



Monadic second-order logic for graphs  
Language and Graph Theoretical applications

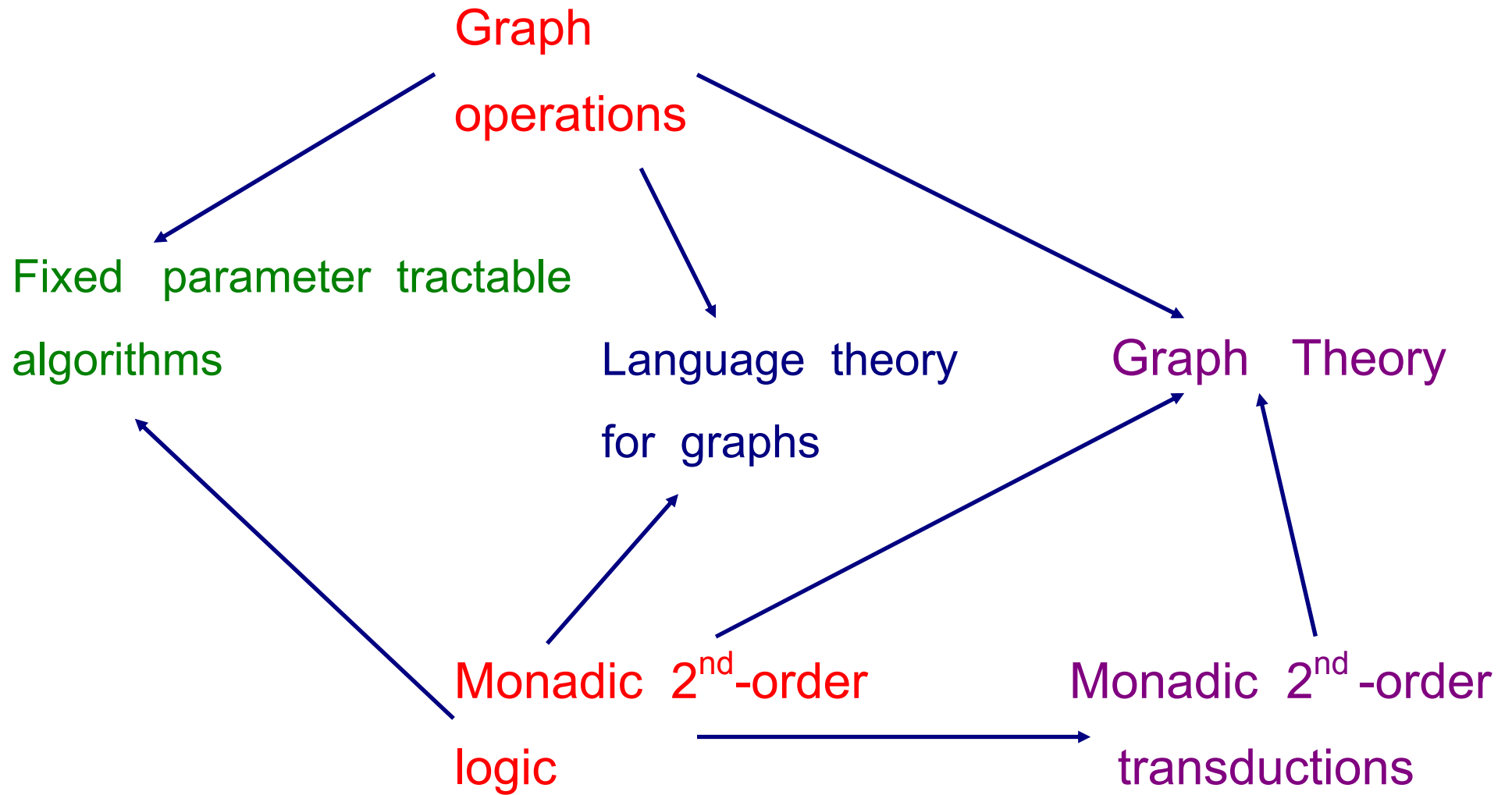
(Part 3)

*Bruno Courcelle*

Université Bordeaux 1, LaBRI & 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

6. Monadic second-order transductions (extension of a classical concept in language theory).
7. *Robustness results* : preservation of classes under direct and inverse monadic second-order transductions.  
Short proofs in graph theory.
8. Logic and *graph structure theory* : Comparing encoding powers of graph classes *via* monadic second-order transductions
9. Graph classes on which monadic second-order logic is *decidable*
10. Some open questions

## 6. Monadic second-order transductions

$\Sigma$  = finite set of relation symbols with fixed arities ( $\rho(R)$  = arity of  $R$ ).

$\text{STR}(\Sigma)$ : finite  $\Sigma$ -relational structures  $S = \langle D_S, (R_S)_{R \in \Sigma} \rangle$ ,  
 $R_S$  relation on  $D_S$  of arity  $\rho(R)$

An **MS-transduction** is a partial function

$\tau : \text{STR}(\Sigma) \times \text{“data”} \rightarrow \text{STR}(\Gamma)$  specified by MS formulas.

Basic case :  $\tau : \text{STR}(\Sigma) \rightarrow \text{STR}(\Gamma)$  ;  $T = \tau(S)$  is defined “inside”  $S$  by MS formulas.

*Examples* : The edge -complement ; the transitive closure of a directed graph.

Next case :  $T = \tau(S, \text{"data"})$  ; the “data” is a tuple  $X_1, \dots, X_p$  of subsets of the domain of  $S$  ; these sets are called the **parameters**.

Parameters  $X_1, \dots, X_p$  are constrained to satisfy an MS property.

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

$(G, X, Y, Z) \longmapsto$  the *minor* of  $G$  having vertex set  $X$ , resulting from the contraction of the edges of  $Y$  and the deletion of the edges and vertices of  $Z$ . (This transduction is **MS<sub>2,2</sub>** ; see below.)

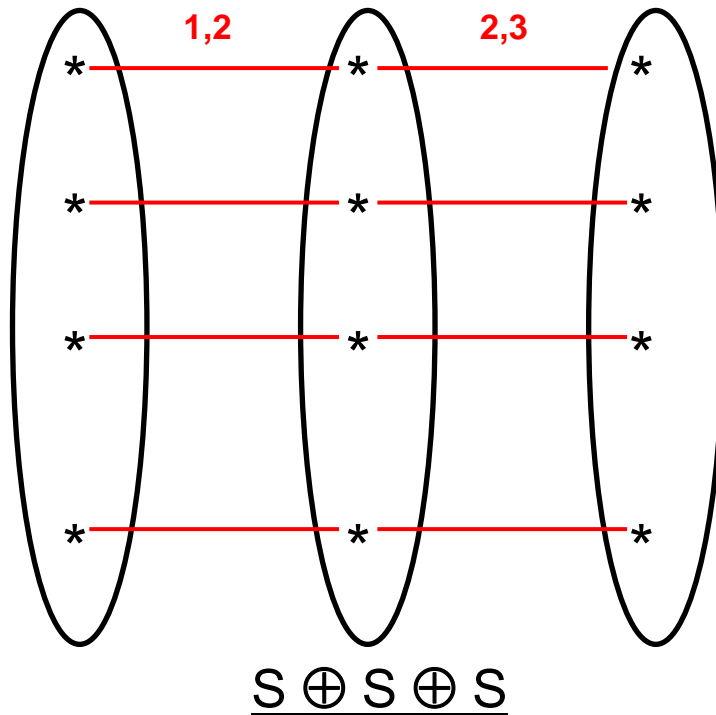
In the second example, no two vertices of  $X$  should be linked by a path of edges in  $Y$ .

$\tau(S) :=$  the set of all  $T = \tau(S, X_1, \dots, X_p)$

for all “good” tuples of parameters.

General case : T is defined as above inside

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



## The fundamental property of MS transductions

If  $S \longrightarrow \tau(S)$

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

Every MS formula  $\psi$  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  $\psi$  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 *in*  $S$  by MS formulas).

*Theorem* : The composition of two MS-transductions is an MS-transduction.

*Example 1* (without parameters) : The *square* mapping  $\delta$  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 :

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

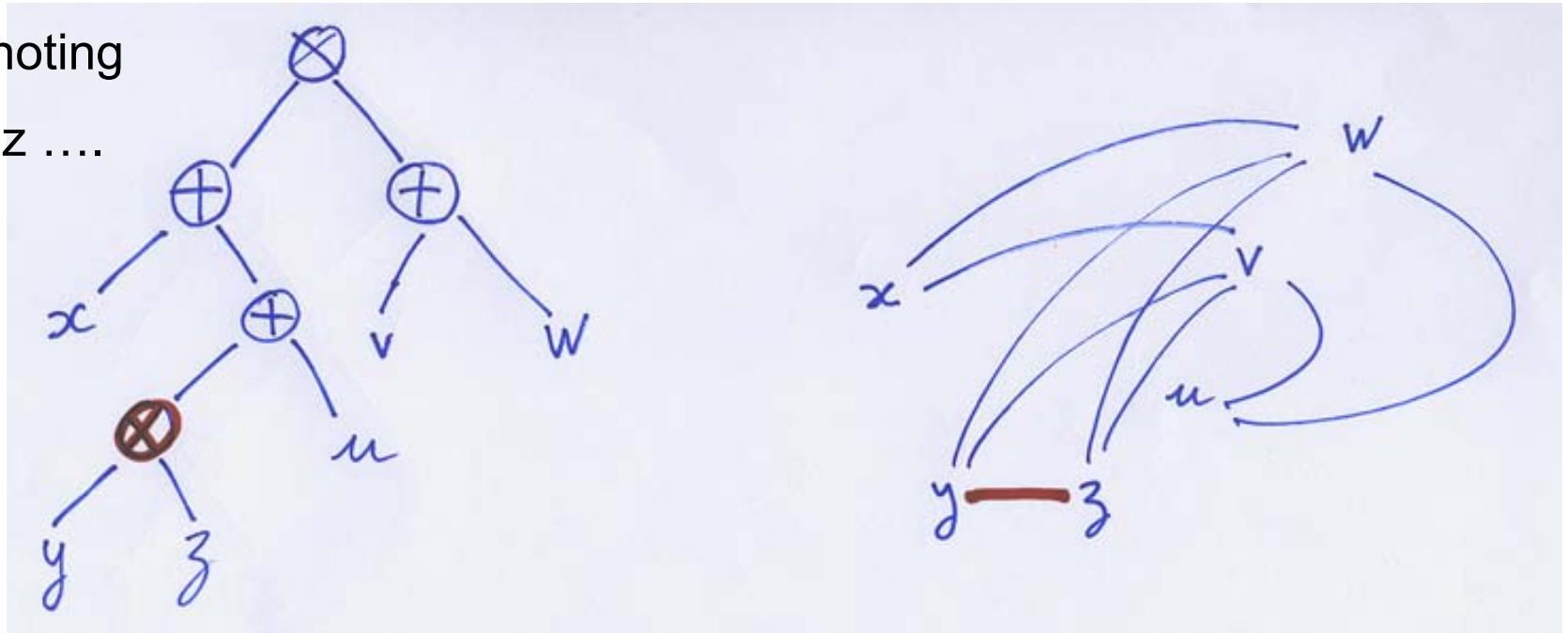
We also remove the "marker" predicates  $p_1, p_2$ .



## Example 2: From a term to a cograph

Terms are 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.

## Edge quantification and edge description

There are 2 representations for an input graph and 2 for the output: type 1 :  $G = (V_G, \text{edg}_G)$  and type 2 :  $\text{Inc}(G) = (V_G \cup E_G, \text{in}_G)$ .

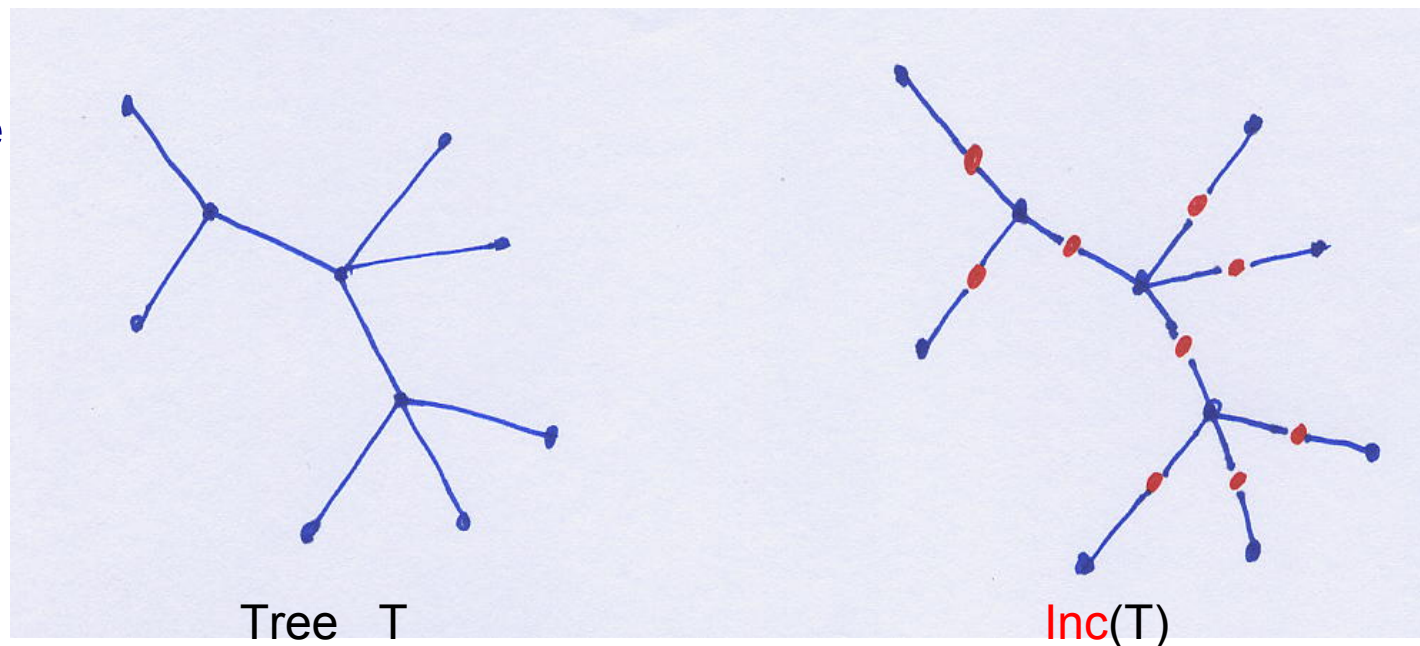
Hence 4 types of graph transductions, denoted by :

$MS_{1,1}$  (or  $MS$  to simplify),  $MS_{1,2}$ ,  $MS_{2,1}$  and  $MS_{2,2}$

$MS_{i,o}$  means  $i$  = type of input,  $o$  = type of output.

For *sparse* graphs, type 2 is equivalent to type 1.

Example 3: From a tree to its incidence graph (also a tree)



$T = \langle N, \text{edg} \rangle$ ; we use parameter  $\{ \mathbf{r} \}$  to make  $T$  rooted and directed

$\tau(T, \{ \mathbf{r} \} ) = \langle N \cup (N - \{ \mathbf{r} \} ) \times \{ 1 \} , \text{inc}(\dots) \rangle$

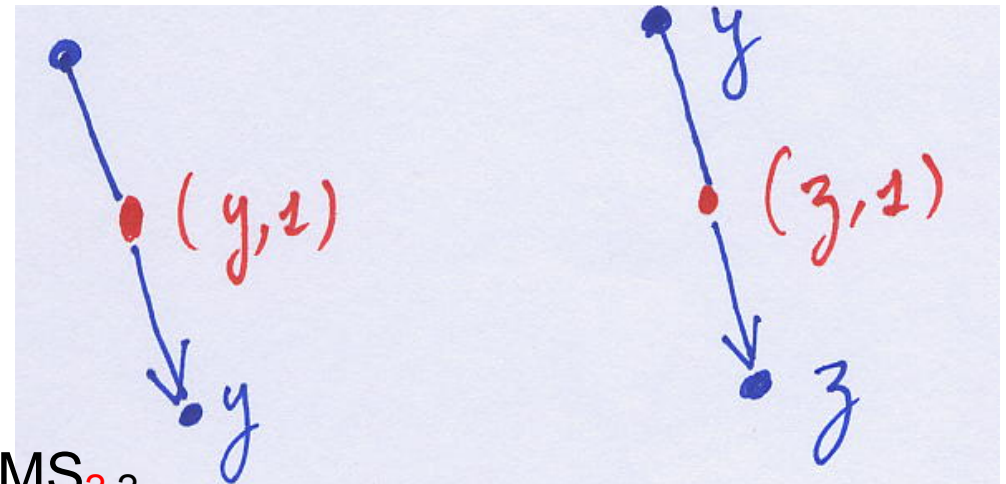
$\text{in}(x,y)$  is defined by :

$x = (y,1) \vee \exists z [ x = (z,1) \wedge \text{edg}(y,z)$

$\wedge$  "y is on the path from  $\mathbf{r}$  to z" ]

From trees (or terms) to graphs :

$$MS_{1,1} = MS_{2,1} \quad \text{and} \quad MS_{1,2} = MS_{2,2} .$$



## Main Results (will be made more precise) :

(1) **MS**-transductions preserve bounded **clique-width** and the (corresponding) class of **VR**-equational sets.

(2) **MS<sub>2,2</sub>**-transductions preserve bounded **tree-width** and the (corresponding) class of **HR**-equational sets.

*Meaning* : Robustness of the *two* graph hierarchies based on clique-width and tree-width.

MS - transductions and  $MS_{2,2}$  - transductions are **incomparable**

*Why ?* For expressing graph properties,  $MS_2$  logic is more powerful than  $MS_1$  logic (the “ordinary” MS logic).

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

**Transitive closure** is  $MS = MS_{1,1}$  but **not**  $MS_{2,2}$

**Edge subdivision** is  $MS_{2,2}$  but **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 simple 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 simple 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 does exist for tree-width.
- 4) Prove that the transformation of a simple graph  $G$  into its *incidence graph*  $\text{Inc}(G)$  is a MS-transduction on graphs of maximal degree  $d$ , 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  $\tau$  is an  $\text{MS}_{2,2}$  – transduction, then  $\tau(\mathcal{C})$  has tree-width  $\leq f_\tau(k)$

*Follows from :*

$\mathcal{C}$  has bounded tree-width  $\Leftrightarrow \mathcal{C} \subseteq \tau(\text{Trees})$  for some  $\text{MS}_{2,2}$  – transduction  $\tau$  (the proof is constructive in both directions)

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

*Follows from :*

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

## *Proof sketch for the logical characterization of bounded clique-width*

1) A ***k*-clique-width** term is a rooted binary tree with each node labelled by one of 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. (Extension of the proof for cographs, cf. page 9.)

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

for some **MS**-transduction  $\tau_k$ .

The converse uses technical tools from model theory (Feferman-Vaught)

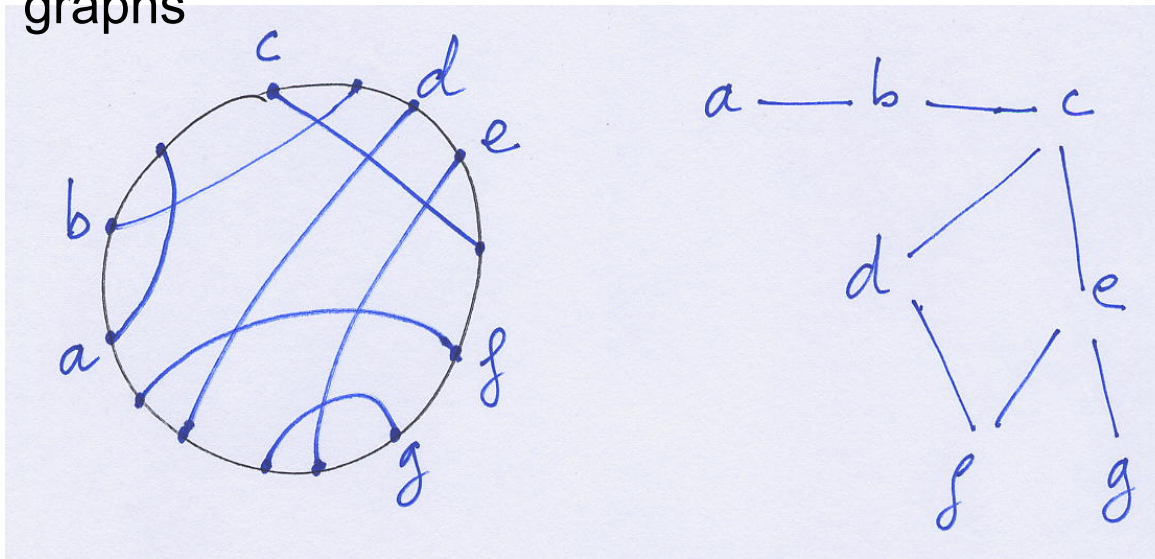
*The proofs for tree-width are similar.*



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  $C$  (directed graphs) has bounded tree-width or clique-width, the **transitive closures** of its graphs have bounded clique-width.
- 3) If  $C$  (directed graphs) 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).
- 5)  $k$ -leaf powers and similar “power” graphs of trees have bounded cwd.

## 6) Circle graphs



Chord diagram  $\Delta$

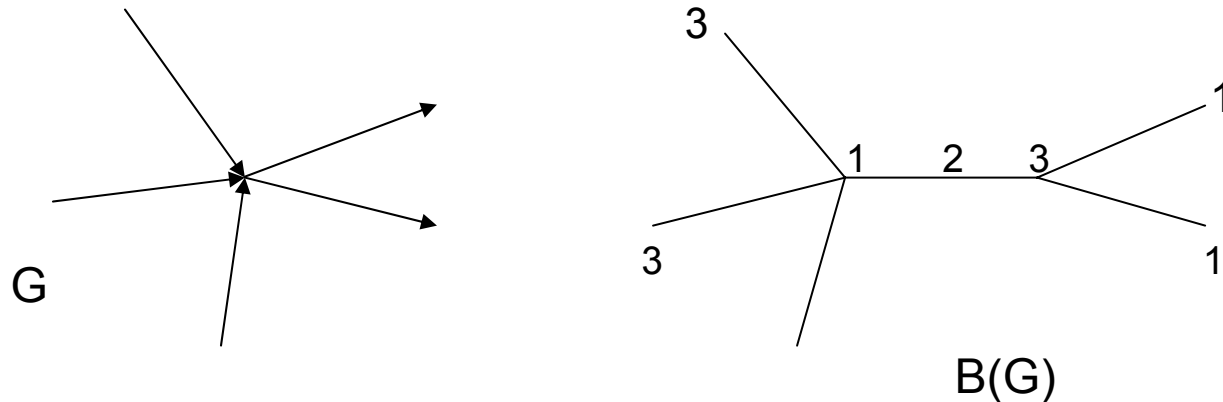
Circle graph  $G(\Delta)$

Theorem: Graphs  $\Delta$  have bounded tree-width  $\Leftrightarrow G(\Delta)$  have bounded clique-width.

- 1) **MS** - transduction from  $G(\Delta)$  to  $\Delta$  ;
- 2) use “*split decomposition*” (Cunningham) and an **MS**-transduction from *prime* circle graphs to their unique chord diagrams.

# Encoding a directed graph into 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, 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)

## Logical characterizations of equational sets

$C$  is HR-equational  $\Leftrightarrow C = \tau(\text{Trees})$  for some

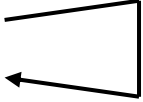

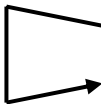
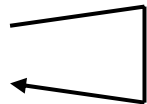
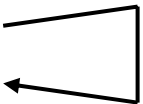
$MS_{2,2}$ -transduction  $\tau$  (for bounded tree-width we have  $\subseteq$ ).

$C$  is VR-equational  $\Leftrightarrow C = \tau(\text{Trees})$  for some

$MS$  - transduction  $\tau$  (for bounded clique-width we have  $\subseteq$ ).

*Consequences* : Closure of equational sets under the corresponding transductions.

# Robustness results for HR- and VR-equational sets


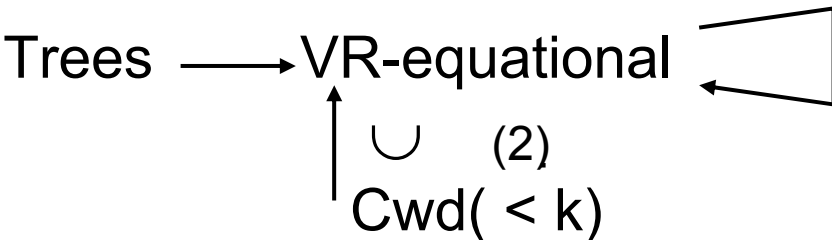

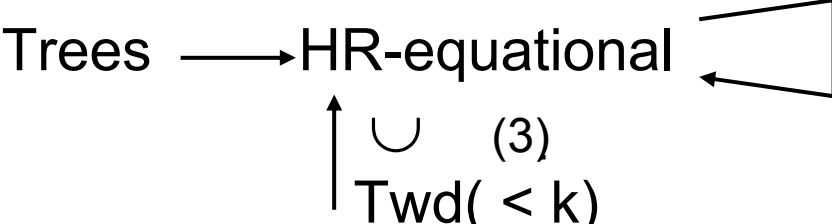
<b>Words : rational transductions (= inverse rational transductions)</b>	
REC 	Dyck lang. $\rightarrow$ Context-free (trees) 
<b>Inverse MS transductions</b>	<b>Direct MS transductions</b>
 MS-def. $\subset$ VR-recog.  (1)	Trees $\rightarrow$ VR-equational  $\cup$ (2) $\uparrow$ Cwd( $\leq k$ )

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. <math>\subset</math> VR-recog. (1)</p>	 <p>Trees <math>\longrightarrow</math> VR-equational (2)  <math>\cup</math>  <math>\uparrow</math> Cwd(<math>&lt; k</math>)</p>
Inverse MS <sub>2,2</sub> transductions	Direct MS <sub>2,2</sub> transductions
 <p>MS<sub>2</sub>-def. <math>\subset</math> HR-recog. (1)</p>	 <p>Trees <math>\longrightarrow</math> HR-equational (3)  <math>\cup</math>  <math>\uparrow</math> Twd(<math>\leq k</math>)</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

unary operations expressed by **quantifier-free formulas (QF)**

Examples of QF operations : Edge-complement, relabellings,

*Add<sub>a,b</sub>* (cf. the definition of clique-width).

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

of 3 particular types : - Copy<sub>k</sub>

- Parameterless and noncopying transduction

- Guessing unary relations

Recognizability is preserved by inverse transductions of each type.

Copy<sub>k</sub> :

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

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

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

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 :

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

$\text{Copy}_k$  is “almost” a homomorphism, and recognizability is preserved under inverse homomorphisms.



## 8. Encoding powers of graph classes *via* MS transductions

An MS-transduction  $\tau$  defines a graph  $H$  *inside* a graph  $G$  with help of parameters (sets of vertices or edges of  $G$ ):

say that  $H$  is *encoded* in  $G$ : the encoding is represented by the parameters and  $\tau$  is the *decoding* function.

The *encoding powers* of graph classes  $C$  and  $D$  can be compared as follows :

$$C \leq D \quad \text{if} \quad C \subseteq \tau(D) \quad \text{for some MS transduction } \tau$$

We get a *quasi-order on graph classes*.

We consider **MS<sub>2,2</sub> - transductions** : (formulas use edge set quantifications and must construct incidence graphs as outputs.)

For graph classes  $C$  and  $D$  we let :

$C \leq D$  if  $C \subseteq \tau(D)$  for some MS<sub>2,2</sub> -transduction  $\tau$

$C \equiv D$  if  $C \leq D$  and  $D \leq C$

$C < D$  if  $C \leq D$  and  $C \not\equiv D$

$C <_c D$  if  $C < D$  and there is no  $E$  with  $C < E < D$

*What is the structure of  $<_c$  (the **covering relation** of  $\leq$ ) ?*

With help of Robertson & Seymour : Graph Minors I and V :

$$\{\bullet\} < \text{Paths} <_c \text{Trees} <_c \text{Grids}$$

These classes **encode** respectively :

finite sets,

sets of graphs of bounded path-width,

sets of graphs of bounded tree-width,

all sets of graphs .

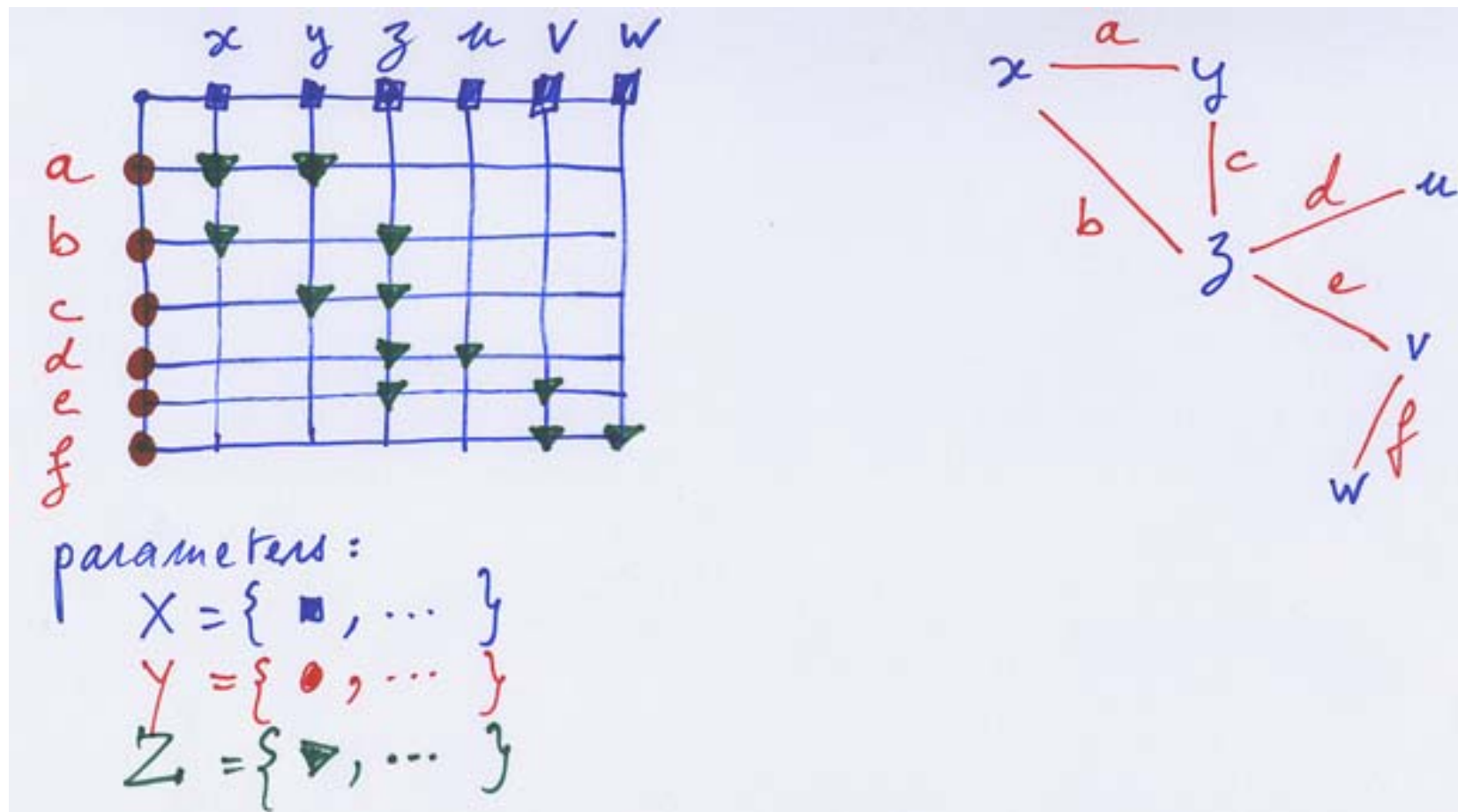
*Proof* :  $\text{Trees} <_c \text{Grids}$ .

If a graph class  $C$  has bounded tree-width, it is  $\leq \text{Trees}$ .

If  $C$  has unbounded tree-width, it contains all grids as minors,

hence :  $\text{Grids} \leq C$  and  $\text{Grids} \equiv C$ , because  $\text{Graphs} \leq \text{Grids}$

Proof : All graphs  $\leq$  Grids



A monadic second-order transduction using parameters  $X, Y, Z$  can transform all grids into all incidence graphs  $\text{Inc}(G)$ .

*More difficult*: What is below Paths ?

*Answer*: (A. Blumensath-B. C., LMCS 2010)

$\{\bullet\} <_c T_2 <_c \dots T_n <_c T_{n+1} <_c \dots < \text{Paths} <_c \text{Trees} <_c \text{Square grids}$

where  $T_n$  is the class of rooted trees of height at most  $n$  (and unbounded degree).

*Idea*:  $T_n$  encodes the classes of graphs having tree-decompositions of *height* at most  $n$  and width at most  $k$  (for all  $k$ ).

**Definition :** *n-depth tree-width of G* =  $\text{tw}_n(G)$  = minimal width of a tree-decomposition of G of height at most n.

Related notion : *tree-depth* (Nesetril , Osona de Mendez).

$\text{td}(G)$  = minimal k such that each connected component of G has a depth-first (normal) spanning tree of height at most k.

**Some properties of these variants of tree-width :**

1)  $\text{pwd}(G) \leq n \cdot (\text{tw}_n(G) + 1)$

2) If G is a minor of H :  $\text{tw}_n(G) \leq \text{tw}_n(H)$  ,  $\text{td}(G) \leq \text{td}(H)$

3)  $\text{td}(G) \leq n$  implies  $\text{tw}_n(G) \leq n$ ,

4)  $\text{tw}_n(G) \leq k$  implies  $\text{td}(G) \leq n \cdot k$

## Excluded Path Theorem

(cf. the Excluded Tree and Grid Theorems of GM1 and GM5)

A class of graphs  $C$  excludes some path as a minor  
(equivalently, as a subgraph)

$\Leftrightarrow$  for some  $n$ ,  $C$  has bounded  $n$ -depth tree-width

$\Leftrightarrow C$  has bounded tree depth.

We use  $n$ -depth tree-width rather than tree-depth to characterize the graph classes encoded by trees of *each height  $n$*

## Logical properties of $n$ -depth tree-width.

*Proposition* : For each  $n$  and  $k$ , there exists an  $MS_{2,2}$ -transduction that maps every graph of  $n$ -depth tree-width at most  $k$  *to all its strict* tree-decompositions of height at most  $n$  and width at most  $k$

(*strict* = with certain connectivity properties ; every tree-decomposition can be made strict without increasing height and width).

*Remark* : The obstruction sets of graphs for  $n$ -depth tree-width  $\leq k$  are computable from each pair  $n, k$  because we have monadic second-order characterizations of these classes and bounds on the tree-widths of the obstruction sets. The same holds for the property “tree-depth  $\leq k$ ”.



In the hierarchy :

$$\{\bullet\} <_c T_2 <_c \dots <_c T_n <_c \dots < \text{Paths} <_c \text{Trees} <_c \text{Grids}$$

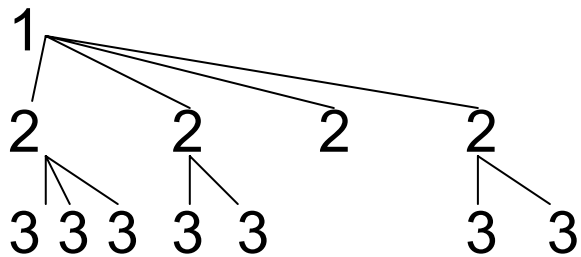
each level  $T_n$  encodes the sets of graphs of bounded  $n$ -depth tree-width.

Proofs to be done :

1)  $T_n \leq \text{Paths}$

Trees of height  $n$  can be encoded as sequences over  $[n]$  and decoded by MS-transductions.

1 2 333 2 33 2 2 33 encodes the tree :



$$2) T_n < T_{n+1}$$

One cannot define by an MS-transduction all trees of height  $n+1$  from all trees of height  $n$ .

The (**technical**) proof uses an analysis of MS definable relations on trees and some counting arguments.

Case  $n = 2$ .

Trees of height 2 correspond (via MS transductions) to sets (without relations).

If a **k-copying** MS-transduction with **p parameters** transforms sets into trees, these trees have less than  $k \cdot 2^p$  internal nodes.

We cannot get all trees of height 3 from sets by a single MS-transduction.

3) Hence, we cannot have  $T_n \equiv \text{Paths}$

## “Dichotomy arguments” :

1) Let  $C$  be a set of bounded pathwidth (i.e.,  $C \leq \text{Paths}$ ):

*Either* : it contains all paths as minors, then  $C \equiv \text{Paths}$

*Or* : (Excluded Path Thm)  $\text{twd}_n(C)$  is bounded and  $C \leq T_n$  for some  $n$

2) Let  $C$  be a set of  $n$ -depth tree-width  $\leq k$  ( $C \leq T_n$ ):

*Either* : for all  $m$ , there is  $G$  in  $C$  s.t., for each  $n$ -depth tree-dec.  $U$  of width  $k$  of  $G$ , the tree  $U$  contains  $T(n,m)$  ( $T(n,m)$  = the  $m$ -ary complete tree of height  $n$ ) and then  $T_n \leq C$  (because  $n$ -depth tree-decompositions of width  $k$  are definable by MS transductions)

*Or* : for some  $m$ , every  $G$  in  $C$  has an  $n$ -depth tree-dec.  $U$  of width  $k$ , s.t.  $U$  does not contain  $T(n,m)$ . By contracting some edges of  $U$ , one gets an  $(n-1)$ -depth tree-dec. of  $G$  of width  $m \cdot (k+1)$ , hence  $C \leq T_{n-1}$ .

*Open question*: What about the hierarchy based for  
MS – transduction ?

Theorem (B.C. & Oum, 2007) :

There exists an MS - transduction (using *even cardinality set predicates*) that transforms every set of undirected graphs of unbounded rank-width into the set of all square grids.

(Uses *vertex-minors* instead of minors)

We need a result corresponding to Graph Minors 1 about “linear rank-width” and excluding a forest as a *vertex-minor*.

We need also something like “n-depth rank-width” and constructions by MS transductions of appropriate rank-decompositions.

## 9. 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 the **even cardinality** set predicate. 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 the result on  $MS_2$ -satisfiability and tree-width :*

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 the result on  $C_2$ MS-satisfiability and clique-width :*

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; cf. page 15).

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

C') If  $C$  has a **decidable  $C_2$ MS-satisfiability** problem, so has  $Vertex-Minors(C)$ , because  $C \longrightarrow Vertex-Minors(C)$  is a  $C_2$ MS-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)

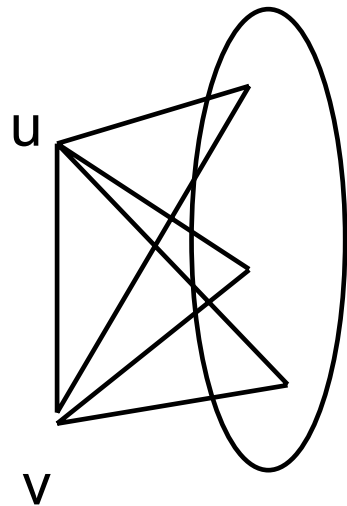
*Vertex-minor* relation :

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



*Proposition (B.C.-Oum, 2004)* : The mapping that associates with  $G$  its locally equivalent graphs is a **C<sub>2</sub>MS**-transduction.

The even cardinality set predicate is necessary :



G      Y

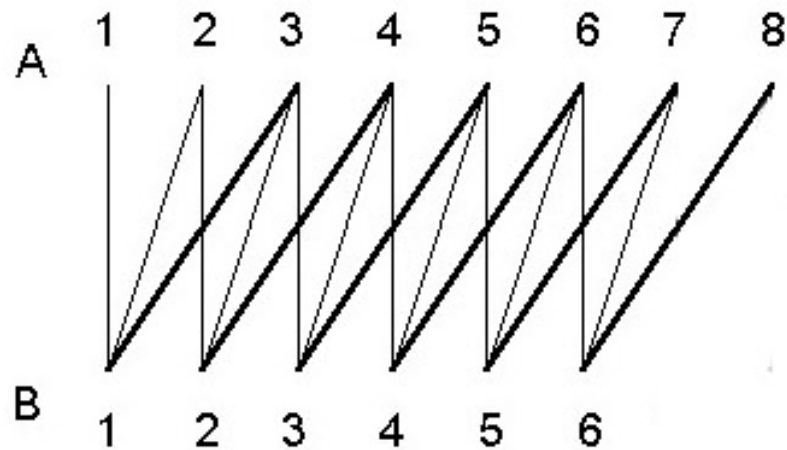
Consider  $G * X$  for  $X \subseteq Y$  :

$u$  is linked to  $v$  in  $G * X$

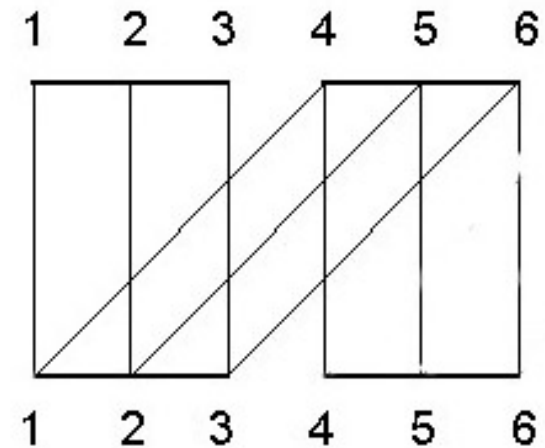
$\Leftrightarrow$   $\text{Card}(X)$  is even

( $G * X$  = 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  $\text{Grid}_{k \times k}$

## 10. A few open questions

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

*Question 2* : 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 :

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