

On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic

B. Courcelle^{a,1}, J.A. Makowsky^{b,*2}, U. Rotics^{c,3}

^a*LabRI, Université de Bordeaux, Talence, France*

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

^c*Department of Computer Science, University of Toronto, Toronto, Canada*

Abstract

We discuss the parametrized complexity of counting and evaluation problems on graphs where the range of counting is definable in monadic second-order logic (MSOL). We show that for bounded tree-width these problems are solvable in polynomial time. The same holds for bounded clique width in the cases, where the decomposition, which establishes the bound on the clique-width, can be computed in polynomial time and for problems expressible by monadic second-order formulas without edge set quantification. Such quantifications are allowed in the case of graphs with bounded tree-width. As applications we discuss in detail how this affects the parametrized complexity of the permanent and the hamiltonian of a matrix, and more generally, various generating functions of MSOL definable graph properties. Finally, our results are also applicable to *SAT* and $\#SAT$. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Fixed parameter complexity; Combinatorial enumeration

1. Prelude: parametrized complexity of permanents and hamiltonians

In this paper we study the complexity of counting or enumeration⁴ functions over graphs which are definable in fragments of second order logic (*SOL*). We also look at evaluation functions which generalize the counting functions in as much as they allow us to compute the total weight of the solutions rather than just counting their number. Special cases of evaluation functions are the permanent and the hamiltonian

E-mail addresses: courcell@labri.u-bordeaux.fr (B. Courcelle), janos@cs.technion.ac.il (J.A. Makowsky), rotics@cs.toronto.edu (U. Rotics).

¹Supported by the European project GETGRATS.

²Partially supported by a Grant of the German Israeli Foundation (1995–1998) and by the Fund for Promotion of Research of the Technion-Israeli Institute of Technology.

³Partially supported by a postdoctoral fellowship of the University of Toronto and Prof. Corneil's grant of the National Science and Engineering Research Council of Canada.

⁴In [73] these functions are called enumeration problems. This might be misleading, as we do not enumerate the solutions but we count them.

of an $(n \times n)$ matrix over a field K . Without further assumptions on the matrices, both these functions are $\#\mathbf{P}$ hard [73]. In this prelude we shall first discuss in detail our results for the case of the permanent and the hamiltonian.

In the sequel of the paper, we extend results from [2,29] and our results about permanents and hamiltonians to a new framework which includes many combinatorial polynomials and we extend them further such as to make them applicable also to the case of bounded clique width (rather than tree-width). A continuation of this work leads into the heart of algebraic complexity, as pursued in [61].

1.1. Tree-width of a matrix

Let $M = \{m_{i,j}\}$ be an $(n \times n)$ matrix over a field K . The permanent $per(M)$ of M is defined as

$$\sum_{\pi \in \mathcal{S}_n} \prod_i m_{i,\pi(i)},$$

where \mathcal{S}_n is the set of permutations of $\{1, \dots, n\}$. The hamiltonian $ham(M)$ of M is defined as

$$\sum_{\pi \in \mathcal{H}_n} \prod_i m_{i,\pi(i)},$$

where \mathcal{H}_n is the set of hamiltonian permutations of $\{1, \dots, n\}$. Recall that a permutation $\pi \in \mathcal{S}_n$ is *hamiltonian* if the relation $\{(i, \pi(i)): i \leq n\}$ is connected and forms a directed cycle. We define a directed graph G_M associated with M , possibly with loops, as follows:

- (1) The vertices of G are the set $V = \{1, 2, \dots, n\}$.
- (2) The edges of G_M are the pairs $E = \{(i, j) \in V^2: m_{i,j} \neq 0\}$
- (3) Each edge $e \in E$ with $e = (i, j)$ has a weight $w(e) = m_{i,j}$.

The tree-width $tw(G)$ of a graph G is the least width of any tree-decomposition of G . For details, cf. [32, chapter 12]. For directed graphs we use the same tree-decomposition. Hence our tree-width of a directed graph G coincides with the tree-width of the graph obtained from G by forgetting the orientation. The tree-width of a matrix M is now defined as the tree-width of G_M , $tw(M) =_{def} tw(G_M)$. Knowing the tree-width of a matrix leads to new algorithms for computing permanents and hamiltonians:

Theorem 1. *Let M be an $(n \times n)$ matrix over a field K .*

- (1) *The permanent of an $(n \times n)$ matrix M of tree-width k can be computed in time $c_k \cdot n^2$ with c_k independent of the matrix but possibly super-exponential in k .*
- (2) *The hamiltonian of an $(n \times n)$ matrix M of tree-width k can be computed in time $d_k \cdot n^2$ with d_k independent of the matrix but possibly super-exponential in k .*

In other words these functions are *fixed parametrized tractable problems* in the sense of [33], where the parameter is the tree-width of the matrix. The algorithms can also be parallelized so as to be in **NC**, cf. [46] but we shall not pursue this further.

1.2. Linear rank of a matrix

Recall that the linear rank of M over a field or ring K is the maximum number of row (column) vectors of M which are linearly independent over K . It is interesting to compare our result with recent results by Barvinok [6].

Theorem 2 (Barvinok [6]). *There are real-valued functions $g_{\text{per}}(r)$ and $g_{\text{ham}}(r)$ such that for $(n \times n)$ matrices M over \mathbb{Z} of linear rank r*

- (1) *the permanent of M can be computed using $g_{\text{per}}(r) \cdot O(n^b)$ arithmetic operations for some constant $b = O(r)$ and*
- (2) *the hamiltonian of M can be computed using $g_{\text{ham}}(r) \cdot O(n^c)$ arithmetic operations for some constant $c = O(r^2)$.*

Kogan in [51] looked at the linear rank of $MM^T - \mathbf{1}$ rather than of M and showed that in fields of characteristic 3 and for matrices M with $rk(MM^T - \mathbf{1}) \leq 1$ the permanent $per(M)$ can be computed in polynomial time. However, he also shows that for $rk(MM^T - \mathbf{1}) \geq 2$ the function is as difficult to compute as the general permanent. So this does not give us another choice of parameters for fixed parameter complexity of computing the permanent.

Barvinok's results and ours are incomparable in the following sense:

Proposition 3. *Tree-width and linear rank are independent of each other. More precisely*

- (1) *For every $n \in \mathbb{N}$ there are $(n \times n)$ matrices with linear rank n and tree-width constant k .*
- (2) *For every $n \in \mathbb{N}$ there are $(n \times n)$ matrices with linear rank 1 and tree-width $n - 1$.*

Proof. (i) Let M be such that G_M is the disjoint union of cliques of size at most $k + 1$ and containing at least one clique of size $k + 1$. Then $tw(G_M) = k$. If K has at least three elements,⁵ we can find $(j \times j)$ matrices M_j with no zero elements which have rank j . For each clique of j elements we choose the matrix M_j and form M by placing these along the diagonal. Now M has linear rank n .

(ii) The $(n \times n)$ matrix E_n which has only one's as its entries has tree-width $n - 1$ and linear rank 1.

The linear rank of a matrix is a matrix property which depends on all the entries in the matrix and which can be computed in polynomial time. The tree-width of a matrix

⁵ As we want the matrix to have only non-zero entries, our field must have at least three elements.

only depends on entries which do not vanish, i.e. only on G_M . However, computing the exact tree-width is **NP**-hard, [1]. On the other hand, there is a polynomial time approximation algorithm for tree-width with performance ratio $O(\log n)$, [11]. Furthermore, checking tree-width for fixed k can be done in linear time, and if the answer is positive, the construction of a tree decomposition can be done in linear time as well, [9,12,13]. All this together makes it attractive to use the tree-width as the parameter for the study of fixed parameter complexity.

1.3. Sparse matrices

There are various notions of sparsity of matrices.

Definition 4. An $(n \times n)$ -matrix M is k -sparse if the number of its non-zero entries is bounded by $k \cdot n$ for a fixed constant k .

It is easy to see that

Proposition 5. *A matrix of tree-width at most k is at most $2k$ -sparse.*

We discuss now how such sparsity assumptions affect the complexity of computing the permanent or hamiltonian of a matrix.

A stronger notion of sparsity is the following:

Definition 6. A matrix is *strongly k -sparse* if in each row (column) there are at most k non-zero entries.

In [52] it is noted that

Proposition 7 (Kogan). *For every $(n \times n)$ -matrix M one can find in polynomial time a strongly 3-sparse matrix A and a strongly 2-sparse matrix A_1 such that $\text{per}(M) = \text{per}(A)$ and $\text{ham}(M) = \text{ham}(A_1)$.*

Note that the problem of computing $\text{per}(M)$ for M strongly 2-sparse is solvable in polynomial time.

Proof. Sketch for $\text{ham}(M)$: Treat the matrix M as an adjacency matrix of a weighted graph G_M where the weight of a path is defined multiplicatively. Like in the construction which shows that hamiltonicity is **NP**-complete for graphs of degree at most 3, we change the graph to make it into a directed graph G'_M where each indegree (outdegree) is at most two. The weights of the new edges are chosen to be 0 or 1 in a way that there is a bijection between the sets of hamiltonian cycles of the two graphs G_M and G'_M .

Corollary 8. *The problems of computing the permanent of strongly 3-sparse matrices and the hamiltonian of strongly 2-sparse matrices are as difficult as computing the permanent in the general case.*

In the light of Theorem 1 we see that there are, assuming that the permanent cannot be computed in polynomial time, strongly k -sparse matrices of unbounded tree-width. Without this assumption it is known that the adjacency matrices of the two-dimensional grids have unbounded tree-width but are strongly 4-sparse.

In [42] $(n \times n)$ -matrices M over $\{0, 1\}$ are considered where one knows in advance that the permanent is bounded by a polynomial, i.e. $\text{per}(M) \leq k \cdot n^q$ for some constants $k, q \in \mathbb{N}$. They prove that

Theorem 9 (Grigoriev and Karpinski [42]). *Let M be an $(n \times n)$ -matrices over $\{0, 1\}$ with $\text{per}(M)$ polynomially bounded. Then $\text{per}(M)$ can be computed in \mathbf{NC}^3 , and hence in \mathbf{P} .*

To the best of our knowledge no similar theorem is known for $\text{ham}(M)$.

The bound on the size of the permanent of M is in no way related to bounded tree-width by k or k -sparsity.

- (1) Let M_2 be the (2×2) matrix with all entries 1. The $(2n \times 2n)$ -matrix A which consists of n M_2 's in the diagonal is strongly 2-sparse, has tree-width 1 and $\text{per}(A) = 2^n$.
- (2) The matrices T_n $t_{i,j} = 1$ iff $i \leq j$ are not k -sparse or of tree-width $\leq k$ for any k independent of n , but $\text{per}(T_n) = 1$.

1.4. Novelty of results

Theorem 1, as stated, is new. In [2,29] certain evaluations of terms over weighted graphs are considered, but their general definition does not include the permanent and hamiltonian. However, Theorem 1 could also be proved using the automata theoretic methods from [2,29].

In [27], the results of [2,29] were extended to optimization problems on graphs of fixed *cliquewidth*, a notion introduced by Courcelle, Engelfriet and Rozenberg [25,30]. We shall give the detailed definitions in Section 3. The proofs in [27] are model theoretic rather than automata theoretic.

The purpose of this paper is to state and prove extensions of Theorem 1. These are Theorem 31 of Section 4 and Theorem 35 of Section 5. The novelty is twofold: We replace in many cases the boundedness of the tree-width by the weaker assumption of bounded clique-width and we apply it to a wide class of *generating functions* of graph properties which are *definable* in monadic second-order logic. We also show that many variations of the classical problem *SAT* are fixed parameter tractable. Both these applications are to the best of our knowledge new. We shall again use a model theoretic proof similar to that presented in [27].

In the statement of Theorems 1 and 2 we were a bit sloppy concerning the model of computation. We count arithmetic operations and manipulations of data structures, but arithmetic operations on real numbers are counted with unit cost. This can be made precise in several ways, but the most convenient model of computation for our purpose is the model introduced by Blum et al. cf. [7]. We shall usually omit the nitty gritty details of computation but they can be provided in this model. In Section 2.5 we shall discuss the choice of computational model in greater detail.

1.5. Organization of the paper

In Section 2 we set up our logical framework of counting and evaluation functions which are definable in monadic second-order logic. In Section 3 we collect the definitions and examples around the notion of clique-width. In Section 4 we state the main results in detail and discuss the advantages and disadvantages of clique-width with respect to tree-width. In Section 5 we apply our main results to generating functions of graph properties and to many variations of the classical problem *SAT*. In Section 6 we give a detailed proof of the Feferman–Vaught–Shelah theorem of monadic second-order logic for disjoint unions of structures and use it to prove our main result. Finally, in Section 7, we draw conclusions and discuss further research.

2. Logical framework

We now present the logical and computational framework in which we work.

2.1. Second-order definable counting and evaluation problems

We assume the reader is vaguely familiar with second order logic (*SOL*) over fixed relational vocabularies τ , cf. [34]. Second-order logic is the natural language of graph theory and most, but not all, graph theoretic concepts are definable in *SOL*. In second-order logic over graphs one allows quantification of typed relation variables. The typing fixes the arity of the relations over which the variables range. It also specifies whether a relation variable ranges over tuples of vertices, edges or a mixture of the two. In the case of relational structures over a fixed vocabulary (sometimes called similarity type or signature) the generalizations are straight forward. In the next subsection we shall introduce two restrictions on *SOL* which will give us two variations of monadic second-order logic, cf. [22] for a survey.

Over finite relational structures with no restrictions on the vocabulary τ (besides having no function symbols and being finite), an *SOL*(τ) definable *counting problem* consists in determining, for each finite τ -structure \mathfrak{U} with universe A , the cardinality

$$|\{X \subset A^m: \langle \mathfrak{U}, X \rangle \models \phi(X)\}|,$$

where ϕ is a second-order τ -formula with X as its free variable which ranges over subsets of A_m . If ϕ has no free variables, this can be interpreted as a decision problem.

The corresponding evaluation problem assumes that the elements of A^m are weighted with values in some ring or field, say the reals \mathbb{R} , given by a function

$$w : A^m \rightarrow \mathbb{R}.$$

Sets $X \subseteq A^m$ are then given either the additive or multiplicative weight

$$|X|_w = \sum_{a \in X} w(a) \quad \text{or} \quad \|X\|_w = \prod_{a \in X} w(a).$$

It consists in determining the value of

$$\sum \{|X|_w : X \subseteq A^n \text{ and } \langle \mathfrak{U}, X \rangle \models \phi(X)\}$$

or

$$\sum \{\|X\|_w : X \subseteq A^n \text{ and } \langle \mathfrak{U}, X \rangle \models \phi(X)\},$$

respectively. The multiplicative version gives us back counting in the case where the graph $G = \langle V, E \rangle$, $w(i, j) = 1$ for $(i, j) \in E$ and $w(i, j) = 0$ for $(i, j) \notin E$ and $X \subseteq E$. The instances of counting and evaluation problems are called *counting* and *evaluation functions*, respectively.

Instead of using an m -ary set variable X we can also use finite sequences of such variables (of fixed length). The generalization is obvious and just complicates the notation. Less obvious is the use of *parametrized families of definable sets* of m -tuples instead of set variables. In this case X ranges over the sets of the form

$$\{\bar{a} \in A^m : \langle \mathfrak{U}, Y_1, \dots, Y_t \rangle \models \theta(\bar{a}, Y_1, \dots, Y_t)\}$$

and Y_1, \dots, Y_t are first or second-order parameters ranging over elements and subsets of A such that

$$\langle \mathfrak{U}, Y_1, \dots, Y_t \rangle \models \psi(Y_1, \dots, Y_t).$$

Clearly, both θ and ψ are required to be $SOL(\tau)$ formulas. In this case we shall write the evaluation term as

$$\sum_{\psi(Y_1, \dots, Y_t)} \left(\prod_{\theta(\bar{a}, Y_1, \dots, Y_t)} w(\bar{a}) \right) \quad \text{or} \quad \sum_{\psi(Y_1, \dots, Y_t)} \left(\sum_{\theta(\bar{a}, Y_1, \dots, Y_t)} w(\bar{a}) \right).$$

2.2. MSOL-definable evaluation problems on graphs

When dealing with graphs we shall impose two kinds of restrictions: on the vocabulary and on the arity of the quantified and free second-order variables.

The logic $MS(\tau_1) = MS_1$ over graphs allows one-sorted structures, the universe of which consists of V , the vertices, and one binary relation symbol R_E for the edges, and an arbitrary but finite number of constant symbols and unary predicate symbols. In this case the set variables range over subsets of V .

The logic $MS(\tau_2) = MS_2$ over graphs allows two-sorted structures, the universes of which consist of V and E , the vertices and edges respectively. We have additionally one binary relation symbol R (in the undirected case) or two binary relation symbol $R_{\text{src}}(v, e)$ and $R_{\text{trg}}(v, e)$ (in the directed case) for the incidence relation or source and target relations between edges and vertices. Furthermore, we allow an arbitrary but finite number of constant symbols and unary predicate symbols, where the set variables range over subsets of V or E .

We speak of an MS_i (*FOL*) definable decision, counting or evaluation problem if the defining *SOL* formula is actually given as an MS_i formula (as a first-order formula).

2.3. *MS-transductions*

An *MS-transduction* f is a function which maps τ -structures \mathcal{A} into σ -structures \mathcal{B} in such a way that the universe of \mathcal{B} and the relations from σ in \mathcal{B} are MS_i -definable in \mathcal{A} . If the universe of \mathcal{B} is definable as a subset of the universe of \mathcal{A} (rather than as an n -ary relation over \mathcal{A}) we call the transduction scalar. An *MS-transduction* f induces a (contravariant) translation f^* of $MS(\sigma)$ -formulas into $MS(\tau)$ -formulas such that for every $MS(\sigma)$ -formula θ and for every τ -structure \mathcal{A}

$$\mathcal{A} \models f^*(\theta) \quad \text{iff} \quad f(\mathcal{A}) \models \theta.$$

Furthermore $f^*(\theta)$ can be computed given θ in linear time (in the size of θ).

MS_i -transductions are *MS-transductions* which map τ_i -structures into τ_i -structures.

2.4. *Some classical examples*

SOL^i denotes second-order logic with second-order variables restricted to relations of arity at most i .

We first look at counting problems.

#Triang: The number of triangles in a graph is *FOL* definable and computable in polynomial time.

k -Cliques: Counting cliques of size k is SOL^2 definable (here k is given as the size of a unary predicate). For fixed k it is computable in polynomial time.

The following are MS_1 definable.

#MaximalClique: Counting maximal cliques (with respect to inclusion).

k -colorings: Counting the number of different k -colorings.

Using well-known model theoretic methods, cf. [34], it is not hard to show that the following are (provably) not MS_1 definable. However, the first four are MS_2 definable, the fifth is SOL^2 definable and the last is SOL^3 definable.

#PHam: Counting hamiltonian paths (circuits).

#PerfMatch: Counting perfect matchings in a bipartite graphs.

#TriangPart: Counting partitions into induced subgraphs isomorphic to triangles.

#H-Part: Counting partitions into induced subgraphs isomorphic to H . This includes **PerfMatch** and **TriangPart** as special cases.

#MaximumCliques: Counting maximum cliques (with respect to size).

#3D-Match: The number of three dimensional matchings in three partite graph.

Each of these counting problems can be turned into an evaluation problem by introducing appropriate weight functions. In particular, the permanent is the evaluation problem arising from $\#PerfMatch$, and the hamiltonian is the evaluation problem arising from $\#PHam$.

2.5. Complexity of definable counting problems

2.5.1. Turing machines

We assume here that the reader is familiar with basic complexity theory for discrete problems to be solved on Turing machines as presented, say, in [50,65]. In particular we have \mathbf{P} (polynomial time), \mathbf{NP} (non-deterministic polynomial time) and $\#\mathbf{P}$ (counting accepting guesses in non-deterministic polynomial time computations). Here we have the inclusions

$$\mathbf{P} \subseteq \mathbf{NP} \subseteq \#\mathbf{P}.$$

The permanent and the hamiltonian are in $\#\mathbf{P}$ if we restrict them to, say, $(0,1)$ -matrices. Then they are also $\#\mathbf{P}$ -complete. We denote by \mathbf{FP} ($\mathbf{FPSpace}$) the class of functions computable in polynomial time (space).

$MS(\tau)$ definable counting problems over arbitrary vocabularies are computable in $\mathbf{FPSpace}$, but it is not clear whether they are in $\#\mathbf{P}$.

Open Problem 1. *Assume that $\mathbf{FPSpace} - \#\mathbf{P} \neq \emptyset$. Are there SOL (MS_i) definable counting problems which are in $\mathbf{PSpace} - \#\mathbf{P}$? Note, that it still may be the case that $\mathbf{FPSpace} - \#\mathbf{P} \neq \emptyset$ but all its members are not SOL -definable.*

By a result of Saluja et al. [68], every counting problem in $\#\mathbf{P}$ is SOL definable, actually by a particularly simple SOL formula with second-order variables for the objects to be counted, and one binary relation variable which ranges over linear orders and is existentially quantified. All other quantifications are first order. Note that already the $MS(\tau)$ decision problems can be arbitrarily complex within the polynomial hierarchy, cf. [62]. Further investigations into logical definability of counting function were pursued by Compton and Grädel [18] and Sharell [69].

2.5.2. Valiant circuits

In dealing with evaluation problems over a field K , say the reals \mathbb{R} , we have to extend the computational model. In the literature there are two such models which deal with

real numbers as atomic entities. The first is the non-uniform model of algebraic circuits and the associated complexity classes $\mathbf{VP}_{\mathbb{R}}$ (non-uniform polynomial time) and $\mathbf{VNP}_{\mathbb{R}}$ (non-uniform non-deterministic polynomial time) introduced by Valiant in [73]. More detailed references would be [17] and [16]. In this setting both the permanent and the hamiltonian (without restrictions) are in $\mathbf{VNP}_{\mathbb{R}}$ and $\mathbf{VNP}_{\mathbb{R}}$ -complete. This requires an elaborate proof. For certain other *SOL*-definable counting problems Bürgisser [16], has determined their complexity in terms of Valiant circuits. But no general result has been proved.

Open Problem 2. *Are there SOL (MS_i) definable evaluation problems which are not in $\mathbf{VNP}_{\mathbb{R}}$?*

We note that over the reals \mathbb{R} we have

$$\mathbf{FP} \subseteq \mathbf{VP}_{\mathbb{R}} \quad \text{and} \quad \#\mathbf{P} \subseteq \mathbf{VNP}_{\mathbb{R}}.$$

2.5.3. Blum–Shub–Smale machines

The other is the uniform model of branching (and looping) programs over a kind of register machines with real numbers in the registers, as introduced by Blum et al., cf. [7]. For short we call it the BSS-model of computation. Its complexity classes are denoted by $\mathbf{P}_{\mathbb{R}}$ (polynomial time), $\mathbf{DNP}_{\mathbb{R}}$ (digital non-deterministic polynomial time) and $\mathbf{NP}_{\mathbb{R}}$ (non-deterministic polynomial time). Here non-determinism consists in guessing an auxiliary input. In the case of $\mathbf{DNP}_{\mathbb{R}}$, this auxiliary input consists of $(0,1)$ -sequences of length polynomial in the size of the input, whereas for $\mathbf{NP}_{\mathbb{R}}$ we may guess sequences of real numbers. It is easy to see that

$$\mathbf{P} \subseteq \mathbf{P}_{\mathbb{R}} \quad \text{and} \quad \mathbf{NP} \subseteq \mathbf{DNP}_{\mathbb{R}} \subseteq \mathbf{NP}_{\mathbb{R}}.$$

It is not clear whether $\mathbf{P}_{\mathbb{R}} \subseteq \mathbf{VP}_{\mathbb{R}}$, $\mathbf{DNP}_{\mathbb{R}} \subseteq \mathbf{VNP}_{\mathbb{R}}$ or $\mathbf{NP}_{\mathbb{R}} \subseteq \mathbf{VNP}_{\mathbb{R}}$ should be expected to be true. The reason being, that in Valiant’s model no branching is allowed.

Meer [64] has introduced an analogue for $\#\mathbf{P}$ over the reals in the BSS-model, denoted by $\#\mathbf{P}_{\mathbb{R}}$. However, this class contains only functions of the form $f: \mathbb{R} \rightarrow \mathbb{N}$, hence the permanent of real matrices is not in the class $\#\mathbf{P}_{\mathbb{R}}$. To the best of our knowledge the description of the exact complexity of generating functions of graph properties has not been studied.

Definition 10. Let $\mathbf{GenP}_{\mathbb{R}}$ consist of the families of generating functions of weighted graph properties which are verifiable in $\mathbf{P}_{\mathbb{R}}$. Formally, $\mathbf{GenP}_{\mathbb{R}}$ consists of evaluation functions of the form

$$\sum_{\langle V, E', w \rangle \in \mathcal{H}, E' \subseteq E} \prod_{(i,j) \in E'} \tau(i,j),$$

where $\tau(i,j)$ is a weight term, \mathcal{H} is a class of weighted graphs and membership in \mathcal{H} is in $\mathbf{P}_{\mathbb{R}}$.

Open Problem 3. *Is the permanent (hamiltonian) complete for $\text{GenP}_{\mathbb{R}}$? If no, find other complete problems for $\text{GenP}_{\mathbb{R}}$.*

2.5.4. Lower and upper bounds

Our analysis of the complexity of graph counting problems yields positive results by placing parametrized versions of these problems in $\mathbf{P}_{\mathbb{R}}$ and also in $\mathbf{VP}_{\mathbb{R}}$. To show that the general version of the problem is hard, we usually quote results which show that they are $\mathbf{VNP}_{\mathbb{R}}$ -hard. The statement that a problem is in $\mathbf{P}_{\mathbb{R}}$ seems to be stronger than the statement that it is in $\mathbf{VP}_{\mathbb{R}}$, because of its implied uniformity. However, the absence of branching in Valiant’s model suggests that $\mathbf{P}_{\mathbb{R}}$ and $\mathbf{VP}_{\mathbb{R}}$ are incomparable.

A purist might object that we work in different models of computation. To remedy this, one has to find an extension of the BSS-model such as to accomodate permanents, hamiltonians and the like. Makowsky and Meer are studying such extensions, cf. [60].

3. Clique width and tree width

3.1. Graph operations and clique-width

In this section we define the notions of graph operations and clique-width, as presented in [30]. Recall that we consider only graphs without loops and without double edges.

Definition 11 (*k-graph*). A *k-graph* is a labeled graph with (vertex) labels in $\{1, 2, \dots, k\}$. A *k-graph* G is represented as a structure $\langle V, E, V_1, \dots, V_k \rangle$, where V and E are the sets of vertices and edges respectively, and V_1, \dots, V_k form a partition of V , such that V_i is the set of vertices labeled i in G . Note that some V_i ’s may be empty. A non-labeled graph $G = \langle V, E \rangle$ will be considered as a 1-graph such that all the vertices of G are labeled by 1.

Definition 12 ($G \oplus H$). For *k-graphs* G, H such that $G = \langle V, E, V_1, \dots, V_k \rangle$ and $H = \langle V', E', V'_1, \dots, V'_k \rangle$ and $V \cap V' = \emptyset$ (if this is not the case, then replace H with a disjoint copy of H), we denote by $G \oplus H$, the disjoint union of G and H such that

$$G \oplus H = \langle V \cup V', E \cup E', V_1 \cup V'_1, \dots, V_k \cup V'_k \rangle.$$

Note that $G \oplus G \neq G$.

Definition 13 ($\eta_{i,j}(G)$). For a *k-graph* G as above, we denote by $\eta_{i,j}(G)$, where $i \neq j$, the *k-graph* obtained by connecting all the vertices labeled i to all the vertices labeled j in G . Formally:

$$\eta_{i,j}(G) = \langle V, E', V_1, \dots, V_k \rangle, \quad \text{where}$$

$$E' = E \cup \{(u, v) : u \in V_i, v \in V_j\}.$$

Definition 14 ($\rho_{i \rightarrow j}(G)$). For a k -graph G as above, we denote by $\rho_{i \rightarrow j}(G)$ the re-naming of i into j in G such that:

$$\rho_{i \rightarrow j}(G) = \langle V, E, V'_1, \dots, V'_k \rangle, \quad \text{where}$$

$$V'_i = \emptyset, \quad V'_j = V_j \cup V_i, \quad \text{and} \quad V'_p = V_p \quad \text{for} \quad p \neq i, j.$$

These graph operations have been introduced in [25] for characterizing graph grammars. For every vertex v of a graph G and $i \in \{1, \dots, k\}$, we denote by $i(v)$ the k -graph consisting of one vertex v labeled by i .

Example 15. A clique with four vertices u, v, w, x can be expressed as

$$\rho_{2 \rightarrow 1}(\eta_{1,2}(2(u) \oplus \rho_{2 \rightarrow 1}(\eta_{1,2}(2(v) \oplus \rho_{2 \rightarrow 1}(\eta_{1,2}(1(w) \oplus 2(x))))))).$$

Definition 16 (k -expression). With every graph G one can associate an algebraic expression, built using $\rho_{i \rightarrow i}$, $\eta_{i,j}$ and \oplus , which defines G . We call such an expression a k -expression defining G , if all the labels in the expression are in $\{1, \dots, k\}$. Clearly, for every graph G , there is an n -expression which defines G , where n is the number of vertices of G .

Definition 17 (*The clique-width of a graph G , $cwd(G)$*). Let $\mathcal{C}(k)$ be the class of graphs which can be defined by k -expressions. The clique-width of a graph G , denoted $cwd(G)$, is defined by $cwd(G) = \text{Min}\{k : G \in \mathcal{C}(k)\}$.

The clique-width is a complexity measure on graphs somewhat similar to tree width, which yields efficient graph algorithms provided the graph is given with its k -expression (for fixed k). A related notion has been introduced by Wanke [74] in connection with graph grammars.

$\mathcal{C}(1)$ is the class of edge-less graphs. Cographs are exactly the graphs of clique width at most 2, and trees have clique width at most 3 (cf. [38]).

Open Problem 4. *Is there a structural characterization of $\mathcal{C}(k)$ for $k \geq 3$?*

In the following sections, when considering a k -expression t which defines a graph G , it will often be useful to consider the tree structure, denoted as $tree_{cw}(t)$, corresponding to the k -expression t . For that we shall need the following definitions.

Definition 18 ($tree_{cw}(t)$). Let t be any k -expression, and let G be the graph denoted by t . We denote by $tree_{cw}(t)$ the parse tree constructed from t in the usual way. The leaves of this tree are the vertices of G , and the internal nodes correspond to the operations of t , and can be either binary corresponding to \oplus , or unary corresponding to η or ρ .

In [30] a notion of *directed clique-width* of a graph G is also studied, which we denote by $dcwd(G)$. If a directed graph G has directed clique-width $\leq k$, then the

graph G' obtained from G by forgetting the direction and disregarding double edges has (undirected) clique-width $\leq k$. We shall use directed clique-width in Section 5.

3.2. Classes of bounded clique-width

In this section we list examples of classes of bounded clique-width, (cf. [30]).

Cliques K_n have clique-width 2. Actually, cographs are exactly the graphs of clique-width ≤ 2 . Trees (and hence the paths P_n) and their complements have clique-width ≤ 3 . The simple cycles C_n have clique-width ≤ 4 and for $n \geq 7$ this bound is sharp, cf. [63].

Cographs can be characterized by the absence of induced P_4 s. The study of graph classes having few P_4 s have been very active in recent years. Example for such graph classes are the classes of cographs, (extended) P_4 -sparse graphs, (extended) P_4 -reducible graphs and P_4 -tidy, studied in [20,38,39,47,48]. Babel and Olariu introduced in [4] the class of (q, t) graphs which for $t = q - 3$ extends all the graph classes mentioned above. In such a graph no set with at most q vertices is allowed to induced more than t distinct P_4 s. Clearly, we assume that $q \geq 4$. In a series of papers (cf. [5,3]) Babel and Olariu studied the classes of $(q, q - 4)$ and $(q, q - 3)$ graphs.

In [67,63] the clique-width of the (q, t) graphs for almost all combinations of q and t was determined.

Theorem 19 (Makowsky, Rotics). *For every $(q, q - 3)$ graph G such that $q \geq 7$, G has clique-width $\leq q$, and a q -expression defining it can be constructed in time $O(|V| + |E|)$.*

3.3. Clique-width vs. tree-width

In the following we compare the strength of the assumptions that a class of graphs has bounded tree-width versus the assumption that it has bounded clique-width. Readers not familiar with tree-width should consult e.g. [9,31]. Given a (not necessarily unique) tree decomposition t_G of a graph G , we denote the parse tree which allows us to reconstruct G from t_G by $tree_{tw}(t_G)$ or, if the context makes it clear, by $tree_{tw}(G)$.

Fact 20. *A class of bounded tree-width has bounded clique-width but not vice-versa.*

If a graph has tree-width at most k , it has clique-width at most $O(4^k)$, and a corresponding parse tree $tree_{cw}(G)$ for G can be constructed in linear time from the a parse tree $tree_{tw}(G)$. On the other hand, cliques have clique-width 2 and are not of bounded tree-width.

The parsing problem for tree (clique) width k is the problem of finding a parse tree for a given graph which is a certificate for k being its tree (clique) width.

Fact 21. *The parsing problem for clique-width k is in general in NP. It is known to be polynomial for clique-width at most 3, [19], but not known to be either polynomial or NP-hard for larger values of k . The parsing problem for tree-width k is solvable in linear time, cf. [13].*

Definition 22. The incidence graph of $G = \langle V, E \rangle$ is a bipartite graph $I(G) = \langle I(V), I(E) \rangle$ of edges and vertices of G with $I(V) = V \cup E$ and for $e = (v, w)$ $(v, w) \in E$ iff $(v, e) \in I(E)$ and $(w, e) \in I(E)$. In other words we replace every edge in E by a path of length 2. The total graph of $G = \langle V, E \rangle$ $T(G) = \langle T(V), T(E) \rangle$ is like the incidence graph, but the original edges remain, and two original edges are linked iff they have a common vertex, i.e., $T(E) = E \cup I(E) \cup E''$ with $E'' = \{(e, e') : \exists v(e, v) \in I(E) \wedge (e', v) \in I(E)\}$.

Fact 23. *If G is an undirected graph and has tree-width at most k , then both its incidence graph $I(G)$ and its total graph $T(G)$ have tree-width at most $k + 1$.*

However, it follows from [63] that the family of incidence graphs $I(K_n)$ of the cliques K_n (which have cliquewidth 2) is not of bounded clique-width. The same can be shown for the total graphs $T(K_n)$.

3.4. MS_2 vs. MS_1 definability

There are properties expressible in MS_2 which are not expressible in MS_1 . For example, the class of finite Hamiltonian graphs, the class of cliques with an even number of vertices, the class of graphs having a spanning tree of out degree at most 2 are definable by MS_2 formulas but (provably) not by MS_1 ones. Proofs are easy by Ehrenfeucht–Fraïssé games or by reduction to the Elgot–Büchi theorem, saying that MS definable languages are regular. For details of these techniques cf. [34].

Fact 24. *For every property of weighted graphs expressible by an MS_2 -formula ϕ there are MS_1 -formulas ψ_1, ψ_2 such that*

$$G \models \phi \text{ iff } I(G) \models \psi_1 \text{ iff } T(G) \models \psi_2.$$

Hence, the MS_2 evaluation problems of a graph G are expressible as MS_1 evaluation problems of the incidence graph $I(G)$ of G . Similarly for total graphs.

Facts 23 and 24 together explain why results based on bounded tree-width can be formulated for MS_2 whereas for bounded clique-width they are restricted to MS_1 .

Although there are formulas in MS_2 which are strictly more expressive than those of MS_1 in general, this is not the case on graphs of special types like planar graphs or graphs of bounded degree or bounded tree-width, cf. [21]. All these cases are subsumed by the following notion:

Recall that a graph $G = \langle V, E \rangle$ is k -sparse if $|E| \leq k \cdot |V|$.

Definition 25. A graph $G = \langle V, E \rangle$ is *uniformly k -sparse* if every subgraph of G is k -sparse. We denote by $US(k)$ the class of all uniformly k -sparse graphs.

Courcelle proved in [23] that for each k , one can construct an MS -transduction that associates with every graph $G \in US(k)$ (considered as a relational structure interpreting the vocabulary τ_1) a structure isomorphic to $I(G)$.

From this and Fact 24 it follows that

Fact 26. *For every property of weighted graphs expressible by an MS_2 -formula ϕ there is an MS_1 -formula ψ such that for every graph $G \in US(k)$*

$$G \models \phi \quad \text{iff} \quad G \models \psi.$$

Furthermore, if $\phi(X)$ is an MS_2 formula expressing a property of a set of edges X of such a graph G , then ϕ can be translated into an MS_1 -formula $\psi(X_1, \dots, X_n)$ where X_1, \dots, X_n denote sets of vertices and

$$|X| = |X_1| + \dots + |X_n|$$

and the integer n depends only on k . It follows that

Fact 27. *MS_2 definable evaluation problems are MS_1 definable on each class of graphs contained in $US(k)$.*

Fact 28. *Let \mathcal{C} be a sub-class of $US(k)$. Then \mathcal{C} has bounded tree-width iff it has bounded clique-width.*

This latter fact follows from the following proposition:

Proposition 29. *If C is a class of graphs of bounded clique-width on which the mapping $G \mapsto I(G)$ is definable by an MS -transduction, then it is of bounded tree-width.*

Proof (Sketch). It is proved in [24, Theorem 2.1] that if C is a class of graphs such that $I(C)$ is a subclass of the image of a set of finite binary trees under an MS -transduction, then it has bounded tree-width.

Now let C be as in the hypothesis of the proposition. Then it is, by [24, Theorem 3.1], a subset of the image of a set of finite binary trees under an MS -transduction. By composing this transduction with the one mapping $G \mapsto I(G)$, we get an MS -transduction defining $I(C)$ from trees, and we conclude the proof by using [24, Theorem 2.1].

4. Main results

4.1. Main results

Definition 30. We say that a class of graphs is *m -effectively of clique-width (tree-width) at most k* if, for every graph in the class of size n , one can construct in time $O((|V| + |E|)^m)$ a parse tree witnessing that its clique-width (tree-width) is at most k .

By Fact 21 from Section 3.3 every class of tree-width at most k is 1-effectively of tree-width at most k (i.e. $m = 1$).

Our main result now is:

Theorem 31. *Let \mathcal{C} be a class of (directed) graphs which is m -effectively of bounded (directed) clique-width k . Then each MS_1 definable counting problem (given by ϕ) can be solved in time $c_k \cdot O((|V| + |E|)^m)$, where c_k is a constant which depends only on ϕ and k .*

The same holds for evaluation problems, provided we assume that the arithmetic operations in \mathbb{R} take constant time (independently of the size of the values). In other words, these evaluation problems are in $\mathbf{P}_{\mathbb{R}}$. They are also in non-uniform $\mathbf{VP}_{\mathbb{R}}$.

Using Facts 20 and 21 from Section 3.3 we get from this the following generalization of the main theorem of [29,2].

Theorem 32. *Let \mathcal{C} be a class of graphs which is of bounded tree-width k . Then each MS_2 definable counting problem (given by ϕ) can be solved in time $c_k \cdot O(|V| + |E|)$, where c_k is a constant which depends only on ϕ and k . The same holds for evaluation problems, provided we assume that the arithmetic operations in \mathbb{R} take constant time (independently of the size of the values). In other words, these evaluation problems are in $\mathbf{P}_{\mathbb{R}}$. They are also in non-uniform $\mathbf{VP}_{\mathbb{R}}$.*

A direct proof of Theorem 32 would be based on a bottom-up traversal of the finite tree representing a tree decomposition of width at most k of the given graph. Hence, one needs to parse the graph, i.e. to construct such a decomposition. This can be done in time $O(|V| + |E|)$ by [13]. The algorithm returns either a failure message if the graph has tree-width more than k or a tree-decomposition of width at most k . The proof of Theorem 31 will use a bottom-up traversal of an appropriate parse tree $tree_{cw}(G)$. The details will be given in Section 6.

4.2. Discussion

4.2.1. Implementability

Neither Theorem 31 nor 32 do yield an implementable algorithm for various reasons:

- (1) Because the bottom-up traversal method based on $tree_{tw}(G)$ or $tree_{cw}(G)$ uses very large sets of auxiliary formulas (of cardinality being a tower of exponentials in the number of quantifiers).
- (2) Because, in the case of bounded tree-width, the algorithm of [13] is not implementable.
- (3) Because, in the case of clique-width $\leq k$, only the case $k \leq 3$, is known to be tractable, cf. Fact 21.

Usable but nonlinear parsing algorithms to obtain tree decompositions exist, see [8–11]. In special cases, graph theoretic structure theorems, as used in [63], yield

feasible algorithms for parse trees of various classes of graphs having few P_4 's, cf. Section 3.

On the other hand, in concrete cases reasonable sets of auxiliary formulas can be constructed ad hoc by hand, rather than being generated in a blind way from the formulas, cf. [33].

4.2.2. Tree-width vs. clique-width

In the following we compare Theorems 32 and 31.

Theorem 31 concerns more classes of graphs than Theorem 32, cf. Fact 20 of Section 3.3.

From Fact 21 of Section 3.3 we know that the parsing problem for clique width is in general in **NP**. For this reason Theorem 31 has an additional hypothesis that the given graphs are parsable (with respect to clique width) in polynomial time, in order to yield a global polynomial algorithm. This hypothesis is perhaps superfluous, but this remains an open problem.

If we leave out the parsing problem, i.e. if we assume that the graphs are given with their appropriate parse trees, then Theorems 31 and 32 have weaker forms. Let us call them 31a and 32a, respectively, where the corresponding algorithms are linear in the sizes of the parse trees. More precisely

Theorem 31a. *Let \mathcal{C} be a class of (directed) graphs and t_G be a parse tree for $G \in \mathcal{C}$ with n_G nodes. Then each MS_1 definable counting problem over \mathcal{C} (given by ϕ) can be solved in time $c_k \cdot O(n_G)$, where c_k is a constant which depends only on ϕ .*

and similarly for 32a.

Then Theorem 32a is also a consequence of Theorem 31a.

However, Theorems 32a and 31a are equally powerful on subclasses of $US(k)$, the uniformly k -sparse graphs.

Let us summarize the discussion for a class of graphs of interest \mathcal{C} :

- either $\mathcal{C} \subseteq US(k)$ for some k . Then Theorems 32 and 31 are applicable iff \mathcal{C} has bounded tree-width, and both these theorems solve the same problems; Theorem 32 gives actually the best complexity if combined with Bodlaender's parsing algorithm [13];
- or $\mathcal{C} \not\subseteq US(k)$ for any k . Then only Theorem 31 is applicable, provided \mathcal{C} has bounded clique-width and a polynomial parsing algorithm is available.

5. Applications: generating functions and satisfiability

Here we give some new applications of our main results.

The first three concern algebraic complexity theory in the style of [17]. We show how a wide class of families of multivariate polynomials over a ring or field K , which in general are VNP_K -complete, are tractable if the indices of the non-zero coefficients

have a certain structure. The examples are all taken from [16, Chapter 3]. Theorem 35 below generalizes Theorem 1 to a large class of multivariate polynomials. None of these examples is covered by the framework of [2]. For these applications we need bounded tree-width, as they concern MS_2 -definable evaluation problems.

The forth application concerns the boolean satisfiability problem SAT , its counting version $\#SAT$, and some variations thereof. All of these are MS_1 -definable and hence the assumption of bounded clique-width suffices.

5.1. Generating functions of graph properties

Let $G = \langle V, E, w \rangle$ be an edge-weighted graph with weights in a field K and \mathcal{E} be a class of (unweighted) graphs closed under isomorphisms. We extend w to subsets of E by defining $w(E') = \prod_{e \in E'} w(e)$. The *generating function corresponding to G and \mathcal{E}* is defined by

$$GF(G, \mathcal{E}) \stackrel{\text{def}}{=} \sum \{w(E') : \langle V, E' \rangle \in \mathcal{E} \text{ and } E' \subseteq E\}$$

Strictly speaking $GF(G, \mathcal{E})$ is a function with argument w and value in K . Furthermore, w is a function

$$w : \{1, \dots, n\}^2 \rightarrow K$$

which can be interpreted as an $(n \times n)$ matrix over K . If we view $w(i, j) = U_{i,j}$ as indeterminates, $GF(G, \mathcal{E})$ is a multivariate polynomial in $K[U_{i,j} : i, j \leq n]$.

The permanent is the generating function for $G = K_n$, the clique on n vertices, and \mathcal{E}_{per} the perfect matchings, and similarly, the hamiltonian is the generating function for \mathcal{E}_{ham} the class of n -cycles.

We denote by $K_{m,n}$ the complete bipartite graph on m and n vertices, by R_n the two-dimensional $(n \times n)$ grid and by C_n the corresponding three-dimensional grid.

In [16] the complexity of many generating functions is discussed. Among his examples we find the following:

Cliques: $\mathcal{E}_{\text{Clique}}$, the class of cliques, is an MS_1 property. By [16], $GF(K_n, \mathcal{E}_{\text{Clique}})$ is $\#P$ hard (or VNP_K -complete).

Maximal clique: $\mathcal{E}_{\text{MaxClique}}$, the class of maximal cliques, is an MS_1 property. By [73], $GF(K_n, \mathcal{E}_{\text{MaxClique}})$ is $\#P$ hard (or VNP_K -complete).

Perfect matchings: $\mathcal{E}_{\text{PerfM}}$, the class of perfect matchings, is an MS_2 property. By [73], $GF(C_n, \mathcal{E}_{\text{PerfM}})$ is $\#P$ hard (or VNP_K -complete).

Partial permanent: $\mathcal{E}_{\text{PartM}}$, the class of partial matchings, is an MS_2 property. By [49], both $GF(K_n, \mathcal{E}_{\text{pm}})$ and $GF(R_n, \mathcal{E}_{\text{pm}})$ are $\#P$ hard (or VNP_K -complete).

5.2. Cycle format polynomials

Let $\rho = (\rho(1), \rho(2), \dots, \rho(n))$ denote a partition of n in frequency notation, i.e., $\rho(i) \in \mathbb{N}$ and $\sum_i i\rho(i) = n$. We associate with ρ the graph property \mathcal{CF}_ρ describing all

graphs on n nodes consisting of $\rho(i)$ disjoint i -cycles. The corresponding generating function

$$CF_\rho = GF(K_n, \mathcal{CF}_\rho)$$

is called the *cycle format polynomial* of ρ . This includes the generating function of perfect matchings (for $\rho(2) = n/2$) and the undirected Hamiltonian cycle polynomial UHC_n (for $\rho(n) = 1$). The property \mathcal{CF}_ρ is MS_2 definable but not MS_1 definable, cf. [57].

The complexity of CF_ρ was determined in [16] as follows:

Theorem 33 (Bürgisser [16]). *Let ρ_n be a sequence of partitions of n such that $n - \rho_n(1) \geq n^\varepsilon$ for all n and some fixed $\varepsilon > 0$. Then the sequence CF_{ρ_n} of cycle format polynomials is \mathbf{VNP}_K complete.*

5.3. F factor polynomials

Let F be a connected graph. Let the graph property $\mathcal{FA}(F)$ describe the graphs all of whose connected components are isomorphic to F . A spanning subgraph of a graph G which has the property $\mathcal{FA}(F)$ is called an F -factor of G . The corresponding generating functions

$$Fact_n(F) = GF(K_n, \mathcal{FA}(F))$$

are called the F -factor polynomials. This subsumes $CF_\rho = Fact_n(F)$ for F an m -cycle and $\rho(m) = r$ and $n = mr$. The property $\mathcal{FA}(F)$ is MS_2 definable but not MS_1 definable.

Deciding the existence of an F -factor is \mathbf{NP} -complete, provided F has at least three nodes, cf. [37, Problem GT12]. Bürgisser showed in [16], that

Theorem 34 (Bürgisser [16]). *The family $Fact_n(F)$ is \mathbf{VNP}_K -complete if F has at least two nodes.*

From Theorem 32 we get immediately

Theorem 35. *Let \mathcal{E} be a class of graphs which is MS_2 -definable and let G_n be a family of graphs of tree-width k and size n . Then $GF(G_n, \mathcal{E})$ can be computed in time $c_k \cdot O(n^p)$. In fact $GF(G_n, \mathcal{E})$ is in \mathbf{VP}_K .*

Remark 36. Bürgisser only considers cases where $w: E \rightarrow K$ is a weight function for edges. For MS_1 definable generating functions it is sometimes more natural to look at weight functions on vertices. In the case of cliques both approaches make sense, but in the case of colorings, the vertices should be weighted.

5.4. Satisfiability of propositional formulas

Our next application is *SAT* and *#SAT*, the corresponding counting problem. We shall phrase it as a graph theoretic problem. But the clique-width of sets of clauses seems an interesting concept in itself.

SAT:

Input: A set of clauses $\Sigma = \{C_1, \dots, C_m\}$ over variables p_1, \dots, p_k . We denote by N the size of Σ , i.e. the sum of the sizes of the clauses.

Output: The cardinality of the set of assignments making Σ true, i.e.

$$|\{z: \{p_1, \dots, p_k\} \rightarrow \{0, 1\}: z \text{ makes all the } C_1, \dots, C_m \text{ true}\}|.$$

Complexity: The counting problem is **#P**-complete, cf. [73].

We can code this as an MS_1 -counting problem by looking at the bipartite directed graph $G_\Sigma = \langle Var, \Sigma; E \rangle$ connecting a variable from Var with a clause from Σ in one direction for positive, and in the other direction for negative occurrence in the clause. Now a satisfying assignment is a subset of vertices V_0 of Var such that for every clause $C \in \Sigma$ there is $v \in V_0$ with $(v, c) \in E$ or there is $v' \in Var - V_0$ with $(c, v') \in E$. Note that in this coding the directed clique-width $dcwd(G_\Sigma)$ of G_Σ somehow measures the complexity of the overlapping occurrences of variables in clauses. From the directed version of Theorem 31 we get

Theorem 37. *There exists a function $\gamma: \mathbb{N} \rightarrow \mathbb{N}$ such that **#SAT** and **SAT** can be solved in $\gamma(dcwd(\Sigma)) \cdot O(N)$ time, where γ depends on the clique-width of G_Σ but not on N .*

We do not know the size of γ but it is not bounded by any polynomial. For fixed clique-width this is ultimately better than the trivial bound given by $2^k \cdot O(N)$.

Open Problem 5. *Is there an exponential lower bound for $\gamma = \gamma(dcwd(G_\Sigma))$?*

Question 1. *What are the families of sets of clauses of fixed clique (tree) width and how can they be characterized in terms relevant to research in satisfiability?*

Note that Horn clauses can have arbitrary large clique-width. Note also that for clauses of bounded size the notions of clique-width and tree-width coincide because they are uniformly sparse, cf. the discussion in Section 3.4.

The following variations of **SAT** from [37, Appendix 9] (and also their corresponding counting problems) are also MS_1 definable.

3SAT: Given Σ where each clause has at most three occurrences, is there an assignment which makes Σ true? Here $N \leq 3m$.

MONOTONE SAT: Given Σ where each clause has only positive or only negative occurrences, is there an assignment which makes Σ true?

NOT-ALL-EQUAL SAT: Is there an assignment A which makes Σ true and such that for every non-isolated clause $C \in \Sigma$ there is either a $v \in Var - A$ which has a positive occurrence in C , or there is a $u \in A$ which has a negative occurrence in C .

ONE-IN-EACH SAT: Is there an assignment A such that for every non-isolated clause $C \in \Sigma$ there is either exactly one $v \in A$ which has a positive occurrence in C and none in A which has a negative occurrence in C , or there is exactly one $u \in Var - A$ which has a negative occurrence in C and none in A which has a positive occurrence.

So Theorem 37 can also be formulated for these problems.

Theorem 38. *There exists a function $\gamma: \mathbb{N} \rightarrow \mathbb{N}$ such that **3SAT**, **MONOTONESAT**, **NOT-ALL-EQUALSAT** and **ONE-IN-EACHSAT** can be solved in $\gamma(\text{dewd}(\Sigma)) \cdot O(N)$ time, where γ depends on the clique-width of G_Σ but not on N .*

This does not apply to

HALF SAT: Is there an assignment A which makes at least half of the non-isolated clauses in Σ true?

as it is not MS_1 definable.

More detailed application of the methods presented in this paper for *SAT* and *MAXSAT* may be found in [58]. They resemble the decomposition methods as discussed in [72] and based on [71], but the exact relationship to these decomposition methods still has to be investigated.

6. Feferman–Vaught–Shelah Theorem

The proof of Theorem 31 makes use of the Feferman–Vaught–Shelah Theorem for Monadic Second Order Logic and disjoint unions of relational structures. The theorem states that the set of $MS(\tau)$ sentences of quantifier rank k true in the disjoint union $G_1 \oplus G_2$ is uniquely determined by the corresponding sets of formulas true in each of the components. As stated here, this can be easily proved using Ehrenfeucht–Fraïssé games, cf. [34]. We need a stronger version which says exactly how the truth of *one* sentence in $G_1 \oplus G_2$ is determined by the truth of certain formulas in each G_i . Such a statement can be obtained either directly by explicit construction or via the use of Hintikka sentences (which describe winning positions in the Ehrenfeucht–Fraïssé games). The first approach is due to Feferman and Vaught, [36] for generalized products and First Order Logic. The second approach, also for products, can be found in [45]. Both proofs can be adapted to disjoint unions, and the first to do so explicitly was Y. Gurevich in [44]. The extension to Monadic Second Order Logic was first stated without proof in [70].

We present here an explicit version of this theorem for disjoint unions, which allows us to solve also evaluation problems. In [66] a systematic treatment of the Feferman–Vaught–Shelah theorem and its use in Database Theory is presented.

Proviso: For the remainder of this section we fix a finite relational vocabulary τ and look at $MS(\tau)$ formulas with all the (free and bound) variables among $x_1, \dots, x_m, y_1, \dots$

$y_{m_2}, X_1, \dots, X_{m_3}$. Put $M = m_1 + m_2 + m_3$. The quantifier rank k of $MS(\tau)$ formulas is defined as usual. We denote by $F_{M,k}^\tau$ the set of $MS(\tau)$ formulas of quantifier rank at most k with M variables as indicated above. If τ is clear from the context we just write $F_{M,k}$. For $\bar{x}, \bar{y}, \bar{X}$ vectors of variables among those M variables we denote by $F_{M,k}(\bar{x}, \bar{y}, \bar{X})$ the subset of $F_{M,k}^\tau$ with free variables $\bar{x}, \bar{y}, \bar{X}$.

The following is folklore, [34].

Proposition 39. *Up to logical equivalence $F_{M,k}$ and $F_{M,k}(\bar{x}, \bar{y}, \bar{X})$ are finite.*

6.1. Hintikka formulas

We can look at the equivalence classes of $F_{M,k}(\bar{x}, \bar{y}, \bar{X})$ as forming a Boolean algebra $B_{M,k}(\bar{x}, \bar{y}, \bar{X})$, with $\wedge, \vee, \neg, \mathbf{T}, \mathbf{F}$ interpreted in the natural way.

A *Hintikka* formula $h(\bar{x}, \bar{y}, \bar{X})$ is a formula equivalent to an atom in $B_{M,k}(\bar{x}, \bar{y}, \bar{X})$. We denote by $H_{M,k}(\bar{x}, \bar{y}, \bar{X})$ the set of Hintikka formulas in $F_{M,k}(\bar{x}, \bar{y}, \bar{X})$. An explicit construction of Hintikka formulas is given e.g. in [45, Chapter 3, Theorem 3.3.2].

The following is straightforward (using Proposition 39):

Proposition 40. (1) $H_{M,k}(\bar{x}, \bar{y}, \bar{X})$ is finite.

(2) The conjunction of any two non-equivalent Hintikka formulas $h_1, h_2 \in H_{M,k}(\bar{x}, \bar{y}, \bar{X})$ is inconsistent.

(3) Every formula $\phi \in F_{M,k}(\bar{x}, \bar{y}, \bar{X})$ is equivalent to a finite disjunction of Hintikka formulas of $H_{M,k}(\bar{x}, \bar{y}, \bar{X})$.

6.2. Disjoint unions

A fundamental property of Hintikka formulas concerns disjoint unions of structures.

$G = G_1 \oplus G_2$ be τ structures with universe V, V_1, V_2 respectively. Furthermore, let z be an assignment of the variables into G with values of x_i in G_1 and of y_i in G_2 . We denote by z_i the assignment with $z_i(X_j) = z(X_j) \cap V_i$ and $z_i(x_j) = z(x_j)$ and $z_i(y_j) = z(y_j)$.

Lemma 41. *Assume $G = G_1 \oplus G_2$. Then for every Hintikka formula $h(\bar{x}, \bar{y}, \bar{X}) \in H_{M,k}(\bar{x}, \bar{y}, \bar{X})$ there are unique Hintikka formulas $h_1(\bar{x}, \bar{X}) \in H_{M,k}(\bar{x}, \bar{X})$ and $h_2(\bar{y}, \bar{X}) \in H_{M,k}(\bar{y}, \bar{X})$ such that every assignment z as above we have*

$$G, z \models h(\bar{x}, \bar{y}, \bar{X})$$

iff

$$G_1, z_1 \models h_1(\bar{x}, \bar{X}) \quad \text{and} \quad G_2, z_2 \models h_2(\bar{y}, \bar{X}).$$

Proof. The proof follows from the connection between Hintikka formulas and Ehrenfeucht-Fraïssé games, as given in, say, [34,45].

Using Proposition 39 and Lemma 41 we get

Corollary 42. (1) For every $G = \langle V, E \rangle$ and $\bar{a} \in V^{m_1}$ and $\bar{U} \in P(V)^{m_2}$ there is a unique Hintikka formula $h_G(\bar{x}, \bar{X}) \in H_{M,k}(\bar{x}, \bar{X})$ such that $G, \bar{a}, \bar{U} \models h_G(\bar{x}, \bar{X})$.

(2) Similarly, with the notation of Lemma 4.1, for any two graphs G_1 and G_2 , if $G = G_1 \oplus G_2$ and $\bar{a} \in V_1^{m_1}$ and $\bar{b} \in V_2^{m_1}$ and $(\bar{U}) \in P(V)^{m_2}$, there are unique Hintikka formulas $h_G(\bar{x}, \bar{y}, \bar{X}), h_{G_1}(\bar{x}, \bar{y}, \bar{X}), h_{G_2}(\bar{x}, \bar{y}, \bar{X}) \in H_{M,k}(\bar{x}, \bar{y}, \bar{X})$ such that $G, \bar{a}, \bar{U} \models h_G(\bar{x}, \bar{y}, \bar{X}), G_1, \bar{a}, \bar{U} \models h_{G_1}(\bar{x}, \bar{X}_1)$ and $G_2, \bar{a}, \bar{U} \models h_{G_2}(\bar{y}, \bar{X}_2)$.

(3) As $H_{M,k}(\bar{x}, \bar{y}, \bar{X})$ is finite, there are only finitely many triples $G = G_1 \oplus G_2$ which differ with respect to their uniquely determined Hintikka formulas. The same is true if we allow parameters \bar{a}, \bar{b} and \bar{U} .

Theorem 43 (Feferman, Vaught and Shelah). For every formula $\psi(\bar{x}, \bar{y}, \bar{X}) \in F_{M,k}(\bar{x}, \bar{y}, \bar{X})$ there are finitely many Hintikka formulas $h_{1,\alpha}(\bar{x}, \bar{X}) \in H_{M,k}(\bar{x}, \bar{X})$ and $h_{2,\alpha}(\bar{y}, \bar{X}) \in H_{M,k}(\bar{y}, \bar{X})$ ($\alpha \leq \beta \in \mathbb{N}$) such that for every τ -structure $G = G_1 \oplus G_2$ and every assignment z as above we have

$$G, z \models \psi(\bar{x}, \bar{y}, \bar{X})$$

iff for at least one $\alpha \leq \beta$

$$G_1, z_1 \models h_{1,\alpha}(\bar{x}, \bar{X}) \quad \text{and} \quad G_2, z_2 \models h_{2,\alpha}(\bar{y}, \bar{X}).$$

The proof follows from Proposition 40 and the lemma.

6.3. Evaluation terms over disjoint unions

Now we want to apply Theorem 43 to compute evaluation terms of the forms

$$\sum_{\psi(Y_1, \dots, Y_t)} \prod_{\theta(\bar{a}, Y_1, \dots, Y_t)} w(\bar{a}) \quad \text{or} \quad \sum_{\psi(Y_1, \dots, Y_t)} \sum_{\theta(\bar{a}, Y_1, \dots, Y_t)} w(\bar{a})$$

in a structure $G = G_1 \oplus G_2$.

We first look at the terms

$$|(Y_1, \dots, Y_t)|_{G, \theta} = \prod_{G, z \models \theta(\bar{a}, Y_1, \dots, Y_t)} w(\bar{a})$$

and

$$|(Y_1, \dots, Y_t)|_{G, \theta} = \sum_{G, z \models \theta(\bar{a}, Y_1, \dots, Y_t)} w(\bar{a}),$$

where z is an assignment of variables and summation is over all \bar{a} in G with $z(\bar{x}) = \bar{a}$. Let $h_{1,\alpha}, h_{2,\alpha}, \alpha \leq \beta$ be the two Hintikka formulas associated with θ by Lemma 41.

Lemma 44. In a τ -structure $G = G_1 \oplus G_2$ we have

$$|(Y_1, \dots, Y_t)|_{G, \theta} = \prod_{\alpha} |(Y_1, \dots, Y_t)|_{G_1, h_{1,\alpha}} \cdot \prod_{\alpha} |(Y_1, \dots, Y_t)|_{G_2, h_{2,\alpha}}$$

and

$$|(Y_1, \dots, Y_t)|_{G, \theta} = \sum_{\alpha} |(Y_1, \dots, Y_t)|_{G_1, h_{1, \alpha}} + \sum_{\alpha} |(Y_1, \dots, Y_t)|_{G_2, h_{2, \alpha}}.$$

We shall use the abbreviation

$$\| (Y_1, \dots, Y_t) \|_{G_i, \theta_i} = \prod_{\alpha} \| (Y_1, \dots, Y_t) \|_{G_1, h_{1, \alpha}}$$

and

$$|(Y_1, \dots, Y_t)|_{G_i, \theta_i} = \sum_{\alpha} |(Y_1, \dots, Y_t)|_{G_1, h_{1, \alpha}}.$$

Theorem 45. Let $h_{1, \alpha}^{\psi}$ and $h_{2, \alpha}^{\psi}$ be the Hintikka formulas associated with ψ as in Theorem 43. Using the notation of the previous lemma we have

$$\begin{aligned} & \sum_{G, z \models \psi(Y_1, \dots, Y_t)} \| (Y_1, \dots, Y_t) \|_{G, \theta} \\ &= \sum_{\alpha} \sum_{\substack{G_1, z_1 \models h_{1, \alpha}^{\psi}(Y_1, \dots, Y_t) \\ G_2, z_2 \models h_{2, \alpha}^{\psi}(Y_1, \dots, Y_t)}} \| (Y_1, \dots, Y_t) \|_{G_1, \theta_1} \cdot \| (Y_1, \dots, Y_t) \|_{G_2, \theta_2} \end{aligned}$$

and

$$\begin{aligned} & \sum_{G, z \models \psi(Y_1, \dots, Y_t)} |(Y_1, \dots, Y_t)|_{G, \theta} \\ &= \sum_{\alpha} \sum_{\substack{G_1, z_1 \models h_{1, \alpha}^{\psi}(Y_1, \dots, Y_t) \\ G_2, z_2 \models h_{2, \alpha}^{\psi}(Y_1, \dots, Y_t)}} |(Y_1, \dots, Y_t)|_{G_1, \theta_1} + |(Y_1, \dots, Y_t)|_{G_2, \theta_2}. \end{aligned}$$

6.4. Transductions and evaluation terms

We now want to compute evaluation terms of graphs $G = \eta_{i,j}(G_1)$ and $G = \rho_{i \rightarrow j}(G_1)$.

The operations $\eta_{i,j}$ and $\rho_{i \rightarrow j}$ are special cases of quantifierfree FOL transductions. The following lemma is implicitly already in [2,29] and explicitly in [28].

Lemma 46. Let G, G_1 be graphs over the same universe V , $G = \eta_{i,j}(G_1)$ and let z be an assignment into elements and subsets of V . Let ψ and θ be formulas in $F_{M,k} \subseteq MS_1$. Then there are formulas $\psi_{i,j}^{\eta}$ and $\theta_{i,j}^{\eta}$ ($\psi_{i,j}^{\rho}$ and $\theta_{i,j}^{\rho}$) in $F_{M,k}$ such that

$$\sum_{G, z \models \psi(Y_1, \dots, Y_t)} \| (Y_1, \dots, Y_t) \|_{G, \theta} = \sum_{G_1, z \models \psi_{i,j}^{\eta}(Y_1, \dots, Y_t)} \| (Y_1, \dots, Y_t) \|_{G_1, \theta_{i,j}^{\eta}}$$

and

$$\sum_{G, z \models \psi(Y_1, \dots, Y_t)} \| (Y_1, \dots, Y_t) \|_{G, z, \theta} = \sum_{G_1, z \models \psi_{i,j}^{\rho}(Y_1, \dots, Y_t)} \| (Y_1, \dots, Y_t) \|_{G_1, z, \theta_{i,j}^{\rho}}.$$

Remark 47. This lemma can be generalized to other operations besides $\eta_{i,j}(G_1)$ and $\rho_{i \rightarrow j}(G_1)$. In fact, any operation defined as a scalar quantifierfree transduction would do.

6.5. Proof of Theorem 31

The main Theorem 31 is now proved by the following steps:

- (1) Let ψ and θ be formulas in $F_{M,k} \subseteq MS_1$. We first compute all the finitely many Hintikka formulas and all instances of Lemma 41 and Corollary 42 of $H_{M,k}(\bar{x}, \bar{y}, \bar{X})$. This step depends only on ψ and θ and not on the graph G . It uses the explicit definition of Hintikka formulas given e.g. in [45, Chapter 3, Theorem 3.3.2].
- (2) For all formulas of $F_{M,k} \subseteq MS_1$ we compute their equivalent presentations as finite disjunctions of Hintikka formulas, using Proposition 40. This step depends only on ψ and θ and not on the graph G .
- (3) Given a graph G defined by a k -expression t , let $tree_c w(t)$ be a parse tree for G . This step depends on G and t . If we have to compute t from G , we need the assumption that t can be computed in polynomial time from G alone.
- (4) We apply recursively, along $tree(t)$, Theorem 45 and Lemma 46 to evaluate

$$\sum_{\psi(Y_1, \dots, Y_r)} \prod_{\theta(\bar{a}, Y_1, \dots, Y_r)} w(\bar{a}).$$

This last step is polynomial (linear) in the number v of vertices of G . There are at most v many disjoint unions to perform, and the number of transductions between the disjoint unions is bounded by the number of formulas in $F_{M,k}$. This establishes that the problem is in $\mathbf{P}_{\mathbb{R}}$. But it the last step can also be used to construct an algebraic circuit which shows that the problem is \mathbf{VP}_K .

7. Conclusions and further research

Tree-width, it turns out, is a feasible parameter to study the fixed parameter complexity of generating functions of MS_2 definable graph properties. Clique-width k of a graph G might be an equally feasible parameter to study the fixed parameter complexity of generating functions of MS_1 definable graph properties. But this requires the following problems to have positive answers:

Open Problem 6. (1) *Can one decide in polynomial time if a graph G has (directed) clique-width k for fixed k ?*

(2) *Under the assumption of a yes answer to the above, and if one has established that G has clique-width at most k , can one construct in polynomial time a corresponding k -expression for G ?*

(3) *Can clique-width k of a graph G be approximated in polynomial time with ration $\log k$?*

For $k = 3$, Corneil et al. [19], have given a positive answer in the undirected case both to the recognition and construction problems.

We have presented here our results for graphs (and problems expressible as graph problems). Tree-width has been generalized to arbitrary relational structures by Feder and Vardi [35]. Interesting applications may be found in [56]. Grohe and Mariño [43] have shown the following remarkable connection between $MS(\tau)$ and the τ -formulas of the fixed point logic (LFP).

Theorem 48 (Grohe and Mariño [43]). *On classes of τ -structures of bounded tree-width every $MS(\tau)$ sentence (formula) is equivalent a $LFP(\tau)$ formula.*

Similarly, clique-width can also be extended to arbitrary relational structures, cf. [26]. Our model theoretic proof easily adapts to this situation and Theorem 31 remains valid.

Open Problem 7. *Is Theorem 48 also true if tree-width is replaced by clique-width?*

Another interesting line of research is the extension of our results to Metafinite Model Theory, as defined in [40]. This combines finite structures (like here the graphs) with infinite structures (like here the ring or field where the weight function takes its values) to form metafinite structures. Metafinite Model Theory gives the framework to extend descriptive complexity theory, cf. [34], to the computational model of Blum, Shub and Smale, cf. [7,41]. Our evaluation problems fit into this framework. Recent applications of the methods presented in this paper include also graph theoretic polynomials like the chromatic polynomials, matching polynomials and Tutte polynomials, and knot theory, cf. [59–61]. For background on graph polynomials cf. [14,15].

A notion of monadic second-order logic for metafinite structures is readily defined, and so the class of $MS(\tau)$ definable classes of metafinite structures is well defined. It seems more general than, but closely related to, the various forms of Extended Monadic Second Logic EMS as discussed in [2,29].

Open Problem 8. *Does the Feferman–Vaught–Shelah Theorem hold for monadic second-order logic over metafinite structures?*

Note that there several ways of giving a positive answer by restricting monadic second-order logic for metafinite model theory to quantification over unary predicates over the finite domain and constant functions from the finite domain to the infinite domain. The real question is whether this is the best we can do?

We are quite sure that a suitable version of the Feferman–Vaught–Shelah Theorem can be proved for first-order logic using the method of [40, Chapter 5], and extending this to monadic second-order logic should be possible as well.

The notion of clique-width and tree-width of a metafinite structures can now be defined as depending only on the underlying finite structure. We conjecture that Theorems 31 and 32 have their suitable analogues.

Acknowledgements

The second author is indebted to Gregory Kogan for introducing him to the world of permanents as documented in [51,53–55]. He would also like to thank Peter Bürgisser, who made an early version of [16] available, to M. Grohe, who pointed out the bound on the treewidth of the incidence graph, and to R. Bar-Yehuda, who reminded us of *SAT*. The final version of this papers was prepared during a visit of the second author at the Fields Institute in Toronto (January 2000), where the last author spent the academic year 1999/2000. They are both indebted to D. Corneil and A. Nabutovsky, who made the visit of the second author possible and provided us with a stimulating environment.

References

- [1] S. Arnborg, D.G. Corneil, A. Proskurowski, Complexity of finding embedding in a k -tree, *SIAM J. Algebraic Discrete Methods* 8 (1987) 227–234.
- [2] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree decomposable graphs, *J. Algorithms* 12 (1991) 308–340.
- [3] L. Babel, Recognition and isomorphism of tree-like P_4 -connected graphs, *Discrete Appl. Math.* 99 (2000) 295–315.
- [4] L. Babel, S. Olariu, On the isomorphism of graphs with few P_4 s, in: M. Nagl (Ed.), *Graph Theoretic Concepts in Computer Science, 21th International Workshop, WG'95, Lecture Notes in Computer Science, Vol. 1017, Springer, Berlin, 1995*, pp. 24–36.
- [5] L. Babel, S. Olariu, On the p -connectedness of graphs — a survey, *Discrete Appl. Math.* 95 (1999) 11–33.
- [6] A.I. Barvinok, Two algorithmic results for the traveling salesman problem, *Math. Oper. Res.* 21 (1996) 65–84.
- [7] L. Blum, F. Cucker, M. Shub, S. Smale, *Complexity and Real Computation*, Springer, Berlin, 1998.
- [8] H. Bodlaender, A tourist guide through tree width, *Acta Cybernet.* 11 (1993) 1–23.
- [9] H. Bodlaender, Treewidth: algorithmic techniques and results, in: I. Privara, P. Ruzicka (Eds.), *Proceedings of the 22th International Symposium on the Mathematical Foundation of Computer Science, MFCS'97, Lecture Notes in Computer Science, Vol. 1295, Springer, Berlin, 1997*, pp. 29–36.
- [10] H. Bodlaender, A partial k -arboretum of graphs with bounded tree width (tutorial), *Theoret. Comput. Sci.* 208 (1998) 1–45.
- [11] H. Bodlaender, J. Gilbert, H. Hafsteinsson, T. Kloks, Approximating treewidth, pathwidth, and minimum elimination tree height, *J. Algorithms* 18 (1995) 238–255.
- [12] H. Bodlaender, T. Kloks, Efficient and constructive algorithms for the pathwidth and treewidth of graphs, *J. Algorithms* 21 (1996) 358–402.
- [13] H.L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, *SIAM J. Comput.* 25 (1996) 1305–1317.
- [14] B. Bollobás, *Modern Graph Theory*, Springer, Berlin, 1999.
- [15] B. Bollobás, O. Riordan, A Tutte polynomial for coloured graphs, *Combin. Probab Comput.* 8 (1999) 45–94.

- [16] P. Bürgisser, *Completeness and Reduction in Algebraic Complexity Theory*, Algorithms and Computations in Mathematics, Vol. 7, Springer, Berlin, 2000. Expanded version of the Habilitationsschrift with the same title, Universität Zürich, 1998.
- [17] P. Bürgisser, M. Clausen, M.A. Shokrollahi, in: *Algebraic Complexity Theory*, Grundlehren, Vol. 315, Springer, Berlin, 1997.
- [18] K. Compton, E. Grädel, Logical definability of counting functions, *J. Comput. System Sci.* 53 (1996) 283–297.
- [19] D.G. Corneil, M. Habib, J.-M. Lanlignel, B. Reed, U. Rotics, Polynomial time recognition of clique-width ≤ 3 graphs, Extended abstract, LATIN'2000, August 1999, to appear.
- [20] D.G. Corneil, H. Lerchs, L. Stewart, Complement reducible graphs, *Discrete Appl. Math.* 3 (1981) 163–174.
- [21] B. Courcelle, Monadic second order graph transductions: a survey, *Theoret. Comput. Sci.* 126 (1994) 53–75.
- [22] B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic, in: G. Rozenberg (Ed.), *Handbook of Graph Grammars and Computing by Graph Transformations*, Vol. 1: Foundations, World Scientific, Singapore, 1997, pp. 313–400 (Chapter 5).
- [23] B. Courcelle, The monadic second order logic of graphs, XIV: uniformly sparse graphs and edge set quantification, submitted for publication, cf. <http://dept-info.labri.u-bordeaux.fr/~courcell/ActSci.html>, 1999.
- [24] B. Courcelle, J. Engelfriet, A logical characterization of the sets of hypergraphs defined by hyperedge replacement grammars, *Math. Systems Theory* 28 (1995) 515–552.
- [25] B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-rewriting hypergraph grammars, *J. Comput. System Sci.* 46 (1993) 218–270.
- [26] B. Courcelle, J.A. Makowsky, Operations on relational structures and their compatibility with monadic second order logic, *Math. Struct. Comput. Sci.* xx (2000) xx–yy.
- [27] 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 International Workshop, WG'98, Lecture Notes in Computer Science, Vol. 1517, Springer, Berlin, 1998, pp. 1–16.
- [28] 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.
- [29] B. Courcelle, M. Mosbah, Monadic second-order evaluations on tree-decomposable graphs, *Theoret. Comput. Sci.* 109 (1993) 49–82.
- [30] B. Courcelle, S. Olariu, Upper bounds to the clique-width of graphs, *Discrete Appl. Math.* 101 (2000) 77–114.
- [31] R. Diestel, in: *Graph Decompositions, A Study in Infinite Graph Theory*, Clarendon Press, Oxford, 1990.
- [32] R. Diestel, in: *Graph Theory*, Graduate Texts in Mathematics, Springer, Berlin, 1996.
- [33] R.G. Downey, M.F. Fellows, *Parametrized Complexity*, Springer, Berlin, 1999.
- [34] H.D. Ebbinghaus, J. Flum, in: *Finite Model Theory, Perspectives in Mathematical Logic*, Springer, Berlin, 1995.
- [35] T. Feder, M. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction, STOC'93, ACM, New York, 1993, pp. 612–622.
- [36] S. Feferman, R. Vaught, The first order properties of algebraic systems, *Fund. Math.* 47 (1959) 57–103.
- [37] M.G. Garey, D.S. Johnson, in: *Computers and Intractability*, Mathematical Series, W.H. Freeman and Company, San Francisco, CA, 1979.
- [38] V. Giakoumakis, F. Roussel, H. Thuillier, On P_4 -tidy graphs, *Discrete Math. Theoret. Comput. Sci.* 1 (1997) 17–41.
- [39] V. Giakoumakis, J. Vanherpe, On extended P_4 -reducible and extended P_4 -sparse graphs, *Theoret. Comput. Sci.* 180 (1997) 269–286.
- [40] E. Grädel, Y. Gurevich Metafinite model theory, *Inform. and Comput.* 140 (1998) 26–81. See also: D. Leivant (Ed.), *Logic and Computational Complexity*, Selected Papers, Springer, Berlin, 1995, pp. 313–366.
- [41] E. Grädel, K. Meer, Descriptive complexity theory over the real numbers, *Lectures in Applied Mathematics*, Vol. 32, Springer, Berlin, 1996, pp. 381–403. A preliminary version has been presented at the 27th ACM-Symposium on Theory of Computing, Las Vegas, 1995.

- [42] D.Y. Grigoriev, M. Karpinski, The matching problem for bipartite graphs with polynomial bounded permanents is in nc , 28th Annual Symposium on Foundations of Computer Science, 1987, pp. 166–172.
- [43] M. Grohe, J. Mariño, Definability and descriptive complexity on databases of bounded tree-width, in: C. Beeri, P. Bunemann (Eds.), *Database Theory-ICDT'99*, Lecture Notes in Computer Science, Vol. 1540, Springer, Berlin, 1999, pp. 70–82.
- [44] Y. Gurevich, Modest theory of short chains, I, *J. Symbolic Logic* 44 (1979) 481–490.
- [45] W. Hodges, in: *Model Theory*, Encyclopedia of Mathematics and its Applications, Vol. 42, Cambridge University Press, Cambridge, 1993.
- [46] H.J. Hoover, R. Greenlaw, W.L. Ruzzo, *Limits to Parallel Computation*, Oxford University Press, Oxford, 1995.
- [47] B. Jamison, S. Olariu, A linear-time algorithm to recognize P_4 -reducible graphs, *Theoret. Comput. Sci.* 145 (1995) 329–344.
- [48] B. Jamison, S. Olariu, Linear-time optimization algorithms for P_4 -sparse graphs, *Discrete Appl. Math.* 61 (1995) 155–175.
- [49] M. Jerrum, Two-dimensional monomer-dimer systems are computationally intractable, *J. Statist. Phys.* 48 (1987) 121–134. Erratum in 59 (1990) 1087–1088.
- [50] D.S. Johnson, A catalog of complexity classes, in: J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science*, Vol. 1, Elsevier Science Publishers, Amsterdam, 1990 (Chapter 2).
- [51] G.P. Kogan, Computing the permanent over fields of characteristic 3: where and why it becomes difficult, *FOCS'96*, IEEE, New York, 1996, pp. 108–114.
- [52] G.P. Kogan, The complexity of Schur functions over finite fields, Research proposal for Candidacy Examination, June 1997, <http://cs.technion.ac.il/~admlogic/TR/readme.html>.
- [53] G.P. Kogan, J.A. Makowsky, The complexity of Schur functions in finite fields of characteristic 2, Technical Report, Technion-Israel Institute of Technology, February 1997, *Comput. Complexity*, submitted for publication. <http://cs.technion.ac.il/~admlogic/TR/readme.html>.
- [54] G.P. Kogan, J.A. Makowsky, Computing Schur functions for Borchardt matrices, in preparation.
- [55] G.P. Kogan, J.A. Makowsky, Computing the permanent over fields of characteristic 3: where and why it becomes difficult, Technical Report, Technion-Israel Institute of Technology, February 1996, *Computational Complexity*, submitted for publication. <http://cs.technion.ac.il/~admlogic/TR/readme.html>.
- [56] P. Kolaitis, M. Vardi, Conjunctive query containment and constraint satisfaction, *PODS'98*, ACM, New York, 1998, pp. 425–435.
- [57] J.A. Makowsky, Model theory and computer science: an appetizer, in: S. Abramsky, D. Gabbay, T. Maibaum (Eds.), *Handbook of Logic in Computer Science*, Vol. 1, Oxford University Press, Oxford, 1992 (Chapter I.6).
- [58] J.A. Makowsky, Logical methods in graph algorithms, Lecture Notes of a course given at ESSLLI'99 in Utrecht, August 1999.
- [59] J.A. Makowsky, Colored tutte polynomials and Kauffman brackets for graphs of bounded tree-width, *Combin. Probab. Comput.* xxx (2000) xx–yy, submitted for publication.
- [60] J.A. Makowsky, K. Meer, On the complexity of combinatorial and metafinite generating functions of graph properties in the computational model of Blum, Shub and Smale, *CSL'00*, Lecture Notes in Computer Science, Vol. xxxx, Springer, Berlin, 2000, pp. xx–yy.
- [61] J.A. Makowsky, K. Meer, Polynomials of bounded tree-width, in: *Formal Power Series and Algebraic Combinatorics*, eds. D. Krob, A.A. Mikhalev and A.V. Mikhalev, Berlin, 2000, pp. 292–703.
- [62] J.A. Makowsky, Y. Pnueli, Arity vs. alternation in second order logic, *Ann. Pure Appl. Logic* 78 (2) (1996) 189–202.
- [63] J.A. Makowsky, U. Rotics, On the cliquewidth of graphs with few P_4 's, *Internat. J. Found. Comput. Sci.* 10 (1999) 329–348.
- [64] K. Meer, On the complexity of quadratic programming in real number models of computation, *Theoret. Comput. Sci.* 133 (1994) 85–94.
- [65] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, MA, 1994.
- [66] E. Ravve, *Decomposition of Databases with Translation Schemes*, Ph.D. Thesis, Department of Computer Science, Technion-Israel Institute of Technology, Haifa, 1998.
- [67] U. Rotics, *Efficient algorithms for generally intractable graph problems restricted to specific classes of graphs*, Ph.D. Thesis, Technion-Israel Institute of Technology, 1998.

- [68] S. Saluja, K. Subrahmanyam, M. Thakur, Descriptive complexity of $\#P$ functions, *J. Comput. System Sci.* 50 (1995) 493–505.
- [69] A. Sharell, *Complexité Descriptive et l'approximation des fonctions de dénombrement*, Ph.D. Thesis, L.R.I.-UPS, Paris, France, 1998.
- [70] S. Shelah, The monadic theory of order, *Ann. Math.* 102 (1975) 379–419.
- [71] K. Truemper, *Matroid Decomposition*, Academic Press, New York, 1992.
- [72] K. Truemper, *Effective Logic Computation*, Wiley, New York, 1998.
- [73] L.G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* 8 (1979) 189–201.
- [74] E. Wanke, k -NLC graphs and polynomial algorithms, *Discrete Appl. Math.* 54 (1994) 251–266.