HAL open science

Efficient Computation of Generalized Ising Polynomials on Graphs with Fixed Clique-Width

Tomer Kotek, Johann A. Makowsky

To cite this version:

Tomer Kotek, Johann A. Makowsky. Efficient Computation of Generalized Ising Polynomials on Graphs with Fixed Clique-Width. 1st International Conference on Theoretical Computer Science (TTCS), Aug 2015, Tehran, Iran. pp.135-146, 10.1007/978-3-319-28678-5_10 . hal-01446257

HAL Id: hal-01446257

https://inria.hal.science/hal-01446257v1

Submitted on 25 Jan 2017

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

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

Distributed under a Creative Commons Attribution 4.0 International License

Efficient computation of generalized Ising polynomials on graphs with fixed clique-width*

Tomer Kotek[1] and Johann A. Makowsky[2]

[1] TU Vienna
Vienna, Austria
tkotek@tuwien.ac.at
[2] Technion — Israel Institute of Technology
Haifa, Israel
janos@cs.technion.ac.il

Abstract. Graph polynomials which are definable in Monadic Second Order Logic (MSOL) on the vocabulary of graphs are Fixed-Parameter Tractable (FPT) with respect to clique-width. In contrast, graph polynomials which are definable in MSOL on the vocabulary of hypergraphs are fixed-parameter tractable with respect to tree-width, but not necessarily with respect to clique-width. No algorithmic meta-theorem is known for the computation of graph polynomials definable in MSOL on the vocabulary of hypergraphs with respect to clique-width. We define an infinite class of such graph polynomials extending the class of graph polynomials definable in MSOL on the vocabulary of graphs and prove that they are Fixed-Parameter Polynomial Time (FPPT or XP) computable, i.e. that they can be computed in time $O(n^{f(k)})$, where n is the number of vertices and k is the clique-width.

1 Introduction

In recent years there has been growing interest in graph polynomials, functions from graphs to polynomial rings which are invariant under isomorphism. Graph polynomials encode information about the graphs in a compact way in their evaluations, coefficients, degree and roots. Therefore, efficient computation of graph polynomials has received considerable attention in the literature. Since most graph polynomials which naturally arise are $\sharp$P-hard to compute (see e.g. [40,25,11]), a natural perspective under which to study the complexity of graph polynomials is that of *parameterized complexity*.

Parameterized complexity is a successful approach to tackling NP-hard problems [18,20], by measuring complexity with respect to an additional *parameter* of the input; we will be interested in the parameters tree-width and clique-width. A

* Tomer Kotek was supported by the Austrian National Research Network S11403-N23 (RiSE) of the Austrian Science Fund (FWF) and by the Vienna Science and Technology Fund (WWTF) grant PROSEED.

computational problem is **fixed-parameter tractable** (FPT) with respect to a parameter k if it can be solved in time $f(k) \cdot p(n)$, where f is a computable function of k, n is the size of the input, and $p(n)$ is a polynomial in n. Many NP-hard problems are fixed-parameter tractable for an appropriate choice of parameter, see [20] for many examples. Every problem in the infinite class of decision problems definable in *Monadic Second Order Logic* (MSOL) is fixed-parameter tractable with respect to tree-width by Courcelle's Theorem [13,9,14] (though the result originally was not phrased in terms of parameterized complexity).

The computation problem we consider for a graph polynomial $P(G; x_1, \ldots, x_r)$ is the following:

$P -$ Coefficients
Instance: A graph G
Problem : Compute the coefficients $a_{i_1,\ldots,i_r}$ of the monomials $x_1^{i_1} \cdots x_r^{i_r}$.

For graph polynomials, a parameterized complexity theory with respect to tree-width has been developed. Here, the goal is to compute, given an input graph, the table of coefficients of the graph polynomial. The Tutte polynomial has been shown to be fixed-parameter tractable [37,8]. [34] used a logical method to study the parameterized complexity of an infinite class of graph polynomials, including the Tutte polynomial, the matching polynomial, the independence polynomial and the Ising polynomial. [34] showed that the class of graph polynomials definable in MSOL in the vocabulary of hypergraphs[3] is fixed-parameter tractable. This class contains the vast majority of graph polynomials which are of interest in the literature.

Going beyond tree-width to clique-width the situation becomes more complicated. [15] studied the class of graph polynomials definable in MSOL *in the vocabulary of graphs*. They proved that every graph polynomial in this class is fixed-parameter tractable with respect to clique-width. However, this class of graph polynomials does not contain important examples such as the chromatic polynomial, the Tutte polynomial and the matching polynomial. In fact, [21] proved that the chromatic polynomial and the Tutte polynomial are not fixed-parameter tractable with respect to clique-width (under the widely believed complexity-theoretic assumption that FPT $\neq$ W[1]). [36] proved that the chromatic polynomial and the matching polynomial are **Fixed-Parameter Polynomial Time (FPPT)**[4] computable with respect to clique-width, meaning that they can be computed in time $n^{f(cw(G))}$, where n is the size of the graph, $cw(G)$ is the clique-width of the graph and f is a computable function. [27] proved an analogous result for the Ising polynomial. The main result of this paper is a meta-theorem generalizing the fixed-parameter polynomial time computability of the matching polynomial and the Ising polynomial to an infinite family of graph polynomials definable in MSOL analogous to [15].

[3] In [14], MSOL in the vocabulary of hypergraphs is denoted MS_2, while MSOL in the vocabulary of graphs is denoted MS_1.

[4] Note that the class XP of slicewise polynomial-time languages (see [18, Chapter 15]) is exactly the class of FPPT languages.

Theorem 1 *Let P be an* MSOL-*Ising polynomial. P is fixed-parameter polynomial time computable with respect to clique-width.*

The class of MSOL-Ising polynomials is defined in Section 2.1.

2 Preliminaries

Let $[k] = \{1, \ldots, k\}$. Let τ_G be the vocabulary of graphs $\tau_G = \langle \mathbf{E} \rangle$ consisting of a single binary relation symbol $\mathbf{E}$. A k-graph is a structure $(V, E, R_1, \ldots, R_k)$ which consists of a simple graph $G = (V, E)$ together with a partition $R_1, \ldots, R_k$ of V. Let τ_G^k denote the vocabulary of k-graphs $\tau_G^k = \langle \mathbf{E}, \mathbf{R}_1, \ldots, \mathbf{R}_k \rangle$ extending τ_G with unary relation symbols $\mathbf{R}_1, \ldots, \mathbf{R}_k$.

The class $CW(k)$ of k-graphs of clique-width at most k is defined inductively. Singletons belong to $CW(k)$, and $CW(k)$ is closed under disjoint union $\sqcup$ and two other operations, $\rho_{i \to j}$ and $\eta_{i,j}$, to be defined next. For any $i, j \in [k]$, $\rho_{i \to j}(G, \bar{R})$ is obtained by relabeling any vertex with label R_i to label R_j. For any $i, j \in [k]$, $\eta_{i,j}(G, \bar{R})$ is obtained by adding all possible edges (u, v) between members of R_i and members of R_j. The clique-width of a graph G is the minimal k such that there exists a labeling $\bar{R}$ for which $(G, \bar{R})$ belongs to $CW(k)$. We denote the clique-width of G by $cw(G)$. The clique-width operations $\rho_{i \to j}$ and $\eta_{i,j}$ are well-defined for k-graphs. The definitions of these operations extend naturally to structures (V, E, S) which expand k-graphs with $S \subseteq v$.

A k-expression is a term t which consists of singletons, disjoint unions $\sqcup$, relabeling $\rho_{i \to j}$ and edge creations $\eta_{i,j}$, which witnesses that the graph $val(\mathsf{t})$ obtained by performing the operations on the singletons is of clique-width at most k. Every graph of tree-width at most k is of clique-width at most $2^{k+1} + 1$, cf. [16]. While computing the clique-width of a graph is NP-hard, S. Oum and P. Seymour showed that given a graph of clique-width k, finding a $(2^{3k+2} - 1)$-expression is fixed parameter tractable with clique-width as parameter, cf. [38,39].

For a formula φ, let $\mathrm{qr}(\varphi)$ denote the quantifier rank of φ. For every $q \in \mathbb{N}$ and vocabulary τ, we denote by $\mathrm{MSOL}^q(\tau)$ the set of MSOL-formulas on the vocabulary τ which have quantifier rank at most q. For two τ-structures $\mathcal{A}$ and $\mathcal{B}$, we write $\mathcal{A} \equiv^q \mathcal{B}$ to denote that $\mathcal{A}$ and $\mathcal{B}$ agree on all the sentences of quantifier rank q.

Definition 1 (Smooth operation). *An ℓ-ary operation* Op *on τ-structures is called* smooth *if for all $q \in \mathbb{N}$, whenever $\mathcal{A}_j \equiv^q \mathcal{B}_j$ for all $1 \leq j \leq \ell$, we have*

$$\mathsf{Op}(\mathcal{A}_1, \ldots, \mathcal{A}_\ell) \equiv^q \mathsf{Op}(\mathcal{B}_1, \ldots, \mathcal{B}_\ell)\,.$$

Smoothness of the clique-width operations is an important technical tool for us:

Theorem 2 (Smoothness, cf. [33])

1. *For every vocabulary τ, the disjoint union $\sqcup$ of two τ-structures is smooth.*

2. *For every* $1 \leq i \neq j \leq k$, $\rho_{i \to j}$ *and* $\eta_{i,j}$ *are smooth.*

It is convenient to reformulate Theorem 2 in terms of *Hintikka sentences* (see [19]):

Proposition 3 (Hintikka sentences) *Let* τ *be a vocabulary. For every* $q \in \mathbb{N}$ *there is a finite set*

$$\mathcal{H}^q_\tau = \{h_1, \ldots, h_\alpha\}$$

of $\mathrm{MSOL}^q(\tau)$*-sentences such that*

1. *Every* $h \in \mathcal{H}^q_\tau$ *has a finite model.*
2. *The conjunction* $h_1 \wedge h_2$ *of any two distinct* $h_1, h_2 \in \mathcal{H}^q_\tau$ *is unsatisfiable.*
3. *Every* $\mathrm{MSOL}^q(\tau)$*-sentence* θ *is equivalent to exactly one finite disjunction of sentences in* $\mathcal{H}^q_\tau$*.*
4. *Every* τ*-structure* $\mathcal{A}$ *satisfies a unique member* $hin^q_\tau(\mathcal{A})$ *of* $\mathcal{H}^q_\tau$*.*

In order to simplify notation we omit the subscript τ in hin^q_τ when τ is clear from the context.

Let $\tau_{\mathbf{S}}$ the be the vocabulary consisting of the binary relation symbol $\mathbf{E}$ and the unary relation symbol $\mathbf{S}$. Let $\tau_{\mathbf{S},k}$ extend $\tau_{\mathbf{S}}$ with the unary relation symbols $\mathbf{R}_1, \ldots, \mathbf{R}_k$. From Theorem 2 and Proposition 3 we get:

Theorem 4 *For every* $k \in \mathbb{N}^+$*:*

1. *There is* $\mathsf{F}_\sqcup : \mathcal{H}^q_{\tau_{\mathbf{S},k}} \times \mathcal{H}^q_{\tau_{\mathbf{S},k}} \to \mathcal{H}^q_{\tau_{\mathbf{S},k}}$ *such that, for every* $\mathcal{M}_1$ *and* $\mathcal{M}_2$*,* $\mathsf{F}_\sqcup(hin^q(\mathcal{M}_1), hin^q(\mathcal{M}_2)) = hin^q(\mathcal{M}_1 \sqcup \mathcal{M}_2)$*.*
2. *For every unary operation* $\mathsf{op} \in \{\rho_{p \to q}, \eta_{p,q} : p, q \in [k]\}$*, there is* $\mathsf{F}_{\mathsf{op}} : \mathcal{H}^q_{\tau_{\mathbf{S},k}} \to \mathcal{H}^q_{\tau_{\mathbf{S},k}}$ *such that, for every* $\mathcal{M}$*,* $\mathsf{F}_{\mathsf{op}}(hin^q(\mathcal{M})) = hin^q(\mathsf{op}(\mathcal{M}_1 \sqcup \mathcal{M}_2))$*.*

2.1 MSOL-Ising polynomials

For every $t \in \mathbb{N}^+$, let $\tau_t = \tau \cup \{\mathbf{S}_1, \ldots, \mathbf{S}_t\}$, where $\mathbf{S}_1, \ldots, \mathbf{S}_t$ are new unary relation symbols.

Definition 5 (MSOL-**Ising polynomials)** *For every* $t \in \mathbb{N}^+$*,* $\theta \in \mathrm{MSOL}(\tau_t)$ *and* $G = (V, E)$ *we define* $P_{t,\theta}(G; \bar{X}, \bar{Y})$ *as follows:*

$$P_{t,\theta}(G; \bar{X}, \bar{Y}) = \sum_{\substack{S_1 \sqcup \cdots \sqcup S_t = V: \\ G \models \theta(S_1, \ldots, S_t)}} \prod_{i=1}^{t} X_i^{|S_i|} \prod_{1 \leq i_1 \leq i_2 \leq t} Y_{i_1,i_2}^{|(S_{i_1} \times S_{i_2}) \cap E|}$$

$P_{t,\theta}$ *is the sum over partitions* $S_1, \ldots, S_t$ *of* V *such that* $(G, S_1, \ldots, S_t)$ *satisfies* θ *of the monomials obtained as the product of* $X_i^{|S_i|}$ *for all* $1 \leq i \leq t$ *and* $Y_{i_1,i_2}^{|(S_{i_1} \times S_{i_2}) \cap E|}$ *for all* $1 \leq i_1 < i_2 \leq t$*.*

Example 1 (Ising polynomial). The trivariate Ising polynomial $Z(G; x, y, z)$ is a partition function of the Ising model from statistical mechanics used to study phase transitions in physical systems in the case of constant energies and external field. $Z(G; x, y, z)$ is given by

$$Z(G; x, y, z) = \sum_{S \subseteq V} x^{|S|} y^{|\partial S|} z^{|E(S)|}$$

where ∂S denotes the set of edges between S and $V \backslash S$, and $E(S)$ denotes the set of edges inside S. $Z(G; x, y, z)$ was the focus of study in terms of hardness of approximation in [24] and in terms of hardness of computation under the exponential time hypothesis was studied in [27]. [27] also showed that $Z(G; x, y, z)$ is fixed-parameter polynomial time computable.

$Z(G; x, y, z)$ generalizes a bivariate Ising polynomial, which was studied for its combinatorial properties in [7]. [7] showed that $Z(G; x, y, z)$ contains the matching polynomial, the van der Waerden polynomial, the cut polynomial, and, on regular graphs, the independence polynomial and clique polynomial.

The evaluation of $P_{2,\mathsf{true}}(G; X_1, X_2, Y_{1,1}, Y_{1,2}, Y_{2,2})$ at $X_1 = x$, $X_2 = 1$, $Y_{1,1} = z$, $Y_{1,2} = y$ and $Y_{2,2} = 1$ gives $Z(G; x, y, z)$ and therefore $Z(G; x, y, z)$ is an MSOL-Ising polynomial.

Example 2 (Independence-Ising polynomial). The independence-Ising polynomial $I_{Is}(G; x, y)$ is given by

$$I_{Is}(G; x, y) = \sum_{\substack{S \subseteq V \\ S \text{ is an independent set}}} x^{|S|} y^{|\partial S|}$$

$I_{Is}(G; x, y)$ contains the independence polynomial as the evaluation $I(G; x) = I_{Is}(G; x, 1)$. See the survey [32] for a bibliography on the independence polynomial. The evaluation $y = 0$ is $I_{Is}(G; x, 0) = (1 + x)^{iso(G)}$, where $iso(G)$ is the number of isolated vertices in G. $I_{Is}(G; x, y)$ is an evaluation of an MSOL-Ising polynomial:

$$I_{Is}(G; x, y) = P_{2,\theta_I}(G; 1, x, 1, y, 1)$$

where $\theta_I(S) = \forall x \forall y \ (\mathbf{E}(x, y) \rightarrow (\neg S_2(x) \vee \neg S_2(y)))$.

Example 3 (Dominating-Ising polynomial). The Dominating-Ising polynomial is given by $D_{Is}(G; x, y, z)$

$$D_{Is}(G; x, y, z) = \sum_{\substack{S \subseteq V \\ S \text{ is a dominating set}}} x^{|S|} y^{|\partial S|} z^{|E(S)|}$$

where ∂S denotes the set of edges between S and $V \backslash S$. $D_{Is}(G; x, y, z)$ contains the domination polynomial $D(G; x)$. $D(G; x)$ is the generating function of its

dominating sets and we have $D_{Is}(G;x,1,1) = D(G;x)$. The domination polynomial first studied in [10] and it and its variations have received considerable attention in the literature in the last few years, see e.g. [5,1,2,4,29,31,3,6,12,26,17]. Previous research focused on combinatorial properties such as recurrence relations and location of roots. Hardness of computation was addressed in [30]. $D_{Is}(G;x,y,z)$ encodes the degrees of the vertices of G: the number of vertices with degree j is the coefficient of xy^j in $D_{Is}(G;x,y,z)$. $D_{Is}(G;x,y,z)$ is an MSOL-Ising polynomial given by $P_{2,\theta_D}(G;x,1,z,y,1)$, where

$$\theta_D = \forall x\,(S_1(x) \vee \exists y\ (S_1(y) \wedge \mathbf{E}(x,y)))\ .$$

2.2 MSOL-Ising polynomials vs MSOL-polynomials

Two classes of graph polynomials which have received attention in the literature are:

1. MSOL-polynomials on the vocabulary of graphs, and
2. MSOL-polynomials on the vocabulary of hypergraphs.

See e.g. [28] for the exact definitions. The former class contains graph polynomials such as the independence polynomial and the domination polynomial. The latter class contains graph polynomials such as the Tutte polynomial and the matching polynomial. Every graph polynomial which is MSOL-definable on the vocabulary of graphs is also MSOL-definable on the vocabulary of hypergraphs.

The class of MSOL-Ising polynomials strictly contains the MSOL-polynomials on graphs, see Fig. 1. The containment is by definition. For the strictness, we use the fact that by definition the maximal degree of any indeterminate in an MSOL-polynomial on graphs grows at most linearly in the number of vertices, while the maximal degree of y in the Ising polynomial $Z(K_{n,n};x,y,z)$ of the complete bipartite graph $K_{n,n}$ equals n^2.

Every MSOL-Ising polynomial $P_{t,\theta}$ is an MSOL-polynomial on the vocabulary of hypergraphs, given e.g. by

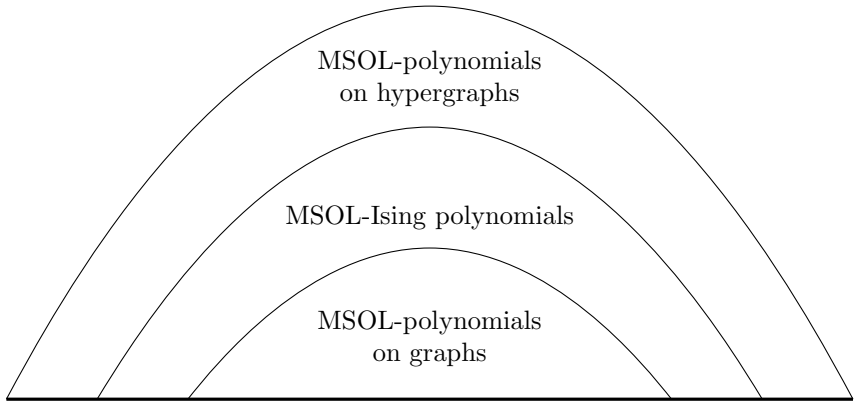
$$\sum_{\bar{S}} \sum_{\bar{B}} \prod_{i=1}^{t} X_i^{|S_i|} \prod_{1 \le i_1 \le i_2 \le t} Y_{i_1,i_2}^{|B_{i_1,i_2}|}$$

where the summation over $\bar{S}$ is exactly as in Definition 5, and the summation over $\bar{B}$ is over tuples $\bar{B} = (B_{i_1,i_2} : 1 \le i_1 \le i_2 \le t)$ of subsets of the edge set of G satisfying $\bigwedge_{i_1,i_2} \psi_{i_1,i_2}$, where

$$\psi_{i_1,i_2} = \forall x \forall y\,(B_{i_1,i_2}(x,y) \leftrightarrow (\mathbf{E}(x,y) \wedge (S_{i_1}(x) \wedge S_{i_2}(y) \vee S_{i_1}(y) \wedge S_{i_2}(x))))$$

We use the fact that $S_1, \ldots, S_t$ is a partition of the set of vertices is definable in MSOL.

Fig. 1. Containments of classes of graph polynomials definable in MSOL.



3 Main result

We are now ready to state the main theorem and prove a representative case of it.

Theorem 6 (Main theorem) *For every* MSOL-*Ising polynomial* $P_{t,\theta}$ *there is a function* $f(k,\theta,t)$ *such that* $P_{t,\theta}(G;\bar{X},\bar{Y},\bar{Z})$ *is computable on graphs* G *of size* n *and of clique-width at most* k *in running time* $O(n^{f(k,\theta,t)})$.

We prove the theorem for graph polynomials of the form

$$Q_\theta(G;X,Y) = \sum_{S\colon G\models\theta(S)} X^{|S|}Y^{|\partial S|}$$

for every $\theta \in \mathrm{MSOL}(\tau_{\mathbf{S}})$. The summation in Q_θ is over subsets S of the vertex set of G. The graph polynomials Q_θ are a notational variation of $P_{t,\theta}$ with $t=2$, $X_2 = 1$ and $Y_{1,1} = Y_{2,2} = 1$: for every $\theta \in \mathrm{MSOL}(\tau_2)$, $P_{2,\theta}(G;X,1,1,Y,1) = Q_{\theta'}(G;X,Y)$, where θ' is obtained from θ by substituting $\mathbf{S}_1$ with $\mathbf{S}$ and $\mathbf{S}_2$ with $\neg\mathbf{S}$. The proof for the general case is in similar spirit.

For every $q \in \mathbb{N}$ there is a finite set $\mathfrak{A}_q$ of $\mathrm{MSOL}(\tau_{\mathbf{S},k})$-Ising polynomials such that, for every formula $\theta \in \mathrm{MSOL}^q(\tau_{\mathbf{S}})$, Q_θ is a sum of members of $\mathfrak{A}_q$ (see below). The algorithm computes the values of the members of $\mathfrak{A}_q$ on G by dynamic programming over the parse term of G, and using those values, the value of Q_θ on G.

More precisely, for every $\beta \in \mathcal{H}^q_{\tau_{\mathbf{S},k}}$, let

$$A_\beta(G;\bar{x},\bar{y}) = \sum_{S\colon G\models\beta(S)} \prod_{1\le c\le k} x_c^{|S\cap R_c|} \prod_{1\le c_1,c_2\le k} y_{c_1,c_2}^{|(R_{c_1}\cap S)\times(R_{c_2}\setminus S)|}$$

and let

$$\mathfrak{A}_q = \{A_\beta : \beta \in \mathcal{H}^q_{\tau_{\mathbf{S},k}}\}.$$

Every $\theta \in \mathrm{MSOL}^q(\tau_{\mathbf{S}})$ also belongs to $\mathrm{MSOL}^q(\tau_{\mathbf{S},k})$, and hence there exists by Proposition 3 a set $\mathcal{H} \subseteq \mathcal{H}^q_{\tau_{\mathbf{S},k}}$ such that

$$\theta \equiv \bigvee_{h \in \mathcal{H}} h$$

Hence,

$$Q_\theta(G; X, Y) = \sum_{h \in \mathcal{H}} A_h(G; \bar{x}, \bar{y}) \tag{1}$$

setting $x_c = X$ and $y_{c_1,c_2} = Y$ for all $1 \le c, c_1, c_2 \le k$.

For tuples $\bar{b} = ((b_c : c \in [k]), (b_{c_1,c_2} : c_1, c_2 \in [k])) \in [n]^k \times [n]^{k^2}$, let $\mathsf{coeff}^G_\theta(\bar{b}) \in \mathbb{N}$ be the coefficient of

$$\prod_c x_c^{b_c} \prod_{c_1,c_2} y_{c_1,c_2}^{b_{c_1,c_2}}$$

in $A_\beta(G; \bar{x}, \bar{y})$.

Algorithm.

Given a k-graph G, the algorithm first computes a parse tree t as in [38,39]. The algorithm then computes $A_\beta(G; \bar{x}, \bar{y})$ for all $\beta \in \mathcal{H}^q_{\tau_{\mathbf{S},k}}$ by induction over t:

1. If G is a graph of size 1, then $A_\beta(G)$ is computed directly.
2. Let G be the disjoint union of H_A and H_B. We compute $\mathsf{coeff}^G_\beta(\bar{b})$ for every $\beta \in \mathcal{H}^q_{\tau_{\mathbf{S},k}}$ and $\bar{b} \in [n]^k \times [n]^{k^2}$ as follows:

$$\mathsf{coeff}^G_\beta(\bar{b}) = \sum_{h_1,h_2 : \mathsf{F}_\sqcup(h_1,h_2) \models \beta} \sum_{\bar{d}+\bar{e}=\bar{b}} \mathsf{coeff}^{H_A}_\beta(\bar{d}) \mathsf{coeff}^{H_B}_\beta(\bar{e})$$

3. Let $G = \rho_{p \to q}(H)$. We compute $\mathsf{coeff}^G_\beta(\bar{b})$ for every $\beta \in \mathcal{H}^q_{\tau_{\mathbf{S},k}}$ and $\bar{b} \in [n]^k \times [n]^{k^2}$ as follows:

$$\mathsf{coeff}^G_\beta(\bar{b}) = \sum_{h : \mathsf{F}_{\rho_{p \to q}}(h) \models \beta} \sum_{\bar{d}} \mathsf{coeff}^H_h(\bar{d})$$

where the inner summation is over $\bar{d}$ such that

$$b_c = \begin{cases} d_c & c \notin \{p, q\} \\ d_p + d_q & c = q \\ 0 & c = p \end{cases}$$

and

$$b_{c_1,c_2} = \begin{cases} d_{c_1,c_2} & c_1, c_2 \notin \{p,q\} \\ 0 & p \in \{c_1, c_2\} \\ d_{q,q} + d_{p,p} + d_{p,q} + d_{q,p} & c_1 = c_2 = q \\ d_{q,c_2} + d_{p,c_2} & c_1 = q, c_2 \notin \{q,p\} \\ d_{c_1,q} + d_{c_1,p} & c_2 = q, c_1 \notin \{q,p\} \end{cases}$$

4. Let $G = \eta_{p,q}(H)$ with $p \neq q$. Let n_G be the number of vertices in G. We compute $\mathsf{coeff}^G_\beta(\bar{b})$ for every $\beta \in \mathcal{H}^q_{\tau_{\mathbf{S},k}}$ and $\bar{b} \in [n]^k \times [n]^{k^2}$ as follows:

$$\mathsf{coeff}^G_\beta(\bar{b}) = \sum_{h:\mathsf{F}_{\eta_{p,q}}(h) \models \beta} \sum_{\bar{d}} \mathsf{coeff}^H_h(\bar{d})$$

where the summation is over $\bar{d}$ such that $b_c = d_c$ and

$$b_{c_1,c_2} = \begin{cases} d_{c_1,c_2} & \{c_1, c_2\} \neq \{p,q\} \\ d_p(n_G - d_q) & c_1 = p,\ c_2 = q \\ d_q(n_G - d_p) & c_1 = q,\ c_2 = p \end{cases}$$

Finally, the algorithm computes Q_θ as the sum from Eq. (1).

3.1 Runtime

The main observations for the runtime analysis are:

- The size of the set $\mathcal{H}^q_{\tau_{\mathbf{S},k}}$ of Hintikka sentences is a function of k but does not depend on n. Let $\mathsf{s}^q_{\tau_{\mathbf{S},k}} = |\mathcal{H}^q_{\tau_{\mathbf{S},k}}|$.
- By definition of A_β, for a monomial $\prod_{1 \leq c \leq k} x_c^{i_c} \prod_{1 \leq c_1, c_2 \leq k} y_{c_1,c_2}^{j_{c_1,c_2}}$ to have a non-zero coefficient, it must hold that $i_c \leq n$ and $j_{c_1,c_2} \leq \binom{n}{2}$, since i_c and j_{c_1,c_2} are sizes of sets of vertices and sets of edges, respectively.
- The coefficient of any monomial of A_β is at most 2^n.
- The parse tree guaranteed in [38,39] is of size $O(n^c f_1(k))$ for suitable f_1 and c.

The algorithm performs a single operation for every node of the parse tree.

Singletons: the coefficients of every $A_\beta \in \mathfrak{A}_q$ for a singleton k-graph can be computed in time $O(k)$, which can be bounded by $O(n^k)$.

Disjoint union, **recoloring and edge additions**: the algorithm sums over (1) $h \in \mathcal{H}^q_{\tau_{\mathbf{S},k}}$ or pairs $(h_1, h_2) \in \left(\mathcal{H}^q_{\tau_{\mathbf{S},k}}\right)^2$ and (2) over $\bar{d} \in [n]^k \times [n]^{k^2}$ or pairs $(\bar{d}, \bar{e}) \in \left([n]^k \times [n]^{k^2}\right)^2$, then (3) performs a fixed number of arithmetic operations on numbers which can be written in $O(n)$ space.

Each node in the parse tree requires time at most $O\left(n^k (\mathsf{s}^q_{\tau_{\mathbf{S},k}})^2 \left([n]^k \times [n]^{k^2}\right)^2\right)$. Since the size of the parse tree is $O(n^c f_1(k))$, the algorithm runs in fixed-parameter polynomial time.

4 Conclusion

We have defined a new class of graph polynomials, the MSOL-Ising polynomials, extending the MSOL-polynomials on the vocabulary of graphs and have shown that every MSOL-Ising polynomial can be computed in fixed-parameter polynomial time. This result raises the question of which graph polynomials are MSOL-Ising polynomials. In previous work [23,35,28] we have developed a method based on connection matrices to show that graph polynomials are not definable in MSOL over either the vocabulary of graphs or hypergraphs.

Problem 1. How can connection matrices be used to show that graph polynomials are not MSOL-Ising polynomials?

The Tutte polynomial does not seem to be an MSOL-Ising polynomial. [22] proved that the Tutte polynomial can be computed in subexponential time for graphs of bounded clique-width. More precisely, the time bound in [22] is of the form $\exp(n^{1-f(cw(G))})$, where $0 < f(i) < 1$ for all $i \in \mathbb{N}$.

Problem 2. Is there a natural infinite class of graph polynomials definable in MSOL which includes the Tutte polynomial such that membership in this class implies *fixed parameter subexponential time* computability with respect to clique-width (i.e., that the graph polynomial is computable in $\exp(n^{1-g(cw(G))})$ time for some function g satisfying $0 < g(i) < 1$ for all $i \in \mathbb{N}$)?

Acknowledgement

We are grateful to Nadia Labai for her comments and suggestions.

References

1. Saeed Akbari, Saeid Alikhani, Mohammad Reza Oboudi, and Yee-Hock Peng. On the zeros of domination polynomial of a graph. *Combinatorics and Graphs*, 531:109–115, 2010.
2. Saeed Akbari, Saeid Alikhani, and Yee-hock Peng. Characterization of graphs using domination polynomials. *European Journal of Combinatorics*, 31(7):1714–1724, 2010.
3. Saieed Akbari and Mohammad Reza Oboudi. Cycles are determined by their domination polynomials. *Ars Comb.*, 116:353–358, 2014.
4. Mehdi Alaeiyan, Amir Bahrami, and Mohammad Reza Farahani. Cyclically domination polynomial of molecular graph of some nanotubes. *Digest Journal of Nanomaterials and Biostructures*, 6(1):143–147, 2011.
5. Saeid Alikhani and Yee-Hock Peng. Dominating sets and domination polynomials of paths. *International journal of Mathematics and Mathematical sciences*, 2009.
6. Saeid Alikhani and Yee-hock Peng. Introduction to domination polynomial of a graph. *Ars Combinatoria*, 114:257–266, 2014.
7. Daniel Andrén and Klas Markström. The bivariate ising polynomial of a graph. *Discrete Applied Mathematics*, 157(11):2515–2524, 2009.

8. Artur Andrzejak. An algorithm for the Tutte polynomials of graphs of bounded treewidth. *DMATH: Discrete Mathematics*, 190, 1998.
9. Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991.
10. Jorge L. Arocha and Bernardo Llano. Mean value for the matching and dominating polynomial. *Discussiones Mathematicae Graph Theory*, 20(1):57–69, 2000.
11. Markus Bläser and Christian Hoffmann. On the Complexity of the Interlace Polynomial. In *STACS 2008*, pages 97–108. IBFI Schloss Dagstuhl, 2008.
12. Jason I. Brown and Julia Tufts. On the roots of domination polynomials. *Graphs and Combinatorics*, 30(3):527–547, 2014.
13. Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and computation*, 85(1):12–75, 1990.
14. Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic: a language-theoretic approach*, volume 138. Cambridge University Press, 2012.
15. Bruno Courcelle, Johann A Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.
16. Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1):77–114, 2000.
17. Markus Dod, Tomer Kotek, James Preen, and Peter Tittmann. Bipartition polynomials, the ising model and domination in graphs. *Discussiones Mathematicae Graph Theory*, 35(2):335–353, 2015.
18. Rod G. Downey and Michael Ralph Fellows. *Parameterized complexity*, volume 3. springer Heidelberg, 1999.
19. Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Springer Science & Business Media, 2005.
20. Jörg Flum and Martin Grohe. Parameterized complexity theory, volume xiv of texts in theoretical computer science. an eatcs series, 2006.
21. Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Algorithmic lower bounds for problems parameterized with clique-width. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 493–502, 2010.
22. Omer Giménez, Petr Hlinený, and Marc Noy. Computing the Tutte polynomial on graphs of bounded clique-width. *SIAM Journal on Discrete Mathematics*, 20(4):932–946, 2006.
23. Benny Godlin, Tomer Kotek, and Johann A. Makowsky. Evaluations of graph polynomials. In *Graph-Theoretic Concepts in Computer Science, 34th International Workshop, WG 2008, Durham, UK, June 30 - July 2, 2008. Revised Papers*, pages 183–194, 2008.
24. Leslie Ann Goldberg, Mark Jerrum, and Mike Paterson. The computational complexity of two-state spin systems. *Random Structures & Algorithms*, 23(2):133–154, 2003.
25. François Jaeger, Dirk L. Vertigan, and Dominic J. A. Welsh. On the computational complexity of the jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, 108(01):35–53, 1990.
26. Sahib Sh Kahat, Abdul Jalil M Khalaf, and Roslam Roslan. Dominating sets and domination polynomial of wheels. *Asian Journal of Applied Sciences*, 2(3), 2014.
27. Tomer Kotek. Complexity of ising polynomials. *Combinatorics, Probability and Computing*, 21(05):743–772, 2012.

28. Tomer Kotek and Johann A. Makowsky. Connection matrices and the definability of graph parameters. *Logical Methods in Computer Science*, 10(4), 2014.
29. Tomer Kotek, James Preen, Frank Simon, Peter Tittmann, and Martin Trinks. Recurrence relations and splitting formulas for the domination polynomial. *Electr. J. Comb.*, 19(3):P47, 2012.
30. Tomer Kotek, James Preen, and Peter Tittmann. Domination polynomials of graph products. *To appear in Journal of Combinatorial Mathematics and Combinatorial Computing*.
31. Tomer Kotek, James Preen, and Peter Tittmann. Subset-sum representations of domination polynomials. *Graphs and Combinatorics*, 30(3):647–660, 2014.
32. Vadim E. Levit and Eugen Mandrescu. The independence polynomial of a graph-a survey. In *Proceedings of the 1st International Conference on Algebraic Informatics*, volume 233254, 2005.
33. Johann A. Makowsky. Algorithmic uses of the Feferman–Vaught theorem. *Annals of Pure and Applied Logic*, 126(1):159–213, 2004.
34. Johann A Makowsky. Coloured Tutte polynomials and Kauffman brackets for graphs of bounded tree width. *Discrete Applied Mathematics*, 145(2):276–290, 2005.
35. Johann A. Makowsky. Connection matrices for MSOL-definable structural invariants. In *Logic and Its Applications, Third Indian Conference, ICLA 2009, Chennai, India, January 7-11, 2009. Proceedings*, pages 51–64, 2009.
36. Johann A. Makowsky, Udi Rotics, Ilya Averbouch, and Benny Godlin. Computing graph polynomials on graphs of bounded clique-width. In *Graph-theoretic concepts in computer science*, pages 191–204. Springer, 2006.
37. Steven D. Noble. Evaluating the Tutte polynomial for graphs of bounded tree-width. In *Combinatorics, Probability and Computing, Cambridge University Press*, volume 7. 1998.
38. Sang-Il Oum. Approximating rank-width and clique-width quickly. In Dieter Kratsch, editor, *WG*, volume 3787 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2005.
39. Sang-Il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96, 2006.
40. Leslie G Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.