bandwidth minimization problem, *SIAM J. Comput.* 13 (1984) 650–667.
- [101] P. Lindström, On extensions of elementary logic, *Theoria* 35 (1969) 1–11.
- [102] M. Lotz, J.A. Makowsky, On the algebraic complexity of some families of coloured Tutte polynomials, *Adv. Appl. Math.* 32 (1–2) (2004) 327–349.
- [103] L. Lovasz, M. Plummer, *Matching Theory*, North-Holland, Amsterdam, 1986.
- [104] S. Mahajan, J.G. Peters, Regularity and locality in  $k$ -terminal graphs, *Dam* 54 (1994) 229–250.
- [105] J.A. Makowsky, Some observations on uniform reduction for properties invariant on the range of definable relations, *Fund. Math.* 99 (1978) 199–203.
- [106] J.A. Makowsky, Compactness, embeddings and definability, in: *Model-Theoretic Logics, Perspectives in Mathematical Logic*, Springer, Berlin, 1985 (Chapter 18).
- [107] J.A. Makowsky, Colored Tutte polynomials and Kauffman brackets on graphs of bounded tree width, in: *Proc. 12th Symp. on Discrete Algorithms*, SIAM, Philadelphia, PA, 2001, pp. 487–495.
- [108] J.A. Makowsky, J.P. Mariño, Farrell polynomials on graphs of bounded tree width, Presented at FPSAC'01, 2000.
- [109] J.A. Makowsky, J.P. Mariño, Farrell polynomials on graphs of bounded treewidth, *Adv. Appl. Math.* 30 (2003) 160–176.
- [110] J.A. Makowsky, J.P. Mariño, The parametrized complexity of knot polynomials, *J. Comput. System Sci.* 67 (4) (2003) 742–756.
- [111] J.A. Makowsky, Y. Pnueli, Arity vs. alternation in second order logic, *Ann. Pure Appl. Logic* 78 (2) (1996) 189–202.
- [112] J.A. Makowsky, E.V. Ravve, Incremental model checking for decomposable structures, in: *Mathematical Foundations of Computer Science (MFCS'95)*, Lecture Notes in Computer Science, Vol. 969, Springer, Berlin, 1995, pp. 540–551.
- [113] J.A. Makowsky, E.V. Ravve, Translation schemes and the fundamental problem of database design, in: *Conceptual Modeling—ER'96*, Lecture Notes in Computer Science, Vol. 1157, Springer, Berlin, 1996, pp. 5–26.
- [114] J.A. Makowsky, E.V. Ravve, Dependency preserving refinement and the fundamental problem of database design, *Data Knowledge Eng.* 24 (3) (1997) 277–312.

- [115] J.A. Makowsky, U. Rotics, On the cliquewidth of graphs with few  $P_4$ 's, *Internat. J. Foundations Comput. Sci.* 10 (1999) 329–348.
- [116] J. Malitz, Infinitary analogs of theorems from first order model theory, *J. Symbolic Logic* 36 (1971) 216–228.
- [117] J. Mighton, *Knot Theory on Bipartite Graphs*, Ph.D. Thesis, Department of Mathematics, University of Toronto, Toronto, Canada, 1999.
- [118] F. Moller, A. Rabinovich, On the expressive power of  $CTL^*$ , in: *LICS'99*, IEEE, London, 1999, pp. 360–369.
- [119] J.D. Monk, *Mathematical logic*, Graduate Texts in Mathematics, Springer, Berlin, 1976.
- [120] A. Mostowski, On direct products of theories, *J. Symbolic Logic* 17 (1952) 1–31.
- [121] A. Mostowski, A. Tarski, Arithmetical classes and types of well-ordered systems. preliminary report, *Bull. Amer. Math. Soc.* 55 (1949) 65, Abstract.
- [122] J. Mycielski, P. Pudlák, A.S. Stern, A Lattice of Chapters of Mathematics, in: *Memoirs of the American Mathematical Society*, Vol. 426, American Mathematical Society, Providence, RI, 1990.
- [123] S.D. Noble, Complexity of graph polynomials, Ph.D. Thesis, New College, Oxford University, England, 1997.
- [124] S.D. Noble, Evaluating the Tutte polynomial for graphs of bounded tree-width, *Combin. Probab. Comput.* 7 (1998) 307–321.
- [125] J. Oxley, On the interplay between graphs and matroids, in: *Surveys in Combinatorics*, 2001, Sussex, London Mathematical Society Lecture Note Series, Vol. 288, Cambridge University Press, Cambridge, 2001, pp. 199–239.
- [126] J. Oxley, What is a matroid? Preprint available at <http://www.math.lsu.edu/preprint/>, 2002.
- [127] J.G. Oxley, D.J.A. Welsh, Tutte polynomials computable in polynomial time, *Discrete Math.* 109 (1992) 185–192.
- [128] M.O. Rabin, A simple method for undecidability proofs and some applications, in: Y. Bar Hillel (Ed.), *Logic, Methodology and Philosophy of Science II*, Studies in Logic, North-Holland, Amsterdam, 1965, pp. 58–68.
- [129] M. Rabin, Decidability of second order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* 141 (1969) 1–35.
- [130] A. Rabinovich, Composition theorems for recursively defined structures, Technical Report, Tel Aviv University, 1999.
- [131] A. Rabinovich, Composition theorems for generalized sums, preprint 2001. Available on <http://http://www.math.tau.ac.il/~rabinoa/>.
- [132] A. Rabinovich, Selection and uniformization in generalized products, to appear in the *Journal of the Interest Group in Pure and Applied Logic*, 2004. Available on <http://http://www.math.tau.ac.il/~rabinoa/>.
- [133] E.V. Ravve, Model Checking for various notions of products, Ph.D. Thesis, Department of Computer Science, Technion-Israel Institute of Technology, 1995.
- [134] E.V. Ravve, Database decomposition with translation schemes, Ph.D. Thesis, Department of Computer Science, Technion-Israel Institute of Technology, 1998.
- [135] J. Riordan, *An Introduction to Combinatorial Analysis*, Wiley, New York, 1958.
- [136] N. Robertson, P.D. Seymour, Graph minors. I. Excluding a forest, *J. Combin. Theory Ser. B* 35 (1983) 39–61.
- [137] N. Robertson, P.D. Seymour, Graph minors. V. Excluding a planar graph, *J. Combin. Theory Ser. B* 41 (1986) 92–114.
- [138] G.C. Robinson, D.J.A. Welsh, The computational complexity of matroid properties, *Math. Proc. Cambridge Phil. Soc.* 87 (1980) 29–45.
- [139] J.G. Rosenstein, *Linear Orderings*, Academic Press, New York, 1982.
- [140] G. Rozenberg (Ed.), *Handbook of Graph Grammars and Computing by Graph Transformations*, Vol. 1: Foundations, World Scientific, Singapore, 1997.
- [141] G. Rozenberg (Ed.), *Handbook of Graph Grammars and Computing by Graph Transformations*, Vol. 2: Applications, Languages, Tools, World Scientific, Singapore, 1999.
- [142] G. Rozenberg (Ed.), *Handbook of Graph Grammars and Computing by Graph Transformations*, Vol. 3: Concurrency, Parallelisms, and Distribution, World Scientific, Singapore, 1999.



- [143] H. Sachs, Beziehungen zwischen den in einem Graphen enthaltenen Kreisen und seinem charakteristischen Polynom, *Publ. Math. Debrecen.* 11 (1964) 119–134.
- [144] J.B. Saxe, Dynamic programming algorithms for recognizing small-bandwidth graphs in polynomial time, *SIAM J. Algebraic Discrete Methods* 1 (1980) 363–369.
- [145] D. Seese, Zur Entscheidbarkeit der monadische Theorie 2, Stufe baumartiger Graphen, *W.Z. der Humbolt-Universität zu Berlin Math.-Nat. R. XXIV* (6) (1975) 768–772.
- [146] D. Seese, Some graph theoretical operations and decidability, *Math. Nachr.* 87 (1979) 15–21.
- [147] D. Seese, The structure of the models of decidable monadic theories of graphs, *Ann. Pure Appl. Logic* 53 (1991) 169–195.
- [148] D. Seese, The structure of the models of decidable monadic theories of graphs, *Ann. Pure Appl. Logic* 53 (1991) 169–195.
- [149] S. Shelah, The monadic theory of order, *Ann. Math.* 102 (1975) 379–419.
- [150] S. Shelah, On the very weak 0-1 law for random graphs with orders, *J. Logic Comput.* 6 (1996) 137–159.
- [151] E. Specker, Application of logic and combinatorics to enumeration problems, in: E. Börger (Ed.), *Trends in Theoretical Computer Science*, Computer Science Press, Rockville, MD, 1988, pp. 141–169, Reprinted in: Ernst Specker, *Selecta*, Birkhäuser, Basel, 1990, pp. 324–350.
- [152] E. Specker, Personal communication, Zurich, August 2002.
- [153] L. Stockmeyer, The polynomial time hierarchy, *Theoret. Comput. Sci.* 3 (1977) 1–22.
- [154] H. Straubing, Finite automata, formal logic, and circuit complexity, *Progress in Theoretical Computer Science*, Birkhäuser, Basel, 1994.
- [155] W. Szmielew, Elementary properties of abelian groups, *Fund. Math.* 41 (1955) 203–271.
- [156] A. Tarski, Arithmetical classes and types of algebraically closed and real-closed fields, *Bull. Amer. Math. Soc.* 55 (1949) 63.
- [157] A. Tarski, Arithmetical classes and types of boolean algebras, *Bull. Amer. Math. Soc.* 55 (1949) 64.
- [158] A. Tarski, Arithmetical classes and types of mathematical systems, *Bull. Amer. Math. Soc.* 55 (1949) 63.
- [159] A. Tarski, Metamathematical aspects of arithmetical classes and types, *Bull. Amer. Math. Soc.* 55 (1949) 63–64.
- [160] A. Tarski, Contribution to the theory of models, I, *Indag. Math.* 16 (1954) 572–581.
- [161] A. Tarski, A. Mostowski, R.M. Robinson, *Undecidable theories*, Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam, 1953.
- [162] W. Thomas, Automata on infinite objects, in: J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science*, Vol. 2, Elsevier, Amsterdam, 1990 (Chapter 4).
- [163] B. Trakhtenbrot, Finite automata and the logic of monadic predicates, *Dokl. Akad. Nauk SSSR* 140 (1961) 326–329.
- [164] Gy. Turán, On the complexity of graph grammars, *Acta Cybernet.* 6 (3) (1983) 271–280.
- [165] W.T. Tutte, A contribution to the theory of chromatic polynomials, *Canad. J. Math.* 6 (1954) 80–91.
- [166] L.G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* 8 (1979) 189–201.
- [167] L.G. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* 8 (1979) 410–421.
- [168] L. Valiant, Reducibility by algebraic projections, in: *Logic and Arithmetic: An International Symposium held in honour of Ernst Specker*, L’enseignement Mathématique, Vol. 30, Université de Genève, 1982, pp. 365–380.
- [169] M. Vardi, The complexity of relational query languages, in: *Proc. 14th ACM Symp. on Theory of Computing (STOC’82)*, ACM, San Francisco, 1982, pp. 137–146.
- [170] M. Las Vergnas, Le polynôme de Martin d’un graphe eulerien, *Ann. Discrete Math.* 17 (1983) 397–411.
- [171] K. Wagner, Über eine Eigenschaft der ebenen Komplexe, *Math. Ann.* 114 (1937) 570–590.
- [172] E. Wanke,  $k$ -NLC graphs and polynomial algorithms, *Discrete Appl. Math.* 54 (1994) 251–266.
- [173] D.J.A. Welsh, *Complexity: Knots, Colourings and Counting*, London Mathematical Society Lecture Notes Series, Vol. 186, Cambridge University Press, Cambridge, 1993.
- [174] A. Wojciechowska, Generalized products for  $q_x$ -languages, *Bull. Acad. Polonaise Sci. Série Sci. Math.* 17 (1969) 337–339.