



Monadic second-order logic for graphs.
Algorithmic and language theoretical applications

Part 2

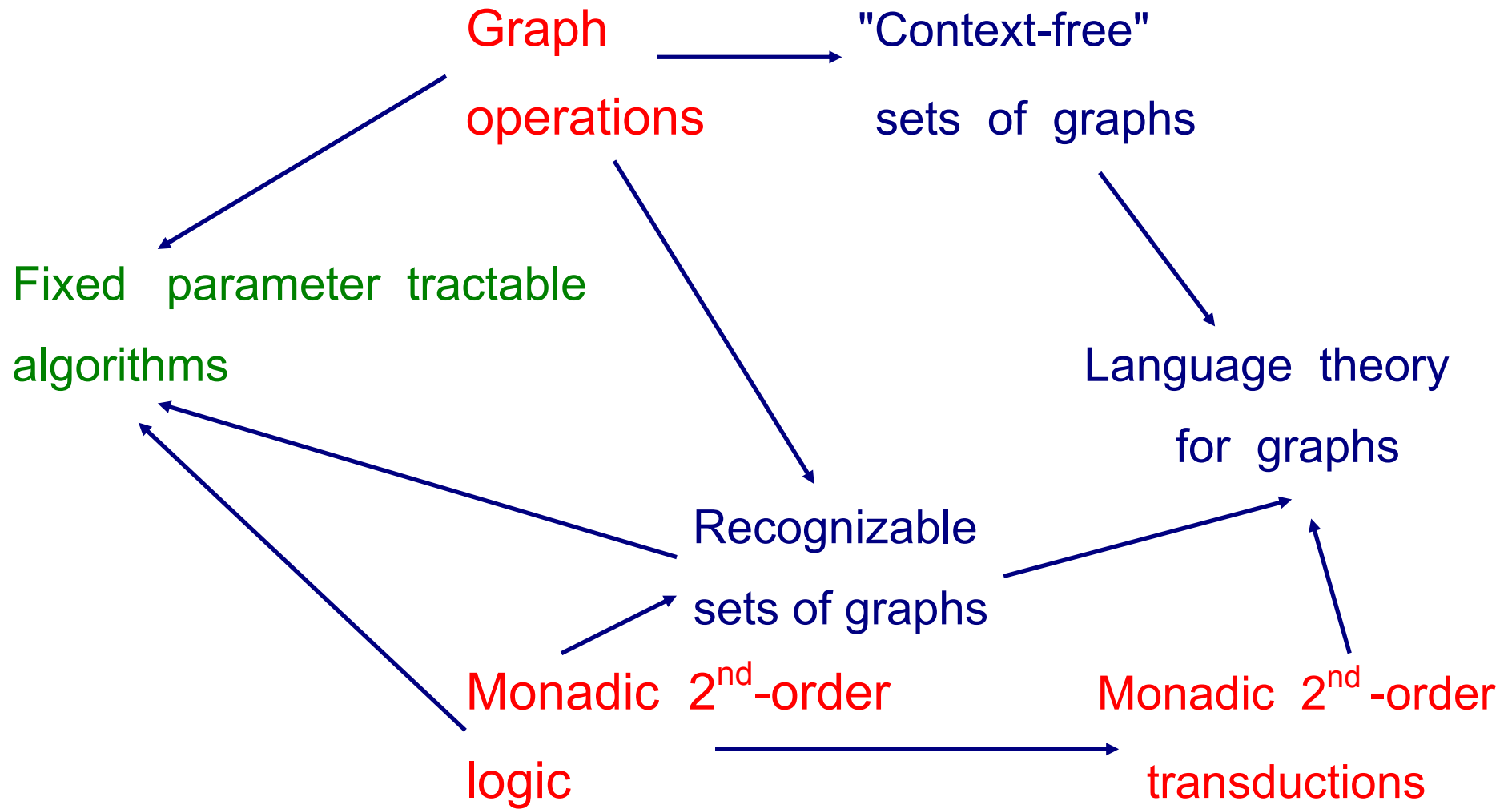
Bruno Courcelle

Université Bordeaux 1, LaBRI, and Institut Universitaire de France



Reference : Graph structure and monadic second-order logic,
book to be published by Cambridge University Press, readable on :
<http://www.labri.fr/perso/courcell/ActSci.html>

An overview chart



Summary

1. Context-free sets defined by equation systems (blue = Language Theor. issues)
2. Two graph algebras. Tree-width and clique-width. (red = new notions)
3. Recognizability : an algebraic notion.

Part 2

4. Monadic second-order sentences define recognizable sets.
5. Fixed-parameter tractable algorithms (green = algorithms)
6. Monadic second-order transductions.
7. Robustness results : preservation of classes under direct and inverse monadic second-order transductions. Short proofs in graph theory. (black= graph theory)
8. *Logic and graph structure theory* : Graph classes on which monadic second-order logic is decidable
9. *Open questions*

4. Monadic Second-Order (MS) Logic

A logical language which specifies inductive properties and functions

= First-order logic on power-set structures

= First-order logic extended with (quantified) variables
denoting subsets of the domains.

MS (expressible) properties : transitive closure, properties of paths, connectivity,
planarity (via Kuratowski, uses connectivity), k-colorability.

Examples of formulas for $G = (V_G, \text{edg}_G(.,.))$, undirected

3-colorability :

$\exists X, Y$ ("X, Y are disjoint" $\wedge \forall u, v \{ \text{edg}(u, v) \Rightarrow$
 $[(u \in X \Rightarrow v \notin X) \wedge (u \in Y \Rightarrow v \notin Y) \wedge (u \in V - (X \cup Y) \Rightarrow v \notin V - (X \cup Y))] \}$)

Non connectivity :

$$\exists X (\exists x \in X \wedge \exists y \notin X \wedge \forall u,v (u \in X \wedge \text{edg}(u,v) \Rightarrow v \in X))$$

Transitive and reflexive closure : $\text{TC}(R, x,y) :$

$$\forall X \{ \text{“X is R-closed”} \wedge x \in X \Rightarrow y \in X \}$$

where “X is R-closed” is defined by : $\forall u,v (u \in X \wedge R(u,v) \Rightarrow v \in X)$

R can be defined by a formula φ_R as in :

$$\forall x,y (x \in Y \wedge y \in Y \Rightarrow \text{TC}(\text{“}u \in Y \wedge v \in Y \wedge \text{edg}(u,v)\text{”},x,y)$$

expressing that $G[Y]$ is connected (Y is free in φ_R).

Application : G contains (fixed) H as a minor where $V_H = \{1, \dots, k\} :$

there exist pairwise disjoint vertex sets X_1, \dots, X_k

in G such that each $G[X_i]$ is connected and whenever

if $i \text{--} j$ in H, there is an edge between X_i and X_j in G.

Consequence : planarity is MS-expressible (no minor K_5 or $K_{3,3}$).

Non expressible properties

- 1) G is isomorphic to $K_{p,p}$ for some p (needs equipotence of two sets, hence quantification over binary relations to find if there is a **bijection**).
- 2) G has a **nontrivial automorphism**, or has all vertices of same degree.
- 3) $\text{Card}(X)$ is a multiple of p .
(but this is possible if the graph is linearly ordered or some linear order is definable by an MS formula).

Definition : One adds these cardinality set predicates to MS logic giving **Counting monadic second-order logic** or **CMS** : all good properties of MS logic hold for CMS logic.

(Adding an *equicardinality* predicate to MS would spoil everything.)

Edge set quantifications *increase* the expressive power

Incidence graph of G undirected, $\mathbf{Inc}(G) = (V_G \cup E_G, \mathbf{inc}_G(\dots))$

$\mathbf{inc}_G(v,e) \Leftrightarrow v$ is a vertex of edge e .

Monadic second-order formulas written with \mathbf{inc} can use **quantifications on sets of edges** : they define \mathbf{MS}_2 -expressible graph properties.

The existence of a *perfect matching* or a *Hamiltonian circuit* is \mathbf{MS}_2 -expressible but **not** MS-expressible.

Definition : A set L of finite graphs is MS-definable if $L = \{ G \text{ finite} / G \models \varphi \}$,

it is \mathbf{MS}_2 -definable if $L = \{ G \text{ finite} / \mathbf{Inc}(G) \models \varphi \}$

for a fixed MS *sentence* (formula without free variables) φ .

Recognizability theorem : (1) A language (set of words or *finite terms*) is recognizable (by congruence or automaton) \Leftrightarrow it is MS definable

(2) A set of finite graphs is VR-recognizable \Leftarrow it is CMS-definable

(3) A set of finite graphs is HR-recognizable \Leftarrow it is CMS₂-definable

Proofs: (1) Büchi, Elgot, Doner, Thatcher, Wright, Trakhtenbrot, others ???

(2) and (3) can be proved *together* in 2 ways :

By using the **Feferman-Vaught paradigm**,

or by constructing an automaton on terms by induction on the structure of the given formula. Better for concrete implementation (**MONA**).

(See **Appendix** and my communication to the Conference on Algebraic Informatics, Greece, 2009 for a comparison of the different proofs).

The Feferman-Vaught paradigm

Main idea : the validity of an MS formula in the disjoint union of two relational structures can be deduced from those of **finitely many auxiliary** formulas of no larger quantification height in each of the two structures.

This is **inductivity / recognizability**.

(A particular simple case of the **Composition Method** for infinite combinations of structures)

The easiest case is that of **VR-recognizability**

Notation: The set of satisfying assignments in G for a formula φ with free variables among X_1, \dots, X_n is :

$$\text{Sat}(G, \varphi, X_1, \dots, X_n) = \{ (V_1, \dots, V_n) \mid G \models \varphi(V_1, \dots, V_n) \}$$

Lemma 1 : If f is a *quantifier-free* mapping on graphs (edge-addition, vertex relabeling, edge complement) every φ has a **Backwards Translation** $f^\#(\varphi)$ relative to f such that for all G :

$$\text{Sat}(f(G), \varphi, X_1, \dots, X_n) = \text{Sat}(G, f^\#(\varphi), X_1, \dots, X_n)$$

where $f^\#(\varphi)$ has no larger quantifier-height than φ

Proof : Routine.

Splitting Theorem : One can construct formulas $\psi_i, \theta_i, i = 1, \dots, p$ of no larger quantifier-height than φ such that for all disjoint G and H :

$\text{Sat}(G \oplus H, \varphi, X_1, \dots, X_n)$ is the *disjoint* union of the sets

$$\text{Sat}(G, \psi_i, X_1, \dots, X_n) \diamond \text{Sat}(H, \theta_i, X_1, \dots, X_n), \quad i = 1, \dots, p,$$

where \diamond combines “partial answers” as follows :

$$\underline{A} \diamond \underline{B} = \{ (A_1 \cup B_1, \dots, A_n \cup B_n) \mid (A_1, \dots, A_n) \in \underline{A}, (B_1, \dots, B_n) \in \underline{B} \}$$

Proof: Induction on the structure of φ and set manipulations.

Lemma 2 : For each n and h there are *finitely many* formulas $\varphi(X_1, \dots, X_n)$ of quantifier-height $\leq h$, up to a *decidable and sound* equivalence.

Proof: Routine. But yields large numbers !

Proof of the Recognizability Theorem :

For each h , the equivalence relation such that :

$$G \approx H \Leftrightarrow \text{Sat}(G, \psi) = \text{Sat}(H, \psi) \quad (= \emptyset \text{ or } (), \text{ the empty sequence})$$

for every sentence ψ of quantifier-height $\leq h$

is a *type-preserving, locally finite* congruence on **VR** that saturates the set defined by φ , for any φ of quantifier-height $\leq h$.

(The same proof works for **HR** ; one more lemma is necessary to handle the fusion of sources in parallel-composition).

Extensions of the proof :

1) For *counting* valid assignments, i.e., for computing, for given G and fixed formula φ the cardinality of $\text{Sat}(G, \varphi, X_1, \dots, X_n)$, the *Splitting Theorem* gives (because of *disjoint unions*) the recursion :

$$\begin{aligned} |\text{Sat}(G \oplus H, \varphi, X_1, \dots, X_n)| &= \\ \sum_{i=1, \dots, p} |\text{Sat}(G, \psi_i, X_1, \dots, X_n)| \cdot |\text{Sat}(H, \theta_i, X_1, \dots, X_n)| \end{aligned}$$

2) Similar fact for *optimizing functions*, defined by :

$$\text{MaxSat}(G, \varphi, X) = \text{Max} \{ |A| \mid G \models \varphi(A) \}$$

Language Theoretical consequences

One can **filter out** from **HR- or VR-**equational sets the graphs which do not satisfy given **MS₂- or MS-**properties : one obtains **HR- or VR-**equational sets.

Generalizes : the intersection of a context-free language and a regular one is context-free ; other logical based extensions to come (MS-transductions).

Consequences for the decidability of logical theories

The **MS₂-theory** of the set of graphs of tree-width $\leq k$ is **decidable**.

(is some given formula true in all graphs ?)

The **MS-theory** of the set of graphs of clique-width $\leq k$ is **decidable**.

5. Algorithmic consequences of the Recognizability Theorem

Fixed-parameter linear algorithms for checking **MS₂** properties on graphs of tree-width $\leq k$; finding a tree-decomposition (an **HR**-term) is possible in **linear time** (Bodlaender)

Fixed-parameter cubic algorithm for checking **MS** properties on graphs of clique-width $\leq k$; finding a **VR**-term defining the graph is possible in **cubic time** (Hlineny, Oum and Seymour).

Other algorithmic consequences

Construction of **Labelling Schemes** for MS or FO properties on graphs of bounded clique-width (even, in some cases **unbounded** clique-width : planar graphs)

Tool for building FPT Model Checking algorithms for First-Order (FO) properties on graph classes of **unbounded** clique-width (Grohe, Frick, Dawar, Nešetřil, Ossona de Mendez).

Structure of these FPT algorithms

Three algorithms :

- 1) *Compiling algorithm* : *Input* : sentence φ and integer k
Output : a finite deterministic automaton $A(\varphi, k)$ recognizing *all* “width- k ” terms over **HR** or **VR** that define graphs satisfying φ .
- 2) *Parsing algorithm* : *Input* : graph **G** and integer k
Output : *some* “width- k ” term **t** over **HR** or **VR** that defines **G** (*failure* if k too small).
- 3) Checking algorithm : Run $A(\varphi, k)$ on **t** ; gives yes or no answer (can also produce “witness” assignments if φ is existential and valid in **G**).

Difficulties

1) The automaton $A(\varphi, k)$ may be too large for being **practically compiled**.

Problems with size of memory for intermediate automata, even if the *unique minimal deterministic* automaton $A(\varphi, k)$ has manageable number of states.

Hopes come from precomputed automata for subformulas.

Soguet *et al.* using MONA have constructed automata for the following cases :

	<i>Clique-width 2</i>	<i>Clique-width 3</i>
MaxDegree ≤ 3	91 states	Space-Out
Degree $\leq 4(x)$	48 states	233 states
Path(x,y)in(X)	26 states	Space-Out
Connectedness	11 states	Space-Out
IsConnComp(X)	48 states	Space-Out
Has ≤ 4 -VertCov	111 states	1037 states
HasClique ≥ 4	21 states	153 states
2-colorable	11 states	57 states

- 2) The best **parsing algorithms** by Bodlaender (for tree-width) and Hlineny-Oum (for clique-width via rank-width) are not implementable.

Bodlaender reports usable algorithms for 50 vertices and tree-width ≤ 35 (not for random graphs).

- 3) The checking algorithm works in time $O(n \cdot \log(q))$,

n = size of term = $O(\text{numbers of vertices})$, q = number of states of $A(\varphi, k)$.

Since q may be much much larger than n , it should not be hidden behind the O -notation.

Exercises

- 1) Write an MS-sentence expressing that the considered graph is a tree.
- 2) Write an MS-formula with free variables x,y,z expressing that, in a tree (undirected and unrooted), if one takes x as root, then y is an ancestor of z .
- 3) Write an MS_2 -sentence expressing that a graph has a Hamiltonian cycle.
- 4) A nonempty word over alphabet $\{a,b\}$ can (also) be considered as a directed path given with unary relations lab_a and lab_b representing the sets of occurrences of letters a and b . Prove that every regular language over $\{a,b\}$ is MS-definable.
- 5) A complete bipartite graph $K_{n,m}$ has a Hamiltonian cycle iff $n=m$. Construct such a graph “over” any word in $\{a,b\}^+$ having at least one a and at least one b . Deduce from 4) that Hamiltonicity is not MS-expressible.
- 6) Write an MS_2 -sentence expressing that a graph has a spanning tree of degree ≤ 3 . Show as in 5) that this property is not MS-expressible.

6. Monadic second-order transductions

Transformations of graphs or, more generally of relational structures specified by **MS** (or **MS₂**) formulas.

Two types : **MS**-transductions, for graphs G handled as (V_G, edg_G)

MS_{2,2}-transductions, for graphs G represented by

their **incidence graphs** $= (V_G \cup E_G, \text{in}_G)$

Results :

MS-transductions preserve bounded **clique-width** and **VR**-equational sets

MS_{2,2}-transductions preserve bounded **tree-width** and **HR**-equational sets

They preserve the decidability of **MS**- (or **MS₂**-) theories.

$\text{STR}(\Sigma)$: finite Σ -relational structures (or directed ranked Σ -hypergraphs).

$S = \langle D_S, (R_S)_{R \in \Sigma} \rangle$, R_S relation on D_S of arity $\rho(R)$

MS transductions are multivalued mappings $\tau : \text{STR}(\Sigma) \rightarrow \text{STR}(\Gamma)$

$$S \longmapsto T = \tau(S)$$

Basic case : T is defined inside S by MS formulas, in terms of **parameters**: subsets X_1, \dots, X_p of the domain of S

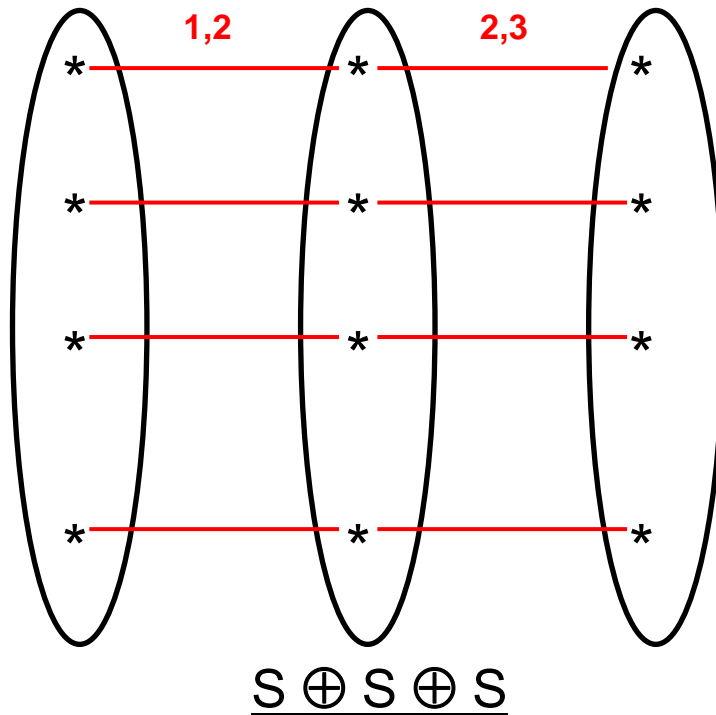
Examples : $(G, \{u\}) \longmapsto$ the connected component containing u .

$(G, X, Y, Z) \longmapsto$ the minor of G with vertex set X , resulting from contraction of the edges in Y and the deletion of the edges and vertices in Z .

Remark : For each tuple of parameters X_1, \dots, X_p satisfying an MS property, T is **uniquely defined**. τ is multivalued by the different choices of parameters.

General case : T is defined in this way inside

$S \oplus S \oplus \dots \oplus S$: disjoint copies of S with "marked"
equalities of copied elements



Proposition : The composition of two MS transductions is an MS transduction.

Example of an MS transduction (without parameters) : The *square* mapping δ on words : $u \mapsto uu$

For $u = aac$, we have S

$$\begin{array}{c} \cdot \rightarrow \cdot \rightarrow \cdot \\ a \quad a \quad c \end{array}$$

$\underline{S \oplus S}$

$$\begin{array}{ccc} \cdot \rightarrow \cdot \rightarrow \cdot & \cdot \rightarrow \cdot \rightarrow \cdot & \text{(marking edges omitted)} \\ a \quad a \quad c & a \quad a \quad c & \\ p_1 \quad p_1 \quad p_1 & p_2 \quad p_2 \quad p_2 & \end{array}$$

$\delta(S)$

$$\begin{array}{c} \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \\ a \quad a \quad c \quad a \quad a \quad c \end{array}$$

In $\delta(S)$ we redefine *Suc* (i.e., \rightarrow) as follows :

$$\begin{aligned} \text{Suc}(x,y) : & \Leftrightarrow (p_1(x) \wedge p_1(y) \wedge \text{Suc}(x,y)) \vee (p_2(x) \wedge p_2(y) \wedge \text{Suc}(x,y)) \\ & \vee (p_1(x) \wedge p_2(y) \wedge \text{"x has no successor"} \wedge \text{"y has no predecessor"}) \end{aligned}$$

We also remove the "marker" predicates p_1, p_2 .

The fundamental property of MS transductions :

$$S \mid \longrightarrow \tau(S)$$

$$\tau\#(\psi) \longleftarrow \mid \psi$$

Every MS formula ψ has an effectively computable

backwards translation $\tau\#(\psi)$, an MS formula, such that :

$$S \models \tau\#(\psi) \text{ if and only if } \tau(S) \models \psi$$

The verification of ψ in the **object structure** $\tau(S)$ **reduces** to the **verification of $\tau\#(\psi)$** in the given structure S
(because S contain all the necessary information to describe $\tau(S)$;
the MS properties of $\tau(S)$ are expressible by MS formulas in S).

Consequence : If $L \subseteq \text{STR}(\Sigma)$ has a **decidable MS theory**, so has its image under an MS transduction.

Comparison between MS - transductions and $MS_{2,2}$ – transductions :
they are **incomparable**

(inputs and outputs of $MS_{2,2}$ – transductions are incidence graphs).

For expressing graph properties, MS logic over incidence graphs (MS_2 logic) is more powerful than “ordinary” MS logic

For building graphs with $MS_{2,2}$ - transductions, we have more possibilities on the input graph, but we want more : to **specify each edge** from some vertex or edge of the input graph.

Transitive closure is an MS-transduction that is **not** $MS_{2,2}$

Edge subdivision is an $MS_{2,2}$ -transduction that is **not** MS.

Proofs : Easy since, if S is transformed into T by an MS-transduction :

$$|D_T| \leq k \cdot |D_S| \quad \text{for fixed } k.$$

Exercises

- 1) Construct an MS-transduction that associates with a directed graph G the *directed acyclic graph* D of its strongly connected components. (The vertices of D are chosen among those of G).
- 2) Let G be undirected. For each k , $G(k)$ is the graph with same vertices and an edge $x—y$ iff x and y are at distance at most k . Define an MS-transduction that transforms G into $G(k)$.
- 3) If G has clique-width $\leq d$, then $G(k)$ has clique-width $\leq f(k,d)$ for some function f . Try to prove this (without looking at the next section). No such function exists for tree-width.
- 4) Prove that the transformation of a graph G into its *incidence graph* $\text{Inc}(G)$ is a MS-transduction on trees, and also on graphs of degree $\leq d$, for each fixed d .

7. Robustness results : Preservation of widths

For every class of graphs \mathcal{C} :

- 1) If \mathcal{C} has tree-width $\leq k$ and τ is an MS_2 – transduction, then $\tau(\mathcal{C})$ has tree-width $\leq f_\tau(k)$

Follows from :

\mathcal{C} has bounded tree-width iff $\mathcal{C} \subseteq \tau(\text{Trees})$ for some MS_2 – transduction τ (the proof is constructive in both directions)

- 2) If \mathcal{C} has clique-width $\leq k$ and τ is an MS – transduction, then $\tau(\mathcal{C})$ has clique-width $\leq g_\tau(k)$.

Follows from :

\mathcal{C} has bounded clique-width iff $\mathcal{C} \subseteq \tau(\text{Trees})$ for some MS – transduction τ (the proof is constructive)

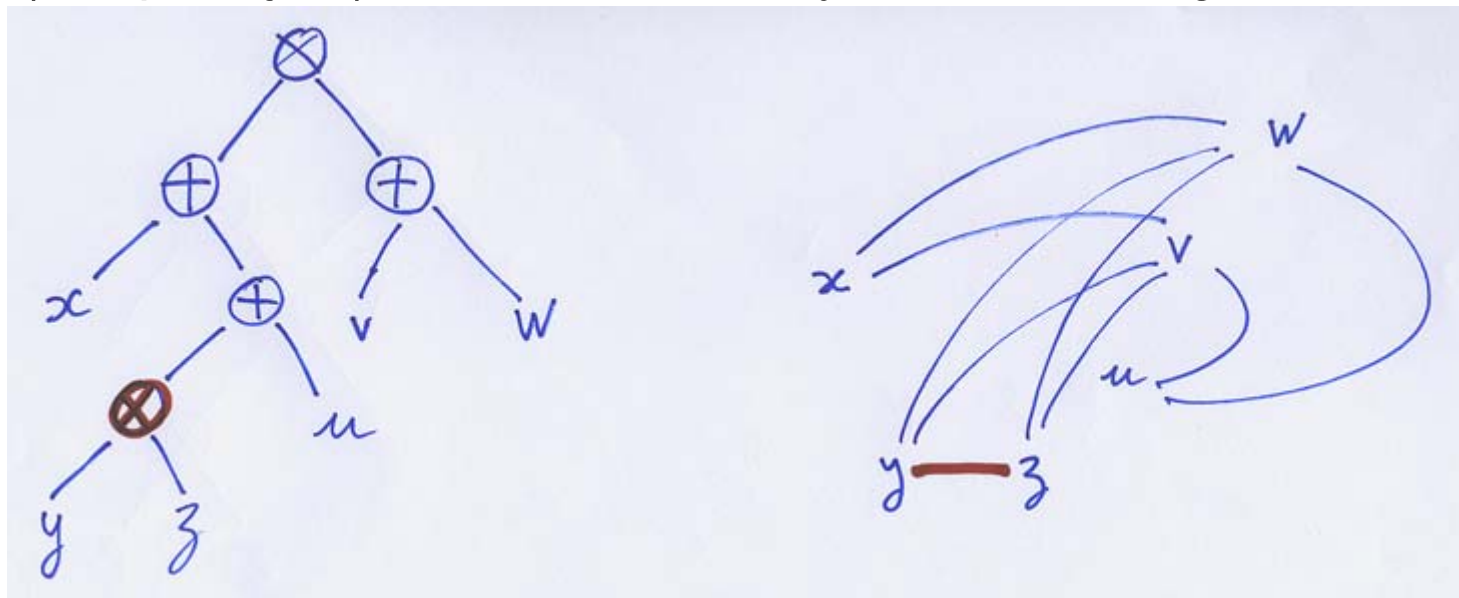
Logical Characterization of bounded clique-width :

- 1) A ***k*-clique-width** term is a rooted binary tree with nodes labelled by the finitely many operations symbols using labels $1, \dots, k$.
- 2) For each k , an MS-transduction can construct the defined graph from this labelled tree.

Hence : If a graph class \mathcal{C} has **clique-width** $\leq k$ then $\mathcal{C} \subseteq \tau_k(\text{Trees})$

for some **MS**-transduction τ_k .

Proof sketch for $k=2$, i.e., for **cographs**, defined by terms written with \oplus (disjoint union), \otimes (complete join) and constants x,y,z, \dots denoting vertices x,y,z, \dots



Vertices = $\{x,y,z,u,v,w\}$ = occurrences of constants in the term

Two vertices are adjacent if and only if their least common ancestor is labelled by \otimes (like y and z , or u and w).

These conditions can be expressed by MS formulas on the labelled tree.

Gives easy proofs (but no good bounds) of facts like :

- 1) If C has bounded tree-width, its **line graphs** have bounded clique-width.
- 2) If C has bounded tree-width or clique-width, the **transitive closures** of its graphs have bounded clique-width.
- 3) If C has bounded clique-width, the **transitive reductions** of its graphs have bounded clique-width.

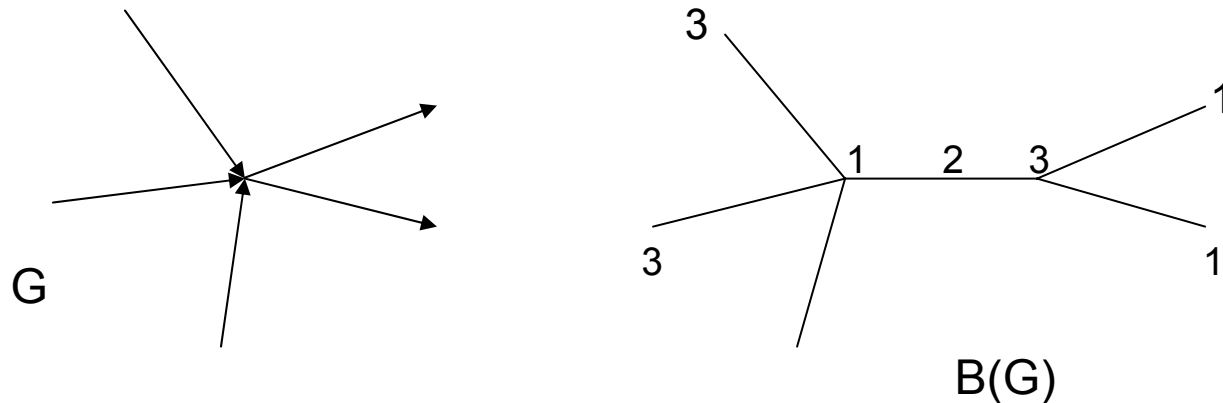
(Not trivial because clique-width is not monotone for subgraph inclusion).

- 4) The set of **chordal graphs** has *unbounded* clique-width

(because an MS transduction can define all graphs from chordal graphs, and graphs have *unbounded* clique-width).

Encoding a directed graph by a vertex-labelled undirected graph

Each vertex of G is split into 3 vertices labelled by 1,2,3 in $B(G)$:

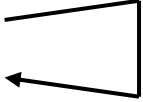

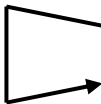
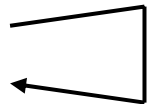
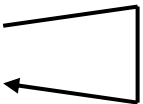


The clique-widths of G and $B(G)$ are related by fixed functions.

(Because the mapping B and its inverse are MS-transductions, even FO-transductions, hence they preserve bounded clique-width.)

Algorithms for checking rank-width of undirected graphs can be transformed into approximation algorithms for clique-width of directed graphs because rank-width and clique-width are related by fixed functions. (Oum, Hlineny, Seymour, Kanté)

Robustness results for HR- and VR-equational sets


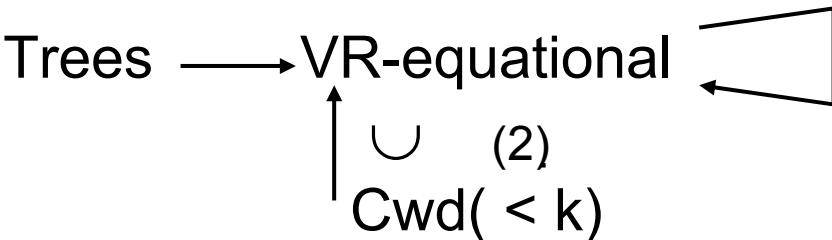

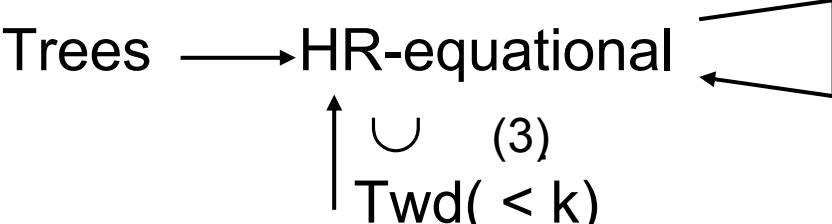
Words : Rational transductions (= inverse rational transductions)	
REC 	Dyck lang. \rightarrow Context-free (trees) 
Inverse MS transductions	Direct MS transductions
 MS-def. \subset VR-recog.  (1)	Trees \rightarrow VR-equational  \cup (2) \uparrow Cwd($\leq k$)

VR-equational \Rightarrow bounded clique-width.

(1) : A. Blumensath -B.C.

(2) : J. Engelfriet.

Robustness results : Preservation and generation (2)

Inverse MS transductions	Direct MS transductions
 <p>MS-def. \subset VR-recog. (1)</p>	 <p>Trees \longrightarrow VR-equational (2) \cup Cwd($< k$)</p>
Inverse $MS_{2,2}$ transductions	Direct $MS_{2,2}$ transductions
 <p>MS_2-def. \subset HR-recog. (1)</p>	 <p>Trees \longrightarrow HR-equational (3) \cup Twd($\leq k$)</p>

VR-equational \Rightarrow bounded clique-width.

HR-equational \Rightarrow bounded tree-width.

(1) : A. Blumensath -B.C.

(2) : J. Engelfriet.

(3) : B.C.-J. Engelfriet

Recognizability is preserved under inverse monadic second-order transductions. (A.Blumensath, B.C., 2004)

Recognizability of sets of relational structures is relative to graph operations consisting of :

disjoint union and

“QF” unary operations expressed by **formulas without quantifiers**.

Examples of QF operations : Edge-complement, relabellings,
edge-addition (cf. the definition of clique-width).

Proof sketch : Every MS transduction is the composition of MS transductions

of 3 “simple” types :

- Copy_k
- Parameterless and noncopying transduction
- Guessing unary relations

Recognizability is preserved by inverse transductions of each type.

Copy_k :

$$S \quad | \rightarrow \quad \underline{S \oplus S \oplus \dots S} \quad (\text{k times})$$

Disjoint union with binary relations $Y_{i,j}$ for $1 \leq i < j \leq k$ defined as

$$\{(x,y) / x \text{ is the } i\text{-copy, } y \text{ is the } j\text{-copy of some } u \text{ in } D_S\}$$

Facts : a) $\text{Copy}_k(S \oplus T) = \text{Copy}_k(S) \oplus \text{Copy}_k(T)$

b) For f quantifier-free, there is a quantifier-free operation g such that :

$$\text{Copy}_k(f(S)) = g(\text{Copy}_k(S))$$

Copy_k is “almost” a homomorphism, recognizability is preserved under inverse homomorphisms.

8. Graph classes with decidable *MS theories* (or *MS satisfiability problems*)

Theorem (Seese 1991): If a set of graphs has a decidable MS_2 satisfiability problem, it has bounded **tree-width**.

Theorem (B.C., Oum 2007): If a set of graphs has a decidable C_2MS satisfiability problem, it has bounded **clique-width**.

Answering a question by Seese : If a set of graphs has a decidable *MS* satisfiability problem, is it the image of a set of trees under an *MS* transduction, equivalently, has it bounded **clique-width** ?

MS_2 = *MS* logic **with** edge quantifications ; C_2MS = *MS* logic with **even cardinality** set predicates. A set C has a **decidable L -satisfiability problem** if one can decide whether any given formula in L is satisfied by some graph in C

Proof of Seese's Theorem :

A) If a set of graphs C has **unbounded tree-width**, the set of its **minors includes all $k \times k$ -grids** (Robertson, Seymour)

B) If a set of graphs **contains all $k \times k$ -grids**, its MS_2 satisfiability problem is **undecidable**

C) If C has **decidable MS_2 satisfiability** problem, so has $\text{Minors}(C)$,
because $C \longrightarrow \text{Minors}(C)$ is an $MS_{2,2}$ transduction.

Hence, if C has **unbounded tree-width** and a **decidable MS_2 satisfiability** problem, we have a contradiction for the **decidability of the MS_2 satisfiability** problem of $\text{Minors}(C)$.

Proof of Courcelle-Oum's Theorem :

D) Equivalence between the cases of all (directed and undirected) graphs and bipartite undirected graphs (with an encoding of directed graphs as labelled bipartite undirected ones).

A') If a set of bipartite graphs \mathcal{C} has **unbounded clique-width**, the set of its *vertex-minors* contains all " S_k " graphs

C') If \mathcal{C} has a **decidable C_2MS satisfiability** problem, so has $\text{Vertex-Minors}(\mathcal{C})$, because $\mathcal{C} \longrightarrow \text{Vertex-Minors}(\mathcal{C})$ is a **C_2MS** transduction.

E) An **MS** transduction transforms S_k into the **$k \times k$ -grid**.

Hence $A' + B + C' + E$ gives the result for bipartite undirected graphs.

The general result follows with the encoding D).

Definitions and facts

Local complementation of G at vertex v

$G * v = G$ with edge complementation of $G[n_G(v)]$,

the subgraph induced by the **neighbours** of v

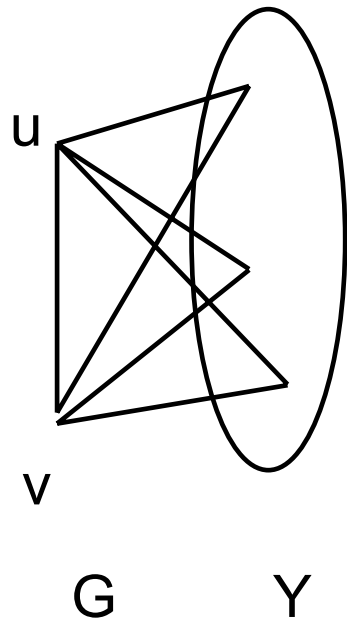
Local equivalence (\approx_{loc}) = transitive closure of local complementation
(at all vertices)

Vertex-minor relation :

$H \leq_{VM} G : \Leftrightarrow H$ is an induced subgraph of some $G' \approx_{loc} G$.

Proposition (B.C. and Oum 2004) : The mapping that associates with G its locally equivalent graphs is a C_2MS transduction.

Why is the even cardinality set predicate necessary ?



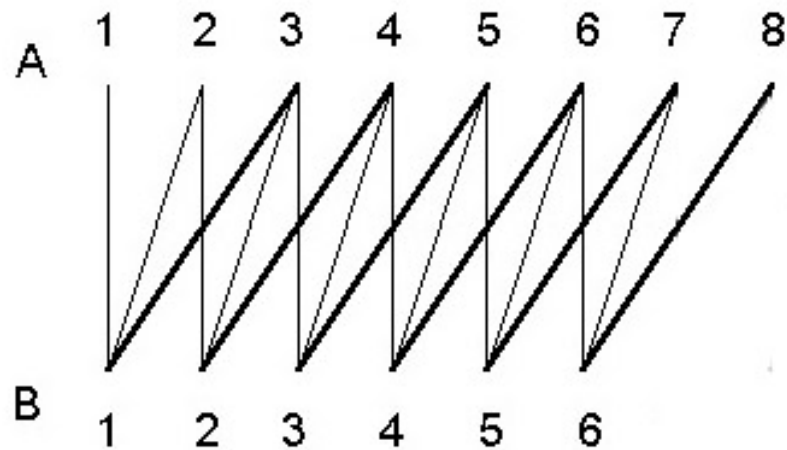
Consider $G * X$ for $X \subseteq Y$:

u is linked to v in $G * X$

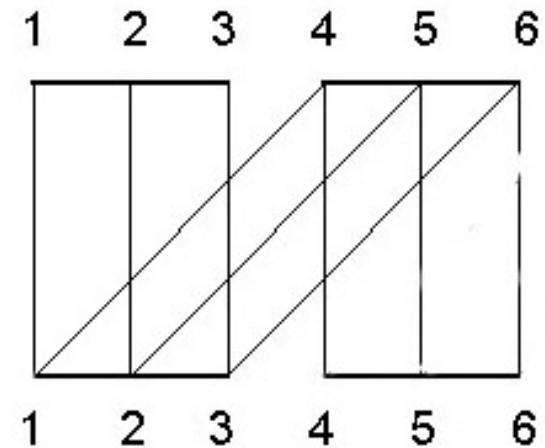
$\Leftrightarrow \text{Card}(X)$ is even

($G * X =$ composition of local complementations at all vertices from X)

Definition of S_k , bipartite : $A = \{1, \dots, (k+1)(k-1)\}$, $B = \{1, \dots, k(k-1)\}$
 From S_k to $\text{Grid}_{k \times k}$ by an MS transduction



S_3



(folded) $\text{Grid}_{3 \times 4}$

The orderings of A and B : x, y are consecutive $\Leftrightarrow \text{Card}(n_G(x) \Delta n_G(y)) = 2$

One recognizes the edges from $i \in B$ to $i \in A$, and from $i \in B$ to $i+k-1 \in A$ (thick edges on the left drawing)

One creates edges (e.g. from $1 \in A$ to $2 \in A$, from $2 \in A$ to $3 \in A$ etc...and similarly for B, and from $1 \in B$ to $4 \in A$, etc...) one deletes others (from $4 \in B$ to $6 \in A$ etc...), and vertices like 7,8 in A, to get a grid containing $\text{Grid}_{k \times k}$

9. A few open questions

Question 1 : What should be the clique-width or rank-width of hypergraphs (or relational structures) ?

Question 2 : Which graph operations, quantifier-free definable or not, yield extensions of the signatures VR, HR that are *equivalent* to them, i.e., that define the same recognizable and equational sets ?

(some answers already given by A. Blumensath, B.C., P. Weil)

Or that yield larger classes of equational sets for which MS logic is decidable ?

Question 3 : Is it true that the decidability of the MS (and not of the C_2 MS) satisfiability problem for a set of graphs implies bounded clique-width, as conjectured by Seese ?

More important (personal opinion) :

Question 4 : What about Question 3 for sets of hypergraphs or relational structures ?

Other questions (and answers) are in the proceedings of ICALP 2008

Appendix : The “Büchi” style proof of the Recognizability Theorem.

The case of terms in $T(F)$, for F *finite* signature.

Logical representation of terms : Nodes of the syntactic tree, *son* binary relation, labels saying “my label is f ” and “I am the i -th brother”.

Construction: automaton $A(\varphi)$ for sentence φ that defines $\text{Models}(\varphi) \cap T(F)$ by induction on sentences (without first-order variables and \forall).

For the case $\exists X. \varphi(X)$, we need $A(\varphi(X))$, and more generally $A(\varphi(X_1, \dots, X_n))$: f in F is replaced by $(f, (w_1, \dots, w_n))$ in $F \times \{0, 1\}^n$ of same arity, a term t in $T(F \times \{0, 1\}^n)$ encodes $\text{pr}_0(t)$ in $T(F)$ and an assignment $v(t) : \{X_1, \dots, X_n\} \rightarrow P(\text{Nodes}(t))$

(if u is an occurrence of $(f, (w_1, \dots, w_n))$, then $w_i = 1$ iff $u \in X_i$)

We construct $A(\varphi(X_1, \dots, X_n))$ that defines

$$\{ t \in T(Fx\{0,1\}^n) \mid (\text{pr}_0(t), v(t)) \models \varphi \}$$

Then : $L(A(\exists X_{n+1} . \varphi(X_1, \dots, X_{n+1}))) = \text{pr}(L(A(\varphi(X_1, \dots, X_{n+1}))))$

where pr is the projection that eliminates the last Boolean.

One obtains a *nondeterministic* automaton ; the case of negation needs determinization. The case of atomic formulas is easy.

The number of states is an *h-iterated exponential*, where h = maximum nesting of negations. This is not avoidable.

The case of graphs : VR_k -recognizability

We fix k , a bound on clique-width of the considered graphs and

F_k = the binary operation \oplus and $O(k^2)$ unary operations (*add-edg_{a,b}*, *relab_{a,b}*)

We use a single constant **1** defining one vertex labelled by 1.

(1 can be changed into another label).

A term t in $T(F_k \cup \{1\})$ defines a graph $G(t)$ with vertex set $V_{G(t)} =$
the set of occurrences of **1** in t . For representing assignments

$$v: \{X_1, \dots, X_n\} \rightarrow P(V_{G(t)})$$

we replace **1** by the constants $(1, (w_1, \dots, w_n))$ where $w_i \in \{0, 1\}$.

A term t in $T(F_k \cup (\{1\} \times \{0, 1\}^n))$ encodes the graph $G(\text{pr}_0(t))$ and
some assignment $v(t)$.

As in the case of terms, we can construct for all relevant k, n and φ a finite deterministic automaton $A(\varphi(X_1, \dots, X_n))$ that defines

$$\{ t \in T(F_k \cup (\{1\} \times \{0, 1\}^n)) \mid (G(\text{pr}_0(t)), v(t)) \models \varphi \}$$

Inductive steps :

$\varphi \wedge \psi, \varphi \vee \psi$: product of two automata

$\neg \varphi$: exchange of accepting / non-accepting states of a **deterministic** automaton

$\exists X_{n+1} \varphi$: “projection” **pr** : makes automata **non-deterministic**

Basic cases : atomic formulas.

Example : the automaton $A(\text{edg}(X_1, X_2))$ with k^2+k+3 states

Graph labels are in $[k] = \{1, \dots, k\}$.

States : 0, Ok, $a(1)$, $a(2)$, ab , Error, for a, b in $[k]$, $a \neq b$

Meanings of states (at occurrence u in term t that defines a graph G)

0 : $X_1 = \emptyset$, $X_2 = \emptyset$

Ok : $X_1 = \{v\}$, $X_2 = \{w\}$, $\text{edg}(v, w)$ true in $G(t/u)$ defined by subterm t/u

$a(1)$: $X_1 = \{v\}$, $X_2 = \emptyset$, v has label a in $G(t/u)$

$a(2)$: $X_1 = \emptyset$, $X_2 = \{w\}$, w has label a in $G(t/u)$

ab : $X_1 = \{v\}$, $X_2 = \{w\}$, v has label a , w has label b

and $\text{edg}(v, w)$ not true, all that in $G(t/u)$,

Error : all other cases.

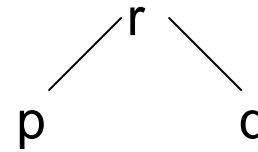
Accepting state : Ok

Transition rules

For the constant **1** :

(1,00) \rightarrow 0 ; **(1,10)** \rightarrow **1(1)** ; **(1,01)** \rightarrow **1(2)** , **(1,11)** \rightarrow Error

For the binary operation \oplus :



If $p = 0$ then $r = q$

If $q = 0$ then $r = p$

If $p = a(1)$ and $q = b(2)$, $a \neq b$ then $r = ab$

If $p = b(2)$ and $q = a(1)$, $a \neq b$ then $r = ab$

Otherwise $r = \text{Error}$

For unary operations *add-edge*_{a,b}

r
|
p

If $p = ab$ then $r = Ok$ else $r = p$

For unary operations *relab*_a \longrightarrow *b*

If $p = a(i)$ where $i = 1$ or 2 then $r = b(i)$

If $p = ac$ where $c \neq a$ and $c \neq b$ then $r = bc$

If $p = ca$ where $c \neq a$ and $c \neq b$ then $r = cb$

If $p = 0$ or Ok or $Error\ c(i)$ or cd or dc where $c \neq a$
then $r = p$

We have **only** proved that for all k and MS sentence φ

1) $\{ t \in T(F_k \cup \{1\}) \mid G(t) \models \varphi \}$ is $T(F_k \cup \{1\})$ -recognizable

2) The set of models of φ of clique-width $\leq k$, is VR_k -recognizable

This is **NOT** the **Recognizability Theorem** : it needs a type-preserving, locally finite congruence on ALL graphs (of unbounded clique-width).

VR_k -recognizability for each k *does not* imply VR -recognizability.

The set of duplicated square grids $G_{m \times m} \oplus G_{m \times m}$ is not VR -recognizable but each of its restrictions to graphs of clique-width at most k is finite hence VR_k -recognizable (by the proved special case).

However this proof is sufficient for **FPT algorithms** and for the **Filtering Theorem** because in these cases only graphs of clique-width at most some k are concerned.