MSOL partitioning problems on graphs of bounded treewidth and clique-width

Michaël Rao

Abstract

We show that a class of vertex partitioning problems that can be expressed in monadic second order logic (MSOL) are polynomials on graphs of bounded clique-width. This class includes COLORING, H-FREE COLORING, DOMATIC NUMBER and PARTITION INTO PERFECT GRAPHS. Moreover we show that a class of vertex and edge partitioning problems are polynomials on graphs of bounded treewidth.

1 Introduction

Many problems can be solved in polynomial time on graphs of bounded treewidth. It is well known that any problem that can be expressed in monadic second order logic can be solved in linear time on this class of graphs [1, 8]. This is the case of STABLE SET, DOMINATING SET and HAMILTONIAN CIRCUIT. An another class of problems that can be solved in polynomial time is given in [26]. The following problems can be also solved in polynomial time on graphs of bounded treewidth: GRAPH ISOMORPHISM [3], decide if the treewidth of a graph is bounded by k' fixed [5], and decide if the clique-width of a graph is bounded by k [15]. An overview of different uses of the treewidth can be found in [4].

Recently a lot of research have been done on the class of graphs of bounded clique-width. Some of the results for the graphs of bounded treewidth have been generalized to this class of graphs. Every problem that can be expressed in monadic second order logic with quantification over the vertices can be solved in linear time [11], [14] gives a collection of problems and [17] gives a class of partitioning problems that can be solved in polynomial time.

We show that a new class of vertex partitioning problems that can be expressed in monadic second order logic is polynomial on graphs of bounded clique-width. This class includes COLORING, H-FREE COLORING, DOMATIC NUMBER and PARTITION INTO PERFECT GRAPHS. Moreover we show that a superclass of this class of problems is polynomial on graphs of bounded treewidth.

2 Preliminaries

2.1 Graphs

Let $G = (V, E)$ be a graph, where V is the set of vertices and E the set of edges. The graph G is *complete* if $E = \{\{u, v\} : u, v \in V, u \neq v\}$. The graph $H = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. Moreover H is a *induced subgraph* of G if $V' \subseteq V$ and $E' = \{\{u, v\} \in E : u, v \in V'\}$. For $V' \subseteq V$, we denote by $G[V'] = (V', \{\{u, v\} \in E : u, v \in V'\})$ the subgraph of G induced by V'. A subset $V' \subseteq V$ is a *clique* if $G[V']$ is complete.

Let $k \in \mathbb{N}$ fixed. The class of *k-trees* is defined recursively: a complete graph with k vertices is a k-tree, and if G is a k-tree then the graph constructed by adding a new vertex v adjacent to a clique of size k is a k-tree. A graph is a *partial k-tree* if it is a subgraph of a k-tree. The *treewidth* of a graph G is the smallest integer k such that G is a partial k-tree.

A *labeled graph* is graph in which every vertex has a label into $\mathbb{N}$. A labeled graph is a *k-labeled graph* if every label are into $\{1, 2, \dots, k\}$. Let define the following operations:

1. create a single vertex with label l (denoted by $\cdot_l$);
2. disjoint union of two labelled graphs (denoted by $\oplus$);
3. add an edge between all vertices labeled i and all vertices labeled j for $i \neq j$ (denoted by $\eta_{i,j}$);
4. relabeling all vertices of label i with label j (denoted by $\rho_{i \to j}$).

A k-expression of a labeled graph G describes a sequence of operations (1)-(4) generating G using labels into $\{1, 2, \dots, k\}$. The *clique-width* of a labeled graph is the smallest integer k such that there is a k-expression describing it. The clique-width of an unlabeled graph G is the minimum of the clique-width over all labeling of vertices of G.

2.2 Structures

A *vocabulary* is a finite set of *relation symbols*. Every relation symbol has an *arity* which is a strictly positive integer. A *relation* of arity a on the set A is a subset of A^a.

Let τ be a vocabulary with relation symbols $R_1, \dots, R_k$, of arity $a_1, \dots, a_k$, respectively. A *structure on the vocabulary* τ (or a τ*-structure*) $\mathcal{A}$ is composed by a set A, called the *domain*, and for every $i \in \{1, \dots k\}$, by a relation $R_i^{\mathcal{A}}$ of arity a_i on A. It is denoted by $\langle A, R_1^{\mathcal{A}}, \dots, R_k^{\mathcal{A}} \rangle$. Let $\mathcal{A}$ and $\mathcal{A}'$ two τ-structures. $\mathcal{A} \bigsqcup \mathcal{A}'$ denote the τ-structure $\langle A \cup A', R_1^{\mathcal{A}} \cup R_1^{\mathcal{A}'}, \dots, R_k^{\mathcal{A}} \cup R_k^{\mathcal{A}'} \rangle$. Without ambiguity on the structure, we will wrote sometimes R instead of $R^{\mathcal{A}}$. Here, every structure is supposed finite (that is, the domain is finite).

We can consider two possibilities to represent a graph by a structure. In the first, elements are the vertices of the graph, and we have one binary relation which is the adjacency relation of the graph. In the second, elements are the vertices and the edges, and the relation is the incidence relation of the graph.

Let τ_1 be the vocabulary $\{E\}$ where E is a relation symbol of arity two. Let $\tau_{1,p}$, where $p \in \mathbb{N}$, be the vocabulary $\{E, U_1, \dots U_p\}$ where E is a binary relation symbol and U_i an unary relation symbol.

For a graph G, $G(\tau_1)$ represents the structure $\langle V, E \rangle$ where V is the domain and E is the adjacency relation of G. For a p-labeled graph G, $G(\tau_{1,p})$ represents the structure $\langle V, E, U_1, \dots U_p \rangle$ where for $v \in V$, $U_i(v)$ if and only if v is labeled i in G.

Let τ_2 be the vocabulary $\{P_V, R\}$ where P_V is an unary relation and R a binary relation. For a graph G, let $G(\tau_2)$ be the structure $\langle V \cup E, P_V, R \rangle$, with domain $V \cup E$, where $R(a, b)$ is true if and only if b is an endpoint of a or a is an endpoint of b, and $P_v(a)$ is true if and only if $a \in V$.

2.3 Clique width of a structure

The notion of clique-width can be naturally be extended to every structures. For that we need an operator $\eta^R_{(i_1, \dots, i_a)}$ for every relation symbol R of arity a in the vocabulary. Moreover we have one operator $\rho_{i \to j}$ which relabel every element of the domain of label i into j, an operator $\oplus$ for the disjoint union and an operator $\cdot_i$ which represents a structure with one element labeled i.

Let τ be a vocabulary, and let τ^k be the vocabulary containing every relation symbol of τ, and k new unary relations symbols $P_1, \ldots, P_k$.

- $\cdot_i$ represents the τ^k-structure of domain $\{v\}$, with $P_i = \{v\}$ and all others relations are empty
- $\rho_{i\to j}(\mathcal{A})$ is the τ^k-structure of domain A, such that $P_i^{\rho_{i\to j}(\mathcal{A})} = \emptyset$, $P_j^{\rho_{i\to j}(\mathcal{A})} = P_i^{\mathcal{A}} \cup P_j^{\mathcal{A}}$, and for all other relation symbol R, $R^{\rho_{i\to j}(\mathcal{A})} = R^{\mathcal{A}}$.
- for $R \in \tau$, $\eta^R_{(i_1,\ldots,i_a)}(\mathcal{A})$ is the τ^k-structure of domain A, such that $R^{\eta^R_{(i_1,\ldots,i_a)}(\mathcal{A})} = R^{\mathcal{A}} \cup P^{\mathcal{A}}_{i_1} \times \ldots \times P^{\mathcal{A}}_{i_a}$, and for all other relation symbol Q, $Q^{\eta^R_{(i_1,\ldots,i_a)}(\mathcal{A})} = Q^{\mathcal{A}}$.
- $\mathcal{A} \oplus \mathcal{A}'$ is the disjoint union of $\mathcal{A}$ and $\mathcal{A}'$, that is the τ^k-structure of domain $A \cup A'$, and for every relation symbol R, $R^{\mathcal{A}\oplus\mathcal{A}'} = R^{\mathcal{A}} \cup R^{\mathcal{A}'}$.

The clique-with of a τ-structure $\mathcal{A}$ is the smallest k such that there is a τ^k-structure $\mathcal{A}'$ with an expression for it with labels into $\{1, \ldots k\}$, and such that $\mathcal{A}$ is the structure $\mathcal{A}'$ without unary relations $P_1, \ldots P_k$.

It is easy to see that the clique-width of the structure $G(\tau_1)$ is exactly the clique-width of G: since $E(v, v)$ is false for every v in the domain, we cannot have an operator $\eta^E_{(i,i)}$ in the expression for $G(\tau_1)$. Moreover, since E is symmetric, we can always add an operation $\eta^E_{(i,j)}$ just after an operation $\eta^E_{(j,i)}$. Thus a k-expression for G comes immediately from a k-expression for $G(\tau_1)$.

2.4 Monadic second order logic

In *first order logic*, we can quantifiate on the elements of the domain, and in *second order logic*, we can also quantifiate on the relations. The *monadic second order logic* is a restriction of the second order logic where we are allowed to quantifiate only on first order variables and on unary relations. Let $\mathrm{MSOL}(\tau)$ denotes the set of all formulas of monadic second order logic on the vocabulary τ.

A variable in a formula is a *free variable* if it is not in the scope of any quantifier. A formula is *closed* if it has no free variable. Let $\mathrm{MSOL}(\tau, \mathfrak{X}, \mathfrak{Y})$ denote the MSOL formulas with first order free variables into $\mathfrak{X}$ and second order free variables into $\mathfrak{Y}$.

Let φ be a MSOL formula. A structure $\mathcal{A}$ with an assignation z of its free variables is a *model* for φ if φ is true for $\mathcal{A}$ with the assignation z. We note $\langle \mathcal{A}, z\rangle \models \varphi$, or $\mathcal{A} \models \varphi$ if φ is closed. The *theory* of $\mathcal{A}$, denoted $\mathrm{Th}_{\mathrm{MSOL}}(\tau)$, is the set of all MSOL formulas φ such that $\mathcal{A} \models \varphi$.

The *quantifier rank* $\mathrm{qr}(\varphi)$ of a formula φ is the maximum number of nested quantifiers occurring in φ. It can be defined inductively:

- $\mathrm{qr}(\varphi) = 0$ if φ is a quantifier free formula,
- $\mathrm{qr}(\varphi \vee \varphi') = \mathrm{qr}(\varphi \wedge \varphi') = \max(\mathrm{qr}(\varphi), \mathrm{qr}(\varphi'))$,
- $\mathrm{qr}(\neg\varphi) = \mathrm{qr}(\varphi)$ and
- $\mathrm{qr}(\forall x, \varphi) = \mathrm{qr}(\exists x, \varphi) = 1 + \mathrm{qr}(\varphi)$.

Let $\mathrm{MSOL}^q(\tau)$ denotes the MSOL formulas with quantifier rank at most q. Let $\mathrm{Th}^q_{\mathrm{MSOL}}(\mathcal{A}) = \mathrm{MSOL}^q(\tau) \cap \mathrm{Th}_{\mathrm{MSOL}}(\mathcal{A})$ (where $\mathcal{A}$ is a τ-structure), and let $\mathrm{MSOL}^q(\tau, \mathfrak{X}, \mathfrak{Y}) = \mathrm{MSOL}^q(\tau) \cap \mathrm{MSOL}(\tau, \mathfrak{X}, \mathfrak{Y})$.

Two formulas φ and ψ are equivalent (denoted by $\varphi \equiv \psi$) if for every structure $\mathcal{A}$, $\mathcal{A} \models \varphi$ iff $\mathcal{A}' \models \psi$. There is infinite number of different formulas of quantifier rank at most q, but there is only finite number of non equivalent formulas. Deciding if two formulas are equivalent is undecidable, but we can define an equivalence relation $\approx$ such that $\approx$ is decidable and the number of equivalence classes of $\approx$ in $\mathrm{MSOL}^q(\tau)$ is finite. This equivalence is defined recursively using the equivalence of boolean formulas. A boolean formula is a formula from $\{0,1\}^k$ to $\{0,1\}$. If $\mathrm{qr}(\varphi) \neq \mathrm{qr}(\varphi')$ then $\varphi \not\approx \varphi'$. If $\mathrm{qr}(\varphi) = \mathrm{qr}(\varphi') = 0$, then $\varphi \approx \varphi'$ if and only if $\varphi \equiv \varphi'$. Else let $q = \mathrm{qr}(\varphi) = \mathrm{qr}(\varphi')$. Let $\varphi = QXB(\varphi_1, \dots, \varphi_k)$ and $\varphi' = Q'XB(\varphi'_1, \dots, \varphi'_{k'})$, where Q and Q' is either $\forall$ or $\exists$, B and B are boolean functions, and for all $i \neq j$, $\varphi_i \not\approx \varphi_j$ and $\varphi'_i \not\approx \varphi'_j$. Then $\varphi \approx \varphi'$ if $Q = Q'$, $k = k'$ and there is a bijection $\sigma : \{1, \dots, k\} \to \{1, \dots, k\}$ such that for all $a_1, \dots, a_k$, $B(a_1, \dots, a_k) = B'(a_{\sigma(1)}, \dots, a_{\sigma(k)})$. It is easy to see that $\varphi \approx \varphi'$ imply $\varphi \equiv \varphi'$, and that the number of equivalence classes of $\approx$ is bounded for fixed q. Since the equivalence of boolean formulas is decidable, $\approx$ is decidable. From now we suppose that every $\mathrm{MSOL}^q(\tau)$ formulas are taken modulo $\approx$, and that $\mathrm{MSOL}^q(\tau, \mathfrak{X}, \mathfrak{Y})$ is finite for a fixed q.

2.5 Translation scheme

Quantifier free translation schemes are special cases of translation scheme of the monadic second order logic [11, 13]. Let $\tau = \{R_1, \dots, R_k\}$ be a vocabulary and for $i \in \{1, \dots, k\}$, let a_i be the arity of R_i. A *quantifier free translation scheme* in τ is $\Phi = \langle \psi_1, \dots, \psi_k \rangle$ where for all $i \in \{1, \dots, k\}$, $\psi_i \in \mathrm{MSOL}^0(\tau)$ (that is, ψ_i is a quantifier free formula on τ), and ψ_i has a_i free first order variables, and no free second order variable.

Let $\mathcal{A}$ be a τ-structure. Then $\Phi^*(\mathcal{A})$ is the τ-structure of domain A and for all $i \in \{1, \dots, k\}$, $R_i^{\Phi^*(\mathcal{A})} = \{(x_1, \dots, x_{a_i}) : x_1, \dots, x_{a_i} \in A \text{ and } \mathcal{A} \models \psi_i(x_1, \dots, x_k)\}$, where a_i is the arity of R_i.

Let φ a formula on the vocabulary τ. $\Phi^\sharp(\varphi)$ is defined recursively:

- $\Phi^\sharp(\forall X, \varphi') \triangleq \forall X, \Phi^\sharp(\varphi)$ and $\Phi^\sharp(\exists X, \varphi') \triangleq \exists X, \Phi^\sharp(\varphi)$.
- $\Phi^\sharp(\varphi_1 \wedge \varphi_2) \triangleq \Phi^\sharp(\varphi_1) \wedge \Phi^\sharp(\varphi_2)$ and $\Phi^\sharp(\varphi_1 \vee \varphi_2) \triangleq \Phi^\sharp(\varphi_1) \vee \Phi^\sharp(\varphi_2)$.
- $\Phi^\sharp(\neg\varphi') \triangleq \neg\Phi^\sharp(\varphi')$.
- $\Phi^\sharp(R_i(x_1, \dots, x_{a_i})) \triangleq \psi_i(x_1, \dots, x_{a_i})$ for all $i \in \{1, \dots, k\}$.
- $\Phi^\sharp(U(x)) \triangleq U(x)$ if U is a second order variable.
- $\Phi^\sharp(x = y) \triangleq x = y$.

If Φ is a quantifier free translation scheme, then $\mathrm{qr}(\Phi^\sharp(\varphi)) = \mathrm{qr}(\varphi)$.

Theorem 1. *[13] Let τ be a vocabulary, $\mathcal{A}$ a τ-structure, Φ a quantifier free translation scheme and φ a formula on the vocabulary τ. Then*

$$\mathcal{A} \models \Phi^\sharp(\varphi) \iff \Phi^*(\mathcal{A}) \models \varphi.$$

2.6 The Feferman-Vaught theorem

The following theorem has been first showed for the first order logic [16], and then extended to the monadic second order logic [19, 25]. A constructive proof can be found in [22].

Theorem 2. *[16, 19, 25] Let τ be a vocabulary and let $\mathcal{A}$ and $\mathcal{A}'$ be two τ-structures. Let $q \in \mathbb{N}$ and $\varphi \in \mathrm{MSOL}^q(\tau)$. Then there is $l \in \mathbb{N}$, a boolean formula $B : \{0,1\}^{2l} \to \{0,1\}$, and for all $i \in \{1,\ldots,l\}$, $\varphi_i, \varphi'_i \in \mathrm{MSOL}^q(\tau)$ such that*

$$\mathcal{A} \bigsqcup \mathcal{A}' \models \varphi \quad \textit{if and only if} \quad B(a_1,\ldots,a_l,a'_1,\ldots,a'_l) = 1$$

where $a_i = 1$ if and only if $\mathcal{A} \models \varphi_i$ and $a'_i = 1$ if and only if $\mathcal{A}' \models \varphi'_i$, for all $i \in \{1,\ldots,l\}$.

In other words, Theorem 2 say that $\mathrm{Th}^q(\mathcal{A} \bigsqcup \mathcal{A}')$ is uniquely determined by $\mathrm{Th}^q(\mathcal{A})$ and $\mathrm{Th}^q(\mathcal{A}')$.

2.7 MSOL properties

Definition 3. A property on τ-structures Π is a MSOL *property* if there is a MSOL τ-formula φ such that Π is true on $\mathcal{A}$ if and only if $\mathcal{A} \models \varphi$.

We apply the previous definition on the τ_1 and τ_2 structures of a graph.

Definition 4. A property Π on graphs is MSOL_1 if there is a MSOL formula φ on the vocabulary τ_1 such that a graph G have the property Π if and only if $G(\tau_1) \models \varphi$. A property Π on graphs is MSOL_2 if there is a MSOL formula φ on the vocabulary τ_2 such that a graph G have the property Π if and only if $G(\tau_2) \models \varphi$.

For example, the property "G is connected" is MSOL_1, with:

$$\varphi \triangleq \forall X, (\forall x \in X, \forall y, E(x,y) \Rightarrow X(y)) \Rightarrow \forall x, x \in X.$$

The property "G is 3-colorable" is also MSOL_1:

$$\varphi \triangleq \exists X \exists Y \exists Z, \mathrm{Stable}(X) \wedge \mathrm{Stable}(Y) \wedge \mathrm{Stable}(Z) \wedge \mathrm{Partition}(X,Y,Z)$$

where

$$\begin{aligned}\mathrm{Partition}(X,Y,Z) \triangleq& \forall x, (x \in X \vee x \in Y \vee x \in Z) \wedge \neg(x \in X \wedge x \in Y) \wedge \\ & \neg(x \in X \wedge x \in Z) \wedge \neg(x \in Y \wedge x \in Z)\end{aligned}$$

and

$$\mathrm{Stable}(X) \triangleq \forall x \in X, \forall y \in X, x \neq y \Rightarrow \neg E(x,y).$$

The property "G have an Hamiltonian circuit" is MSOL_2. We check if there is a subset X of E such that (V, X) are 2-regular and connected.

$$\begin{aligned}\varphi \triangleq& \exists X \subseteq V, \\ & [[\forall x \in V, \exists e, f \in X, (e \neq f \wedge R(x,e) \wedge R(x,f) \wedge \\ & \qquad \forall g \in X, R(x,g) \Rightarrow (g = e \vee g = f))] \wedge \\ & [\forall V' \subseteq V, [\forall u \in V', \forall v \in V, (\exists e \in X, (R(e,u) \wedge R(e,v))) \Rightarrow y \in V') \\ & \qquad \Rightarrow V \subseteq V]]].\end{aligned}$$

It is easy to see that a MSOL_1 property is a MSOL_2 property. The opposite is not always true: the property "G have an Hamiltonian circuit" is not MSOL_1 [9].

LinEMSOL problems are an equivalent of MSOL properties for optimisations problems.

Definition 5. [11] A optimisation problem on a class of τ-structures is a LinEMSOL problem if it can be expressed in the following way:

Given a τ-structure $\mathcal{A}$, and m weight functions $f_1, \ldots, f_m$ over A, find an assignment z of free variables of φ such that

$$\sum_{\substack{1 \leq i \leq i \\ 1 \leq j \leq m}} a_{ij} |z(X_i)|_j = \mathrm{opt} \left\{ \sum_{\substack{1 \leq i \leq i \\ 1 \leq j \leq m}} a_{ij} |z'(X_i)|_j : \langle \mathcal{A}, z' \rangle \models \varphi(X_1, \ldots, X_l) \right\}$$

where φ is a MSOL formula with free set variables $X_1, \ldots X_l$, opt is either $\min$ or $\max$, a_{ij} is an integer for all $1 \leq i \leq l$ and $1 \leq j \leq m$, and $|z(X_i)|_j = \sum_{a \in z(X_i)} f_j(a)$.

A optimisation problem on graphs is $\mathrm{LinEMSOL}_1$ if it is a LinEMSOL problem on the τ_1 structure of the graph, and it is $\mathrm{LinEMSOL}_2$ if it is a LinEMSOL problem on the τ_2 structure of the graph. For example, STABLE SET, DOMINATING SET, FEEDBACK VERTEX SET are $\mathrm{LinEMSOL}_1$ problems. On the other hand CHROMATIC NUMBER is not $\mathrm{LinEMSOL}_1$ [20].

Theorem 6. *[1, 8] Let $k \in \mathbb{N}$ and Π be a fixed* $\mathrm{LinEMSOL}_2$ *problem. Then Π can be solved in linear time on graphs of treewidth bounded by k (if the decomposition is given with the graph).*

If a graph G has treewidth at most k but we don't have a decomposition, a k-decomposition of the graph can be found in linear time [5].

Theorem 7. *[11] Let $k \in \mathbb{N}$ and Π be a fixed* $\mathrm{LinEMSOL}_1$ *problem. Then Π can be solved in linear time on graphs of clique width bounded by k (if the k-expression is given).*

If a graph G has a clique-width at most k, then a $2^{k+1} - 1$-expression for it can be computed in polynomial time using the *rank-width* [12, 23, 24].

3 MSOL partitioning problems

3.1 Definition and examples

Definition 8. A problem Π is a MSOL *partitioning problem* if there is a MSOL τ-formula φ with free set variable X such that Π can be expressed in the following way:

Given a τ-structure $\mathcal{A}$ and an integer r, can A be partitioned into $\{A_1, A_2 \ldots, A_r\}$ such that

$$\forall i \in \{1, 2 \ldots, r\}, \langle \mathcal{A}, A_i \rangle \models \varphi(A_i) \ ?$$

We say that a problem on graphs is a MSOL_1 *partitioning problem* if it is a MSOL partitioning problem on the τ_1-structure of the graph, and a MSOL_2 *partitioning problem* if it is a MSOL partitioning problem on the τ_2-structure of the graph.

It is easy to see that COLORING and PARTITION INTO CLIQUE are MSOL_1 partitioning problems (with $\varphi \triangleq \forall x, y \in X, \neg E(x, y)$ and $\forall x, y \in X, E(x, y)$ respectively). The following problems are also MSOL-partitioning problems.

Co-coloring

INSTANCE: Graph $G = (V, E)$, and an integer $r \leq |V|$.

QUESTION: Is there a partition of V into r disjoint sets $V_1, \dots, V_r$, such that for each $i \in \{1, 2, \dots k\}$, V_i is either an independent set or a clique ?

Domatic number
INSTANCE: Graph $G = (V, E)$, and an integer $r \leq |V|$.
QUESTION: Is there a partition of V into r disjoint sets $V_1, \dots, V_r$, such that for each $i \in \{1, 2, \dots k\}$, V_i is a dominating set of G ?

For DOMATIC NUMBER, we can take $\varphi(X) \triangleq \forall x, \exists y \in X, E(x, y)$, and for CO-COLORING, $\varphi(X) \triangleq (\forall x, y \in X, E(x, y)) \vee (\forall x, y \in X, \neg E(x, y))$.

Let H be a graph. A graph $G = (V, E)$ is *H-free* if there is no $V' \subseteq V$ such that $G[V']$ is isomorphic to H. The following problem is also a MSOL_1 partition problem.

H-free coloring (for H fixed)
INSTANCE: Graph $G = (V, E)$, and an integer $r \leq |V|$.
QUESTION: Is there a partition of V into r disjoint sets $V_1, \dots, V_r$, such that for each $i \in \{1, 2, \dots k\}$, $G[V_i]$ is H-free ?

A graph G is a *Berge* graph if neither G nor $\overline{G}$ contains an induced cycle of size $2k+1$, for every $k \geq 2$. Thus the property "G is a Berge graph" is MSOL_1 (check if G and $\overline{G}$ have no subgraph H such that H is connected, 2-regular, not bipartite and has at most 5 vertices). Strong Perfect Graph Theorem show that Berge graphs are exactly perfect graphs [6], so the following problem is a MSOL_1 partitioning problem.

Partition into perfect graphs
INSTANCE: Graph $G = (V, E)$, and an integer $r \leq |V|$.
QUESTION: Is there a partition of V into r disjoint sets $V_1, \dots, V_r$, such that for each $i \in \{1, 2, \dots k\}$, $G[V_i]$ is perfect ?

Every MSOL_1 partitioning problem is a MSOL_2 partitioning problem. The problem PARTITION INTO HAMILTONIAN SUBGRAPHS is an MSOL_2 partitioning problem.

3.2 MSOL partitioning problems on graphs of bounded clique width

If we fix $k \in \mathbb{N}$, deciding if V is partitionable into at most k subsets having the MSOL_1 property Π is a MSOL_1 problem. But to solve in polynomial time the corresponding MSOL partitioning problem on a graph of bounded clique-width, it is not sufficient to use the algorithm given in [11] for k form 1 to n, since the constant in the execution time of the algorithm depends of the formula. For example ACHROMATIC NUMBER for k fixed is MSOL_1, but ACHROMATIC NUMBER is not a MSOL_1 partitioning problem, and is NP-complete on cographs [2] (which are graphs of clique-width at most 2).

We show that every MSOL_1 partitioning problem can be solved in polynomial time on graphs of bounded clique-width. We prove it in the more general case of structures of bounded clique-width.

Theorem 9. *Let k be an integer, τ be a vocabulary, and $\varphi \in \mathrm{MSOL}(\tau, \emptyset, \{X\})$, fixed. There is a polynomial algorithm for the following problem:*
INPUT*: a τ^k-structure $\mathcal{A}$ of clique-width at most k, and a k-expression for $\mathcal{A}$.*
OUTPUT*: $C \subseteq \{1, \dots, |V|\}$ such that $r \in C$ if and only if there is a partition $\{A_1, \dots, A_r\}$ of A such that:*

$$\forall i \in \{1, \dots, r\}, \langle \mathcal{A}, A_i \rangle \models \varphi(A_i).$$

Proof. Let q be the quantifier rank of φ. A *multiset* is a set in which an element may occurs more than once. It is denoted by $[x_1, x_2, \dots, x_n]$ or $[x_i]_{i\in\{1,\dots,n\}}$. The idea is to compute recursively on the k-expression the set of all multisets of theories of the structures, for all partitions of the domain A. More formally, we compute recursively on the k-expression the set:

$$\mathcal{S}(\mathcal{A}) = \left\{ [\mathrm{Th}^q(\langle \mathcal{A}, A_i\rangle)]_{i\in\{1,\dots,r\}} : \{A_1, \dots, A_r\} \text{ is a partition of } A \right\}.$$

The set $\mathcal{S}$ for the structure $\cdot_i$ is the following:

$$\mathcal{S}(\cdot_i) = \{[\mathrm{Th}^q(\langle \mathcal{A}, \{v\}\rangle), \mathrm{Th}^q(\langle \mathcal{A}, \emptyset\rangle), \dots, \mathrm{Th}^q(\langle \mathcal{A}, \emptyset\rangle)]\}$$

since there is only one possible partition of the domain of $\cdot_i$. Theories can be computed in constant time since the structure is of constant size and there is a fixed number of formulas to check.

For all $R \in \tau$ and $i_1, \dots, i_a \in \{1, \dots, k\}$, $\Phi_{\eta^R_{(i_1,\dots,i_a)}} = \langle \psi_{R'} : R' \in \tau^k\rangle$ is a quantifier free translation scheme for $\eta^R_{(i_1,\dots,i_a)}$, where $\psi_{R'} = R'(x_1, \dots x_{a'})$ for all $R' \in \tau^k \setminus \{R\}$ and $\psi_R = R(x_1, \dots x_a) \vee (P_{i_1}(x_1) \wedge P_{i_2}(x_2) \dots \wedge P_{i_a}(x_a))$ (a and a' are the arity of R and R', respectively).

Moreover, for all $i, j \in \{1, \dots, k\}$, $\Phi_{\rho_{i,j}} = \langle \psi_{R'} : R \in \tau^k\rangle$ is a quantifier free translation scheme for $\rho_{i\to j}$, where $\psi_{R'} = R'(x_1, \dots x_{a'})$ for all $R' \in \tau^k \setminus \{P_i, P_j\}$, $\psi_{P_i} = \text{False}$ and $\psi_{P_j} = P_i(x_1) \vee P_j(x_1)$.

Let f be an operator $\rho_{i\to j}$ or $\eta^R_{(i_1,\dots,i_a)}$, and Φ its corresponding quantifier free translation scheme. Then:

$$\mathcal{S}(f(\mathcal{A})) = \left\{ \left[\Phi^\sharp(T_i)\right]_{i\in\{1,\dots,r\}} : [T_i]_{i\in\{1,\dots,r\}} \in \mathcal{S}(\mathcal{A}) \right\}$$

where $\Phi^\sharp(T) = \{\Phi^\sharp(\varphi) : \varphi \in T\}$.

Since the multisets are members of an universe of at most $(r+1)^{|\mathrm{MSOL}^q(\tau,\emptyset,\{X\})|}$ elements, $\mathcal{S}(f(\mathcal{A}))$ can be computed in time $(r+1)^{|\mathrm{MSOL}^q(\tau,\emptyset,\{X\})|+1}$.

For the operator $\oplus$ we use the Feferman-Vaught theorem. Let $\mathcal{A}$ and $\mathcal{A}'$ two τ^k-structures, and let $A_1 \subseteq A$ and $A'_1 \subseteq A'$. The theorem says that we can compute $\mathrm{Th}^q(\langle \mathcal{A} \bigsqcup \mathcal{A}', A_1 \cup A'_1\rangle)$ from $\mathrm{Th}^q(\langle \mathcal{A}, A_1\rangle)$ and $\mathrm{Th}^q(\langle \mathcal{A}', A'_1\rangle)$. Moreover, as the size of theses sets is bounded by a fixed constant (dependent of τ^k and q), this computation can be done in constant time.

Let $\mathcal{M}_a \in \mathcal{S}(\mathcal{A})$ and $\mathcal{M}_b \in \mathcal{S}(\mathcal{A}')$. Each set T_a in $\mathcal{M}_a$ corresponds to $\mathrm{Th}^q(\langle \mathcal{A}, A_1\rangle)$ for a $A_1 \subseteq A$ and each set T_b in $\mathcal{M}_b$ corresponds to $\mathrm{Th}^q(\langle \mathcal{A}', A'_1\rangle)$ for a $A'_1 \subseteq A'$. Thus we can compute $\mathrm{Th}^q(\langle \mathcal{A} \bigsqcup \mathcal{A}', A_1 \cup A'_1\rangle)$, which will be denoted by $T_a \boxplus T_b$.

We start with the set $\mathcal{D} = \{[]\} \times \mathcal{S}(\mathcal{A}) \times \mathcal{S}(\mathcal{A}')$, and we extend $\mathcal{D}$ by all triplets which can be obtained from a triplet $(\mathcal{M}, \mathcal{M}_a, \mathcal{M}_b) \in \mathcal{D}$ by removing a set T_a in $\mathcal{M}_a$ and a set T_b in $\mathcal{M}_b$, and adding $T_a \boxplus T_b$ to $\mathcal{M}$. When $\mathcal{D}$ cannot be extended anymore, we have $\mathcal{S}(\mathcal{A} \oplus \mathcal{A}') = \{\mathcal{M} : (\mathcal{M}, [], []) \in \mathcal{D}\}$.

Operations $\boxplus$ take constant time. There is at most n^2 ways to combine one element from $\mathcal{M}_1$ with one element from $\mathcal{M}_2$. Since $\mathcal{M}$, $\mathcal{M}_1$ and $\mathcal{M}_2$ are in an universe with at most $(r+1)^{|\mathrm{MSOL}^q(\tau,\emptyset,\{X\})|}$ elements, $\mathcal{D}$ has at most $(r+1)^{3\times|\mathrm{MSOL}^q(\tau,\emptyset,\{X\})|}$ elements, and thus can be computed in time $(r+1)^{3\times|\mathrm{MSOL}^q(\tau,\emptyset,\{X\})|+2}$. □

Remark 1. It is not necessary to compute $\mathrm{Th}^q(\mathcal{A})$: it is sufficient to found a subset $\Psi \subseteq \mathrm{MSOL}^q(\tau)$ such that:

- for every translation scheme Φ (corresponding to an operation ρ or η), $\Phi^\sharp(\psi) \in \Psi$ and

- for every formula ψ_i appearing in the Feferman-Vaught for the formula ψ, $\psi_i \in \Psi$.

This set of formula can be constructed starting with the set $\{\varphi\}$ (where φ is the MSOL property to check), and enlarging the set by adding $\Phi^\sharp(\psi)$ and ψ_i for all $\psi \in \Psi$. When the set cannot be extended anymore then it has the desired property.

Remark 2. A partition can be found in the same time using common backtracking techniques.

The theorem applied to the structure τ_1 of the graph gives immediately:

Corollary 10. *Every fixed* MSOL_1 *partitioning problem can be solved in polynomial time on graphs of bounded clique-width (if the* k*-expression is given with the graph).*

If the k-expression is not given, we can compute in polynomial time a $2^{k+1}-1$-expression using results in [12, 23, 24].

3.3 Application to graphs of bounded treewidth

The *incidence graph* of $G=(V,E)$ is the graph $I(G)$ of vertex set $V \cup E$, and $v \in V$ is adjacent to $e \in E$ in $I(G)$ if v is an endpoint of e in G. The clique width of $G(\tau_2)$ is the clique width of $I(G)$, since the structure $G(\tau_2)$ without the unary relation P_V is isomorphic to the structure $I(G)(\tau_1)$. If the treewidth of G is bounded by k, the treewidth of $I(G)$ is at most k [21], and thus the clique-width of $I(G)$ and $G(\tau_2)$ is at most $3 \times 2^{k-1}$ [7]. We can apply previous theorem on the τ_2-structure of the graph.

Corollary 11. *Every fixed* MSOL_2 *partitioning problem can be solved in polynomial time on graphs of bounded treewidth (if the tree decomposition is given with the graph).*

If the decomposition is not given, it can computed in linear time [5].

References

[1] S. ARNBORG, J. LAGERGREN, D. SEESE, Easy problems for tree-decomposable graphs, *Journal of Algorithms* 12 (1991) 308-340

[2] H. L. BODLAENDER, Achromatic number is NP-complete for cographs and interval graphs, *Information Processing Letters* 31 (1989) 135–138.

[3] H. L. BODLAENDER, Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees, *Journal of Algorithms* 11 (1990) 631–643

[4] H. L. BODLAENDER, A tourist guide through treewidth, *Acta Cybernetica* 11 (1993) 1–23

[5] H. L. BODLAENDER, A Linear Time Algorithm for Finding Tree-decompositions of Small Treewidth, *SIAM Journal on Computing* 25 (1996) 1305–1317

[6] M. CHUDNOVSKY, N. ROBERTSON, P. SEYMOUR, R. THOMAS, The strong perfect graph theorem, *Annals of Mathematics* 164 (2006) 51–229

[7] D. G. CORNEIL, U. ROTICS, On the Relationship Between Clique-Width and Treewidth, *SIAM Journal on Computing* 34 (2005) 825–847

[8] B. COURCELLE, The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues, *Informatique Théorique et Applications* 26 (1992) 257–286

[9] B. COURCELLE, The monadic second-order logic of graphs VI: On several representations of graphs by relational structures, *Discrete Applied Mathematics* 54 (1994) 117–149

[10] B. COURCELLE, J. ENGELFRIET, G. ROZENBERG, Handle-rewriting hypergraph languages, *Journal of Computer and System Sciences* 46 (1993) 218–270

[11] B. COURCELLE, J. A. MAKOWSKY, U. ROTICS, Linear time solvable optimization problems on graphs of bounded clique-width, *Theory of Computing Systems* 33 (2000) 125–150

[12] B. COURCELLE, S. OUM, Vertex-Minors, Monadic Second-Order Logic, and a Conjecture by Seese, *Journal of Combinatorial Theory Series B*, to appear (2006)

[13] H.-D. EBBINGHAUS, J. FLUM, Finite model theory, Springer–Verlag, Berlin (1995)

[14] W. ESPELAGE, F. GURSKI, E. WANKE, How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time, *Proceedings of WG 2001*, LNCS 2204 (2001) 117–128

[15] W. ESPELAGE, F. GURSKI, E. WANKE, Deciding clique-width for graphs of bounded tree-width, *Journal of Graph Algorithms and Applications* 7 (2003) 141–180

[16] S. FEFERMAN, R. VAUGHT, The first order properties of algebraic systems, *Fundamenta Mathematicae* 47 (1959) 57–103

[17] M. U. GERBER, D. KOBLER, Algorithms for vertex-partitioning problems on graphs with fixed clique-width, *Theoretical Computer Science* 299 (2003) 719–734

[18] D. KOBLER, U. ROTICS, Edge dominating set and colorings on graphs with fixed clique-width, *Discrete Applied Mathematics* 126 (2003) 197–221

[19] H. LÄUCHLI, A decision procedure for the weak second order theory of linear order, *Logic Colloquium '66*, North Holland, Amsterdam, (1968) 189–197

[20] C. LAUTEMANN, Logical definability of NP-optimization problems with monadic auxiliary predicates, LNCS 702 (1993) 327–339

[21] V. V. LOZIN, D. RAUTENBACH, The Tree- and Clique-width of Bipartite Graphs in Special Classes, *Australasian Journal of Combinatorics* 34 (2006) 57–68

[22] J. MAKOWSKY, Algorithmic uses of the Feferman-Vaught theorem, *Annals of Pure and Applied Logic* 126 (2004) 159–213

[23] S. OUM, Approximating Rank-width and Clique-width Quickly, *Proceedings of WG 2005*, LNCS 3787 (2005) 49–58

[24] S. OUM, P. SEYMOUR, Approximating Clique-width and Branch-width, *Journal of Combinatorial Theory Series B*, to appear (2006)

[25] S. SHELAH, The monadic theory of order, *Annals of Mathematics* 102 (1975) 379–419

[26] J. A. TELLE, A. PROSKUROWSKI, Algorithms for Vertex Partitioning Problems on Partial k-Trees, *SIAM Journal on Discrete Mathematics* 10 (1997) 529–550