Alcom-FT Technical Report Series
ALCOMFT-TR-01-36

Polynomials of bounded tree-width

Janos A. Makowsky[1⋆] and Klaus Meer[⋆⋆2]

[1] Department of Computer Science, Technion - Israel Institute of Technology, Haifa, Israel
e-mail: janos@cs.technion.ac.is

[2] SDU Odense, Department of Mathematics and Computer Science, Campusvej 55, DK-5230 Odense M, Denmark, e-mail: meer@imada.sdu.dk

Dedicated to Steve Smale on the occasion of his 70th birthday

Abstract. We introduce a new sparsity conditions, the *tree-width*, on multivariate polynomials in n variables (over some ring R) and show that under this condition many otherwise intractable computational problems involving these polynomials become solvable in polynomial (in some cases even linear) time in n in the Blum-Shub-Smale-model over R. To define our sparsity condition we associate with these polynomials a hypergraph and study classes of polynomials where this hypergraph has tree-width at most k for some fixed $k \in \mathbb{N}$.
We are interested in three cases:
(1) The evaluation of multivariate polynomials where the number of monomials is $O(2^n)$. Examples are the permanent or the hamiltonian polynomials.
(2) For finite fields $\mathbb{F}$ the question whether a system of n polynomials $p_i(\bar{x}) \in \mathbb{F}[\bar{x}]$ of fixed degree d in n variables has a root in $\mathbb{F}^n$.
(3) For infinite ordered rings (or fields) R_{ord}, a polynomial of fixed degree d in n variables $p(\bar{x}) \in R_{ord}[\bar{x}]$ and a finite subset $A \subset R_{ord}$ we want to know whether $p(\bar{a}) > 0$ for all $\bar{a} \in R^n_{ord}$.
Our method uses graph theoretic and model theoretic tools developed in the last 15 years and applies them to the algebraic setting. This work is an extension of work by B. Courcelle, J.A. Makowsky, and U. Rotics and by Arneborg, Lagergren, and Seese.

Key words: Blum-Shub-Smale model, meta-finite structures, bounded tree-width, monadic second order logic

1 Introduction and results

It is well known that deciding the solvability of polynomial systems and approximating solutions, if they exist, are in general difficult computational tasks. In a complexity theoretic framework for real or algebraically closed fields Blum, Shub, and Smale [BSS89] substantiated this experience by introducing a computational model over $\mathbb{R}, \mathbb{C}$ and more general ring structures and showing **NP**-completeness results for the above mentioned

⋆ Partially supported by a Grant of the Fund for Promotion of Research of the Technion–Israeli Institute of Technology

⋆⋆ Supported in part by the IST Programme of the EU under contract number IST-1999-14186 ALCOM-FT. Both authors were partially supported by EU ESPRIT project NeuroCOLT 2

decision problem. The same is true over finite fields. For an introduction into the Blum-Shub-Smale (shortly BSS) model of computation see [BCSS98].

Another interesting problem is the computation of families of polynomials having exponentially many monomials. Prominent examples are the permanent and the hamiltonian polynomials. Their computational complexity has been studied by Valiant [Val79].

A lot of work has been done in analyzing subclasses of such hard problems hoping for better algorithms when problem instances are restricted to these classes. For example, in relation with polynomial systems sparsity conditions were previously used in order to analyze location of zeros of multivariate polynomials. One of the most spectacular stems from the *Newton polytope* associated with a system of multivariate polynomials over the complex numbers. Bernstein's theorem then relates the number of isolated zeros of this system to the *mixed volume of the Newton polytope*; for more on this subject see [CLO97,GKZ94,Kh91].

In our approach we are more interested in deciding the existence of zeros than in counting. For this purpose we introduce a new sparsity condition, the tree-width, on systems of multivariate polynomials in n variables (over some ring R) and show that under this condition many otherwise intractable computational problems involving these polynomials become solvable in polynomial or even linear time in n (in the BSS-model over R).

We associate with these polynomials a hypergraph and study classes of polynomials where this hypergraph has tree-width at most k for some fixed $k \in \mathbb{N}$. Tree-width of graphs is a useful concept with a long history and a plethora of results, cf. [Die96]. A definition for hypergraphs is given in section 2 below.

We are interested in three cases:

(i) The evaluation of multivariate polynomials where the number of monomials is $O(2^n)$, such as the permanent of a matrix, the permanent of the Hadamard powers of a matrix, the hamiltonian or many other *generating functions of graph properties*, see [Bür00]. In general most of these polynomials are not known to allow evaluation in polynomial time. Here, the sparsity condition we impose is the bound k on the tree-width of the underlying graph. We show that all these generating functions can be evaluated in time $O(n)$ where the constant depends (super-exponentially) on k.

(ii) The feasibility problem over a finite field $\mathbb{F}$. Here, the question is whether a system of n polynomials $p_i(\bar{x}) \in \mathbb{F}[\bar{x}]$ of fixed degree d in n variables has a root in $\mathbb{F}^n$.
This problem is $NP_{\mathbb{F}}$ hard for large enough degree d. The sparsity condition we impose is the bound k on the tree-width of the d-hypergraph of non-vanishing monomials. We show that for finite fields the problem is solvable in time $O(n)$ where the constant depends super-exponentially on k and the size of the field $\mathbb{F}$. The same is true for finite rings.

(iii) We analyze an extension of (ii) to infinite fields or rings. It turns out that we have to impose further conditions on the decision problem being considered. For infinite ordered rings R_{ord}, a polynomial of fixed degree d in n variables $p(\bar{x}) \in R_{ord}[\bar{x}]$ and a finite subset $A \subset R_{ord}$ we want to know both whether there exists an $\bar{a} \in A^n$ such that $p(\bar{a}) = 0$ or whether $p(\bar{a}) > 0$ for all $\bar{a} \in A^n$. Though not known to be NP_R complete any more, these problems are important members of the subclass DNP_R of NP_R where the search space for verification is restricted to be finite. Feasibility and positivity turn out to be decidable in polynomial time in n for polynomials of tree-width at most k if we impose some further restriction on the coefficients of p and A.

For positivity we finally show how this additional coefficient condition can be avoided.

Our approach applies to a general setting where properties being expressible in a specific logical manner are considered. It uses methods from graph theory and model theoretic tools developed in the last 15 years and applies them to the algebraic setting. This work is an extension of work by B. Courcelle, J.A. Makowsky and U. Rotics [CMR00b], which extends [CM93] and [ALS91]. The main new aspect with respect to those works is the ability to deal with a much larger class of algebraic properties captured by the logical framework we are going to define. This allows treatment of problems like the existence of zeros for polynomials, linear programming and many more.

The paper is organized as follows. In section 2 we introduce tree-width of matrices, polynomials and systems of polynomials. In section 3 we state a result from [CMR00b] to illustrate the definition. Section 4 collects problems in relation with polynomial systems our approach applies to. The main results are then stated. The mathematical development begins in section 5. We define the logical framework in which we express our problems. This mainly refers to developing so-called existential monadic second-order logic for meta-finite structures. The latter is crucial in order to combine algebraic issues with the concept of bounded tree-width. Proofs of the theorems are given in section 6 and further discussions follow in section 7. Since the arguments in sections 5 and 6 use a lot of previous work which we could not find presented in a compact fashion in literature, a detailed appendix is added. With the appendix the paper is self-contained. It explains the construction of parse-trees starting from tree-decompositions of hypergraphs; full proofs of the crucial theorems by Fraisse-Hintikka and Feferman-Vaught are given together with its algorithmic use.

2 Tree-width of polynomials and matrices

Let $V = \{0, 1, 2, \ldots, n\}$ be the index set of the variables of

$$p(x) = p(x_0, x_1, \ldots, x_n) = \sum_{(i_1, \ldots i_d) \in E} c_{i_1,\ldots,i_d} x_{i_1} \cdot x_{i_2} \cdot \ldots \cdot x_{i_d}$$

with $E \subseteq V^d$ and $x_0 = 1$. E is the set of d-tuples of indices $(i_1, \ldots i_d)$ such that the coefficient $c_{i_1,\ldots,i_d} \neq 0$.

Definition 1 *With $p(x)$ we associate the d-hypergraph $G = \langle V, E \rangle$ and define the tree-width of $p(x)$ as the tree-width of G.*

For systems of polynomials p_j of degree d we look at the hypergraph of the non-vanishing coefficients $c_{j,\alpha}$, i.e. the induced $d+1$-hypergraph.

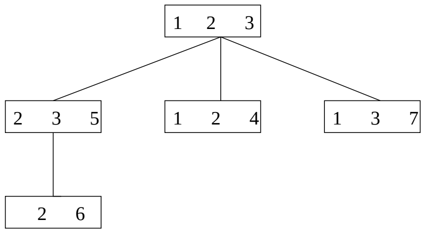
Definition 2 (Tree-width of a d-hypergraph) *A k-tree decomposition of G is defined as follows:*

(i) $\mathcal{T} = \langle T, <_T \rangle$ is a tree with $t <_T s$ expressing that t is a child of s.
(ii) For each $t \in T$ we have a subset $V_t \subseteq V$ of size at most $k+1$.
(iii) For each hyperedge $(i_1, \ldots, i_d) \in E$ there is a $t \in T$ such that $\{i_1, \ldots i_d\} \subseteq V_t$.
(iv) For each $i \in V$ the set $V(i) = \{t \in T \mid i \in V_t\}$ forms a (connected) subtree of $\mathcal{T}$.

G has tree-width at most k if there exists a k-tree decomposition of G. The tree width *of G is the smallest such k.*

Examples 3 *(i) The polynomial $p_1(x) = \sum_{i=1}^{n} x_i^4$ has tree-width 0.*
Note that the Newton polytope of p_1 is maximal with respect to polynomials of degree 4 in n variables.

(ii) Consider the polynomial $p(x_1, \dots, x_7) := x_1 \cdot x_2 \cdot x_3 + x_2^3 \cdot x_3^5 + x_1^3 \ + x_2^2 \cdot x_6 + x_2 \cdot x_3 \cdot x_5 + x_1 \cdot x_2^5 \cdot x_4 + x_1 \cdot x_3 \cdot x_7 + x_7^2 + x_3 \cdot x_7 - x_1 \cdot x_3 - x_3^5$. It has tree-width 2 according to the following tree decomposition:



Note that there is no tree decomposition of width 1 because there are monomials involving 3 factors.

(iii) The polynomials

$$p_2(x) = \sum_{i=1}^{n} c_{i,i+_3 1,i+_3 2} x_i x_{i+_3 1} x_{i+_3 2}$$

and

$$p_3(x) = \sum_{i=1}^{n} d_{i,i+_3 1,i+_3 2} x_i^3 x_{i+_3 1}^5 x_{i+_3 2}$$

(where $+_3$ is addition $\pmod 3$) have tree-width 2.

(iv) For $p_1(x)$ from above, the polynomial $p_1^2(x)$ has tree-width $n-1$. This is so because all monomials appear and hence the 4-hypergraph associated with the polynomial is a hyperclique.

Boundedness of the tree-width is a sparsity condition. If $p(x)$ in n variables of degree d has tree-width k, the number of monomials is $O(n)$ with a constant depending on k, d only.

Remark 1. The tree-width of a polynomial depends on its particular representation. We can easily see that p_1 and p_1^2 have different tree-width but they represent the same variety. The tree-width of a system of polynomials which consists of a single polynomial does differ at most by 1 from the tree-width of the polynomial as such.

Similarly, we can define the tree-width of an $(n \times n)$ matrix $M = (m_{i,j})$.

Definition 4 *The tree-width of an $(n \times n)$ matrix $M = (m_{i,j})$ is the tree-width of the graph $G_M = \langle V_M, E_M \rangle$ with $V_M = \{1, 2, \dots, n\}$ and $(i, j) \in E_M$ iff $m_{i,j} \neq 0$.*

Examples 5
(1) The $(n \times n)$ *matrix* $M_1 = (m_{i,j})$ *with* $m_{i,j} = 1$ *for all* i, j *has tree-width* $n-1$. *Note that* M_1 *has linear rank* 1.
(2) The $(n \times n)$ *matrix* $\mathbf{1} = (m_{i,j})$ *with* $m_{i,i} = 1$ *and* $m_{i,j} = 0$ *for* $i \neq j$ *has tree-width* 0. *Note that* $\mathbf{1}$ *has linear rank* n.

Theorem 6 (Bodlaender).

a) *There is a linear time algorithm (with bad constants) which decides, given a hypergraph* G *whether it has a* k*-tree decomposition, and if yes, constructs one.*
b) *If a hypergraph* G *over* n *vertices has a* k*-tree decomposition, then one can construct in linear time a balanced* $O(k)$*-tree decomposition of depth* $O(logn)$.

Though originally formulated for graphs, the extension of the above results to hypergraphs is straightforward.

A survey of such results may be found in [Bod98,Bod97].

3 Generating functions of graph properties

The first use of tree-width of a matrix was presented in [CMR00b]. There the computational complexity of computing the *permanent* and *hamiltonian* of a matrix was studied.

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

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

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 $\{1, \ldots, n\}$. 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 cycle.

In general, both the *permanent* and the *hamiltonian* are hard to compute and the best algorithms known so far are exponential in n, [BCS97,Bür00]. This applies also for the computational model due to Blum, Shub and Smale (BSS model), cf. [BCSS98]. Barvinok in [Bar96] has shown that if the (linear) rank of the matrix is bounded by r both the *permanent* and the *hamiltonian* can be computed in polynomial (not linear) time. Hence these problems are parametrically tractable in the sense of [DF99]. Linear rank and tree-width are independent notions: The $(n \times n)$ matrix consisting of 1's only has rank 1 but tree-width $n-1$ (it is a clique). The corresponding unit matrix has rank n but tree-width 1, as the graph consists of isolated points. Tree-width of a matrix also makes the *permanent* and the *hamiltonian* parametrically tractable.

Theorem 7 (Courcelle, Makowsky, Rotics, 1998). *Let* M *be a real* $(n \times n)$ *matrix of tree-width* k. *Then* $per(M)$ *and* $ham(M)$ *can be computed in time* $O(n)$.

The same technique can also be applied to other families of multivariate polynomials such as *cycle format polynomials*, and, more generally, *generating functions of graph properties*, cf. [Jer87,Bür00].

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

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

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

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

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. The *hamiltonian*, similarly, is the generating function for $\mathcal{E}_{ham}$, the class of n-cycles.

The proof of theorem 7 relies on the observation that in these two (and many more) cases $\mathcal{E}$ is definable in Monadic Second Order Logic. This allows us to apply techniques first used in a more restrictive framework in [ALS91] and extended in [CMR00b]. For more details, cf. section 5.

In [GK87] $(n \times n)$-matrices M over $\{0, 1\}$ are considered where one knows in advance that the permanent is bounded by a polynomial, i.e. $per(M) \leq k \cdot n^q$ for some constants $k, q \in \mathbb{N}$. Grigoriev and Karpinski prove that under this assumption $per(M)$ can be computed in $\mathbf{NC^3}$, and hence in $\mathbf{P}$. To the best of our knowledge no similar result is known for $ham(M)$.

More about the complexity of generating functions of graph properties in the BSS framework can be found in [MaM00,Mak00].

4 Feasibility and positivity of polynomial systems

We now want to explore how far these techniques can be pushed further. We will state our main results; proofs then follow in the next sections.

Is the tree-width of a system of polynomials an appropriate tool to decide the existence of zeros?

We look at the following problems:

Definition 8 *Let $\mathbb{F}$ be a field (finite or infinite). Let $A \subseteq \mathbb{F}$ be finite of cardinality a. Let $p(x)$ be a polynomial in $\mathbb{F}[x]$ in the variables $x = (x_1, \dots, x_n)$ of degree d and tree-width k, and Σ be a system of such polynomials, whose $d+1$-hypergraph is of tree-width at most k.*

$(d,k) - FEAS_{\mathbb{F}}$**:** *Does p have a zero in $\mathbb{F}^n$?*
$(d,k) - FEAS(A)_{\mathbb{F}}$**:** *Does p have a zero in A^n?*
$(d,k) - HN_{\mathbb{F}}$**:** *Does Σ have a common zero in $\mathbb{F}^n$?*
$(d,k) - HN(A)_{\mathbb{F}}$**:** *Does Σ have a common zero in A^n?*

For $\mathbb{F}$ an ordered field we can also ask

$(d,k)-POS_{\mathbb{F}}$: *Is $p(r)>0$ for all $r \in \mathbb{F}^n$?*
$(d,k)-POS(A)_{\mathbb{F}}$: *Is $p(r)>0$ for all $r \in A^n$?*

If the tree-width is not bounded we write $(d,\infty)-FEAS(A)_{\mathbb{F}}$, etc. If R is an (finite, infinite, ordered) ring rather than a field, we use the analogous notation $(d,k)-FEAS(A)_R$, etc.

Remark 2. The finite set $A \subseteq \mathbb{F}$ can be viewed as condition of zero-dimensionality. A can be encoded by an additional polynomial which has exactly the elements of A as its zeros. Such problems are considered in [PRS93].

Remark 3. The way we have set up the definitions there is a minor discrepancy between $(d,k)-FEAS_{\mathbb{F}}$ and $(d,k)-HN_{\mathbb{F}}$ for systems consisting of one polynomial. We have $(d,k)-FEAS_{\mathbb{F}} \subseteq (d,k+1)-HN_{\mathbb{F}}$, but not necessarily $(d,k)-FEAS_{\mathbb{F}} \subseteq (d,k)-HN_{\mathbb{F}}$. This does not affect the results.

$(d,\infty)-FEAS_{\mathbb{F}}$, $(d,\infty)-HN_{\mathbb{F}}$ and, for ordered fields $(d,\infty)-POS_{\mathbb{F}}$ are discussed in [BCSS98,MM97]. In general they are $\mathbf{NP}_{\mathbb{F}}$ resp. co-$\mathbf{NP}_{\mathbb{F}}$ hard, and no subexponential algorithms are known for their solution. If we relativize the problems to $(d,\infty)-FEAS(A)_{\mathbb{F}}$ and $(d,\infty)-POS(A)_{\mathbb{F}}$ they are in $\mathbf{DNP}_{\mathbb{F}}$ resp. co-$\mathbf{DNP}_{\mathbb{F}}$, the classes *digital* $\mathbf{NP}$ *resp. digital co-*$\mathbf{NP}$ *over* $\mathbb{F}$.

It is known that $\mathbf{P}_{\mathbb{F}} \subseteq \mathbf{DNP}_{\mathbb{F}} \subseteq \mathbf{NP}_{\mathbb{F}}$ for any field $\mathbb{F}$ but it is not known whether the inclusions are proper, cf. [Poi95,MM97].

Problem 1. Are the problems $(d,\infty)-FEAS(A)_{\mathbb{R}}$ and $(d,\infty)-POS(A)_{\mathbb{R}}$ $\mathbf{DNP}_{\mathbb{R}}$ resp. co-$\mathbf{DNP}_{\mathbb{F}}$ complete over the reals? Is $(d,\infty)-HN_{\mathbb{C}}(A)$ is $\mathbf{DNP}_{\mathbb{C}}$ complete over the complex numbers?

The problem with proving one of these questions is the following: If we mimic the original proof of $\mathbf{NP}_{\mathbb{R}}$ completeness of $(4,\infty)-FEAS_{\mathbb{R}}$ in [BSS89] it does not apply to $(4,\infty)-FEAS_{\mathbb{R}}(\{0,1\})$ if we restrict to problems in $\mathbf{DNP}_{\mathbb{R}}$. The reason is that intermediate results of a computation are also used as guesses. They might be reals even though the initial guesses are zeros and ones. However, some seemingly hard problems can be reduced to $(4,\infty)-FEAS_{\mathbb{R}}(A)$ for $|A| \geq 2$. This includes for example the real Knapsack problem and, of course, also the classical SAT problem. It is therefore reasonable considering $(4,\infty)-FEAS_{\mathbb{R}}(A)$ to be a difficult problem in the BSS setting. Complete problems for $DNP_{\mathbb{R}}$ do exist ([CucMat]).

Our first main theorem is

Theorem 9. *For finite fields $\mathbb{F}$ of size f the problems $(d,k)-FEAS_{\mathbb{F}}$ and $(d,k)-HN_{\mathbb{F}}$ can be solved in time $O(n)$ where the constant depends on k,d,f.*
The same holds for finite rings.

The proof exploits the finiteness of the field (ring) by making all elements of it part of the underlying logic. In this way the problem becomes a problem of the weighted hypergraph of the non-vanishing monomials (with the coefficients as weights). The proof has no particular algebraic content, but shows that the notion of tree-width of systems of polynomials has surprising algebraic applications. Vardi pointed out that, using the methods of [FV99], the result can be sharpened a little by making it independent of the size of the finite field. Unfortunately, this does not help in the case of infinite structures, cf. the conclusion section.

The same technique as for the proof of Theorem 9 can also be used for infinite fields like $\mathbb{R}$ when dealing with so called $\exists$-MSO$_{\mathbb{R}}$ decision and evaluation problems (to be defined later on). This includes the evaluation of polynomials with exponentially many monomials like the permanent of a matrix of bounded tree-width and many other. We obtain

Theorem 10. *For $\mathbb{R}$-structures of tree-width at most k, $\exists$-MSO$_{\mathbb{R}}$-decision and evaluation problems can be solved in linear time (in the size of the structure).*

We next generalize the discussion of the feasibility and positivity problems to infinite fields. Here it turns out that the tree-width condition alone seems not to be strong enough to allow similar results. We can afford infinite fields (or rings), but restrict the set of tuples for which we want to evaluate the polynomial. This results in decision problems belonging to the class DNP$_{\mathbb{R}}$. Without bounded tree-width no polynomial algorithm is know for these problems.

We then prove

Theorem 11. *a) For arbitrary fields $\mathbb{F}$ the problem $(d,k)-FEAS(A)_{\mathbb{F}}$ can be solved in polynomial time in n where the constants of the polynomial bound depend on k, d and a condition concerning the number of different values the monomials take over A.*

b) For ordered fields $\mathbb{F}$ the problem $(d,k)-POS(A)_{\mathbb{F}}$ can be solved in polynomial time in n where the constants of the polynomial bound depend on k, d and a condition concerning the number of different values the monomials take over A.
The same holds for any ordered ring R.

The condition concerning the coefficients and A basically requests that the set of partial sums of the monomials evaluated in A be bounded by a polynomial function in n.

A similar theorem can be proved also for $(d,k) - PIS(A)_R$, systems of polynomial inequalities of polynomials of bounded degree $d \geq 2$ over an ordered ring R.

The most general result on infinite fields in this paper is on $\exists$-MSO$_{\mathbb{R}}$ extended decision problems (to be defined):

Theorem 12. *For structures of tree-width at most k $\exists$-MSO$_{\mathbb{R}}$-extended decision problems can be solved in time $O(n \cdot t(n))$ provided the number of possible values of the subterms appearing in the formalization of the problem is bounded by $O(t(n))$ where n is the size of the structure.*

Finally, we discuss how the coefficient condition mentioned in Theorems 11 and 12 in some cases can be avoided. A kind of linearization of the positivity problem in fact allows us to strengthen Theorem 11, b). Here, by linearization we mean that the weight terms are linear in the variables. We obtain

Theorem 13. *For ordered fields $\mathbb{F}$ the problem $(d,k) - POS(A)_{\mathbb{F}}$ can be solved in linear time in n where the constants of the linear bound depend on k, d.*
The same holds for any ordered ring R.

5 Meta-finite Monadic Second Order Logic

In this and the next section the main mathematical ideas behind the results stated above are developed. Complete proofs are given.

Before going into details let us first outline the overall foregoing: the major point of our approach is to extend the (previously known) handling of MS_2 properties over finite structures to algebraic issues, i.e. decision and evaluation problems as they naturally appear in the Blum-Shub-Smale framework. To this aim we consider problem instances as specific finite, relational structures together with real-valued weight functions. The latter are called $\mathbb{R}$-structures in [GG98,GM96]. Problems are then given as conjunction of two formulas; one is expressed in monadic second order logic MS_2 over the underlying finite structure and the other is given in existential monadic second order logic over the corresponding $\mathbb{R}$-structure (a logic to be defined). This generalizes the framework of *Extended Monadic Second order Logic EMSOL* proposed in [ALS91] and unifies it with the framework of *Meta-finite Model Theory* of [GG98].

On the underlying finite structure we evaluate MS_2 properties using the Feferman-Vaught theorem. The latter is rigorously proved in the Appendix together with its use for algorithmic purposes. Those readers being not familiar with it are strongly encouraged to study the Appendix before continuing.

The results of this paper rely on a careful definition of the meta-finite monadic second order logic for expressing algebraic issues. It has to be strong enough to capture interesting problems; on the other hand, it must be defined in such a way that we can perform a decomposition on the meta-finite structure in parallel to the one given by the bounded tree-width decomposition and the Feferman-Vaught theorem on the underlying finite structure. This will be done now.

5.1 Hypergraphs and parse trees

Let us formalize the kind of meta-finite structures and decompositions of them we are interested in. Some missing proofs are in the Appendix.

We consider problem instances as logical structures representing particular hypergraphs. To capture the combinatorial aspects of a problem we start with a finite relational structure $(V, E, R_1, \ldots, R_\ell)$ of signature τ. Here, (V, E) is a hypergraph, i.e. $V := \{1, \ldots, n\}$ for some $n \in \mathbb{N}$ and any element of E is a (non-void) subset of V (of arbitrary but fixed arity). Every relation R_i is a subset of V^{n_i}, where $n_i \in \mathbb{N}$ denotes its corresponding arity.

We consider $(V, E, R_1, \ldots, R_\ell)$ as a two-sorted structure with universe $V \uplus E$; by convention, there is a relation $R_{inc} \subseteq V \times E$ among the symbols in τ which gives the incidence relation between vertices and edges. If more comfortable, we can also choose $V \uplus V^2$ as the two-sorted universe and include a relation $R_E \subseteq V^2$ representing the edge set, see below.

The monadic second order logic $MS_2(\tau)$ over hypergraphs is defined as sublogic of second order logic, where we allow quantified and free second order variables of arity 1 only. In addition, set variables range over subsets of V or E.

Example 14 *(cf. [CMR00b]). In order to give an idea about the expressiveness of the logics we are using two examples related to graph properties are studied. As we will see both can be captured by MS_2 logic.*

We will be very precise here concerning the logical representation of graphs as structures. Later on, we will just keep in mind how to represent (weighted) hypergraphs as logical structures without going through all the details every time. Firstly, a graph $G = (V, E)$ is represented as a two-sorted finite structure. The universe U of this structure consists of the union of two sorts, namely the vertex set V and the set V^2 of possible edges (sometimes, we also directly take E as second sort). Several relations are included in the structure whose interpretation will give the graph G. More precisely, a unary relation $R_V \subseteq U$ is interpreted as $R_V(x) \Leftrightarrow x \in V$, a unary relation $R_E \subseteq U$ is interpreted as $R_E(e) \Leftrightarrow e \in E$, and a binary incidence relation R_{inc} is interpreted as $R_{inc}(v, e) \Leftrightarrow R_V(v) \wedge R_E(e) \wedge v$ is incident with e. Thus, the graph G is represented as the finite structure $\langle V \cup V^2, R_V, R_E, R_{inc}\rangle$. We can as well add unary predicates either on V or on V^2 (the two sorts); for example, such unary relations might indicate labels of either the vertices or the edges.

Note that for a subset $X \subseteq U$ we can easily express in MS_2 logic that $X \subseteq V$ (resp. $X \subseteq E$) by

$$\forall x \in U \; X(x) \;\Rightarrow\; R_V(x) \text{ resp. } \forall x \in U \; X(x) \;\Rightarrow\; R_E(x) \; .$$

This will be used implicitly throughout the paper.

a) *3-Colorability: given a graph G as $\langle V \cup V^2, R_V, R_E, R_{inc}\rangle$ we define a MS_2 formula Φ as follows:*

$$\begin{aligned} \Phi \equiv \; & \exists \; X_1 \subseteq V, X_2 \subseteq V, X_3 \subseteq V \text{ such that} \\ & PART(X_1, X_2, X_3) \wedge NO\text{-}EDGE(X_1) \wedge NO\text{-}EDGE(X_2) \wedge NO\text{-}EDGE(X_3). \end{aligned}$$

Here, $PART(X_1, X_2, X_3)$ is true if and only if the sets X_1, X_2, X_3 build a partition of V, i.e.

$$\begin{aligned} PART(X_1, X_2, X_3) \equiv \; & \forall v \in V \, (X_1(v) \vee X_2(v) \vee X_3(v)) \; \wedge \\ & \neg\exists \; u \in V \, (\{X_1(u) \wedge X_2(u)\} \vee \{X_1(u) \wedge X_3(u)\} \vee \\ & \{X_2(u) \wedge X_3(u)\}) \end{aligned}$$

and $NO\text{-}EDGE(X)$ is true if and only if no two vertices in X are incident with the same edge of G :

$$NO\text{-}EDGE(X) \equiv \forall u, v \in V \, (X(u) \wedge X(v) \Rightarrow \neg E(u, v))$$

Clearly, G is 3-colorable iff $G \models \Phi$. In the above example we do not really need the full power of $MS_2(\tau)$ logic; the quantification is just extending to subsets of the first sort V of the universe. Therefore we could represent G as well as a one-sorted structure only and still express colorability. This will be different with the next example.

b) *Another MS_2-definable property is that of being a perfect matching. This time we need set quantifiers over both sorts of the universe. Again, let a graph G being given as $\langle V \cup V^2, R_E, R_V, R_{inc}\rangle$. For a subset $X \subseteq V^2$ we want to find a MS_2 formula $\Phi(X)$ such that $(G, X) \models \Phi(X)$ iff X is a perfect matching of G. Using R_{inc} we can either deal with an edge $e \in E$ or with the two incident vertices; just consider the MS_2 formula*

$$\varphi(u,v,e) \equiv R_{inc}(u,e) \wedge R_{inc}(v,e) \wedge u \neq v$$

which gives the incident vertices of an edge. Having this in mind we define the formula

$$\begin{aligned}\Phi(X) \equiv \quad & \forall u,v, \in V\,\{(u,v) \in X \Longrightarrow E(u,v)\}\\ \wedge \ & \forall v \in V\ \exists u \in V\,\{(u,v) \in X \ \vee \ (v,u) \in V\}\\ \wedge \ & \forall u,v,w \in V\,\{((u,v) \in X \ \wedge \ (u,w) \in X) \Longrightarrow v = w\}\\ \wedge \ & \forall u,v,w \in V\,\{((v,u) \in X \ \wedge \ (w,u) \in X) \Longrightarrow v = w\}\end{aligned}$$

which satisfies the properties we are looking for.

Besides the combinatorial part of our structures the use of weights in some algebraic structure (ring, field, ordered ring, etc.) has to be incorporated. To simplify our notation we assume here that weights are in the ordered field of real numbers $\mathbb{R}$. Structures of this kind are particular $\mathbb{R}$-structures in the sense of [GG98,GM96].

Towards this aim a weight function $C : E \mapsto \mathbb{R}$ is added to the structure, thereby turning it into a meta-finite one. The ordered structure
$(\mathbb{R}, +, *, \leq\ , 1, 0, -1, r_1, \ldots, r_s)$ is included as well. Here, the r_i are fixed real constants.
We could also think about more than one weight function or weights on the vertices, but the above is sufficient for our purposes.

The properties which will be checked on such $\mathbb{R}$-structures are twofold. One is combinatorial and expressed by a $MS_2(\tau)$ formula. The other involves the weight functions and the real number part. It is given as well as a specific monadic second order property, this time defined for the real number part of the structure.

For structures of bounded tree-width one major ingredient of our algorithm is a decomposition. This is first done on the underlying finite structure (the hypergraph without weights).

Definition 15 (cf. [DF99]) *A hypergraph G is k-boundaried if exactly k of its vertices are labelled by $\{1, \ldots, k\}$ (i.e. every label appears with one vertex). The labelled vertices are called the boundary $\delta(G)$ of G.*

We next define the gluing operations we are interested in:

a) **create** : this operation creates a $k+1$ boundaried hypergraph with no edges (i.e. $k+1$ vertices all of which are labelled);
b) **join** : the join $\oplus$ operates on two $k+1$ boundaried hypergraphs $G_1 = (V_1, E_1)$ with boundary $\delta(G_1)$ and $G_2 = (V_2, E_2)$ with boundary $\delta(G_2)$. The graph $G_1 \oplus G_2$ is obtained by joining V_1 and V_2 in such a way that those vertices in $\delta(G_1)$ and $\delta(G_2)$ having the same label are identified. The vertices in $\widehat{V_1} := V_1 \setminus \delta(G_1)$ and those in $\widehat{V_2} := V_2 \setminus \delta(G_2)$ obtain an own copy. After this identification the hyperedges E_1 and E_2 are unified;
c) $\mathbf{change}_{i,j}(G)$: in the $k+1$ boundaried hypergraph G labels i and j are interchanged;
d) $\mathbf{add}_{i_1,i_2,\ldots,i_s}(G)$: adds an hyperedge between vertices with labels $i_1, i_2, \ldots, i_s$;
e) $\mathbf{new}_i(G)$: adds a new vertex, labels it i and removes label i from the previously labelled vertex.

Remark 4. Later on we decompose a given structure into substructures using the above operations in an inverse manner. Note that in that situation a decomposition is not always unique. Most important for our purposes is the case where a structure G is divided into substructures A and B such that $G = A \oplus B$. We fix by convention that hyperedges being made of vertices in the common boundary $\delta(A) = \delta(B)$ then are only transferred to substructure A, not to B.

Definition 16 *Let G be a $k+1$ boundaried hypergraph. A parse-tree for G is a tree whose vertices are labelled by one of the above operators such that the following holds:*

- *the leafs are labelled by create;*
- *the branch nodes are labelled by $\oplus$;*
- *the other nodes are labelled by one of the other operations;*
- *G is the hypergraph obtained at the root after performing all the operations along the tree bottom-up.*

The main relation between parse-trees and hypergraphs of bounded tree-width is the following:

Theorem 17 (cf. [DF99]). *Let G be a hypergraph of tree-width at most k. Starting from a tree-decomposition of G we can compute in linear time w.r.t. $|G|$ a parse-tree of G.*

A tree-decomposition itself can also be obtained in linear time according to theorem 6.

The Feferman-Vaught theorem gives information about the way $MS_2(\tau)$ properties on a finite structure can be evaluated on substructures if the relation between these structures are given by one of the above operations. We state it here for the join operation $\oplus$. The proof can be found in the Appendix.

Let $\mathfrak{A}$ and $\mathfrak{B}$ be $k+1$ boundaried hypergraphs over τ with universes A and B resp., and boundaries $\delta(A) \subset A, \delta(B) \subset B, \widehat{A} := A \setminus \delta(A), \widehat{B} := B \setminus \delta(B)$. Let $\mathfrak{C} := \mathfrak{A} \oplus \mathfrak{B}$ be the join of $\mathfrak{A}$ and $\mathfrak{B}$.

Let z be an assignment of the variables in a $MS_2(\tau)$ formula $\Phi(\underline{x}, \underline{y}, \underline{w}, \underline{X})$ into the universe C of $\mathfrak{C}$ such that the following is true:

- z maps the variables x_i of block $\underline{x}$ to the set $\widehat{A}$
- z maps the variables y_i of block $\underline{y}$ to the set $\widehat{B}$
- z maps the variables w_i of block $\underline{w}$ to the boundary $\delta(A) = \delta(B)$ (recall that the two boundaries were identified). In particular, $z(w_i)$ is member of both the universes A and B.

Denote by z_A, z_B the assignments with $z_A(X) = X \cap A, z_A(s) = z(s)$ for $s \in A$ and $z_B(X) = X \cap B, z_B(s) = z(s)$ for $s \in B$.

Theorem 18 (Feferman-Vaught).

For every $MS_2(\tau)$ formula $\psi(\underline{x}, \underline{y}, \underline{w}, \underline{X})$ there are finitely many so called Hintikka formulas $h_{1,\alpha}(\underline{x}, \underline{w}, \underline{X})$ and $h_{2,\alpha}(\underline{y}, \underline{w}, \underline{X})$ such that for every τ-structure $\mathfrak{C}$ given as join $\mathfrak{A} \oplus \mathfrak{B}$ and every assignment z as above we have

$$(\mathfrak{C}, z) \models \psi(\underline{x}, \underline{y}, \underline{w}, \underline{X}) \;\Leftrightarrow\; \exists\ \alpha\ (\mathfrak{A}, z_A) \models h_{1,\alpha}(\underline{x}, \underline{w}, \underline{X}) \wedge (\mathfrak{B}, z_B) \models h_{2,\alpha}(\underline{y}, \underline{w}, \underline{X}).$$

If ψ itself is a Hintikka formula then there exist uniquely determined Hintikka formulas h_1 and h_2 such that

$$(\mathfrak{C}, z) \models \psi(\underline{x}, \underline{y}, \underline{w}, \underline{X}) \;\Leftrightarrow\; (\mathfrak{A}, z_A) \models h_1(\underline{x}, \underline{w}, \underline{X}) \wedge (\mathfrak{B}, z_B) \models h_2(\underline{y}, \underline{w}, \underline{X}).$$

5.2 MSO logic over $\mathbb{R}$-structures

In order to extend the Feferman-Vaught approach to meta-finite structures we have to look for an appropriate definition of monadic second order logic $\mathrm{MSO}_{\mathbb{R}}$ for these structures. The major task is to define it in such a way that the decomposition operations of section 5.1 extend in a natural manner also to weighted structures and the $\mathrm{MSO}_{\mathbb{R}}$ formulas.

Let us inductively define terms and formulas for obtaining this monadic second order logic over weighted hypergraphs (V, E, C). In each step we first define the class of terms or formulas which will be used. Then, it is shown how the corresponding terms and formulas can be decomposed to the substructures involved. We restrict the presentation again to the $\oplus$ operation. For the other parse operations similar statements hold true.

Simple terms are of the following form:

- for any edge $e \in E$ the expression $C(e)$ is a simple term;
- the constants $r_1, \ldots, r_s \in \mathbb{R}$ are simple terms;
- if t_1 and t_2 are simple terms so are $t_1 + t_2$ and $t_1 \cdot t_2$.

Thus, simple terms are of the form $pol(C(e_1), \ldots, C(e_m))$ for a polynomial pol and edges $e_1, \ldots, e_m$.

Decomposition: For a decomposition $\mathfrak{C} = \mathfrak{A} \oplus \mathfrak{B}$ a term $C(e)$ is taken over to that substructure e belongs to. Note that for $e \in \mathfrak{P}(\delta(A))$ by convention e appears in the second sort of the substructure $\mathfrak{A}$ only. The constants can be used in both substructures. A polynomial $pol(C(e_1), \ldots, C(e_m))$ is evaluated on $\mathfrak{C}$ by first evaluating $C(e_i), 1 \leq i \leq m$ (and further parts as far as possible) on the corresponding substructure and then putting together the results on $\mathfrak{C}$. Note that the complexity of the latter step only depends on the size of the polynomial pol, not on $\mathfrak{C}$.

Summation and product terms are expressions of the following form:

- for a subset $U \subseteq E$ a summation term has the form $T(U) := \sum\limits_{e \in U} C(e)$ and a product term has the form $T(U) := \prod\limits_{e \in U} C(e)$; for the empty set we can define $T(\emptyset)$ to be any fixed real value.
- if $T_1(U_1)$ and $T_2(U_2)$ are summation and product terms so are $T_1(U_1) + T_2(U_2)$ and $T_1(U_1) \cdot T_2(U_2)$.

Decomposition: For $\mathfrak{C} = \mathfrak{A} \oplus \mathfrak{B}$ and $T(U) = \sum\limits_{e \in U} C(e)$ denote $U^A := U \cap E^A$ and $U^B := U \cap E^B$. Then $T_1(U^A) := \sum\limits_{e \in U^A} C(e)$ and $T_2(U^B) := \sum\limits_{e \in U^B} C(e)$ can be evaluated on $\mathfrak{A}$ and $\mathfrak{B}$ resp. Their sum gives the result of T on $\mathfrak{C}$. The same holds true for $T(U) := \prod\limits_{e \in U} C(e)$. Polynomials in summation or product terms are evaluated as it was explained for simple terms.

Remark 5. Note that according to our convention $E^A \cap E^B = \emptyset$. Thus, there will be no hyperedge taken into account twice. We could have modeled the decomposition by including all hyperedges relating vertices from the boundary in both substructures and giving them the weight 0 in one of it. But this would raise some technical problems; for example, in a product term we would obtain the value 0 above.

Min/max terms are defined as summation and product terms but using

$$T(U) := \max_{e \in U} C(e) \text{ and } T(U) := \min_{e \in U} C(e)$$

instead of $\sum$ and $\prod$.

Monadic second order $\text{MSO}_{\mathbb{R}}$ **terms** are all terms above together with terms of the following form:

- for a summation or product term of the form $T(U)$ (not a polynomial in such terms!) and a MS_2 formula ϱ on the underlying finite structure the term $\sum_{U, \varrho(U)} T(U)$ is a $\text{MSO}_{\mathbb{R}}$ term;
- for a summation or product term of the form $T(U)$ and a univariate polynomial *pol* the term $\sum_{U, \varrho(U)} pol(T(U))$ is a $\text{MSO}_{\mathbb{R}}$ term;
- for $1 \leq i \leq m$ let $T_i(U_i) := \prod_{e \in U_i} C(e)$ be product terms and $\varrho(X_1, \ldots, X_m)$ be a MS_2 formula on the underlying finite structure. Then $\sum_{(U_1, \ldots, U_m), \varrho(\underline{U})} T_1(U_1) \cdot \ldots \cdot T_m(U_m)$ is a $\text{MSO}_{\mathbb{R}}$ term;
- sums and products of $\text{MSO}_{\mathbb{R}}$ terms are $\text{MSO}_{\mathbb{R}}$ terms.

Decomposition: Let $\mathfrak{C} = \mathfrak{A} \oplus \mathfrak{B}$ and consider as example a $\text{MSO}_{\mathbb{R}}$ term of the form $\sum_{(U_1, \ldots, U_m), \varrho(\underline{U})} T_1(U_1) \cdot \ldots \cdot T_m(U_m)$, where all T_i are product terms. The evaluation on $\mathfrak{C}$ uses the Feferman-Vaught theorem for ϱ as well as evaluations on $\mathfrak{A}$ and $\mathfrak{B}$ as follows. W.l.o.g. suppose ϱ to be a Hintikka formula and h_1 as well as h_2 to be the corresponding unique formulas for $\mathfrak{A}$ and $\mathfrak{B}$ given by theorem 18. Let $\underline{U}_j := (U_{1j}, \ldots, U_{mj}), j \in J$ be those assignments with $(\mathfrak{C}, \underline{U}_j) \models \varrho(\underline{U}_j)$. Furthermore, let $\underline{U}_j^A$ and $\underline{U}_j^B$ be its corresponding decompositions for the substructures $\mathfrak{A}$ and $\mathfrak{B}$, see the definition of $\oplus$ and remark 4.
Now

$$\begin{aligned}
&\sum_{\underline{U}, \varrho(\underline{U})} T_1(U_1) \cdot \ldots \cdot T_m(U_m) = \sum_{j \in J} T_1(U_{1j}) \cdot \ldots \cdot T_m(U_{mj}) \\
&= \sum_{j \in J} T_1(U_{1j}^A) \cdot T_1(U_{1j}^B) \cdot \ldots \cdot T_m(U_{mj}^A) \cdot T_m(U_{mj}^B) \\
&= \sum_{j \in J} T_1(U_{1j}^A) \cdot \ldots \cdot T_m(U_{mj}^A) \cdot T_1(U_{1j}^B) \cdot \ldots \cdot T_m(U_{mj}^B) \\
&= \left(\sum_{\underline{U}^A, h_1(\underline{U}^A)} T_1(U_1^A) \cdot \ldots \cdot T_m(U_m^A) \right) \cdot \left(\sum_{\underline{U}^B, h_2(\underline{U}^B)} T_1(U_1^B) \cdot \ldots \cdot T_m(U_m^B) \right)
\end{aligned}$$

where the latter equality holds because of the Feferman-Vaught theorem.
For the other types of $\text{MSO}_{\mathbb{R}}$ terms the decomposition can be done in the same manner.

Remark 6. The above definition of $\text{MSO}_{\mathbb{R}}$ terms probably is not the most general one. However, the reader might check the difficulties arising when performing a decomposition for a term like $\prod_{\underline{U}, \varrho(\underline{U})} \sum_{e \in U} C(e)$ or one like $\sum_{(U_1, U_2), \varrho(U_1, U_2)} T_1(U_1) + T_2(U_2)$ where T_1 and T_2 are product terms. The above setting nevertheless is strong enough to capture important problems.

For the other operations $op \in \{add_{i_1,\dots,i_s}, change_{ij}, new_i\}$ the evaluation of a $\mathrm{MSO}_{\mathbb{R}}$ term on a structure $op(\mathfrak{C})$ can be done similarly by first evaluating a corresponding term on $\mathfrak{C}$ and then extending the result to $op(\mathfrak{C})$.

After having introduced monadic second order terms we turn to monadic second order formulas built from these terms.

Basic $\mathrm{MSO}_{\mathbb{R}}$ formulas are expressions of the form $T(v_1 \dots, v_k, U_1, \dots, U_t)\Delta 0$ with $\Delta \in \{=, >, \geq\}$; here, T is a monadic second order term, the v_i are elements of the universe and the U_i are subsets of E.

$\mathrm{MSO}_{\mathbb{R}}$ formulas are all basic $\mathrm{MSO}_{\mathbb{R}}$ formulas together with

- if $\psi(v_1, \dots, v_k, W_1, \dots, W_t)$ is a $\mathrm{MSO}_{\mathbb{R}}$ formula and $U_1, \dots, U_k$ are subsets of E, then $\varrho(W_1, \dots, W_t) \equiv \bigwedge\limits_{v_i \in U_i} \psi(v_1, \dots, v_k, W_1, \dots, W_t)$ is a $\mathrm{MSO}_{\mathbb{R}}$ formula;
- if $\psi(v_1, \dots, v_k, U_1, \dots, U_s, W_1, \dots, W_t)$ is a $\mathrm{MSO}_{\mathbb{R}}$ formula and $\varrho(v_1, \dots, v_k, U_1, \dots, U_s)$ is a $MSOL(\tau)$ formula then $\phi(W_1, \dots, W_t) \equiv \bigwedge\limits_{\underline{v}, \underline{U}, \varrho(\underline{v}, \underline{U})} \psi(\underline{v}, \underline{U}, W_1, \dots, W_t)$ is a $\mathrm{MSO}_{\mathbb{R}}$ formula (for $W_i \subseteq E$);
- the closure of the above construction scheme under logical conjunction, disjunction and negation gives the set of $\mathrm{MSO}_{\mathbb{R}}$ formulas.

$\exists$-$\mathrm{MSO}_{\mathbb{R}}$ (Existential monadic second order logic) over $\mathbb{R}$-structures is obtained from $\mathrm{MSO}_{\mathbb{R}}$ logic by defining

- all formulas in $\mathrm{MSO}_{\mathbb{R}}$ to belong to $\exists$-$\mathrm{MSO}_{\mathbb{R}}$
- if $\psi(W_1, \dots, W_t) \in \exists$-$\mathrm{MSO}_{\mathbb{R}}$ then $\exists W_1, \dots, W_t\ \psi(W_1, \dots, W_t) \in \exists$-$\mathrm{MSO}_{\mathbb{R}}$.

The problems considered in this paper are of the following type:

Decision problems: For a fixed MS_2 formula ψ, decide whether $\mathcal{G} \models \psi$.
Evaluation problem: For a fixed $\mathrm{MSO}_{\mathbb{R}}$ term T, compute its value over $\mathcal{G}$.
Extended decision problem: Given a $MS_2(\tau)$ formula ψ as well as a $\exists$-$\mathrm{MSO}_{\mathbb{R}}$ formula Φ we want to decide whether $\mathcal{G} \models \psi \wedge \Phi$.
Optimization problems: These are like the extended decision problems, but with Φ quantifier free, but possibly involving the functions max and min.

For other fields $\mathbb{F}$ the definition of $\exists$-$\mathrm{MSO}_{\mathbb{F}}$ is done on a similar way.

6 Proofs

After clarifying the above way to define $\exists$-$\mathrm{MSO}_{\mathbb{R}}$ logic by dealing with some examples we will turn to rigorous proofs of the theorems stated in section 4.

6.1 Guiding examples

Let us consider some examples and the way they fit into the formal setting of the previous section.

Example 19 *The generating functions of section 3*

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

with $\mathcal{E}$ $MSOL$-definable by $\psi(E')$ can be written as a Monadic Second Order Term

$$\sum_{\psi(E') \wedge E' \subseteq E} \prod_{e \in E'} w(e)$$

Hence they are $\exists$-$\mathrm{MSO}_{\mathbb{R}}$-evaluation problems.

Example 20 **(#4-$FEAS(A)$)** *In order to formalize the #4-$FEAS(A)$ problem as an $\exists$-$\mathrm{MSO}_{\mathbb{R}}$ extended decision problem we use the representation of degree 4 polynomials as given in [GM96]: let $V = \{0, 1, \ldots, n\}, E := V^4$ and $C : E \to \mathbb{R}$ be a weight function giving the coefficients of f in the following sense: for $(i, j, k, l) \in E$ the value $C(i, j, k, l)$ is the coefficient of the monomial $x_i \cdot x_j \cdot x_k \cdot x_l$ in f. Note that here we assume f to be homogeneous of degree 4. Later on f is dehomogenized by adding the condition $x_0 := 1$.*

Let $A := \{s_1, \ldots, s_m\}$; for every $1 \le i \le m$ define a function $w_i : V \to A$ such that $\forall x \in V \; w_i(x) := s_i$. We are looking for disjunct subsets $U_1, \ldots, U_m$ of V such that the following holds: $\bigcup_{i=1}^{m} U_i = V$ and if we assign to every $x \in U_i$ the value $w_i(x) = s_i$ then this assignment of variables gives a zero of f.

For any quadruple $\lambda = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)$ where $\varepsilon_i \in \{1, \ldots, m\}$ define the set $E_\lambda := E \cap (U_{\varepsilon_1} \times U_{\varepsilon_2} \times U_{\varepsilon_3} \times U_{\varepsilon_4})$ (i.e. a point $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) \in V^4$ belongs to E_λ iff every component α_i lies in U_{ε_i}).

For $\alpha \in E_\lambda$ the corresponding monomial $C(\alpha) \cdot x^\alpha$ gives the value $C(\alpha) \cdot s_{\varepsilon_1} \cdot s_{\varepsilon_2} \cdot s_{\varepsilon_3} \cdot s_{\varepsilon_4}$ under the above assignment. The $MS_2(\tau)$ formula for #4-$FEAS(A)$ simply is

$$\psi \equiv \exists U_1, \ldots, U_m \subseteq V \text{ such that } \forall i, j \in \{1, \ldots, m\} U_i \cap U_j = \emptyset \wedge U_1 \cup \ldots \cup U_m = V$$

(note that m is independent of the structure).

The real number part of the logical description is $\Phi \equiv \sum_{\lambda \in \{1, \ldots, m\}^4} T(E_\lambda) = 0$, where $T(E_\lambda) := \sum_{\alpha \in E_\lambda} C(\alpha) \cdot s_{\varepsilon_1} \cdot s_{\varepsilon_2} \cdot s_{\varepsilon_3} \cdot s_{\varepsilon_4}$ is a summation/product term. Again note that m is independent of the structure; hence, the same is true for the size of the first sum above. For dehomogenization one can introduce a further subset $U_0 \subseteq V$ which only consists of the element 0 and put $s_0 := 1$. It follows that f has a zero in A if and only if $(V, E, C) \models \psi \wedge \Phi$.

Note that over a finite *field $\mathbb{F}$ this logical description works as well taking $A := \mathbb{F}$, but could of course be simplified. We used the above description in order to cover already the infinite field case we are interested in later on.*

In a completely similar fashion the feasibility of polynomial inequality systems 2-$PIS(A)_{\mathbb{R}}$ fits into the framework of $\exists$-$\mathrm{MSO}_{\mathbb{R}}$ extended decision problems and $(d, \infty) - POS(A)_{\mathbb{F}}$ can be coded as an optimization problem. If the ring (field) is finite the coding yields an $\exists$-$\mathrm{MSO}_{\mathbb{R}}$ decision problem.

Example 21 *Another example fitting into the framework is the computation of the determinant of a square matrix $A \in \mathbb{R}^{n \times n}$ (whose computation is not that interesting in its own* *but in relation with more complicated problems like, for example, the linear programming problem). It can be expressed as a* $\mathrm{MSO}_{\mathbb{R}}$ *term as follows.*

Since A *is not necessarily symmetric we represent it as a directed weighted graph* (G, c) *with vertices* $V := \{1, \ldots, n\}$*. The weight function* $c : V^2 \to \mathbb{R}$ *for* $(i, j) \in V^2$ *gives the value* $c(i, j) := a_{ij}$*. This implies that in the model theoretic representation of* A *we use a vocabulary* τ *with two incidence relations* R_1, R_2 *instead of* R_{inc} *(and the obvious meaning:* $R_1(i, e) \Leftrightarrow i$ *is the start node for edge* e*). Moreover, we include a linear order* $<$ *on* V *(with the same interpretation as the natural order* $1 < 2 < \ldots < n$*). The interested reader might try to define such an order in* MS_2 *logic.*

In the first step we now define a formula $PERM(\Pi)$ *such that* $(G, \Pi) \models PERM(\Pi)$ *iff* $\Pi \subseteq V^2$ *is a permutation of* V *(where* $(i, j) \in \Pi$ *stands for* $\Pi(i) = j$*). This formula can easily be written down in* MS_2 *logic.*

Secondly, for a permutation Π *we define the set* $INV(\Pi) \subseteq V$ *of its inversions by*

$$i \in INV(\Pi) \;\Leftrightarrow\; \exists j \in V : (i, j) \in \Pi \;\wedge\; i > j \;.$$

Here, we use the linear order on V*. Thirdly, we need two product terms* T_1 *and* T_2*; the first is defined on subsets* U *of* V *:*

$$T_1(U) := \prod_{i \in U} (-1) \qquad (and\ T_1(\emptyset) := 1);$$

the second is defined on subsets Π *of* V^2 *(the second sort of the universe)*

$$T_2(\Pi) := \prod_{(i,j) \in \Pi} c(i, j) \;.$$

If now $\varrho(U, \Pi)$ *is the* MS_2 *formula expressing*

$$\varrho(U, \Pi) \equiv U \subseteq V \;\wedge\; \Pi \subseteq V^2 \;\wedge\; U = INV(\Pi) \;\wedge\; PERM(\Pi)$$

we obtain the determinant of A *as*

$$DET(A) \;=\; \sum_{U, \Pi, \varrho(U, \Pi)} T_1(U) \cdot T_2(\Pi)$$

which is a $\mathrm{MSO}_{\mathbb{R}}$ *term according to the definition of section 5.2.*

Although the determinant of a matrix can be computed quickly, the example can be modified, such as to include the permament and other matrix functions, which are #P hard to compute, cf. [CMR00b].

We will now turn to the proof of Theorem 9.

Proof of Theorem 9:

We restrict ourselves to the case $(4, k) - FEAS_{\mathbb{F}}$. The other cases can be treated in the same way. Let the input polynomial be represented as in example 20 by a structure $\mathfrak{D} = (V, E, C)$ of tree-width at most k and size n; let ψ and Φ be the corresponding MS_2 and $\exists$-$\mathrm{MSO}_{\mathbb{F}}$ formulas.

Perform the Feferman-Vaught decomposition of $\mathfrak{D}$ and ψ according to theorem 18. Suppose ψ to be a disjunction $\bigvee_\gamma h_\gamma$ of Hintikka formulas h_γ. We check for each of it whether $\mathfrak{D} \models h_\gamma \wedge \Phi$ holds. With respect to the MS_2 formulas h_γ we do this as shown in the Appendix: We precompute all triples of Hintikka formulas of a given quantifier rank and number of variables which are linked by the Feferman-Vaught theorem according to the gluing operation $\oplus$. The corresponding tables are precomputed as well for the other operations. Climbing up the parse tree for $\mathfrak{D}$ by looking into these precomputed tables we check in linear time whether $\mathfrak{D} \models h_\gamma$.

For the $\exists$-MSO$_{\mathbb{F}}$ formula Φ we proceed as follows: first, we decompose the MS_2 formulas involved in Φ in the same way using Theorem 18. Next, we begin the evaluation at the leave structures of the parse tree for $\mathfrak{D}$. Since all of them have a universe of size at most k we can use a brute algorithm which evaluates the corresponding MSO$_{\mathbb{R}}$ terms for the leave structures by considering all possible choices for second order variables. This takes exponential time only in the fixed parameter k. We store the computed results in order to use them again on the further structures appearing when we glue substructures and climb up the parse tree.

Note that there are at most $|\mathbb{F}|$ many different subresults, i.e. a constant number.

We continue along the parse tree until the root, representing $\mathfrak{D}$, is reached. We check the possible results of the evaluation (which are at most $|\mathbb{F}|$ many) and decide whether $\mathfrak{D} \models h_\gamma \wedge \Phi$ for all γ. Climbing up the tree according to the decomposition has running time $O(n)$.

A technical problem appears during the evaluation of MSO$_{\mathbb{F}}$ terms. Here, joining two substructures $\mathfrak{A}$ and $\mathfrak{B}$ and identifying their boundaries makes it necessary to consider only assignments treating the boundary elements the same in $\mathfrak{A}$ and $\mathfrak{B}$. If we decompose a structure top down that's of course no problem. But in the above algorithm we compute bottom up. Suppose we want to evaluate a term $\sum_{\underline{U},\varrho(\underline{U})} T(\underline{U})$ over $\mathfrak{A}\oplus\mathfrak{B}$ using the Feferman-Vaught theorem. Suppose furthermore w.l.o.g. ϱ to be a Hintikka formula and

$$(\mathfrak{A} \oplus \mathfrak{B}, \underline{U}) \models \varrho(\underline{X}) \;\Leftrightarrow\; (\mathfrak{A}, \underline{U}^A) \models h_1(\underline{X}) \;\wedge\; (\mathfrak{B}, \underline{U}^B) \models h_2(\underline{X}) \;.$$

If we evaluate bottom up then only assignments $\underline{U}^A, \underline{U}^B$ can be combined which give the same pattern for elements in the boundary of the two structures. That is, only if $\forall x \in \delta(A) = \delta(B)$ and $\forall i \in \{1, \ldots, m\}$ $x \in U_i^A \Leftrightarrow x \in U_i^B$. This control can be organized either by enlarging the vocabulary or by keeping track of the possible patterns arising when checking which boundary elements belong to which of the U_i. Because at every step there are only $k+1$ boundary values this bookkeeping does not effect the overall linear running time. $\square$

For decision and evaluation problems even over infinite fields the above proof can be adapted as well:

Proof of Theorem 10:

Let $\mathfrak{D} = (V, E, C)$ be a $\mathbb{R}$-structure of bounded tree-width. If we are dealing with decision problems we just have to check the validity of $\mathfrak{D} \models \psi$ for a MS_2-formula ψ. We do that in the same manner as explained in the proof of Theorem 9. If a MSO$_{\mathbb{R}}$ term T has to be computed over $\mathfrak{D}$, on every substructure of the tree decomposition we evaluate that term corresponding to the decomposition of T; climbing up the tree we combine the two values computed at the two substructures to obtain the value on the joined structure. This needs constant time for every substructure in the decomposition. □

6.2 Extended decision problems over infinite fields

If we consider extended decision problems like the feasibility problem over infinite fields there arise some problems with the tree decomposition approach.

The first observation is that in $\exists$-$\text{MSO}_{\mathbb{R}}$ logic we cannot quantify over the real numbers. Thus, at first sight it is not clear whether the existence of a real zero of a polynomial can be expressed in this logic. We therefore restrict ourselves to ask for zeros the components of which belong to a fixed finite subset $A \subset \mathbb{R}$ (cf. Remark 2). Nevertheless, there are still many interesting and potentially hard problems captured in this situation, recall the discussion in section 4. Furthermore it turns out that for some problems we have to require an additional condition on the values of the weight function. The necessity of such a condition will become evident in the proof of Theorem 11. A precise definition is then provided after it.

Proof of Theorem 11:

We again restrict to the case $(4,k) - FEAS(A)$ and the representation of an input polynomial by $\mathfrak{D} = (V, E, C)$ as in Example 20. Let ψ and Φ be the corresponding formulas in MS_2 and $\exists$-$\text{MSO}_{\mathbb{R}}$. The decomposition of $\mathfrak{D}, \psi$ and Φ is done as in the proof of Theorem 10. The same is true for checking $\mathfrak{D} \models h_\gamma$ for at least one of the Hintikka formulas ψ decomposes into. The validity of an additional condition has to be required when climbing up the parse tree in order to evaluate $\text{MSO}_{\mathbb{R}}$ terms in Φ. The number of different intermediate values which are taken by the $\text{MSO}_{\mathbb{F}}$ terms during the evaluation process has to be bounded by a polynomial in the size of the input-structure.

Note that in the current situation this condition is satisfied if the set of partial sums of the monomials evaluated on the finite set A is bounded by a function in $O(p(n))$ for p a polynomial. This is true because the decomposition of Φ according to the definition of $\text{MSO}_{\mathbb{R}}$ logic in the previous section implies that the $\text{MSO}_{\mathbb{R}}$ terms evaluated on substructures of the input polynomial precisely are such partial sums.

At the root of the parse tree we check all possible results of the evaluation, which are now polynomially many. The rest of the proof again works as the one of Theorem 9.

The proof of part b) works precisely the same. Instead of checking whether at least one result is zero we only have to check the positivity of all results (which are at most polynomially many).

□

The above proof substantiates the interest in the following condition on a meta-finite structure $\mathfrak{D}$ together with a MS_2 formula ψ and a $\exists$-$\text{MSO}_{\mathbb{R}}$ formula Φ :

Definition 22 *Let p be a polynomial and $\mathfrak{D}, \psi, \Phi$ as above. The triple $(\mathfrak{D}, \psi, \Phi)$ satisfies the "coefficient condition" w.r.t. polynomial p if the number of intermediate results of* $\text{MSO}_{\mathbb{R}}$ *terms which have to be evaluated when applying the above algorithm is bounded by* $p(size(\mathfrak{D}))$.

In some situations we will restrict ourselves to structures $\mathfrak{D}$ of bounded tree-width which in addition satisfy the coefficient condition with respect to a given polynomial and the decision problem $\mathfrak{D} \models \psi \wedge \Phi$ we are interested in.

Let us comment on the coefficient condition. Since it depends on ψ and Φ as well as on the used algorithmic implementation of the Feferman-Vaught theorem it might be possible to strengthen the results by carefully considering the formulation of a problem and the evaluation process. However, this might result in quite complicated conditions. For example, consider the proof of theorem 11. Instead of dealing with all partial sums of monomials it would actually be sufficient to take into account only those appearing along the decomposition. Nevertheless, it seems to be more practical to look for sufficient conditions like the one mentioned in the proof.

It is not always obvious whether the additional coefficient condition on the partial sums of a polynomial implies a real restriction. For an example like the real Knapsack problem (given n real numbers $x_1, \ldots, x_n$, is there a subset $S \subseteq \{1, \ldots, n\}$ s.t. $\sum\limits_{i \in S} x_i = 1$) it implies a serious restriction because under the additional hypothesis that all sums $\sum\limits_{i \in \tilde{S}} x_i$ only take polynomially many different values the problem lies in $\mathrm{P}_{\mathbb{R}}$. On the other hand, if we reduce a problem like 3-SAT to an instance of $4 - FEAS(\{0,1\})$ the coefficient condition is automatically fulfilled because of the lemma below. Thus, in related situations our approach gives linear algorithms on structures of bounded tree-width without any additional assumption on the weights.

Lemma 23 *Let q be a polynomial, $A \subset \mathbb{R}^n$ finite and $d \in \mathbb{N}$. Then for all polynomial functions $f : \mathbb{R}^n \to \mathbb{R}$ of degree d which only have integer coefficients in $\{-q(n), -q(n) + 1, \ldots, q(n)\}$ the number of different values for the partial sums of the monomials of f if evaluated on A is bounded by $O(N \cdot q(n))$. Here, N is the number of non-zero monomials in f.*

Proof. For fixed degree d the normalized monomials $x_{i_1} \cdot \ldots \cdot x_{i_d}, i_1, \ldots, i_d \in \{1, \ldots, n\}$ of f only take polynomially in $|A|$ many different values if evaluated in elements of A. Let S be the set of these results. Each value in S is multiplied by the sum of the coefficients of those monomials actually giving the value (or by 0). Since the number of monomials is bounded by a N (which itself is bounded by $O(n^d)$) every such sum gives an integer in $\{-q(n) \cdot N, \ldots, q(n) \cdot N\}$. Thus, there are only polynomially many different possibilities to multiply one of the finitely many elements in S by a particular sums of the coefficients. The assertion follows. □

Recall that in particular for polynomials of bounded tree-width the number of non-zero monomials is linear in the variable number.

From the previous proofs it should be clear that Theorem 12 summarizes the general way for obtaining polynomial time algorithms by exploiting the above methods. We therefore omit its proof.

6.3 Getting around the coefficient condition

For some of the problems studied above it is possible to avoid the coefficient condition over infinite fields. The ideas are already present in [ALS91] (in an automata theoretic framework) and [CMR00a] (in a logical framework). Therefore we just outline how some of these problems can be putted into their framework.

The problems we can handle that way have to be optimization problems where the objective function has a linear structure. In the framework of weighted hypergraphs $\mathfrak{D} = (V, E, C)$ linear structure means that we want to compute

$$\min_{\phi(\underline{X})} \sum_{t=1}^{s} a_t \cdot |X_t| \ , \text{ where } \underline{X} = (X_1, X_2, \ldots, X_s).$$

Here, ϕ is a MS_2 formula and $|X_t|$ is an abbreviation for $\sum_{e \in X_i} C(e)$. In such a situation analyzing the Feferman-Vaught theorem for all of the parse operations shows that the special linear structure of the evaluation term (which is a $\mathrm{MSO}_{\mathbb{R}}$ term according to our definition) allows to compute the minimum (or maximum) of a $\mathbb{R}$-structure obtained after applying a parse operation in constant time from the corresponding extremal values on the substructure (see [ALS91,CMR00a]). In particular, we do not have to store too many intermediate results. This ideas can be applied to the $POS(A)$ problem:

Theorem 15. *For ordered fields $\mathbb{F}$ the problem $(d,k) - POS(A)_{\mathbb{F}}$ can be solved in linear time in n where the constants of the linear bound depend on k, d.*

Proof. We restrict ourselves to proving how $(d,k) - POS(A)$ can be expressed as a linear optimization problem in the above sense. The linear algorithm can then be obtained by the algorithm given in the proofs of Theorems 9 and 11 and the above remarks.

Let $A := \{s_1, \ldots, s_m\} \subset \mathbb{R}$ and $\mathfrak{D} = (V, E, C)$ be a $\mathbb{R}$-structure representing a polynomial f in n variables of degree d. Obviously, $f(x) > 0 \ \forall x \in A^n$ iff $\min\{f(x) | x \in A^n\} > 0$. We enlarge $\mathfrak{D}$ by d incidence relations $R_1, \ldots, R_d \subset V \times E$ having the interpretation $R_i(j, e) \Leftrightarrow$ the i-th factor in monomial $e \in E$ is x_j. Note that for this enlarged structure the tree-width concept and the Feferman-Vaught approach work similarly as before.

In order to make the problem linear we consider all (at most m^d many) values $a_1, \ldots a_{m^d}$ which can be obtained from A by multiplying d elements. The a_t thus give possible values of a monomial (with coefficient 1) if an assignment from A^n is chosen.

For any a_t there is a set $\mathfrak{P}_t$ of finitely many patterns in $\{1, \ldots, m\}^d$ such that for every $(k_1, \ldots, k_d) \in \mathfrak{P}_t$ we have $s_{k_1} \cdot \ldots \cdot s_{k_d} = a_t$. Now consider m^d subsets $X_t \subset E$. The intended meaning of the MS_2 formula $\phi(X_1, \ldots, X_{m^d})$ we want to be fulfilled on $\mathfrak{D}$ is a decomposition of all monomials in E into sets X_t such that there is a minimizing assignment which for every monomial $e \in X_t$ gives the value $a_t, 1 \le t \le m^d$. Thus

$$\begin{aligned} \phi(X_1, \ldots, X_{m^d}) \equiv \ & \exists U_1, \ldots, U_m \subset V, \bigcup_i U_i = V, \text{ all disjoint} \\ & \bigwedge \ \forall 1 \le t \le m^d \ \forall \ i_1, \ldots, i_d \in V : \\ & \text{if } e \in X_t \text{ and } R_1(i_1, e) \wedge \ldots \wedge R_d(i_d, e) \text{ then } (i_1, \ldots, i_d) \in \mathfrak{P}_t \end{aligned}$$

If $(\mathfrak{D}, \underline{X}) \models \phi(\underline{X})$ then there is an assignment $(U_1, \ldots, U_m)$ for $x_1, \ldots, x_n$ from A^n which is compatible with $\underline{X}$ in the sense that for every monomial $e \in X_t$ the assignment given by the U_i yields the corresponding value a_t. Finally, the linear objective function to be minimized is $\sum_{t=1}^{m^d} a_t \cdot \sum_{e \in X_t} C(e)$. □

7 Conclusions and further research

We have shown how the concept of tree-width of multivariate polynomial systems can be used in finding polynomial time algorithms for some otherwise difficult computational problems provided the tree-width is bounded by a constant. The method has further extensions to linear and quadratic programming, previously analyzed in [Mee94]. This will be worked out in a future paper.

Our proofs use a detour through logic. An alternative route would be through automata theory, as in [ALS91]. But automata theory and Monadic Second Order Logic are just two faces of the same definability phenomenon, cf. [Cou97,Cou90,Tho90].

Using the methods developed in [FV93,FV99,KV98] theorem 9 can be improved to

Theorem 24. *For finite rings R of size r, $(d,k)-FEAS_R$ and $(d,k)-HN_R$ can be solved in time $O(n \cdot r)$ where the constant depends on k, d.*

The point is to view feasibility of systems of polynomials over finite structures as *constraint satisfaction problems.* However, this method does not give improvements for infinite structures.

Another direction of further research is the restriction of our feasibility problems to finite subsets A for the components of possible zeros.

Using very effective versions of quantifier elimination over the reals $\mathbb{R}$ one might ask:

Problem 2. Can $\mathbb{R}$ $(d,k) - POS_{\mathbb{R}}$ and $(d,k) - FEAS_{\mathbb{R}}$ be solved in time $O(n)$ over the reals, where the constant in $O(n)$ depends on k, d only?

For the best known algorithms to solve $(d,\infty) - FEAS_{\mathbb{R}}$ and for quantifier elimination the reader should consult the surveys [BPR96,Roy96,Bas99].

It remains a challenging problem to find direct algebraic proofs and to overcome the limitations imposed by our coding technique.

8 Appendix

In this Appendix we want to give a rigorous presentation of the results used in the previous sections. This will make the paper self-contained; though well-known to logicians people in complexity theory of algebraic problems might not be that familiar with the Feferman-Vaught theorem. However, a lot of different results and concepts have to be put together which to our knowledge are hard to find at a single place in literature. So even for readers familiar with the background we believe it might be useful having all this available in an Appendix.

Moreover, we adapted the proofs to the framework in which we need the corresponding statements.

8.1 MSO logic over finite structures

We suppose the reader to be familiar with the notions of a finite structure and second order logic SOL over a given signature (see [EF95]). the finite structures we are interested in are hypergraphs. These are two-sorted structures the universe of which consists V and E, the vertices and hyperedges respectively. In addition, there is one binary relation symbol R_{inc} for the incidence relation between edges and vertices. (This relation is not always needed, for example when dealing with the zero-existence problem for a polynomial). In case E is the set of edges of a directed graph we might also include another binary relation in order to distinguish the direction of an edge. Furthermore, we allow an arbitrary but finite number of constant symbols and unary predicate symbols. In MSO logic the set variables range over subsets of V or E only. We denote the vocabulary by τ and the MSO logic over τ by $MS_2(\tau)$ (the index "2" indicates the two sorts), see also [CMR00b].

8.2 Hintikka formulas; the Fraisse-Hintikka theorem

Fix a relational, finite vocabulary τ as above and consider the monadic second order logic $MS_2(\tau)$ over τ. In $MS_2(\tau)$ we are interested in formulas with first-order variables among $x_1, \ldots, x_{n_1}$ and second-order variables among $X_1, \ldots, X_{n_2}$. Later on it will be convenient to split the block $x_1, \ldots, x_{n_1}$ into further blocks.

Proviso: In order to avoid confusion by mixing too many things we reduce our description of theorem 25 and 18 to $MS_2(\tau)$ formulas with free first order variables for vertices only and free second order variables for subsets of vertices.

However, it should be clear from the presentation that a generalization including the edges as second sort of our structures is straightforward from that.

Let $n := n_1 + n_2$. We denote by $F^{\tau}_{n,r}$ the set of $MS_2(\tau)$ formula with n variables and quantifier rank at r (the quantifier rank is the maximal number of nested quantifiers in a formula).

Theorem 25. *Let τ be a signature. For each $r, n \in \mathbb{N}$ we can effectively find a finite set $H_{n,r}$ of unnested formulas in $F^{\tau}_{n,r}$ such that the following is true:*

a) for every τ-structure $\mathfrak{A}$ and for every tuple $(\underline{a}, \underline{M}) := (a_1, \ldots, a_{n_1}, M_1, \ldots, M_{n_2})$, where the a_i are elements of the finite universe A of $\mathfrak{A}$ and the M_j are subsets of A there exists exactly one formula $\theta \in H_{n,r}$ such that $(\mathfrak{A}, \underline{a}, \underline{M}) \models \theta(\underline{a}, \underline{M})$.
b) For every $r \in \mathbb{N}$ and every un-nested formula $\phi \in F^{\tau}_{n,r}$ there is a disjunction $h_1 \vee \ldots \vee h_m$ of formulas $h_i \in H_{n,r}$ which is equivalent to ϕ for any τ-structure. The h_i can be computed effectively.

Proof. We follow closely [Hod], where this theorem is established for first-order logic.

Ad a) We define the sets $H_{n,r}$ recursively with respect to the parameter r. First, consider all unnested atomic formula in n variables. There are only finitely many such atomic formulas, say $\psi_1, \ldots, \psi_s$. Define $H_{n,0}$ to be the set of all formulas

$$\psi_1^{\alpha_1} \wedge \ldots \wedge \psi_s^{\alpha_s} \ ,$$

where $\alpha_i \in \{0,1\}$ and $\psi^0 := \neg\psi, \psi^1 := \psi$. Obviously, $\mathfrak{A}$ together with a tuple $(\underline{a}, \underline{M})$ satisfies either ψ_i or $\neg\psi_i$ (of course, we regard only those formulas with the same shape of variables as $(\underline{a}, \underline{M})$). Thus, $(\mathfrak{A}, \underline{a}, \underline{M})$ satisfies exactly one of the formulas in $H_{n,0}$.

Now let the set $H_{l,r}$ be defined for all $l \in \mathbb{N}$ according to the assertion of part a). In particular, suppose the set $H_{n+1,r}$ to consist of formulas $\theta_1, \ldots, \theta_m$. Define the set $H_{n,r+1}$ as follows:

First, divide $H_{n+1,r}$ into a part $H^1_{n+1,r}$ where the $n+1$-st free variable is first-order and a part $H^2_{n+1,r}$ where the $n+1$-st free variable is second-order. That way one obtains two disjunct index sets I_1, I_2 such that $I_1 \cup I_2 = \{1, \ldots, m\}$. We consider two ways to define formulas in $MS_2(\tau)$ with n variables and quantifier rank $r+1$ from those in $H^1_{n+1,r}$ and in $H^2_{n+1,r}$:

for any nonempty subset $X \subseteq I_1$ build the formula

$$\bigwedge_{i \in X} \exists a_{n+1} \in A\ \theta_i(\underline{a}, \underline{M}, a_{n+1}) \ \wedge\ \bigvee a_{n+1} \in A\ \exists i \in X\ \theta_i(\underline{a}, \underline{M}, a_{n+1}) \ . \tag{1}$$

Accordingly, for any nonempty subset $\tilde{X} \subseteq I_2$ build the formula

$$\bigwedge_{i \in \tilde{X}} \exists M_{n+1} \subseteq A\ \theta_i(\underline{a}, \underline{M}, M_{n+1}) \ \wedge\ \bigvee M_{n+1} \subseteq A\ \exists i \in \tilde{X}\ \theta_i(\underline{a}, \underline{M}, M_{n+1}) \ . \tag{2}$$

Now, combine any formula of type (1) with a formula of type (2) by conjunction. This gives for any pair $(X, \tilde{X})$ the formula

$$\left\{ \begin{array}{l} \bigwedge_{i \in X} \exists a_{n+1} \in A\ \theta_i(\underline{a}, \underline{M}, a_{n+1}) \ \wedge\ \bigvee a_{n+1} \in A\ \exists i \in X\ \theta_i(\underline{a}, \underline{M}, a_{n+1}) \ \bigwedge \\ \\ \bigwedge_{i \in \tilde{X}} \exists M_{n+1} \subseteq A\ \theta_i(\underline{a}, \underline{M}, M_{n+1}) \ \wedge\ \bigvee M_{n+1} \subseteq A\ \exists i \in \tilde{X}\ \theta_i(\underline{a}, \underline{M}, M_{n+1}) \end{array} \right. \tag{3}$$

where $X \subseteq I_1$ and $\tilde{X} \subseteq I_2$ and not both of them are empty.

Consider a structure $\mathfrak{A}$ over τ and assignments $a_1, \ldots, a_{n_1} \in A, M_1, \ldots, M_{n_2} \subseteq A$.

For every $a_{n+1} \in A$ the induction hypothesis yields the existence of a formula $\theta_{i_0} \in H^1_{n+1,r}$ such that $(\mathfrak{A}, \underline{a}, \underline{M}, a_{n+1}) \models \theta_{i_0}(\underline{a}, \underline{M}, a_{n+1})$. Define $X \subseteq I_1$ as the set of all indices $i \in I_1$ such that there is a $a_{n+1} \in A$ with $(\mathfrak{A}, \underline{a}, \underline{M}, a_{n+1}) \models \theta_i(\underline{a}, \underline{M}, a_{n+1})$. Similarly, for every $M_{n+1} \subseteq A$ the induction hypothesis yields the existence of a formula $\theta_{j_0} \in H^2_{n+1,r}$ such that $(\mathfrak{A}, \underline{a}, \underline{M}, M_{n+1}) \models \theta_{j_0}(\underline{a}, \underline{M}, M_{n+1})$. Define $\tilde{X} \subseteq I_2$ as the set of all indices $i \in I_2$ such that there is a $M_{n+1} \subseteq A$ with $(\mathfrak{A}, \underline{a}, \underline{M}, M_{n+1}) \models \theta_i(\underline{a}, \underline{M}, M_{n+1})$. Now it is easy to see that $(\mathfrak{A}, \underline{a}, \underline{M})$ satisfies exactly that formula built according to rule (3) in which the sets X and $\tilde{X}$ are chosen as explained above. This gives claim a).

Ad b) Again via induction over r : For $r = 0$ let ψ be a quantifier free formula in $MS_2(\tau)$ with n variables; ψ is a disjunction of conjunctions of atomic formulas. Let $\varrho_1 \wedge \ldots \wedge \varrho_t$ be such a conjunction; consider all elements $\theta \in H_{n,0}$ containing the ϱ_i in exactly the same form. Now, $\varrho_1 \wedge \ldots \wedge \varrho_t$ is equivalent to the disjunction of all the θ's chosen above. The formula ψ itself then is equivalent to all the corresponding disjunctions.

For the induction step from r to $r+1$ we assume the set $H^2_{n+1,r}$ to be $\{\theta_1, \ldots, \theta_m\}$. Without loss of generality let $\phi \equiv \theta_1 \vee \ldots \vee \theta_s$ (i.e. we assume the $n+1$-st variable to be a set-variable). Consider the formula

$$\psi(\underline{x}, \underline{Y}) := \exists\, Y_{n+1}\ \phi(\underline{x}, \underline{Y}, Y_{n+1})\ .$$

In order to express ψ as a disjunction of formulas in $H_{n,r+1}$, among the formulas in (3) take all those where X is the empty set and $\tilde{X} \subseteq I_2$ contains at least one index in $\{1, \ldots, s\}$. This gives a new set of formulas

$$\tilde{\theta}_1, \ldots, \tilde{\theta}_{\tilde{s}} \in H_{n,r+1}. \tag{4}$$

<u>Claim:</u> $\psi\ \equiv \tilde{\theta}_1 \vee \ldots \vee \tilde{\theta}_{\tilde{s}}$.

To prove the claim suppose $(\mathfrak{A}, \underline{x}, \underline{Y}) \models\ \tilde{\theta}_1(\underline{x}, \underline{Y}) \vee \ldots \vee \tilde{\theta}_{\tilde{s}}(\underline{x}, \underline{Y})$ holds. Then there exists a formula, say $\tilde{\theta}_1$, such that $(\mathfrak{A}, \underline{x}, \underline{Y}) \models\ \tilde{\theta}_1(\underline{x}, \underline{Y})$. According to rule (3) we find a set $\tilde{X}$ containing an element $i_0 \in \{1, \ldots, s\}$ and

$$(\mathfrak{A}, \underline{x}, \underline{Y}) \models\ \bigwedge_{i \in \tilde{X}} \exists\, Y_{n+1}\ \tilde{\theta}_i(\underline{x}, \underline{Y}, Y_{n+1}).$$

In particular, we obtain $(\mathfrak{A}, \underline{x}, \underline{Y}) \models\ \exists\, Y_{n+1}\ \tilde{\theta}_{i_0}(\underline{x}, \underline{Y}, Y_{n+1})$ which implies $(\mathfrak{A}, \underline{x}, \underline{Y}) \models \psi(\underline{x}, \underline{Y})$.

To show the opposite, assume $(\mathfrak{A}, \underline{x}, \underline{Y}) \models\ \psi(\underline{x}, \underline{Y})$, say

$$(\mathfrak{A}, \underline{x}, \underline{Y}) \models\ \exists\, M_{n+1}\ \tilde{\theta}_{i_1}(\underline{x}, \underline{Y}, M_{n+1}).$$

We define $\tilde{X}$ as set of all those indices $i \in \{1, \ldots, m\}$ for which $\theta_i \in H^2_{n+1,r}$ as well as $(\mathfrak{A}, \underline{x}, \underline{Y}) \models\ \exists\, M_{n+1}\ \tilde{\theta}_i(\underline{x}, \underline{Y}, M_{n+1})$. According to the definition we get $i_1 \in \tilde{X}$; thus, the formula obtained by choosing our particular $\tilde{X}$ together with $X = \emptyset$ in (3) belongs to the formulas under (4), which immediately gives the reverse direction.

If ψ has the shape $\forall M_{n+1}\ \phi(\underline{x}, \underline{Y}, M_{n+1})$, for building the corresponding set of formulas in (4) consider all choices $\tilde{X} \subseteq \{1, \ldots, s\}$.

If ψ is obtained by quantification of a first-order variable x_{n+1} the proof works similarly after replacing $\tilde{X}$ by X and the formulas in $H^2_{n+1,r}$ by those in $H^1_{n+1,r}$. □

Remark 7. In the above proof there are several ways to obtain a conjunction of atomic formulas which is false over any structure with an equality relation, for example by including $x \neq x$ for one or several variables. By convention, we include only one such formula into each of the sets $H_{n,0}$. This will be needed later on in order to avoid ambiguities.

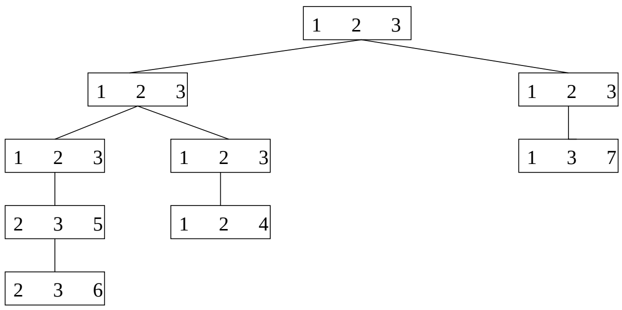
8.3 Parse-trees for hypergraphs of bounded tree-width

Let G be a hypergraph of bounded tree-width k. Our goal is to create G from some elementary hypergraphs by applying finitely many operations to the latter. The initial hypergraphs will be structures of size at most k. The operations have to be chosen carefully in order to guarantee validity of the theorem by Feferman-Vaught.

The hypergraphs in-between the building procedure of G all are $(k+1)$-boundaried hypergraphs in the in the sense defined in section 5. The following example shows how the decomposition of a hypergraph of tree-width k into $(k=1)$ boundaried hypergraphs works.

Example 26 *We consider once again the polynomial p from Example 3, ii). Starting with a tree decomposition of p the latter is first turned into a binary tree. To this aim branch nodes are simply duplicated sufficiently many times. By adding further nodes to those subsets V_t with cardinality less than $k+1$ (without destroying the tree decomposition requirements) we can assume all sets V_t to be of cardinality $k+1$. Furthermore, by duplication once again branch nodes every branch has identical children. Finally, including additional nodes we can guarantee that two neighbored sets V_t differ by at most one element.*

For the polynomial p from Example 3,ii) this results in the following normalized tree decomposition:



The parse tree is now constructed bottom up. At the leaves we apply the create operations for the $k+1$ vertices included in the corresponding sets V_t. To include new hyperedges the *add* operation is applied. To change an already treated vertex with a new one we use the operation *new* ; *change* is used to fit labels. Finally, due to the normalization of the tree decomposition *join* is only applied to substructures with the same labels on the same vertices (the boundary).

Full details can be found in [DF99].

A tree-decomposition itself can also be obtained in linear time according to theorem 6.

8.4 Feferman-Vaught theorem for finite structures of bounded tree-width

We combine the previous subsections in order to obtain the main result for the structures we are interested in.

The overall idea is to decide a $MS_2(\tau)$ formula for a given structure by reducing it to substructures and deciding corresponding formulas there. The substructures are obtained by analyzing the parse-tree of the given structure. The formulas to be decided on the substructures are determined through the given one according to theorem 25.

We will explicitly proof the according statement for a decomposition related to a join operation $\oplus$. The according statements for the other parse operations can be established similarly.

For the following recall our general proviso!

Let $\mathfrak{A}$ and $\mathfrak{B}$ be $k+1$ boundaried hypergraphs over τ with universes A and B resp., and boundaries $\delta(A) \subset A, \delta(B) \subset B, \widehat{A} : +A \setminus \delta(A), \widehat{B} := B \setminus \delta(B)$. Let $\mathfrak{C} = \mathfrak{A} \oplus \mathfrak{B}$ be the join of $\mathfrak{A}$ and $\mathfrak{B}$ as explained in subsection 8.4.

Let z be an assignment of the variables in a $MS_2(\tau)$ formula $\Phi(\underline{x}, \underline{y}, \underline{w}, \underline{X})$ into the universe C of $\mathfrak{C}$ such that the following holds:

- z maps the variables x_i of block $\underline{x}$ to the set $\widehat{A}$
- z maps the variables y_i of block $\underline{y}$ to the set $\widehat{B}$
- z maps the variables w_i of block $\underline{w}$ to the boundary $\delta(A) = \delta(B)$ (recall that the two boundaries were identified). In particular, $z(w_i)$ is member of both the universes A and B.

Denote by z_A, z_B the assignments with $z_A(X) = X \cap A, z_A(s) = z(s)$ for $s \in A$ and $z_B(X) = X \cap B, z_B(s) = z(s)$ for $s \in B$.
Then the following is true

Theorem 27. *Let $\mathfrak{C}$ be the join of $\mathfrak{A}$ and $\mathfrak{B}$: $\mathfrak{C} = \mathfrak{A} \oplus \mathfrak{B}$ and let $h(\underline{x}, \underline{y}, \underline{w}, \underline{X})$ be a $MS_2(\tau)$ formula which is a Hintikka formula in some $H_{n,r}(\underline{x}, \underline{y}, \underline{w}, \underline{X})$. Then there are unique Hintikka formulas $h_1(\underline{x}, \underline{w}, \underline{X}) \in H_{n,r}(\underline{x}, \underline{w}, \underline{X})$ and $h_2(\underline{y}, \underline{w}, \underline{X}) \in H_{n,r}(\underline{y}, \underline{w}, \underline{X})$ such that for every assignment z as above we have $(\mathfrak{C}, z) \models h(\underline{x}, \underline{y}, \underline{w}, \underline{X}) \Leftrightarrow (\mathfrak{A}, z_A) \models h_1(\underline{x}, \underline{w}, \underline{X}) \wedge (\mathfrak{B}, z_B) \models h_2(\underline{y}, \underline{w}, \underline{X})$.*

Proof. Induction on the quantifier rank r of h:

$\underline{r = 0:}$
According to the proof of theorem 25 the quantifier free Hintikka formula h is a conjunction of all atomic formulas (where the latter appear either negated or not). We indicate with some examples how h_1 and h_2 are being built given these parts in h:

i) suppose an atomic formula $x_i = y_j$ appears unnegated in h. Then $(\mathfrak{C}, z) \not\models h$ because z assigns different values to x_i and y_j. This holds independently of $\mathfrak{C}$. We are therefore done by including in h_1 (or in h_2) an atomic formula $x \neq x$ which is always false as well (according to the uniqueness condition mentioned in remark 7).
ii) Suppose an atomic formula $w_i \neq w_j$ appears in h. Then we include it both in h_1 and in h_2. A formula like $x_i \neq x_j$ is only included in h_1, similarly for variables from the block $\underline{y}$ and h_2.
iii) For a unary relation R in τ a formula $R(x_i)$ is maintained in h_1, whereas $R(y_j)$ is transferred to h_2. An atomic formula $R(w_i)$ will again be both present in h_1 and in h_2.
Thus, given as unary relations which only hold true on exactly one element of the boundary, the labels are transformed to both substructures.

Note that by convention in the general case where we deal with two-sorted structures we decompose the set E in such a way that multi-edges from $\mathfrak{P}(\delta(A))$ are only maintained in $\mathfrak{A}$, not in $\mathfrak{B}$.

$\underline{r - 1 \to r:}$ Let

$\psi(\underline{x}, \underline{y}, \underline{w}, \underline{M}) :=$

$$\bigwedge_{i \in X} \exists a_{n+1} \in A \ \theta_i(\underline{x}, \underline{y}, \underline{w}, \underline{M}, a_{n+1}) \ \wedge \ \forall a_{n+1} \in A \ \exists i \in X \ \theta_i(\underline{x}, \underline{y}, \underline{w}, \underline{M}, a_{n+1}) \ \bigwedge$$

$$\bigwedge_{i \in \tilde{X}} \exists M_{n+1} \subseteq A \ \theta_i(\underline{x}, \underline{y}, \underline{w}, \underline{M}, M_{n+1}) \ \wedge \ \forall \ M_{n+1} \subseteq A \ \exists i \in \tilde{X} \ \theta_i(\underline{x}, \underline{y}, \underline{w}, \underline{M}, M_{n+1})$$

where $\theta_i \in H^1_{n+1,r-1}(\underline{x},\underline{y},\underline{w},\underline{M},a_{n+1})$ for $i \in X$ and $\theta_i \in H^2_{n+1,r-1}(\underline{x},\underline{y},\underline{w},\underline{M},M_{n+1})$ for $i \in \tilde{X}$.

Suppose $(\mathfrak{A} \oplus \mathfrak{B}, z) \models \psi$. Then

- for every $i \in X$ there exists a $c_i^{\mathfrak{A}\oplus\mathfrak{B}} \in A \oplus B$ such that $(\mathfrak{A}\oplus\mathfrak{B}, z) \models \theta_i(\underline{x},\underline{y},\underline{w},\underline{M},c_i^{\mathfrak{A}\oplus\mathfrak{B}})$ and
- for every $i \in \tilde{X}$ there exists a $M_i^{\mathfrak{A}\oplus\mathfrak{B}} \subseteq A \oplus B$ such that $(\mathfrak{A}\oplus\mathfrak{B}, z) \models \theta_i(\underline{x},\underline{y},\underline{w},\underline{M},M_i^{\mathfrak{A}\oplus\mathfrak{B}})$.

The induction hypothesis yields the existence of four Hintikka formulas $\theta^{\mathfrak{A}}_{i,1}, \theta^{\mathfrak{A}}_{i,2} \in H_{n_1,r-1}$ and $\theta^{\mathfrak{B}}_{i,1}, \theta^{\mathfrak{B}}_{i,2} \in H_{n_2,r-1}$ such that

$$(*) \quad \begin{cases} (\mathfrak{A}, z_A) \models \theta^{\mathfrak{A}}_{i,1}(\underline{x},\underline{w},\underline{M},c_i^{\mathfrak{A}}) \ \wedge \theta^{\mathfrak{A}}_{i,2}(\underline{x},\underline{w},\underline{M},M_i^{\mathfrak{A}}) \\ \\ (\mathfrak{B}, z_B) \models \theta^{\mathfrak{B}}_{i,1}(\underline{x},\underline{w},\underline{M},c_i^{\mathfrak{B}}) \wedge \theta^{\mathfrak{B}}_{i,2}(\underline{x},\underline{w},\underline{M},M_i^{\mathfrak{B}}) \end{cases}$$

(where either $c_i^{\mathfrak{A}}$ or $c_i^{\mathfrak{B}}$ might be a "dummy" constant according to the membership of $c_i^{\mathfrak{A}\oplus\mathfrak{B}}$ to $\widehat{A}, \widehat{B}$ or $\delta(A) = \delta(B)$.)

We define h_1 : let X_A be the set of all indices of Hintikka formulas in $H_{n_1,r-1}$ such that this index appears among the formulas $\theta^{\mathfrak{A}}_{i,1}$ in $(*)$. Let $\tilde{X_A}$ denote the corresponding set of indices appearing among one of the formulas $\theta^{\mathfrak{A}}_{i,2}$ in $(*)$. Then h_1 is defined to be built according to the general scheme (cf. equation 3 in the first subsection of this paragraph) and with X_A and $\tilde{X_A}$ as the chosen index sets. Formula h_2 is given in the same manner. It's now straightforward to check that h_1 and h_2 satisfy the requirements. □

The theorem together with finiteness of $H_{n,r}$ implies that there are only finitely many triples $\mathfrak{C} = \mathfrak{A} \oplus \mathfrak{B}$ with respect to the uniquely determined Hintikka formulas. The same is true if we allow parameters.

The Feferman-Vaught theorem generalizes the previous theorem by getting independent of the particular structures involved. It can be proved now as follows:

Proof of theorem 18:

According to theorem 25 ψ is equivalent to a disjunction $\theta_1 \vee \ldots \vee \theta_s$ of finitely many Hintikka formulas. For each θ_i there exist finitely many triples $(\theta_i, h_1^j, h_2^j), j \in J_i, |J_i|$ finite such that for given structures $\mathfrak{C} = \mathfrak{A} \oplus \mathfrak{B}$ theorem 27 is true for exactly one $j \in J_i$. If $\mathfrak{C}, \mathfrak{A}, \mathfrak{B}$ are varied then $\mathfrak{C} \models \theta_i$ iff $\bigvee\limits_{j \in J_i} \mathfrak{A} \models h_1^j \ \wedge \ \mathfrak{B} \models h_2^j$. The formula ψ itself is equivalent to the disjunction $\bigvee\limits_{i=1}^{s} \bigvee\limits_{j \in J_i} \mathfrak{A} \models h_1^j \ \wedge \ \mathfrak{B} \models h_2^j$. This gives the formulas $h_{1,\alpha}$ and $h_{2,\alpha}$. □

8.5 The algorithm

The Feferman-Vaught theorem holds also true in a corresponding way for the other parsing operations introduced above. It is true as well for other classes of parameters like the clique-width of a graph and its corresponding graph operations, see [CMR00b].

It is then used for computational purposes as follows. Given a $MS_2(\tau)$ formula $\psi \in F_{n,r}$ we precompute a table of all Hintikka formulas in $H_{n,r}$ and how three of them fit together with respect to the Feferman-Vaught theorem for all the parsing operations. The formula ψ is written in its equivalent form as disjunction of Hintikka formulas, say $\psi = \theta_1 \vee \ldots \vee \theta_s$.

Now, given a structure of bounded tree-width we first construct its tree decomposition and the corresponding parse tree according to theorems 6 and theorem:DowneyFellows99. From the precomputed table of Hintikka formulas we obtain a corresponding decomposition of the θ_i 's. Then we evaluate bottom-up from the leaves of the parse tree to its root the Hintikka formulas in the decomposition (using again the precomputed table) and check whether ψ is true on the input. Since the parse tree has linear size in the size of the structure the running time is linear (though with large constants).

Some technicalities have to be taken care of. We demonstrate them again for the join operator $\oplus$.

Theorem 18 depends on the shape $(\underline{x}, \underline{y})$ an assignment z induces on the free first-order variables. Therefore, if the above algorithm is performed bottom up, we have to take into account all possible assignments for z, that is all possible patterns z induces on the free first-order variables. Suppose there are s many of them (where s only depends on the given formula). If we consider all decomposition patterns of these s many variables along the parse tree of a given structure of bounded tree-width we see that their number only depends on a function $f(s)$ in s. For every fixed pattern we perform the above algorithm in linear time, giving an $O(n)$ algorithm in total.

The interested reader might try to perform a proof of Feferman-Vaught theorem and the subsequent algorithm for the operator new_i!

The main contribution of the present paper is to combine the theory over finite structures presented in the Appendix with algebraic issues. To this aim a meta-finite monadic second order logic is defined carefully. It is constructed in such a way that on the one hand side a lot of important problems can be expressed by it. On the other hand side it is not too general in the sense that we can decompose a given meta-finite structure together with a $\exists$-$\mathrm{MSO}_{\mathbb{R}}$ property in parallel to the decomposition of the underlying finite structure described in this Appendix.

Acknowledgements

The first author would like to thank D. Grigoriev, I. Shparlinski, M. Vardi and V. Weisspfenning for their interest and comments.

References

[ALS91] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.

[Bar96] A.I. Barvinok. Two algorithmic results for the traveling salesman problem. *Mathematics of Operations Research*, 21:65–84, 1996.

[Bas99] S. Basu. New results on Quantifier Elimination Over Real Closed Fields and Applications to Constraint Databases. *Journal of the ACM*, Vol. 46, No. 4: 537–555, 1999.

[BCS97] P. Bürgisser, M. Clausen, and M.A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren.* Springer Verlag, 1997.

[BCSS98] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer Verlag, 1998.

[BGHM97] B. Bank, M. Giusti, J. Heintz, and G.M. Mbakop polar varieties, real equation solving, and data structures: The hypersurface case. *Journal of Complexity*, 13: 5–27, 1997.

[Bod97] H. Bodlaender. Treewidth: Algorithmic techniques and results. In I. Privara, P. Ruzicka, editors, Proc. of the 22nd Symposium *Mathematical Foundation of Computer Science*, volume 1295 of *Lecture Notes in Computer Science*, 19–36. Springer, 1997.

[Bod98] H. Bodlaender. A partial k-arboretum of graphs with bounded tree-width (tutorial). *Theoretical Computer Science*, 208:1–45, 1998.

[BPR96] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43(6):1002–1045, 1996.

[BSS89] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc.*, vol. 21, 1–46, 1989.

[Bür00] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory.* Algorithms and Computation in Mathematics, vol. 7 Springer Verlag, 2000.

[CLO97] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Graduate Texts in Mathematics. Springer Verlag, 1997.

[CM93] B. Courcelle and M. Mosbah. Monadic second–order evaluations on tree–decomposable graphs. *Theoretical Computer Science*, 109:49–82, 1993.

[CMR00a] B. Courcelle, J.A. Makowsky, and U. Rotics. Linear Time Solvable Optimization Problems on Graphs of Bounded Clique Width. *Theory of Computing Systems*, vol. 33 No.2, 125–150, 2000.

[CMR00b] B. Courcelle, J.A. Makowsky, and U. Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second order logic. *Discrete and Applied Mathematics*, xx:xx–yy, 2000.

[Cou90] B. Courcelle. Graph rewriting: An algebraic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, chapter 5. Elsevier Science Publishers, 1990.

[Cou97] Bruno Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, editor, *Handbook of graph grammars and computing by graph transformations, Vol. 1: Foundations*, pages 313–400. World Scientific, 1997.

[CucMat] F. Cucker, M. Matamala. On digital nondeterminism. *Mathematical Systems Theory*, 29:635–647, 1996.

[DF99] R.G. Downey and M.F Fellows. *Parametrized Complexity.* Springer, 1999.

[Die96] R. Diestel. *Graph Theory.* Graduate Texts in Mathematics. Springer, 1996.

[FV93] T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction. In *STOC'93*, 612–622, ACM, 1993.

[FV99] T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28: 57–104, 1999.

[EF95] H.D. Ebbinhaus and J. Flum. *Finite Model Theory* Perspectives in Mathematical Logic, Springer, 1995.

[GG98] E. Grädel and Y. Gurevich. Metafinite model theory. *Information and Computation*, 140:26–81, 1998. See also: D. Leivant (Ed.), Logic and Computational Complexity, Selected Papers, Springer 1995, 313-366.

[GK87] D.Y. Grigoriev and M. Karpinski. The matching problem for bipartite graphs with polynomial bounded permanents is in NC. In *28th Annual Symposium on Foundations of Computer Science*, pages 166–172, 1987.

[GKZ94] I. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants.* Birkhäuser Verlag, 1994.

[GM96] E. Grädel and K. Meer. Descriptive complexity theory over the real numbers. *Lectures in Applied Mathematics*, 32:381–403, 1996. A preliminary version has been presented at the 27th ACM-Symposium on Theory of Computing, Las Vegas, 1995.

[Hod] W. Hodges. Model theory. *Encyclopedia of Mathematics and its Applications*, vol. 42. Cambridge University Press, 1993.

[Jer87] M. Jerrum. Two–dimensional monomer–dimer systems are computationally intractable. *Journal Stat. Phys.*, 48:121–134, 1987. Erratum in vol. 59, 1087–1088, 1990.

[Kh91] A.G. Khovanskii Fewnomials. *Translations of Mathematical Monographs*, vol. 88. American Mathematical Society, 1991.

[KV98] P. Kolaitis and M. Vardi. Conjunctive query containement and constraint satisfaction. In *PODS'98*, 425–435, ACM, 1998.

[LL91] Y.N. Lakshman and D. Lazard. On the complexity of zero-dimensional algebraic systems. In T. Mora and C. Traverso, editors, *Effective methods in Algebraic Geometry*, volume 94 of *Progress in Mathematics*, 217–225, Birkhäuser, Basel, 1991.

[Mak99] J.A. Makowsky. Logical methods in graph algorithms. Lecture Notes of a course given at ESSLLI'99 in Utrecht, August, 1999.

[Mak00] J.A. Makowsky. Colored Tutte Polynomials and Kauffman Brackets for Graphs of Bounded Tree Width. Extended Abstract , submitted MFCS'00; revised further, March 31, 2000, submitted to Combinatorics, Probability and Computation

[MaM00] J.A. Makowsky and K. Meer On the Complexity of Combinatorial and Metafinite Generating Functions of Graph Properties in the Computational Model of Blum, Shub and Smale. Extended abstract, to appear in: Proc. CSL 2000, LNCS, Springer.

[Mee94] K. Meer. On the complexity of quadratic programming in real number models of computation. *Theoretical Computer Science*, 133:85–94, 1994.

[MM97] K. Meer and C. Michaux. A survey on real structural complexity theory. *Bulletin of the Belgian Math. Soc.*, 4:113–148, 1997.

[Poi95] B. Poizat. *Les Petits Cailloux: Une approche modèle-théorique de l'algorithmie*. Aléas, Paris, 1995.

[PRS93] P. Pedersen, M.-F. Roy and A. Szpirglas. Counting real zeros in the multivariate case. In F. Eysette and A. Galligo, editors, *Computational Algebraic Geometry*, volume 109 of *Progress in Mathematics*, 203–224. Birkhäuser, Basel, 1993.

[Roy96] M.-F. Roy. Basic algorithms in real algebraic geometry and their complexity: from Sturm's theorem to the existential theory of reals. In F. Broglia, editor, *Lectures in Real Geometry*. Walter de Gruyter, 1996.

[Tho90] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, chapter 4. Elsevier Science Publishers, 1990.

[Val79] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.