Seminar Algorithmic Meta-Theorems and Parameterized Complexity: Clique–Width

Jonathan Lennart Schneider

RWTH University, Aachen, Germany

Abstract. We present the meta-theorem for graph problems over graphs of bounded clique-width which are expressible in monadic second-order logic, originally given by Courcelle et al. (see [1]), in combination with the clique-width approximation algorithm proposed by Oum [9]. The purpose of this paper is to provide an introduction into the graph complexity measure clique-width and parameterized complexity.

Keywords: Clique–Width · MSOL

1 Introduction

It the year 2000, Courcelle and Olariu introduced a complexity measure on vertex labeled graphs, called clique-width, using expressions built with so called graph operations. These expressions induce a unique labeled graph up to isomorphism. That is, a graph with labels assigned to its vertices. By posing a bound on the number of different operations of a specific operation type, namely those operations which define the singleton graph with a specified vertex label, Courcelle and Olariu define the clique-width of a labeled graph G. The clique-width of G, also denoted $cwd(G)$, is the (well defined) minimum number of different labels used to construct an expression which corresponds to G [3].

Roughly speaking, the naming clique-width comes from the fact that this measure, analogous to the notion of tree-width, describes how clique-like a given graph is. Graphs which are cliques have a clique-width of 2 while trees have clique-width 3 [3]. Furthermore, other classes of graphs of bounded clique-width, like so called P_4-tidy graphs, have been studied and efficient algorithms for constructing corresponding expressions have been found [1]. Golumbic and Rotics have provided graph classes with unbounded clique-width [5]. It has also been shown that deciding whether or not a given graph has clique-width at most k, for some input $k \in \mathbb{N}_{>0}$, is NP-complete [4].

Courcelle, Makowsky and Rotics provide a well known meta-theorem with regards to the notion of clique-width in their paper *Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width* [1]. They show that problems which can be expressed in monadic second-order logic (MSOL) on graphs with clique-width at most $k \in \mathbb{N}$ and second-order quantification only over sets of vertices can be solved in $\mathcal{O}(f(|G|))$-time, if there exists a known $\mathcal{O}(f(|G|))$-time algorithm for constructing a corresponding expression witnessing the clique-width bound of G.

In fact, they show the above property more generally for so called LinEMSOL optimization problems, in which the problem consists of finding optimal sets of vertices with regards to some evaluation function and conformance with some MSOL formula [1]. In this paper, we present this result by showing the implication derived in combination with Oum's clique-width approximation algorithm, which computes an approximate expression, for graphs of clique-width bounded by some constant, in polynomial time [9].

For didactic reasons, some general definitions or notions introduced in the reference paper will be presented below in a specific, narrowed down context.

2 Preliminaries

In order to define the clique-width of an undirected, vertex labeled graph, we first have to introduce the notion of so-called p-graphs and algebraic expressions defining them.

Definition 1. *Let $p \in \mathbb{N}$. A* ***p-graph*** *$G = (V, E)$ is an undirected, finite graph with labels from $\{1, \dots, p\}$ assigned to its vertices.*

Definition 2. *An arbitrary p-graph G can be represented as a* ***logical structure*** *$G(\tau_{1,p})$ with the domain V and signature $\tau_{1,p} = (E, U_1, \dots, U_p)$, where the relation symbol $E \subseteq V \times V$ corresponds to exactly the adjacent pairs of vertices in G and for all $i \in \{1, \dots, p\}$, $U_i \subseteq V$ is a relation symbol corresponding to exactly those vertices that are labeled i in G. That is, $G(\tau_{1,p}) = (V, E, U_1, \dots, U_p)$.*

It is to note that not every $\tau_{1,p}$-structure corresponds to a p-graph. Throughout the rest of the paper, we will omit the focus on these structures and not differentiate between p-graphs and their corresponding logical representations. This definition allows us to evaluate formulas from MSOL on p-graphs. These are formulas from first-order logic with the addition of quantification over subsets of the domain, typically represented with capital letters (e.g. $\exists X(\dots)$). Furthermore, for some first-order and second-order variables x and X, membership testing is expressed by the atomic formula $X(x)$, which is valid iff $x \in X$ holds.

Let C be an arbitrary class of p-graphs and let φ and $\theta(X_1, \dots, X_n)$ be MSOL($\tau_{1,p}$) formulas over the same signature such that φ has no and θ has $n \in \mathbb{N}$ free set variables.

Definition 3. *A* ***MSOL($\tau_{1,p}$) decision problem*** *over C is defined by φ and C. The problem is to decide, given a problem instance $G \in C$, whether or not $G(\tau_{1,p}) \vDash \varphi$ holds, i.e. the given graph's logical structure is a model for φ.*

Let $a_{i,j} \in \mathbb{Z}$ for $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$ be arbitrary coefficients for some $m, n \in \mathbb{N}$ and for an arbitrary $G \in C$, let $f_1, \dots, f_m : V_G \to \mathbb{N}$ be functions assigning integers to the vertices of G. Any assignment of free variables of θ $z : \{X_1, \dots, X_n\} \to \mathcal{P}(V_G)$ can be represented as an n-tuple $(D_1, \dots, D_n)$ with $D_1, \dots, D_n \subseteq V_G$ corresponding to the n assignments of free variables to subsets of V_G. Symmetrically, every such tuple corresponds to an assignment z_D.

With this, we can define $\boldsymbol{sat(G,\theta)}$ as the set of n-tuples corresponding to assignments which satisfy θ in combination with $G(\tau_{1,p})$:

$$sat(G,\theta) = \{(D_1,\dots,D_n) \in \mathcal{P}(V_G)^n \mid (G(\tau_{1,p}), z_D) \vDash \theta(X_1,\dots,X_n)\}.$$

Furthermore, we define $\boldsymbol{h(D_1,\dots,D_n)}$ and $\boldsymbol{Max\_h(X)}$, mapping assignment tuples to their so-called evaluation value and computing a maximal evaluation value respectively:

$$h(D_1,\dots,D_n) = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}(\sum_{v\in D_i} f_j(v)),$$

$$Max\_h(X) = \max\{h(D_1,\dots,D_n) \mid (D_1,\dots,D_n) \in X\}.$$

In the following, we will define optimization problems with constraints provided as MSOL formulas by combining two notions presented by Courcelle et al. and thus deviate from the original notation (see [1]).

Definition 4. *A* ***LinEMSOL($\boldsymbol{\tau_{1,p}}$) optimization problem*** *over C is defined by $\theta(X_1,\dots,X_n)$, $a_{i,j} \in \mathbb{Z}$ for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,m\}$ and C, where $n,m \in \mathbb{N}$. An input instance for this problem consists of a p-graph $G \in C$ and m so called evaluation functions $f_1,\dots,f_m : V_G \to \mathbb{N}$.*

The problem for such an instance is to find an optimal assignment z^* of the free set variables of θ, i.e. $z^* = (D_1^*,\dots,D_n^*) \in \mathcal{P}(V_G)^n$ such that the following holds:

$$h(D_1^*,\dots,D_n^*) = Max\_h(sat(G,\theta)).$$

That is, in plain English: find a satisfying assignment z^* of $X_1,\dots,X_n$ such that it has the maximum evaluation value $h(D_1^*,\dots,D_n^*)$ among all candidate assignments satisfying θ. It follows that each MSOL($\tau_{1,p}$) problem can be expressed as a LinEMSOL($\tau_{1,p}$) optimization problem [1].

2.1 Clique-Width

Through the notion of graph complexity measures, one can derive algorithmic results for solving LinEMSOL($\tau_{1,p}$) problems over classes of graphs with bounded such complexity. Courcelle et al. used a measure called clique-width to formulate the meta-theorem presented in this paper (see [1]). In order to define the clique-width of a p-graph, we will need algebraic expressions consisting of so-called graph operations. Every such expression defines a set of isomorphic p-graphs and every p-graph can be constructed by at least one expression.

Definition 5. *Let $G = (V,E,U_1,\dots,U_p)$ and $H = (V',E',U_1',\dots,U_p')$ be p-graphs with disjoint sets of vertices, represented as logical structures. There are four types of* ***graph operations*** *which are defined as follows:*

- *Union:* $G \oplus H = (V \cup V', E \cup E', U_1 \cup U_1', \dots, U_p \cup U_p')$
- *Edge connection:* $\eta_{i,j}(G) = (V, E \cup \{(u,v) \mid u \in U_i, v \in U_j\}, U_1,\dots,U_p)$

- *Vertex relabeling:* $\rho_{i\to j}(G) = (V, E, U_1^*, \ldots, U_p^*)$ *such that* $U_i^* = \emptyset$, $U_j^* = U_i \cup U_j$ *and* $U_k^* = U_k$ *for* $k \in \{1, \ldots, p\} \setminus \{i, j\}$
- *Singleton creation:* $i(v) = (\{v\}, \emptyset, U_1^*, \ldots, U_p^*)$ *such that* $U_i^* = \{v\}$ *and* $U_k^* = \emptyset$ *for* $k \in \{1, \ldots, p\} \setminus \{i\}$

Since these operations take p-graphs and return them, they can be chained together to form expressions, where expressions of the form $i(v)$ *are atomic expressions. Such a term is called a* ***k-expression*** *if all labels present in the expression are in* $\{1, \ldots, k\}$ *for some* $k \in \mathbb{N}$.

Definition 6. *Let* G *be a p-graph. The* ***clique-width*** *of* G, $cwd(G)$, *is defined as the smallest* $k \in \mathbb{N}$ *such that there is a k-expression defining* G. *That is,*

$$cwd(G) = \min\{k \in \mathbb{N} \mid G \in C(k)\},$$

where $C(k)$ *denotes the class of p-graphs defined by arbitrary k-expressions.*

Clique-width is a complexity measure of graphs similar to that of tree-width, in the sense that algorithmic properties of certain graph problems can be derived when imposing a fixed upper-bound on the clique-width of the instance-graphs. It is known that a class of graphs with bounded tree-width also has bounded clique-width, while the converse does not hold [2].

A k-expression defining a p-graph G can be seen as a decomposition of G which yields desirable results with regards to model checking, i.e. we will be able to derive assignments that satisfy a formula in combination with G from the subexpressions that G is composed of. This composition method uses a generalization of the Feferman-Vaught theorem (see [8]) and so-called translation schemes for generating formulas and inferring results on the structure of a k-expression.

2.2 Translation Schemes and the Feferman–Vaught Theorem

Let τ and σ be arbitrary signatures. Courcelle et al. introduce the notion of a τ-σ-translation scheme $\Phi = (\varphi, \psi_1, \ldots, \psi_m)$, which is a tuple of MSOL(τ) formulas respecting certain properties of the signature σ. For the purpose of this paper, we will avoid the general, generic definition and assume that $\tau = \sigma = \tau_{1,p}$ holds.

This implies that, Φ is of the form $\Phi = (\varphi, \psi_E, \psi_1, \ldots, \psi_p)$ where $\varphi, \psi_1, \ldots, \psi_p$ have one and ψ_E has two free first-order variables. Such a translation scheme induces two mappings, $\Phi^* : Structure(\tau_{1,p}) \to Structure(\tau_{1,p})$ (transduction) and $\Phi^{\#} : \mathrm{MSOL}(\tau_{1,p}) \to \mathrm{MSOL}(\tau_{1,p})$ (backwards translation).

Given a $\tau_{1,p}$-structure $\mathcal{A}$, $\Phi^*(\mathcal{A})$ defines a structure, where the universe of $\mathcal{A}$ is restricted to those elements x satisfying $\mathcal{A} \vDash \varphi(x)$. Similarly, the interpretation of E (or U_i) in $\Phi^*(\mathcal{A})$ is defined as those element pairs x, y (elements x) in the universe of $\Phi^*(\mathcal{A})$ such that $\mathcal{A} \vDash \psi_E(x, y)$ ($\mathcal{A} \vDash \psi_i(x)$) holds.

For some MSOL($\tau_{1,p}$) formula θ, a backwards translation $\Phi^{\#}(\theta)$ is defined inductively such that first-order and second-order quantification is restricted to the elements present in the transduction's universe, expressed via φ. Also, every subformula of the form $E(x, y)$ (or $U_i(x)$) is replaced by the formulas encoding the interpretation of these symbols in the transduction, i.e. $\psi_E(x, y)$ ($\psi_i(x)$). This concept will be demonstrated in example 1.

Lemma 1 (Courcelle et al. [1]). *For every edge connection or vertex renaming graph operation $\alpha = \eta_{i,j}$ ($\alpha = \rho_{i\to j}$), one can compute a $(\tau_{1,p})$-$(\tau_{1,p})$-translation scheme Φ_α, such that for any MSOL($\tau_{1,p}$) formula $\theta(X_1, \ldots, X_n)$ and p-graph G, the following holds:*

$$sat(\alpha(G),\ \theta) = sat(G,\ \Phi^{\#}_{\alpha}(\theta)).$$

The formula $\Phi^{\#}_{\alpha}(\varphi)$ has quantifier depth less than or equal to that of θ.

Corollary 1. *For $\alpha = \eta_{i,j}$ or $\alpha = \rho_{i\to j}$, any MSOL($\tau_{1,p}$) formula θ and any p-graph G,*

$$Max\_h(sat(\alpha(G),\ \theta)) = Max\_h(sat(G,\ \Phi^{\#}_{\alpha}(\theta)))$$

holds for arbitrary h. Specifically, any optimal assignment from $sat(G,\ \Phi^{\#}_{\alpha}(\theta))$ is also an optimal assignment from $sat(\alpha(G), \theta)$.

As an example, we provide a translation scheme for an edge connection graph operation and the corresponding backwards translation of an example formula.

Example 1. Let $\theta(X_1) = \exists x(\forall y(X_1(y) \to E(x, y)))$ be a MSOL($\tau_{1,p}$) formula expressing that an element exists such that all elements from X_1 have an edge to this node in the corresponding graph. Let G be an arbitrary p-graph and $\eta_{i,j}$ be an edge connection graph operation for $i, j \leq p$.

We define $\psi_E(x, y) := E(x, y) \vee (U_i(x) \wedge U_j(y))$, while $\psi_k(x) := U_k(x)$ for all $k \in \{1, \ldots, p\}$ and $\varphi(x) := (x = x)$. This induces the translation scheme $\Phi_{\eta_{i,j}} = (\varphi, \psi_E, \psi_1, \ldots, \psi_p)$, where $sat(\alpha(G),\ \theta) = sat(G,\ \Phi^{\#}_{\alpha}(\theta))$ holds.

Specifically, θ's backwards translation $\Phi^{\#}(\theta)$ is defined as:

$$\exists x(\varphi(x) \wedge \neg(\exists y(\varphi(y) \wedge \neg((X_1(y) \wedge \varphi(y)) \to (\psi_E(x, y) \wedge \varphi(x) \wedge \varphi(y))))))$$
$$\wedge \forall y(X_1(y) \to \varphi(y)),$$

which can be simplified to:

$$\exists x(\varphi(x) \wedge \forall y(\neg\varphi(y) \vee (X_1(y) \to \psi_E(x, y)))) \wedge \forall y(X_1(y) \to \varphi(y)).$$

The subformula $\forall y(X_1(y) \to \varphi(y))$ expresses that all elements occurring in θ's free set variables need to present in the transduction's universe.

Lemma 1 allows us to modify formulas such that the evaluation result with regards to a structure, that is composed of a smaller structure using unary graph operations, can be inferred from the evaluation result of the modified formula on the smaller structure. As indicated by Courcelle et al., a similar result for the $\oplus$-operation can be derived from the generalization of the Feferman-Vaught theorem. This theorem infers the first-order theory of a structure composed of other structures from the first-order theories of these other structures [1, 8, 7].

Lemma 2 (Courcelle et al. [1]). *Let $\theta(X_1, \ldots, X_n)$ be a MSOL($\tau_{1,p}$) formula. One can construct two families of MSOL($\tau_{1,p}$) formulas, $\varphi'_i(X_1, \ldots, X_n)$ and $\psi'_i(X_1, \ldots, X_n)$ for $i \in \{1, \ldots, m\}$ and some $m \in \mathbb{N}$, such that for all p-graphs G and H having disjoint sets of vertices, the following holds:*

$$sat(G \oplus H, \theta) = \bigcup_{i=1}^{m} sat(G, \varphi'_i) \boxtimes sat(H, \psi'_i),$$

where $X \boxtimes Y = \{(D_1 \cup E_1, \ldots, D_n \cup E_n) \mid (D_1, \ldots, D_n) \in X, (E_1, \ldots, E_n) \in Y\}$. Furthermore, φ'_i and ψ'_i all have quantifier depth less than or equal to that of θ.

Corollary 2. *For G, H and $\theta(X_1, \ldots, X_n)$ as above,*

$$Max\_h(sat(G \oplus H, \theta)) = \max\{Max\_h(sat(G, \varphi'_i)) + Max\_h(sat(H, \psi'_i)) \mid i \in \{1, \ldots, m\}\}$$

holds for arbitrary h. This can be derived from Lemma 2 and minor statements given by Courcelle et al. [1]. Furthermore, one can infer an optimal assignment $(D_1 \dot\cup E_1, \ldots, D_n \dot\cup E_n) \in sat(G \oplus H, \theta)$, where $(D_1, \ldots, D_n) \in sat(G, \varphi'_i)$ and $(E_1, \ldots, E_n) \in sat(H, \psi'_i)$ are optimal assignments for the pair of formulas φ'_i, ψ'_i having the largest added Max_h value.

3 Solving MSOL Problems over Graphs of Bounded Clique-Width

The above Lemma 1 and Lemma 2 allow one to compute $Max\_h(sat(G, \theta))$ and a corresponding optimal assignment inductively on the structure of the so-called parse tree of G, which corresponds to an expression g that defines G. Such a tree is denoted by $tree(g)$. The nodes in $tree(g)$ correspond to the graph operations present in g, where $i(v)$-operations hold no children and make up the tree's leafs, while η- and ρ-operations have one and $\oplus$-operations have two children. Every root node n of a subtree t of $tree(g)$ induces a labeled graph, $graph(n)$ via the sub-expression corresponding to t. We will now describe the main algorithm presented in the reference paper in combination with a result on approximate clique-width decompositions as presented by Oum [9].

Let $k \in \mathbb{N}$ and let C be a class of p-graphs having bounded clique-width k. That is, all graphs in C have clique-width at most k. Furthermore, let P be a LinEMSOL($\tau_{1,p}$) optimization problem over C defined by some formula $\theta(X_1, \ldots, X_n)$ and coefficients $a_{i,j} \in \mathbb{Z}$ for $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$.

Theorem 1 (Courcelle et al. [1] and Oum [9]). *For fixed k, the optimization problem P over graphs of clique-width bounded by k can be solved in polynomial time.*

Proof. Let k be fixed. Let $G \in C$ be an arbitrary graph instance with evaluation functions $f_1, \ldots, f_m$. As shown by Oum, one can approximate the clique-width of a given graph, i.e. output a $(8^k - 1)$-expression defining G in $\mathcal{O}(|V_G|^3)$-time. This is achieved via an intermediary step, where a so-called rank-decomposition is computed, which is then used to create the $(8^k - 1)$-expression [9]. Let g be such a $(8^k - 1)$-expression computed with this algorithm. As described by Courcelle et al., the parse tree of g will be traversed two times:

1. In the first, top-down traversal, one recursively assigns a set of formulas to each node of the tree by utilizing Lemma 1 and 2:
   Let θ be assigned to the root node of $tree(g)$. If the current node corresponds to a unary graph operation, one assigns the backwards translation of each currently assigned formula to the child node of the current node. This principle is shown in Fig. 2. Otherwise, if the current node corresponds to the $\oplus$-operation, one assigns the formulas $\varphi'_1, \ldots, \varphi'_m$ and $\psi'_1, \ldots, \psi'_m$ to the left and right child for all currently assigned formulas, as shown in Fig. 1. At the leaf nodes, which correspond to singleton operations, no computation needs to be done.
2. The second subroutine traverses $tree(g)$ from the bottom up. For each formula φ present at a node x, the value $Max\_h(sat(graph(x), \varphi))$ and an assignment tuple from $sat(graph(x), \varphi)$ having the maximal evaluation value are computed from those solutions computed at x's children:
   If the current node x is a leaf, one can compute $Max\_h(sat(graph(x), \varphi))$ and an optimal assignment via brute force, since $graph(x)$ defines only a singleton graph. If the current node corresponds to a unary graph operation, one can simply copy the maximal value $Max\_h(sat(graph(x), \varphi))$ and an optimal assignment from the computation done at φ's backwards translation in the child node of x (see Corollary 1). Otherwise, x corresponds to a $\oplus$-operation and one can compute Max_h and an optimal assignment from the previously computed values in the left and right child of x, as shown in Corollary 2.

After traversing $tree(g)$ for the second time, an optimal assignment z^* and the corresponding evaluation value with regards to the optimization problem P have been computed at the root node of the parse tree.

We provide an intuition for the running time of the algorithm described above. Note that k is fixed. It is clear that the size of $tree(g)$ is in $\mathcal{O}(|V_G|^3)$. Courcelle et al. argue that the maximal size of the set of formulas assigned to each node of $tree(g)$ is bounded by a constant depending on θ and $p \leq 8^k - 1$. They make this argument, since up to tautological equivalence, there are only finitely many MSOL$(\tau_{1,p})$ formulas with quantifier depth less than or equal to that of θ having n free set variables that express properties of p-graphs [1]. Furthermore, they state that the computations in the second traversal can be performed in constant time. Here, the quintessential task is to find a solution to $Max\_h(sat(graph(x), \varphi))$, where $graph(x)$ corresponds to a singleton graph and φ is one of the previously generated formulas.

If one was able to provide a constant bound on the total number of formulas occurring, these statements follow immediately. But since the above statement only provides a bound for formulas up to tautological equivalence, it is not clear how one assures that the number of syntactically different formulas occurring is also bounded. The author of this paper assumes that through some intermediate, syntactic normalization of formulas, as mentioned by Grohe (see [6]), one is able to avoid the creation of arbitrarily many equivalent formulas, given a class of p-graphs having bounded clique-width.

Altogether, we get that for fixed k, one can compute a clique-width approximation in $\mathcal{O}(|V_G|^3)$-time while the computations from above, under the previously stated assumption, can be conducted in time constant in the size of $tree(g)$. Thus, for a fixed clique-width bound k, the whole procedure has a runtime in $\mathcal{O}(|V_G|^3)$ as well. That is, the running time can be written as $\mathcal{O}(f(k) \cdot |V_G|^3)$ where $f(k)$ is a function exponential only in k.

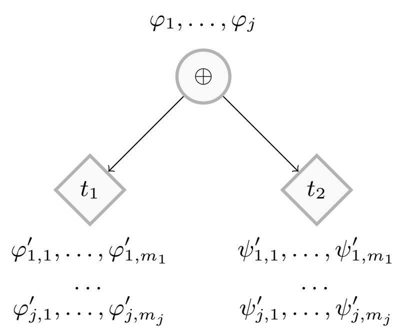

Fig. 1. Downward inductive step on $\oplus$-operations, where $\varphi'_{i,p}$ and $\psi_{i,p}$ are computed from φ_i using Lemma 2.

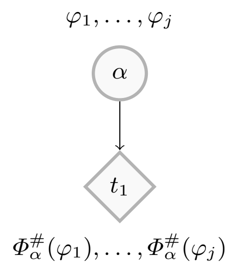

Fig. 2. Downward inductive step on η- and ρ-operations, where $\Phi^{\#}_{\alpha}(\varphi_i)$ is computed as shown in Lemma 1.

4 Conclusion

We have presented the meta-theorem for graph problems expressible in MSOL with second-order quantification over sets of vertices, where the class of graphs considered is of bounded clique-width. The original approach was proposed by Courcelle et al. [1]. We deviate from the exact formulation of their theorem by providing a combined result with Oum's clique-width approximation algorithm [9]. While the original algorithm relies on another algorithm computing a k-expression for a graph instance, Oum's result implies that this is no longer a requirement. Section 3 highlights a perceived shortcoming with regards to Courcelle et al.'s argumentation on the algorithm's runtime, which the author of this paper encountered while researching the topic. We believe that a more straightforward argumentation on the algorithms's running time and connection with the Feferman-Vaught theorem might be fruitful areas for further research.

References

1. 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.
2. Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete applied mathematics*, 108(1-2):23–52, 2001.
3. Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.
4. Michael R Fellows, Frances A Rosamond, Udi Rotics, and Stefan Szeider. Clique-width minimization is np-hard. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 354–362, 2006.
5. Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, 11(03):423–443, 2000.
6. Martin Grohe and Stephan Kreutzer. Methods for algorithmic meta theorems. *Model Theoretic Methods in Finite Combinatorics*, 558:181–206, 2011.
7. Yuri Gurevich. Monadic second-order theories. *Model-theoretic logics*, pages 479–506, 1985.
8. Johann A Makowsky. Algorithmic uses of the feferman–vaught theorem. *Annals of Pure and Applied Logic*, 126(1-3):159–213, 2004.
9. Sang-il Oum. Approximating rank-width and clique-width quickly. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 49–58. Springer, 2005.